FSUIPC7 User Guide

Flight Simulator Universal Inter-Process Communication
by John Dowson, © May 2024

An application interfacing module for MSFS 2020

Support Forum: [John Dowson's FSUIPC7 Support Forum](#)

For Versions 7.4.12 and later of FSUIPC

Note: All our Windows based software is always available in the latest versions from from the **Download Links** subforum in our Support Forum (link above), and normally also from

https://fsuipc.com

(Selected modules are also available elsewhere).

Assorted extras and interim updates will only be available on my Support Forum.

For changes since the previous version, please review the History and Changes documents if provided

<u>Contents</u>

Introduction: what is FSUIPC?

FSUIPC7 is a new version of FSUIPC built as a 64-bit application especially tailored to run with Microsft's Flight Simulator version 11 (released in 2020) [known throughout this document as MSFS]. The original version of FSUIPC dates back as far as FS98 and was built using hooks and direct access into the FS code—FSUIPC7 is not like that, but depends upon facilities in "SimConnect".

The function of FSUIPC, now that developers have an official Microsoft interface they can use, is still two fold, as it has been over the many years. It is just that one side of it may see less application as time goes on. The two aspects are:

1. The provision of an interface to MSFS for application programs and add-on aircraft which is compatible with that provided by previous FSUIPC versions for FS2000, FS2002, FS2004, FSX, FSX-SE and P3Dv1-5 (and also, if you go back to FS6IPC days, FS98). The advantage of this is that programs already working on FS2004 (say) should, if they use FSUIPC, work immediately with MSFS .

    In time I would expect newer versions of FSUIPC-using application, and, indeed, complete new applications, to arise that interface to SimConnect directly. Hence, this part of FSUIPC is unlikely to see a great deal of new development (additional offsets and the like)—at least certainly not as much as has occurred in the past.

    This aspect of FSUIPC is available to applications irrespective of the user's status. In other words, user purchase and registration of FSUIPC has no bearing on the local MSFS PC support of FSUIPC-interfacing programs (*support for Networked clients of WideFS does however still need purchase of the WideFS key*).

2. The main part of FSUIPC, and the one which actually justifies it as a payware add-on, are the user features which is the subject of this document. It is these features you get if you purchase a Key to register. As well as those facilities familiar to previous users, for programming buttons, axes and performing more precise joystick calibrations, the functions of the original AutoSave and GPSout modules are now part of the benefits of subscribing to FSUIPC, and are built in facilities with on-line option setting.

Note that some of the main features of previous versions of FSUIPC, such as weather reading (and manipulation) and AI traffic control (the traffic limiter and zappimg facilities) are currently not available due to lack of functionality in the current MSFS SDK.

Paying for FSUIPC Registration

Why should I pay for FSUIPC?

The short answer is that you do not have to. The registration by users of their copy of FSUIPC is not compulsory. Here are the pros and cons:

What you get if you don't pay:

*   The ability to run all FSUIPC-dependent application programs.

What you get if you pay:

*   Facilities for joystick calibration, some fancy assignment capabilities, and button programming, all with aircraft-specific assignment capabilities so that you can do different things with the same controls on different aircraft.

*   Key press programming facilities, also aircraft-specific if desired.

*   A few frills like facilities to load Flights or Plans from *any* folder in reach of Explorer, keeping the MSFS time in sync with the system clock, pausing automatically after the flight is restored following a crash, extending battery life, throttle syncing, allowing intervening actions during pushback without losing direction selection, and others.

*   AutoSave options—to save a Flight regularly, as a background task so you can resume from a few minutes before in the event of a crash or a need to reproduce some problem.

*   GPSout facilities: optional output of NMEA or AV400 sentences to mapping programs either via a serial port (COM) connection, or via WideFS.

*   Full adaptability in myriads of ways via Macro files and Lua plug-ins. Lua is a very capable scripting language which is well supported in a Registered install of FSUIPC. Many things can be done very efficiently with Macros and Plug-ins which would otherwise necessitate full external application programs to achieve.

- Support (via the Support Forum) for any queries or problems arising from the use of FSUIPC, or identifiable as FSUIPC even when using an application. I would still expect the application developer to be the prime support for his own application, and it is easier for him to sort out FSUIPC interfacing problems with me rather than between the user and me, if you see what I mean.

How do I pay?

There is only one place for payment and registration, and that is via SimMarket, at:

<div align="center">

FSUIPC7 purchase at SimMarket

</div>

A word about **WideFS7:** this is an extra facility within FSUIPC, in the sense that the FSUIPC module contains the equivalent of "WideServer", the Server part of the WideFS package. The Client part is the same as before, and downloadable separately in the usual WideFS.ZIP package. Just use the WideClient.EXE from the latest WideFS.ZIP but not the WideServer.DLL. The full WideFS documentation is also only in the WideFS.ZIP package.

If you are an existing WideFS7 user, you don't need to buy it again – use your existing Key for that. Even the name and email from that registration can be used if different.

Note that the WideFS part of FSUIPC has its own Logging, but its settings are in a section of the FSUIPC7.INI file. Once you registered WideFS7 through the FSUIPC7 installer (as described in the Installation document) you will be able to enable and disable it whilst running MSFS. Each time you re-enable it the parameters from the INI file are re-read, so you can actually change things there without reloading MSFS.

For details of the different ways of paying at SimMarket please see
<div align="center">

http://secure.simmarket.com/paymentoptions.php.
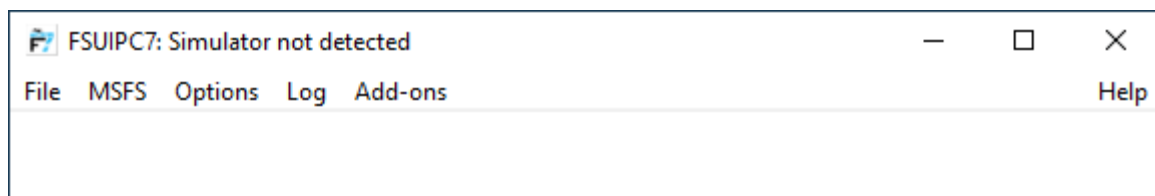
</div>

Running FSUIPC7

Unlike previous versions of FSUIPC, FSUIPC7 is an executable application and not a module loaded by the FS.
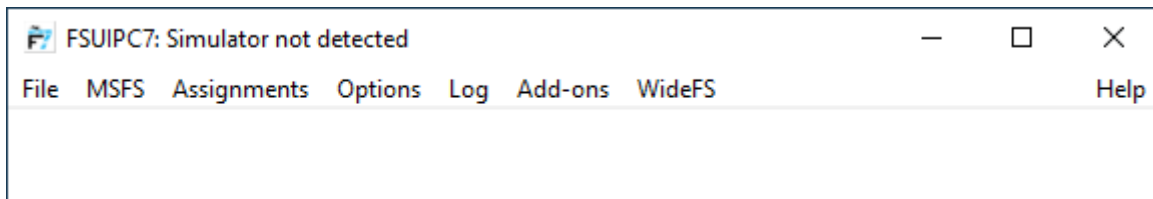
However, FSUIPC7 is now (from v7.0.4 onwards) automatically started when you start MSFS. You can still start FSUIPC7 manually if you so wish, by simply running (i.e. double click) the FSUIPC7.exe. You can do this with or without MSFS running. FSUIPC7 will automatically connect to the MSFS (in its default configuration – see later) when started.

When you start FSUIPC7, you will see a splash screen indicating that FSUIPC is running. The splash screen will disappear automatically after a second or two. FSUIPC7 will then sit as an icon in your system tray. To see the main window, either double-click the tray icon, select the **Show...** option from the icon right-click menu, or double click the icon. You can also use the default hotkey, **Alt + F**, to open/close FSUIPC7. Note that this hot key can also be changed (see **FSUIPC7 Hot Key** section on next page).

The following main window will be displayed for unregistered versions:

F7 FSUIPC7: Simulator not detected	—	□	×
File MSFS Options Log Add-ons			Help

and for registered versions you will see the following:

```
FSUIPC7: Simulator not detected                    —    □    ×

File  MSFS  Assignments  Options  Log  Add-ons  WideFS        Help
```

Note that the **WideFS** menu bar option will only be displayed if you registered your WideFS license during the installation process, and the **Add-ons** menu bar option will only be displayed if you have a compatible add-on installed (e.g. the WebSocket server, PFCcom64 driver or the FSUIPC7 WASM module). Both of these menus can be present in the registered and unregistered versions of FSUIPC.

If, during installation, you opted to install the desktop link to start MSFS and FSUIPC7, then you can use this to start MSFS. This shortcut displays a splash screen for 30 seconds while MSFS is initialising.

FSUIPC7 Hot Key

You can use a "hot key" to show and hide FSUIPC7's main window at any time, regardless of which application has the current focus. This hot key is registered when you start FSUIPC7 and is determined by the ini parameter **InvokeFSUIPCOptionsKey**. The default hot key assigned is **Alt+F**, by the following line in the *[General]* section of your FSUIPC7.ini file:

```
        InvokeFSUIPCOptionsKey=70,12
```

where in the above example, 70 is the F key (0x46) and 12 is the modifier (which is
actually a hex value, 0x12 which is VK_MENU, i.e. the Alt key).
For a list of the key codes and modifier codes, please see the windows Virtual-Key Codes list, available at:
        https://docs.microsoft.com/en-us/windows/win32/inputdev/virtual-key-codes
Note that the main key is specified as an integer value, whereas the modifier is given as a hex value but not preceded by an *x*.

For no hot key, you can use
```
        InvokeFSUIPCOptionsKey=No
```

FSUIPC7 must be restarted for any changes with this ini parameter to take affect.

De-Installation

If you ever want to remove FSUIPC7 from your system, just run the uninstaller (**uninstallFSUIPC7.exe** file) file from the FSUIPC7 installation folder, or by using the standard Windows Apps uninstall panel. This will remove that installation of FSUIPC7 from your system, including all files installed with FSUIPC. However, your FSUIPC7.INI and FSUIPC7.KEY files, as well as any lua or macro files will be left intact. You may wish to manually remove these if you do not plan on re-installing FSUIPC7.

Finding and Editing files relating to FSUIPC and WideFS

Normally, when asked to check, post, edit or otherwise handle a file in a Windows folder, you will use Windows Explorer to find it and select it. There are some things you need to know here, because Microsoft, in its wisdom, has decided that it should hide some important parts of filename from you, by default.

I recommend you change this nasty habit. Open Explorer and find the Tools-Folder Options selection. Then select the View tab. Some way down the displayed list, find and uncheck "Hide extensions for known file types". After you do this, INI files will show as ".ini" and not just as "configuration settings", and LOG files will show as .log and not just as "text file".

All FSUIPC related files, apart from the provided documentation, are contained within the FSUIPC installation folder, which is the one selected when you ran the FSUIPC Installer. You can easily see its contents whilst running MSFS by selecting the **Open Installation Folder...** option from the FSUIPC **File** menu. This opens Windows Explorer with the folder contents shown.

Note that the FSUIPC documentation is now installed in a subfolder of your windows **Documents** folder, called **FSUIPC7.**

FSUIPC related files like .INI, .LOG, .KEY, .MCRO and .Lua are plain text files. When viewing them, editing them or printing them, use a basic text editor like Notepad, *not* WordPad or any other word processor. The special Joystick device scanning log, FSUIPC7.Joyscan.csv can be viewed in the same way, but if double-clicked will invoke a spreadsheet program if one is installed – this displays it in a columnar format.

MOST OF THE REST OF THIS DOCUMENT IS VALID ONLY FOR REGISTERED COPIES OF FSUIPC

The only facilities available to unregistered users are the Logging and Monitoring facilities (described later)

Setting FSUIPC options

Options for registered installations of FSUIPC are controlled by parameters in a file called FSUIPC7.INI. This is not supplied with FSUIPC, but it is generated the first time you run FSUIPC7. It sits with FSUIPC in the installation folder. Most of the options of general interest are accessible whilst running FSUIPC7. This is by far the best and easiest way to access the more popular options. There are still some you can only access by editing the FSUIPC7.INI file, but most of these are not as useful to most users and are mostly only described in the Advanced User's Guide. The main exception is the facility for running additional programs automatically when running FSUIPC7.

You don't need to understand all parts of the Options and Settings to use FSUIPC. But if you want to get maximum benefit it would be worth making a little effort. You can experiment anyway without doing any real harm, and you can always click **Reset to Defaults** (located in the **Options** menu) at any time, and start again. Note that doing this does not affect assignments and calibrations, only the assorted options elsewhere.

IMPORTANT – please read!!!

Although FSUIPC7 should run reasonably well without any manual configuration, there can be connection issues if you have a slow start-up time   (i.e. the time taken between starting MSFS and arriving at the main menu), usually due to having a lot of add-ons installed in your Community folder. Since FSUIPC7 version 7.4.12, FSUIPC7 will monitor for any such issues and adjust some start-up parameters appropriately. This parameters will take affect the next time FSUIPC7 is auto-started and you do not need to take any action. However, if you have a very long start-up time (i.e. time taken from starting MSFS to it arriving in the main menu takes longer than 5 minutes or so) then it may be a good idea to restart MSFS for these new parameters to take affect.

More on auto-tuning can be found in the Advanced User manual. It is also possible to manually set these parameters – see the following FAQ entry on how to do this:

https://forum.simflight.com/topic/98307-fsuipc7-configuring-connection-parameters-for-optimal-use-and-to-prevent-simconnect-issues/

POWER USERS note: if you have a lot of investment in your Buttons, Keys, Axes or Joystick Calibrations via FSUIPC version 4, 5 or 6, then you will want to retain most, if not all, of that work. This is okay—those parameter sections are identical in FSUIPC, though many [General] section parameters no longer apply. These won't be deleted, so to create a tidy file, use an editor to copy just the [Buttons …], [Keys …], [Axes …] and [JoystickCalibrations …] sections over from your FSUIPC4/5/6.INI to your new FSUIPC7.INI file. After that, check they all work and possibly re-assign those which have changed.

Main Window Overview

Currently FSUIPC7 consists of a main window containing a menu bar only. This menu bar gives access to all facilities provided by FSUIPC7 (apart from a few more advanced facilities where you will need to edit the FSUIPC7.ini file directly).

NB: I will add a facility to use the main window to also display a user-defined *button style* interface (as provided by WideClient), but this is currently still under development.

Title Bar

The FSUIPC7 title bar will always contain text showing the current status of the simulator ('Simulator not detected' or 'Simulator is available') and FSUIPC's connection status if the simulator is running ('Not connected', 'Trying to connect...' or 'Connected').

File Menu
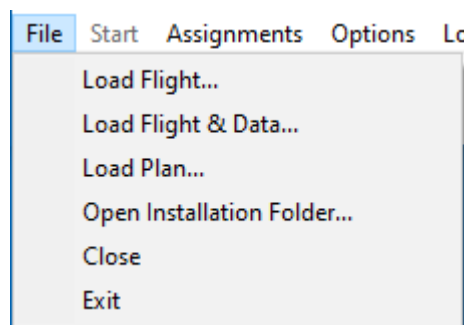
The **File** menu contains the following options:

- **Load Flight...**
  This option allows loading of Flights (.FLT files) from anywhere within reach of a standard Explorer file selection dialogue. Not that in previous versions of FSUIPC, this menu entry was added to the flight Simulator's **Add-on** menu.
  This option is only available in a registered version of FSUIPC7. It is only available (i.e. active) when connected to the FS.

- **Load Flight & Data...**
  This option allows loading of Flights and FSUIPC offset data (when saved/available) from anywhere within reach of a standard Explorer file selection dialogue.
  This option is only available in a registered version of FSUIPC7. It is only available (i.e. active) when connected to the FS.

- **Load Plan...**
  This option allows loading of Plans (.PLN files) from anywhere within reach of a standard Explorer file selection dialogue. Not that in previous versions of FSUIPC, this menu entry was added to the flight Sumulator's **Add-on** menu.
  This option is only available in a registered version of FSUIPC7. It is only available (i.e. active) when connected to the FS.

- **Open Installation Folder...**
  This option opens a standard Explorer file selection dialogue on the folder where FSUIPC7 was installed (and where your FSUIPC7.log, FSUIPC7.ini and FSUIPC7.JoyScan.csv files are located).

- **Close**
  This option closes the FSUIPC7main window. It does not close FSUIPC7 completely – it will still be available from the system tray.

- **Exit**
  This option exits FSUIPC7.

MSFS Menu
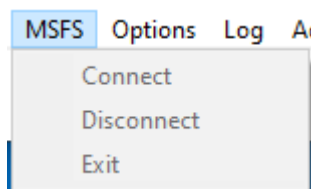
The **MSFS** menu contains options to **Connect**, **Disconnect** and **Exit** MSFS. The options available depend on the current state of MSFS and whether FSUIPC7 has an active connection. Unavailable options are present but disabled.

Note that to use the **Save Previous Flight** functionality (which is active by default), then the **Exit** option in this menu must be used. If you exit MSFS from within MSFS itself, a **Previous Flight** file will not be saved.

Note also that if you quit MSFS with this **Exit** menu and you also have the **Exit with FS** option set (which is the default), then FSUIPC7 will also exit once MSFS has closed.

Assignments Menu
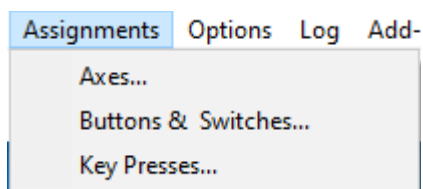
This menu is only available to registered/licensed versions of FSUIPC7, and contains access to the assignment features of FSUIPC:

- **Axes...** : use to assign your joystick device axes to MSFS nad FSUIPC's axes controls, and also to calibrate axes assigned both in FSUIPC7 and in MSFS
- **Buttons & Switches...**: use to assign your joystick device' buttons/switches to either MSFS or FSUIPC added controls, or to send key presses to MSFS
- **Key Presses**: use to assign your keyboard keys to MSFS or FSUIPC added controlled

All these options are described in detail in the following sections.

Options Menu
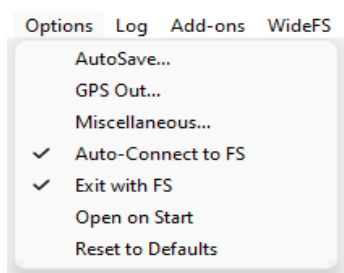
This menu provides access to the following features:

- **AutoSave...**
  Provides access to a UI panel to configure the **AutoSave** functionality. This option is only available to registered versions of FSUIPC7. The **AutoSave** panel is described in the next section.

- **GPS Out...**
  Provides access to a UI panel to configure the **GPS Out** facilities provided by FSUIPC, which allow MSFS to

9

look like a GPS device emitting positional and other interesting information to any program (or device) equipped to have a real GPS connected via the standard type of NMEA serial port connection.
This option is only available to registered versions of FSUIPC7and is described in detail in the next section.

- **Miscellaneous...**
  Provides access to various assorted facilities that don't fit in else where. This option is only available to registered versions of FSUIPC7 and is described in detail in a later section.

- **Auto-Connect to FS**
  A checkbox that when selected (i.e. checked) will instruct FSUIPC7 to automatically connect to MSFS when it is started. If this is checked, then the **Start** menu entry will be disabled.

- **Exit with FS**
  A checkbox that when selected (i.e. checked) will instruct FSUIPC7 to automatically exit when MSFS exits. Note that this option is only available when manually starting FSUIPC7. If FSUIPC7 is auto-started by MSFS, then this option is unavailable and FSUIPC7 will automatically exit with the FS.

- **Open on Start**
  A checkbox that when selected (i.e. checked) will keep the FSUIPC7 main window open when FSUIPC7 is started and not in the system tray.

- **Reset to Defaults**
  This changes all FSUIPC7 options back to there default values. This does not affect your assignments or calibration, (and is equivalent to the '**Normal Defaults**' option in previous FSUIPC versions).

Log Menu
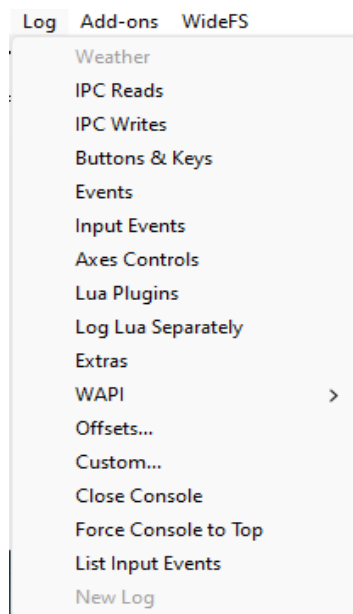
The **Log** menu allows access to FSUIPC's logging facilities. You shouldn't need to use these unless you run into problems with an application and want to help the author sort it out. The text files produced by the logging facility are called FSUIPC7.LOG (or FSUIPC7.n.LOG files where 'n' is a sequence number) and are located in the installation folder.

The following logging options are available:

- **Weather**: to log a complete history of weather changes. However, please note, as there are currently no facilities provided by MSFS to read or update the weather, this option has been disabled.

- IPC **Reads**: to log reading of offset data from FSUIPC external/3rd part clients

- **IPC Writes**: to log writes to offsets from FSUIPC external/3rd part clients

- **Buttons & Keys**: to log button and key events. This will be primarily of use to those of you who make good use of the comprehensive key and (especially) button programming facilities available in FSUIPC, especially

when editing the FSUIPC7.INI file for more powerful and conditional options. The logging will help you understand exactly what is going on.

- **Events**: to log events or controls (all MSFS controls other than those from Axes or Input Events). Use if you are interested in seeing the results of pressing buttons or keys, to see how MSFS is interpreting these actions. Also, many of the more complex Panels do send a lot of controls very frequently, for reasons which, when you look at what is sent, seem pretty obscure. You can prevent such controls from being logged using the **DontLogThese** ini parameter (see Advanced User guide for details).

- **Input Events**: to log Input Events. Similar to the Events option (above) but specific to Input Events.

- **Axis Controls**:to log all events related to axes controls

- **Lua Plugins**: to enable logging of Lua plug-in operations. *More information about Lua plug-ins is provided in separate documents you will find installed in your Documents "FSUIPC7" folder.*

- **Log Lua Separately**: checking this option will log lua operations (see previous option) in its own separate log file. These are cumulative (each time the same plug-in is run it adds new entries to any existing log file it has), so you don't get a huge proliferation of files. If you want the file to be overwritten on each run of lua scrupts, you can set the ini parameter **LogLuaNew=Yes** (see Advanced User guide for details).

- **Extras**: check for additional 'extra' logging. This logs things not covered by the other options (such as the threads used and the thread of each message). You would normally activate this when asked to provide further information for a support request.

- **WAPI**: a sub-menu to set the WAPI log level.

- **Offsets...**: this option allows logging of specific offsets. This is a more technical developer-oriented logging facility, and is described in the **Advanced Users** document.

- **Custom...**: this is for more detailed custom logging and should be used as and when instructed by us to help track down issues raised in support requests.

- **Open Console**: this option allows you to see what is being written to the log in real time - provided you run MSFS in Windowed mode. The log is updated in a "console window", which you can size to suit your needs. Note that when selected and the console is open, this menu entry will change to **Close Console**.

  WARNING: Do not attempt to close the console window directly! To remove it, just go back to the Log options and select 'Close Console'.

- **Force Console to Top**: when selected/checked, this will attempt to keep the console window on top of other windows.

- **List Input Events**: use this to list the Input Events available for the current aircraft. Note that Input Events are ONLY available when an aircraft is loaded, and not all aircraft support such Events (the list will be empty for such aircraft).

- **New Log**: this option is now disabled (by default), but can be enabled via the **EnableNewLog** ini file option. When enabled, this will close the current log file and start a new one. The current (latest) log file is always called "FSUIPC7.LOG", previous ones being renamed "FSUIPC7.n.LOG" where 'n' is a sequence number. Although this will have a repeat of some of the initialisation data as the original, starting a New Log is not advised if you intend to supply the log for support to help with a problem, as some vital information will by then be missing. I have received too many such logs (continuation log files) for support issues, whuch is why this is now disabled by default.
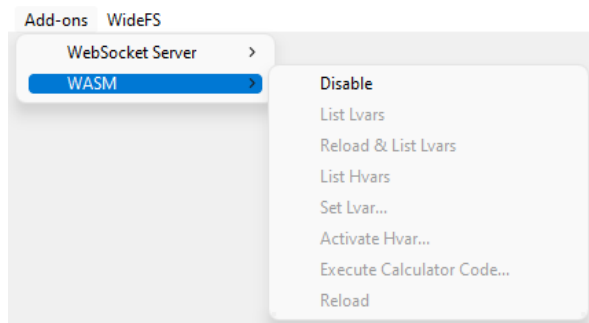
Add-ons Menu
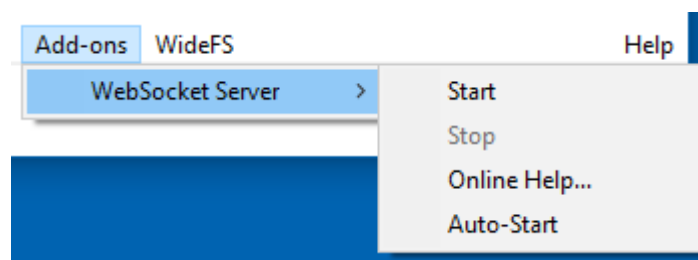
The **Add-ons** menu item will only be displayed when one or more compatible add-ons are found when FSUIPC7 is started. This menu can include (amongst others) a menu entry to access the PFCcom64 user interface (when the PFCcom64.dll is located in the FSUIPC7 installation folder), and a menu entry for the WebSocket server (provided by Paul Henty and included in the FSUIPC7 installer).

The menu for the WebSocket server contains menu entires to start and stop the server, visit the online help page, and a checkbox to auto-start the server when FSUIPC7 is started.

Add-ons WASM Menu

If you opted to install the FSUIPC WASM module during installation, then you will see a **WASM** menu under the **Add-ons** menu. The WASM will be automatically enabled if installed (as of version 7.3.7) – you can **Disable** the WASM from thus menu if you do not want to use the facilities provided by the WASM.

Note: you MUST have the FSUIPC7 WASM module installed and enabled in FSUIPC7 if you wish to use either lvarsm hvars and calculator code, including presets (such as those provided by MobiFlight).

The remaining menu items will only be enabled once you have an aircraft loaded and ready-to-fly, and provided access to test functions for helping to discover and use lvars and hvars:

   List Lvars: this lists the lvars currently available lvars to FSUIPC in the main window as well as the log file. Note that this lists the lvars currently known by FSUIPC, and not necessarily all that may have been created by the aircraft.

   Reload & List Lvars: as the above **List Lvars** option but a reload command is first issued so that all lvars currently created by the aircraft will be listed.

   List Hvars: this lists the currently available hvars to the main FSUIPC window as well as the log file. Note that there is no automatic discovery of lvars – you need to provide an aircraft-specific hvar file to make these known to FSUIPC – see the the FSUIPC WASM section in the Advanced User guide for more details.

   Set Lvar...: shows a panel where you can select and set a value for an lvar

   Set Hvar...: shows a panel where you can select and activate an hvar

   Execute Calculator Code...: allows any code to be executed in the FS using the execute_calculator_code function

   Reload: instructs FSUIPC to tell the WASM to re-scan for lvars and reload the hvar file for the current aircraft

For further information on the WASM module, please see the Advanced User guide.

WideFS menu
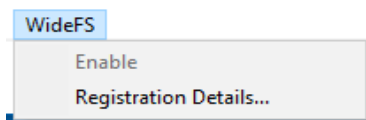
The **WideFS** menu will only be available if you have registered your WideFS license during the FSUIPC7 installation process, It contains a menu entry to Enable or Disable WideServer (the WideFS component embedded into FSUIPC7), as well as a menu item to view your WideFS registration details.

Note that WideServer will be enabled/started automatically once connected to MSFS and 'ready to fly'. You can only manually enable the WideServer component if it has been manually disabled.

When WideFS is registered, its status will be displayed in the title bar of MSFS itself.

Help menu
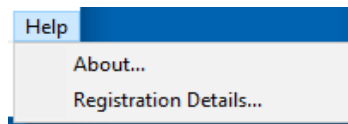
The **Help** menu contains two menu items:

- **About...**: this will display the version and build number of your installed FSUIPC7, together with your registration status.

- **Registration Details...**: this displays your registration status and details.

AutoSave

AutoSave saves flights ("FLT", "FLT.FSSAVE" and "SPB" files, plus optional "IPCBIN" files) at regular intervals, so you can retry that crashed landing or quickly recover from other problems.

By default, when enabled, AutoSave saves a flight every 60 seconds, and keeps the last 10 files. These files are placed in the standard user data folder selected by MSFS. They are named **AutoSave day hhmmss,** where 'day' is day of the week. The time is the computer's local system time, NOT the one on the MS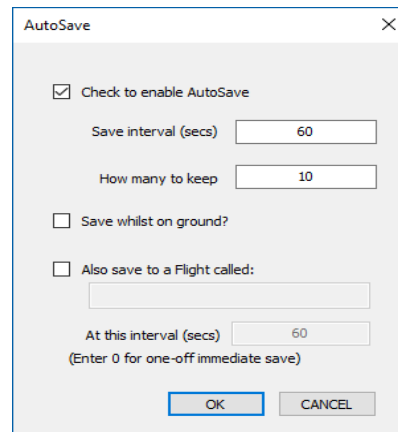FS clock. This format allows you to easily select a suitable file to restart from. With 10 files at one minute intervals you have the last 10 minutes covered.

Note that AutoSave does not normally save these files whilst the aircraft is on the ground (EXCEPT maybe in slew mode if you started this mode in the air). This allows you to take a breather and recoup after a crash! If you really want it to save files when on the ground, you can check that option (see the picture above). They still only save when the aircraft is moving (or moved).

The files aren't saved whilst MSFS is paused, either, to avoid overwriting all those useful ones with identical copies of a paused simulator!

You can also select an option to save an additional file set which has a fixed name and which is saved regardless of MSFS's mode or state. For this check the "Also save .." option and provide a name, such as "My Restart". The default interval is 60 seconds, but you can edit that too. The idea here is that you can set "My Restart" or whatever as your default flight, so after an program or hardware crash, or power outage, you can simply load up MSFS and continue from a few seconds before the problem.

If you want only this behaviour, and no timed files, disable the main AutoSave option but still enable the "Also save" one . Then only the "also file" will operate.

Note that there is also an FSUIPC-added control called **'auto-save toggle'** that acts in addition to the AutoSave facility that allows this feature to be dynamically controlled (i.e. enabled/disabled) but only once the AutoSave facility has been activated.

Finally, note also that complex add-ons may save their own data in files in other places. each time an AutoSave occurs. These will accumulate and NOT be deleted by AutoSave unless you expressly request this to happen. For that you will need to edit the FSUIPC7.INI file. A special section on this is provided in the Advanced User's document. Please check its contents list.

GPS Out
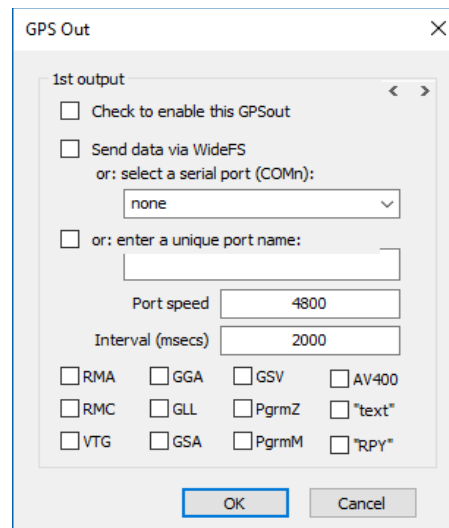
GPS Out provides the facilities to make MSFS look like a GPS device emitting positional and other interesting information to any program (or device) equipped to have a real GPS connected via the standard type of NMEA serial port connection. By this method you can have moving maps and other guidance software, normally running on a separate PC, following your flights in MSFS.

You will need to connect your flying PC to your moving map PC with a serial "null modem" cable (i.e. send and receive crossed over), and configure your map program accordingly. Alternatively, if you use WideFS you can tell GPS Out to send the GPS data to the Client PCs. You will need to configure those in the WideClient.ini files. A virtual serial port program (freeware) from MixW is available from my Support Forum. You can use this to generate a pair of linked virtual serial ports to connect between GPS Out and a receiving program on the same PC, or between WideClient and the receiving program on a client PC. Either way, no cables are needed with this method.

[*Note that the MixW virtual port program might not work on all systems. It does not work on Vista nor on 64-bit operating systems.An alternative is the one from Eterlogic. Their 32-bit version is free, and the 64-bit version is a reasonable $US 25*].

FSUIPC can simultaneously support *two* devices, on separate ports, or one WideFS connection and one local port. The second connection is setup in the same options screen: just use the little scroll-bar type arrows in the top right corner of the GPS Out option page to scroll between the two outputs. All of the parameters can be different between the two except for the Interval. Whatever interval you set for one output will also apply to the other. This is because FSUIPC needs to keep the data synchronised.

GPS Out *may* not support all other moving map and atlas programs, but it has been tested with:

> AutoRoute 2001: use GGA sentences
> AutoRoute Express 98: use GGA
> Anywhere Map, by Control Vision: use RMC, GGA and GSA
> MapPoint 2001: use GGA or RMC
> Map'n'Go: use RMC
> Jeppesen FliteMap: use RMC and PgrmZ with a Garmin NMEA setting,
>   or (for versions 8 & 9) RMC, GGA and GSA.

It is also known to work with real Garmin GPS units which support input in the "Series 400 Aviation" format (this may also be known as ARNAV, or KING format). For this you should set just AV400 as the only sentence, and the speed is normally 9600 in this format, but it may be adjustable.

I've received confirmation that the "Aviation In" mode works with the AV400 protocol, with the Garmin GPS III Pilot and the GPS x96 series aviation models. The GPS has to be in simulator mode, and then shows the correct track and vertical data as well as position and speed. One Garmin 496 user provided this information:

Garmin 496

First of all, you will have to buy the optional cable: Part Number: 010-10514-00 Description: "PC interface cable (RS232 serial port connector) for Garmin 496". The 496 comes with a USB cable, but I could not make it work. The GPSout settings are:

Sentences=AV400
Interval=1500
Port=COMn (whatever COM port you connected it to)
Speed=9600

In the Garmin go to 'Main menu' then 'Set up', and then to the 'Interface' tab. For 'Serial data format' select 'Aviation in'. Don't forget to put the GPS in 'Simulation mode' in the 'system mode' settings.

Use with PDAs may also be possible—in fact I've recently succeeded with this myself, using PocketFMS on an Ipaq Pocket PC.

For those who have an older Palm PDA, one with serial PC connection rather than the now ubiquitous USB, you can get software that will turn the PDA into a moving map device, working with GPSout, from www.gpspilot.com. Look for "Atlas" or "Fly" applications. For these you need to run the link at 4800 baud and sentences RMC and GGA should be specified for GPSout.

For the IPAQ, try "PocketFms" (pocketfms.com) like me, formerly a Donationware program but now apparently quite expensive. This is reported to run well in an Ipaq 3670 with the following settings:

Sentences: RMC, PgrmZ and GGA
Interval: 100
Speed: 52600

Though I use it on my Ipaq with

Sentences: RMC, GGA, GSA and VTG
Interval: 500
Speed: 38400

For PDAs on a USB connection I've received the following information which may help those with only USB connections on their Pocket PC:

- Check the unique port name option, and enter the name as \\.\WCEUSBSH001. If there are more than one serial-type USB devices, I think they are numbered 002, 003 etc.

- Kill ActiveSync on both the PC (Connection Settings: uncheck 'Allow USB connections'), and on the PocketPC, if necessary, then fire up your PocketPC GPS software and select COM9/USB as your GPS receiver from within the PocketPC software. Hopefully everything should then be working together. In the case of PocketFMS on my Ipaq I just left it to find the COM port and speed automatically, and it did so without any problem.

This has been tried (with older versions of GPSout) on a Medion PDA (rebranded Mitac Mio 168) and an HP iPAQ without problems, and at baud rates up to 19200 using both the Medion Navigator (for cars) software and the PocketFMS software. Apparently both worked particularly well.

GPS Out can also be used with some GPS emulator software—the Lowrance 500 and 1000 emulators from www.lowrance.com. The following settings are reported to be okay:

Sentences: RMA, RMC, GGA, GSA
Speed: 4800

though I think "RMA" should be left off as it is a total subset of RMC.

Another hint has been published for these emulators: you can set it up to accept NMEA in, but don't do this, This will probably work if you have a real Lowrance 500 or 1000, but just start the emulator and it should get a position.

Some notes on the NMEA sentences:

RMC:	best for FliteMap, provides course and ground speed, but no altitude
RMA:	similar
PgrmZ:	Garmin NMEA 0183 extension for Altitude only: use with RMC
GLL:	used by Microsoft Autoroute 2001 but GGA works better
VTG:	usually used with GLL to provide course and ground speed data
GGA:	also works with FliteMap and others, and provides altitude, but not course and speed
GSA:	provides no useful information from GPSout, but appears to be needed by some applications (such as "Anywhere") in order for it to be able to use the altitude in the GGA sentence.
GSV:	similarly, nothing useful from GPSout but fixed satellite data for some programs which seem to need it.

On my part, GPS Out has only been thoroughly tested with FliteMap, Memory Map and PocketFMS.

With version 7 of FliteMap, in the LORAN/GPS connection, select "NMEA 0183 RMC (9600)", or find a Garmin NMEA setting. In versions 8 and 9 of Flitemap there are far more options, under "Connections–External Device Setup". Those I now find work best are the Garmin NMEA settings. All the values are recognised if you send RMC and PGRMZ sentences, or RMC, GGA and GSA, and you can choose a really decent speed like 19200 or better, as well.

NOTE for Apple IOS devices

These devices (iPhone or iPad), when used with suitable connectors, can receive and understand some of the NMEA sentences produced by GPS Out, but they need this parameter in the [GPSout] section of the FSUIPC7.INI file:

SimModeIndicator=Yes

Miscellaneous

This is called "Miscellaneous" because at present it contains several options that simply don't really fit anywhere else.
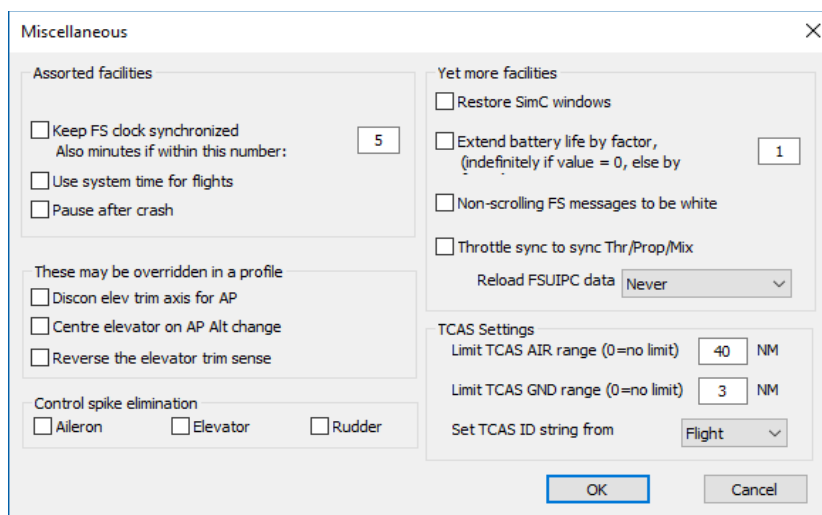
Let's look at the left hand column first:

- **Keep FS clock synchronised**: This facility compensates for the occasional phenomenon of the FS losing time. This was an issue with P3D, but should (hopefully) not be needed with MSFS. It just synchronises the seconds values (and optionally the minutes) with that of your PC's system clock. Provided you always keep to the normal 1x simulation speed this should stop MSFS time running slower and slower compared to real time (or, rather, PC time, which may not be quite the same as real time either). At other than 1x simulation speeds the facility is temporarily disabled to avoid odd things happening on the MSFS clock.

  The time is only adjusted on the minute (i.e. seconds = 0), because MSFS only provides a seconds zeroing facility. Further, it won't attempt to do this if the time difference is greater than the value shown (default 5 minutes). If you set that value to 59 minutes or more you effectively synchronise both minutes and seconds. BUT please note, any time adjustment of more than a minute or so will cause MSFS to reload textures and you will get the occasional progress bar. This often happens, for example, after exiting from the menu system, whilst MSFS's clock has not been updating.

- **Use System Time for flights:** This option when enabled will automatically set the sim time to the current PC system time (and date) each time a flight is loaded.

- **Pause after crash**: This option merely makes MSFS enter 'paused' mode when it has reloaded a flight after a crash. I found this very useful in some situations where the reloaded flight places the aircraft part way into an airport building instead of correctly on the ramp. I think this also happens with some of the "Go To Airport" parking selections. At least with it paused you are able to select slew, unpause, and move the aircraft to a safe location.

- **Disconnect elevator trim axis for A/P**: This option is aimed at those who are using an analogue trim axis instead of the more usual trim up/down buttons, wheels or levers. When the MSFS autopilot has control vertically (such as in altitude hold or glideslope acquired modes), it disconnects the elevator axis so that joystick inputs don't upset

things, but doesn't do the same with the trim axis. This FSUIPC option fixes that. If it is enabled it also disconnects the trim axis if an add-on or external autopilot disconnects the elevator axis via the special facilities in the FSUIPC interface (ref: offset 310A for programmers among you).

- **Centre elevator on AP Alt change** makes FSUIPC centre (zero) the elevator input when the Autopilot ALT hold mode is enabled or disabled. This is mainly of use for keyboard flyers, where the original keyboard-set value for the elevator is retained, ineffectively, whilst the A/P controls the vertical modes, but which then undesirably re-asserts itself when the A/P control is relinquished

- **Reverse elevator trim sense**: This provides corrective action for some panels which try to control the aircraft pitch with the trim axis on a setup where this has been reversed.

- **Control Spike Elimination** provides controls to ignore signals specifying maximum possible deflection on rudder, aileron or elevator.

Please note that if you fly with any of these options set you should also consider calibrating your controls in FSUIPC (see the Joysticks section, later), setting a "dead zone" at either extremes of the travel. If you don't do this you may find it impossible to obtain maximum deflection. Calibrate first, with the "spike removal" option off, then test it with the option back on.

Facilities listed in the right-hand column are:

- **Restore SimC Windows:** FSUIPC reads the position and size of all three SimConnect window types (Text line, Message Window and Menu) and will save them in the FSUIPC7.INI file. However, this data is not used to restore those positions and sizes unless this option is enabled. Even then, it only does this in full screen more (trying to do it in Windowed mode creates odd and variable problems with those Windows).

  However, please note that SimConnect text windows are not currently functioning correctly in MSFS: these functions have now been deprecated in the SDK provided by Asobo. I am investigating a replacement for this.

- **Extend battery life** keeps the voltage on the battery above failure point for longer. The battery discharge can be prevented from reducing the voltage too low for a specified multiple of the normal MSFS run-down time. The default is '0'—the battery voltage will be kept up indefinitely. The factor can be 1–999. For example, if the battery normally fails after 90 seconds, then a factor of 40 will keep it going for one hour. A factor of 1 is the same as disabling the facility.

Note that once the battery has discharged MSFS records an electrical failure in any case. FSUIPC cannot re-charge a battery once the voltage has been allowed to drop. MSFS may not let you clear the electrical fault in its menu— you must recover power by either reloading the aircraft or, better, getting an engine started to re-charge the battery.

- **Throttle sync to sync Thr/Prop/Mix:** When selected, this makes the added Throttle Sync controls synchronise the Prop pitch and Mixture settings to the Engine 1 values, as well as the throttles.

- **Reload FSUIPC data:** This operates an option to save application program settings from FSUIPC data "offsets", with saved MSFS Flights, and optionally reload them when flights are reloaded. This can enable some FSUIPC client programs to continue running when reloading and recovering from an aircraft crash or even from an MSFS crash or hang. It can especially apply to WideFS client applications on networked PCs, if they are left running whilst MSFS is reloaded.

There are four variations, selectable in the little drop-down list:

Never: the FSUIPC data is not saved nor reloaded. This is the default, so there's no change for those not interested.

Menu: FSUIPC data is always saved with a flight (an "ipcbin" file is created), but it isn't loaded unless the flight is loaded using the FSUIPC' File menu item "Load Flight & Data".

Auto: FSUIPC data is always saved with a flight (an "ipcbin" file is created), and it is automatically loaded if the flight being loaded is an AutoSaved one (or "AlsoSaved"). Any flight with data can also still be loaded using the FSUIPC-Added menu item "Load Flight+Data".

Always: FSUIPC data is always saved with a flight (an "ipcbin" file is created), and it is always re-loaded when the flight is loaded. Generally this is not recommended, as some of the program data may refer to programs not currently running. No special menu entry is created for this option as it isn't needed.

Note that all of the FSUIPC offsets are saved, in a binary file exactly 65536 bytes in size (the size of the offsets data memory exactly). However, only those pertaining to application options and settings are reloaded—the rest are

re-supplied by SimConnect, and so on, after the flight is fully loaded. The rest of the data may well be useful for diagnostics, however.

- **Limit TCAS AIR range** sets the maximum range at which AI *air* aircraft will be added to the AI traffic tables. If this is set to 0 no limit is applied, but there is then a danger that busy areas will have more aircraft than can be accommodated in FSUIPC's tables (currently these can cope with 96). The default is 40 nm, which should be ample. There is little point in setting it much greater than 80nm as that seems to be the range in which MSFS generates flights in any case. When the tables are full, those aircraft which are furthest away won't be listed, so nearby TCAS warnings should still be good.

- **Limit TCAS GND range** is as above but applies applies to *ground* AI traffic. The default is also 40nm, although a fixed limit of 3nm is applied when the user aircraft is on the ground. When the tables are full, those aircraft which are furthest away won't be listed, so nearby TCAS warnings should still be good.

- **Set TCAS id string from** … allows you to determine how FSUIPC provides A.I. aircraft identification to any program using the TCAS data. Probably not many strict TCAS displays show aircraft ID or type, but the information is available. You can choose from the following ways of distinguishing the aircraft:

  Flight   for airline plus flight number, or tail number, as available (this is the default and recommended)
  Tail   for tail numbers only
  Type   for the ATC aircraft type, generally only the Make, at least for default aircraft
  Title   from the aircraft title (in the .CFG file), truncated to 17 characters
  Type+   for the type as above, truncated if necessary, plus the last 3 characters of the tail number
  Model   for the aircraft model

User profiles for all control settings

Much of the usefulness of FSUIPC these days springs from its very flexible and powerful provisions for assigning and programming control inputs for your aircraft. There are section (each with its own Chapters in this document) for assigning Key Presses, Buttons and Switches, and Joystick Axes, and for calibrating and copying the latter for a variety of purposes.

Part of the power of this comes from the facilities to make such settings specific to an aircraft, or selection of several aircraft. Controls for helicopters are obviously rather different from those for light prop aircraft and those again are different from jets. Even within those categories there can be strong differences, such as between the calibration needs for fast jet fighters and the more lumbering heavy airliners. And, of course, even within the airliner "group" between the users of joysticks (e.g. Airbus) and yokes (e.g. Boeing).

FSUIPC enables this with "Profiles". A "profile" is a user-named set of parameters, applying to Key Presses ("Keys"), Buttons and Switches ("Buttons"), Axis Assignments ("Axes") and Calibrations ("JoystickCalibration"). Think of the profile name as a generic term for the type of aircraft you want to use those settings with.

So, for example, you could have Profiles called "Prop", "Jet", "Helo" with appropriate assignments in each, and then for each aircraft you simply select which profile you wish to use for it.

If you make extensive use of Profiles then you might find managing all the settings easier by using the **UseProfiles=Files** setting, which splits all the settings for each profile into a separate file, one per profile, and puts those in a separate folder on disk. Full details of this facility are provided in a separate document in your FSUIPC documents sub-folder.

In the FSUIPC assignment dialogues (for 'Axes', 'Buttons & Switches' and 'Key Presses'), you will see a "profile specific" checkbox. There will be no other outward difference, until you actually use one of those to select a Profile for the currently loaded aircraft.

When you do you will see something like this:

You can then select Cancel, to change your mind, or New to create a new Profile. If there are already existing profiles you can also select option "New, based on ...", which will lead to another selection where the Profile to be copied can be selected. Both "**New**" and "**New, based on**" will then require you to give it a name, so this pops up:

Think of a suitable name for your profile, type it in and press OK. The name will appear in the title bar, and the currently loaded aircraft will be its first user. Now any assignments you make, here or in any of the Tabs, with the "profile specific" option checked, will be associated with this profile. The name of the active profile will always be shown in the title bar of the options dialogue in these tabs.

Next time you load a new aircraft and elect to use Profile specific options for it, the Profiles you've already created will be available to you to select from, or, again, you can create another. If you use a profile which has already been created, any profile-specific settings will become available to this newly loaded aircraft straight away. FSUIPC will have to load them, and it will tell you it is doing so.

Changing or removing Profile assignments

If you ever want to change the profile assigned to an aircraft, or remove it from such as assignment, it will be necessary to do this outside of MSFS, by editing the FSUIPC configuration settings file (FSUIPC7.INI). Use a text file editor such as Notepad++ (*not* a word processor). You will see that the file is divided into many sections with headings in square brackets, [...]. find the one with the Profile name, as follows:

> [Profile.<name>]

where <name> is the name of the profile. Within that section you will see a list of aircraft names which are associated with that profile. just delete the line containing the aircraft name you wish to withdraw.

Then, on the next MSFS session you will find you can reassign that aircraft or create a new profile for it.

Profile matching on aircraft with different liveries

When an aircraft name is added to a profile, the full aircraft name is used. This can cause issues if you load the same aircraft with a different livery, as the aircraft name may not match completely.

By default, FSUIPC performs substring matching on the aircraft profile name to the loaded aircraft name. To match all aircraft, regardless of the livery used, you can edit the aircraft profile names in your FSUIPC7.ini so that they match every livery.

From FSUIPCv 7.2.15 and onwards, there is also a new way to match a profile to the current aircraft, by using the name of the folder where the aircraft.cfg file is located. To use this profile matching strategy, you need to add the following ini parameter to the [General] section of your FSUIPC7.ini file:

> UseAirLocForProfiles=Yes

Depending upon which livery is loaded when you create the profile, you may also need to edit the profile name in your FSUIPC7.ini to make sure that is also a match for all other versions of the same aircraft.

Keeping track of multiple control devices ("Joy Letters")

The "Joy Letters" facility is now activated by default in FSUIPC7, and so letters will be automatically assigned to your devices. If you wish to assign your own letters, you can change the assigned letter by editing your *FSUIPC7.ini*. It is recommended to do this before you start assigning to the device, otherwise you will also need to change the joyletter used in those assignments. In general you would choose letters suggesting the type of control—J for Joystick, Y for Yoke, T for throttle or Q for Quadrant, G for G-stick, etc.

Devices listed with assignments subsequently removed altogether

This is handled by FSUIPC up to a point. If you have assignments to an alpha ID and that joystick is removed, so no longer being named, the assignments remain included in the INI file but are not loaded into FSUIPC. The JoyNames list will show the letter as an error. For example:

> C=<!!! MISSING JOYSTICK !!!>

It cannot show the missing name as it cannot read the name for a disconnected device, but it does indicate to you that those assignments, the ones in this example to C's buttons and axes, will not operate until the device is reconnected ... and then, of course, it may not be the same one. If you re-connect whilst MSFS is running it may not be seen in any case, but try going to the axes and buttons Tabs in FSUIPC options and using the "reload" button. You may get everything back as before!

If you remove devices because they are no longer used or owned, the letters associated for those will remain unusable for new devices. There are only 24 letters available (A-Z less I and O to avoid ambiguity), so it is then possible for the Auto option to run out of letters. You should therefore consider manually deleting unused entries.

Buttons & Switch Assignments
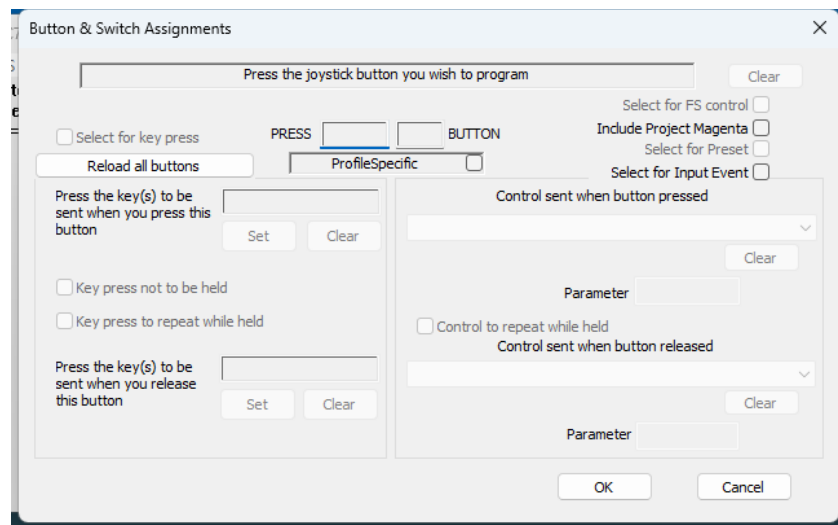
This page provides facilities for programming joystick buttons. Whilst this is particularly suited to GoFlight, VRInsight and PFC users, it is by no means specific, and so can be useful for any joystick system with 'spare' buttons. The only proviso is that, except for the special support for GoFlight, VRInsight and PFC, these are visible through the Windows joystick interface (the Game Controllers applet in the Control Panel), using the standard Windows facilities. FSUIPC can detect up to 128 buttons. The first are numbered 0-31, 32-39 are served for POV buttons, and standard buttons > 31 will be numbered from 132 (i.e. button 32 will be recognised as button 132). Note that Windows Game Controller numbers buttons from 1, so FSUIPC buttons 0-31 are seen by windows as buttons 1-32.

Note that the supported serial port VRInsight devices need special configuring in the FSUIPC7.INI file before you can assign Buttons and Switches in FSUIPC7. This is not hard and step by step instructions are provided for this, but this is outside the scope of this general User Guide, so please refer to the Advanced User's guide where you will find VRInsight mentioned in the Contents list.

The Saitek Pro-Flight Radio, Multi- and Switch panels can be programmed via FSUIPC's "virtual buttons" facility using a neat driver called "SPAD" (Saitek Panels Advanced Driver) written by Massimo ("Maxx"). Download this from http://fstools.weebly.com. There is also a payware version called SPAD.Next which also integrates well with FSUIPC.

You might well ask why this facility is provided here at all when MSFS offers something similar. Well, these are the additional features available in FSUIPC:

- The actions can be programmed differently for different aircraft or profiles. Just check the box "aircraft specific" or "profile specific" (if the profiles option has been pre-selected), and then everything you program will operate for the currently loaded aircraft only. Anything programmed without that checkbox selected will also be available, unless overridden by an aircraft-specific assignment. If a button already has a "profile specific" assignment for the current aircraft, then this will be show regardless of the state of the "profile specific" checkbox, and the checkbox will be automatically checked.

- Buttons can be made to cause key presses. This is useful for accessing those add-ons which do not provide "controls" as such, but do react to keystrokes—for example, advanced panels such as those from PMDG and PSS, external programs such as Radar Contact, and many assorted utilities.

- The range of MSFS controls assignable here exceeds those that can be assigned in the MSFS Assignments dialogue (and the CFG file), and fixed parameter values can also be supplied with them so they can be used to set (fixed) values as well as operate switches.

  However, please note that there are also many controls that you can currently assign in MSFS but not in FSUIPC7, as the controls have not been exposed via the SimConnect interface. To use such a control in FSUIPC7, you can assign the control to a key press in MSFS, and assign the button in FSUIPC7 to that assigned key press.

- Controls aimed at other programs can be included. Currently these include Roger Wilco and Advanced Voice Client (both for push to talk functions), Project Magenta and generally any WideFS client programs via the WideClient "KeySend" facility.

21

- Sequences of actions, whether key presses, or MSFS controls, or a mixture, can be programmed. However, this facility can only be accessed by editing the **FSUIPC7.INI** file, as it would have made the FSUIPC options page too complex for most users. Full details for editing the Button programming in the INI file are provided in the Advanced User's Guide.

- Compound (or conditional) button actions can be programmed—with the action of one or more buttons being determined by the state of one or more other buttons or switches. This can be used, for example, to program a bank of buttons for different things depending on a toggle or multiple position switch, hence multiplying the effective number of buttons available. The same facility can also be used to cope with those types of rotary switches that indicate which direction the spindle is being turned by the phase relationship between two button lines. Again, this more complex facility can only be accessed by editing the **FSUIPC7.INI** file, and is described in the Advanced User's Guide.

- Button actions can be made conditional on values or bits in FSUIPC's offsets, opening up a whole realm of possibilities, such as different things to be done when on the ground as opposed to flying, or with engines running or not, and so on. These also need editing in the INI file and the facilities are described in the Advanced User's Guide.

Serial port PFC equipment controlled by PFCcom64

If you are using PFC equipment driven by my PFCcom64.dll driver, then all the buttons, switches and digital rotary knobs on that equipment will also be visible and programmable here, just as if they were on joystick inputs. They will have Joystick numbers of 16 or higher, above the 0–15 range allowed for normal Game Port or USB joysticks in the Windows joystick interface. If you do program any PFC buttons here, the normal function in PFCcom64.DLL will not be applied. Normal PFC functions will be automatically resumed, however, when you delete the programming in FSUIPC (either by using the Clear button here or by deleting the entries in FSUIPC7.INI).

GoFlight equipment

GoFlight buttons and rotary dials can also be programmed here—but take care that you don't have both Goflight and FSUIPC7 acting on the same buttons as FSUIPC is not able to override the original programming. For GoFlight buttons to be recognised you must have the GoFlight module (**GFDev64.dll**) installed. If you install the latest version of the GoFlight software the correct DLL may be installed automatically for you, but otherwise download it from the Download Links subforum on the FSUIPC Support Forum. Just place a copy into the FSUIPC installation folder, alongside FSUIPC7.exe.

GoFlight buttons will all have high "joystick" numbers in FSUIPC.

Note that GoFlight devices can be connected and disconnected and FSUIPC can handle the changes without reloading, *providing* you go to the Buttons tab and use the "reload" button there. This re-scans all USB joystick connections.

Remote buttons through WideFS

If you are a WideFS user, then with versions 6.22 or later of WideClient.EXE on the Client PCs, FSUIPC will recognise any Windows, EPIC or GoFlight buttons on any of the connected Client PCs. Each Client PC will add 1000 to the "joystick number", so each will still be unique. More details about this feature are provided in the WideFS documentation.

Before attempting to use these facilities, please make sure that the buttons you wish to program are ***not*** already programmed to provide MSFS facilities in MSFS's own assignments. If you want to program them here rather than there you must delete the assignments there, otherwise you will get both actions when you press the buttons. It is recommended that you *create an empty profile* in MSFS for your device before you start assigning in FSUIPC. However, you can mix and match, and have some assignments in MSFS and others in FSUIPC. This is useful as there are many assignable controls in MSFS that are not curently available via FSUIPC as they have not been exposed via the SimConnect API.

Programming the buttons

First, please note that it is always advisable to program buttons when MSFS is running and you have an aircraft loaded and ready-to-fly (i.e. not when in the MSFS main menu system). Also be aware that lvars, hvars and Input Events are only available when MSFS is in this state, as they are aircraft-specific.

Okay, assuming now that you have a 'spare' button to program, and this is definitely not already assigned in the MSFS assignments, you can proceed as follows:

1. First, please make sure that your device, the one with the buttons, is plugged in and working BEFORE running Flight Simulator. The Windows joystick interface used by FSUIPC to scan the buttons may not recognise any devices that are plugged in after Flight Sim has started. I know USB is supposed to be "hot pluggable", but this doesn't always seem to be the case with respect to running programs. You could try the "Reload" button in the options, which forces FSUIPC to re-scan the devices too, where it may pick it up.

2. With the FSUIPC "Buttons" page showing, as above, check or uncheck the "Profile Specific" option, according to whether you want to edit buttons for this aircraft/profile only, or for global use. If you do select aircraft or profile specific you will see the name of the current aircraft or profile showing in the title bar of the options. (Note: *if you are using the abbreviated aircraft names option, this may be the abbreviated name that will be used. This option requires editing the FSUIPC7.INI file after programming, and is described only in the **Advanced User's** document).*

3. Now press the button you want to program. If it is visible to FSUIPC then it will be identified on the screen by Joystick Number (0–255 for local buttons) and Button Number (0–228). Please take careful note of the fact that this numbering starts at 0 *not* 1. Windows "Game Controllers" actually numbers from 1. Button numbers 33-39 are reserved for POV buttons

   For buttons being pressed on a WideFS client PC there will be a much larger joystick number: plus 1000 for Client 1, 2000 for Client 2, and so on.

   Note that FSUIPC uses the original FS convention of using button numbers 32–39 for the possible 8 directions on a "Point Of View" (POV) Hat. The directions are 32 (forward), then 45 degree increments clockwise to 39 = forward left. Please see the Box below for details of how to assign these buttons to imitate a full "point of view" Hat in MSFS.

   If the "button" is in fact a toggle or rotary then you may need to toggle it or turn it twice. This is because, when programming (only) FSUIPC is only looking for changes from "off" to "on", so it won't see a change from "on" to "off". This is necessary because of the way some multi-way switches work.

4. If you had already programmed this button in FSUIPC for this aircraft (or globally if the aircraft/profile specific option isn't checked), then the current programming details will be shown. If you had programmed sequences or mixed actions in the INI file, only the first action will show here and will not be editable—you can only do that in the INI file. Otherwise you can 'clear' all the programming or edit it as desired here.

   Buttons assigned to custom controls, using numbers not names, and which don't have a name known to FSUIPC, are shown as <n> where n is the number in decimal.

   For an un-programmed button, select first whether you want to make it produce a keystroke, or an MSFS control, an input event or a preset. The actions then are different and can be selected using the provided checkboxes. Once selected, keystroke assignments can be programmed using the left-hand side of the dialog box and FS controls, Input Events and Presets can be programmed using the right-hand side.

   Input events are only available when an aircraft is loaded and ready-to-fly in MSFS. Note also that there are no input events available for some aircraft.

5. **Key press programming** is done by pressing the 'Set' button, for button press or release (or both, if you want) then entering the key press or combination you want. The value won't appear until the key press is complete, but you can use combinations of **Ctrl**, **Shift**, **Tab** and one normal 'graphic' key.

   For a key press occurring when you press the button, you can program it to operate transiently—i.e. the key(s) are pressed and released almost immediately—otherwise the keys will be kept pressed until you release the button.

   If you have the keys kept pressed whilst the button is pressed, you can opt for the keys to be repeated, just as on the real keyboard. However, the repeat rate is fixed at around 6 per second.

   If you set a key press to occur when the button is released, this is always a transient key press, with no repeats. The key press for button release does not have to be the same as the one on button press, and can be used on its own.

6. **FS Control, Input Event or Preset programming** involves simply selecting the FS Control, Input Event or Preset you want to occur from the drop-down list. You can switch between the list of available presets, input events and the list of MSFS controls by using the provided checkboxes.

   Input Events are only available for assignment when MSFS is active with an aircraft loaded and ready-to-fly (i.e. not when in the MSFS main menu system. Input events are a relatively new addition, and there will be none available for some aircraft. Many can take a parameter, which is always a floating point value. To determine what values an input event can take, you should use the logging facilities provided (**Log->Input Events**), open the

logging console (**Log->Open Console**) and see what values are logged when you trigger the input event in the virtual cockpit. You can also list the available Input Events to the FSUIPC7 main window using the **Log-List Input Events** menu option. This will also tell you if the input event tales a parameter, but there is currently an issue with this in that many input events listed as not taking a parameter actually do. This has been reported to Asobo and will hopefully be rectified in a future update. For the time being, it is better to log the input events to determine if a parameter is needed, and the range of values (always 64-bit floating-point numbers) supported.

It is also possible to add an input event to an FSUIPC offset. This allows you to read the value of the Input Event, and also to write to an offset to trigger/set an Input Event. Please see the Advanced User guide for details on how to add an Input Event to an FSUIPC offset.

The preset list should only be used if you have installed and activated the FSUIPC WASM module. If you also installed the MobiFlight preset list (the *events.txt* file in your FSUIPC7 installation folder, which is installed by default) then you will automatically have access to these presets. The preset list is very large (>10000 entries) and it can take a second or two to populate the drop-down menu, and any specific preset can be difficult to locate. Therefore a '**Find Preset...**' button is shown, which will present a pop-up box containing a tree-view of the available presets, broken down by source (MobiFlight or Personal), provider/vendor, aircraft and sub-system. You can also use the MobiFlight HubHop resource (see https://hubhop.mobiflight.com/presets/) which has a good user interface for searching the available presets by vendor, aircraft, subsytem, etc. Please see the **FSUIPC7 Advance User** manual for further details on presets, including defining your own presets.

The FS controls list is also very long, and is rather cryptic—the names here are mainly the names actually used in MSFS internally. The exceptions to this are custom controls, and special commands added by FSUIPC.

Custom controls, without a name known to FSUIPC, can be assigned by selecting **<custom control>** from the drop-down list, and then entering its number in decimal (range 65536 to 131070 inclusive) or, if you wish, in hexadecimal preceded by an **x** (range x10000 to x1FFFE). Decimal will be displayed when re-viewed no matter how it is entered.

The special controls or commands added by FSUIPC are all listed in the Advanced User's guide, but include all these and more:

- Two controls to operate the Roger Wilco (RW) and Advanced Voice Client (AVC) "push to talk" facilities (more below).

- Controls to operate the "PVT" private voice push-to-talk facility in Squawkbox 4.

- Controls to operate Squawkbox 4's transponder standby/mode c signal, and the transponder ident.

- A control for switching Comms transmission between COM1 and COM2.

- Three controls to handle FSUIPC's "button flags", used in conditional button programming. *Button flag clear, set and toggle* are programmable both here and in the **Keys** programming page, but they can only be used in complex button programs by editing the FSUIPC7.INI file. This is described fully in the Advanced User's Guide.

- A control for sending **KeySend** messages to WideFS clients running on other PCs. The specific KeySend number is set as a parameter (1–255) and is equated to a key press directed to a client program in the relevant WideClient.ini file. Just be sure to set the parameter for the KeySend control to the same number ("N" in KeySendN=…) used in the WideClient.ini file, as it is this which related the two. (Please see the WideFS documentation for more details of its KeySend facilities).

- A "**Nothing**" control which can be assigned in aircraft-specific or profile-specific cases to prevent operation of a generally assigned button which is needed for purposes outside of FSUIPC for those specific aircraft.

- Optionally many extra controls for Project Magenta modules. To see the PM controls in the drop-down lists check the option box on the page. All the Project Magenta entries start with "Pm" and the Airbus-specific ones "Pm Ab". A list of those supported is provided in the FSUIPC Advanced Users Guide.

- An **Autobrake Set** control, to directly set the autobrake (0=RTO, 1=off, 2–5=settings to Max).

- A set of "fast" autopilot value increment/decrement controls. These add to the default MSFS "inc" and "dec" controls with ten times the increment. Values handled are Altitude, Speed, Mach, V/S, Heading, and both OBIs (Courses).

- A set of controls to adjust the in-use radio frequencies directly, as well as controls to set the transponder using only two dials instead of four.

• **Spoiler inc** and **Spoiler dec** controls, which raise or lower the spoilers (speed brakes) by a small amount on each use. The increment is set, by default, to give 32 steps in the whole range, but you can change that in the FSUIPC7.INI file—see the Advanced User's document.

• **Freeze pos** controls (on, off and toggle) to freeze the aircraft's geographical position (latitude/longitude) but allowing it to otherwise fly normally, including altitude changes. [*However, please note that MSFS actually comes with its own control for this, known as 'Active Pause'*].

• **Engine N Autostart:** Separate engine auto-start controls.

• **Throttles on, off, toggle:** Controls to disconnect and re-connect throttle inputs.

• An assortment of controls to handle Lua plug-ins, including starting and 'killing' them, and setting flags for them to test and parameters for them to use.

• **Offset** controls, which allow values to be written to FSUIPC's offsets, for all sorts of actions and effects. There are controls for byte (8 bit), word (16-bit), double word (32-bit) and both 32-bit and 64-bit floating point values, and facilities for bit setting, clearing, toggling, incrementing and decrementing (the latter two only for byte and word values). When using these, another input field appears on screen for the entry of the specific offset, which is best entered in hexadecimal (precede the value with the letter 'x'). The value is provided as a parameter and this too can be entered in hexadecimal, preceded by 'x'. For full information on these please refer to the Advanced User's documentation.

Offset Increment/Decrement Controls

The increment/decrement controls operate on signed (Sbyte, Sword) or unsigned values (Ubyte, Uword), and have a more complex parameter that specifies both the increment/decrement (always positive) and a limit, thus:

<increment value> / <limit>

This allows things like Trim adjustments to be programmed to be whatever speed you like, or even multiple speed if you have enough buttons or levers to spare. Taking Trim as the example:

The elevator trim is a signed WORD at offset x0BC0. For this sort of information you should refer to the Offsets list in your FSUIPC Documents folder. The range is –16383 (full trim down) to 16383 (full trim up). So, when programming your button/levers:

1. Select the "Offset SWord Increment" control to program the nose up trim.

2. Enter x0BC0, or just xBC0 into the offset edit box.

3. Enter 256/16383 in the parameter box. The 256 is the increment and 16383 is the limit. This will give 128 steps between –16383 and +16383 inclusive (32768 / 256 = 128). If you want a faster, coarser trim adjustment specify a larger increment, and of course vice versa for a slower, finer adjustment.

4. You'll probably want it repeating whilst held, so check that too (*but never set repeat for rotary switches which can be left in an 'on' position, nor latching switches*).

5. Do the same for the decrement, with a parameter of 256/–16383 (–16383 being the lower limit). Note that the decrement is still positive—you can only provide positive numbers for this part. It is the definition of increment or decrement which controls the addition or subtraction.

6. When reviewing such assignments you may see the parameters showing in hexadecimal (preceded by 'x'). This is an optional way of inputting these in the first place, just as with the offsets.

Note: This method can also be applied to the rudder & aileron trims. Select "Offsets sword increment" with 256/16383 as the parameter to trim to the right, and select "Offset sword decrement" with256/-16383 as the parameter to trim to the left. Enter x0C04 for rudder trim or x0C02 for aileron trim into the offset edit box, and then proceed as 3 to 5 above.

For some help in untangling the real Flight Sim controls you can refer to my "MSFS Controls" document, which is included in the files installed for you into the "FSUIPC7" folder under your Windows Documents folder. It is in PDF form (for reading with Adobe Acrobat). For FSUIPC added controls refer instead to the Advanced User's Guide.

The best way to work them out, really, is to try them—pick one that seems to indicate that it may do what you want, and see if it does. I know for sure that some of them do not work, or at least do not work the way you might hope or expect.

Most if not all of those with the word "set" in their name are controls that take a parameter. Axis controls are like this of course, but so are, for example, the controls "MAGNETO SET" and the equivalents for each of four separate engines.

Naturally, there is no way FSUIPC can dynamically provide different parameters for a single button. You can have a single button send a control with a fixed parameter—you specify this in the parameter field. A good example is a button to set the "standard barometer" setting (1013.2 hPa or 29.92"), for flying Flight Levels. For this, select:

Kohlsman Set	as the control, and
16211	for the parameter (1013.2 mb x 16 = 16211)

Another useful example is a button to exactly centre the elevator trim:

Elevator Trim Set	is the control, and
0	is the parameter

Again, as with key presses, you can have a separate control sent when you release the button, and you can also specify that the earlier control should be sent repeatedly whilst the button is held. A typical use of different controls on button press and release is for Views, from a Hat. For example:

Button 33 (forward right) might be programmed with

View Forward Right	on press
View Forward	on release

If you want to mix key presses and MSFS controls, or send a sequence, you need to edit the FSUIPC7.INI file. This is described in the Advanced User's Guide.

For Roger Wilco or Advanced Voice Client (AVC) users, at least those running the voice program on the same PC as MSFS, you do not need to try to allocate the correct keypress as assigned in that program. Some versions of RW do not obtain keys in a way that FSUIPC's key press facilities can get to in any case. Instead, just scroll down the MSFS controls list for the pair "Ptt Transmit Off" and "Ptt Transmit On". Set the press of your button to operate the "On" and the release to operate the "Off". This is known to work will all currently available versions of Roger Wilco and with recent versions of Advanced Voice Client.

Finally, if you want to clear ALL the FSUIPC button programming, without pressing each button and pressing "Clear" in turn, just delete the complete [Buttons] section (for global buttons) and [Buttons.<aircraft name>] sections (for aircraft-specific buttons) in FSUIPC7.INI. Try to do this when MSFS is not running. Most changes to the INI file are only noticed by FSUIPC when MSFS is started, though both the Keys and Buttons sections are reloaded when you change aircraft.

Key Press Assignments
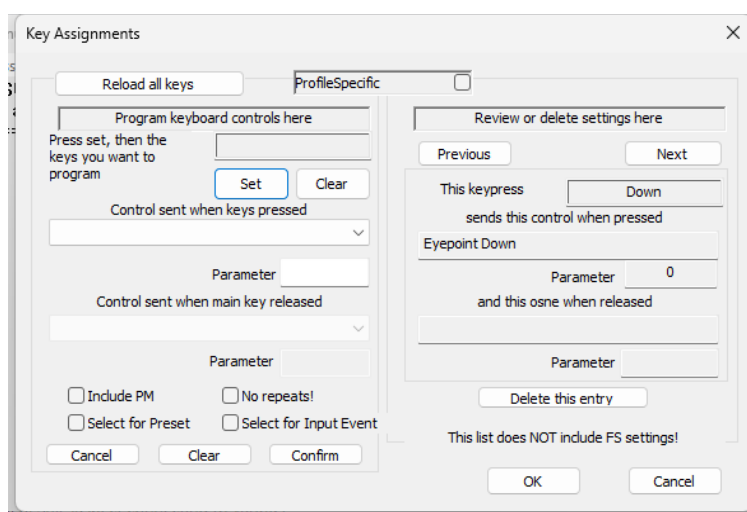
Normally you will assign keys to Flight Sim operations in the MSFS assignment facilities. However, there are more controls than are assignable in that way, and the assignments cannot be made conditional on the current aircraft (for instance)..

FSUIPC therefore offers its own way of assigning keystrokes. In this page you can assign simply single keys or combinations of those with Shift, Control, Tab, Alt and Menu. (The "Menu" key is the one with a menu icon on it, to the right of the right-hand Windows key). The left/right Shift, Control and Alt keys are distinguished and are treated as individual keys.

If you assign keystrokes to MSFS controls here, those keystrokes will also be seen by MSFS's own keystroke assignment programming at the moment, due to masking issues with MSFS. Therefore, please take care and check any key assigned in FSUIPC is not also assigned in MSFS.

Key press actions can be made conditional on values or bits in FSUIPC's offsets, opening up a whole realm of possibilities, such as different things to be done when on the ground as opposed to flying, or with engines running or not, and so on. These need editing in the INI file and the facilities are described in the Advanced User's Guide.

Looking at the Keys option page, above (or on your screen), you'll see that it is divided into two parts. The left side is where you will add assignments or modify existing ones. The right side is only used to review or delete existing assignments (in FSUIPC *only*).

You can assign keys to operate globally, with all aircraft, or to operate only when the current aircraft (profile) is loaded. This selection is by unchecking or checking the "profile specific" check box. Do this first, before starting to program or review key settings. If you program a key both for global and specific use, the latter prevails whenever the aircraft is loaded—you don't get both actions simultaneously. If you do select "profile specific" you will see the name of the current aircraft or profile showing in the title bar of the options. (Note: *if you are using the abbreviated aircraft names option, this may be the abbreviated name that will be used. This option requires editing the FSUIPC7.INI file after programming, and is described only in the **Advanced User's** document).*

To assign an action to a keypress, press the **Set** button in the left hand panel. The edit box above displays "PRESS KEY". Press your key combination, for example TAB+Q, and you will see it displayed. Nothing is displayed until you have completed the combination with a graphic, cursor or function key. Shift-type keys (and Tab) cannot be used alone.

You can now assign an action to that key press. You can also assign an action to be carried out upon the key release— but this is not implemented *unless* you have an action for the press. In other words, you can have a single action, which occurs when the key is pressed, or a dual action, one when it is pressed and another, probably different, when released. The dual facility tends to be only useful for holding temporary situations, like views or additional windows.

Additionally, whilst, like MSFS itself, FSUIPC will normally act upon repeated keypresses (the result of holding a key down) as if they are requests for repeated "press" actions, you can ask it to ignore such repeats. To do this, just check the **No repeats!** checkbox which you will see near the bottom.

Use the drop-down control list to find the control you want to assign. You can switch between the list of available presets, the list of available Input Events and the list of controls (both standard and add-on) by checking/unchecking the **Select for Preset** and **Select for Input Event** checkboxes.

The preset list is very long (>10000 entries) and can take a second or two to populate once opened. Rather than using the drop-down menu, where it can be difficult to locate a specic preset due to the number of entries, you should use the **'Find Preset...'** button. This will show an additional box containing a tree-view of available presets, broken down by vendor/providor, aircraft and sub-system. Select the preset in this view and the preset drop-down will be populated with your selection.

Note that the controls list is also very long (approx 1800 entries), and is rather cryptic—the names here are mainly the names actually used in MSFS. The exceptions are some special commands added by FSUIPC. The latter include:

- Controls to operate the Roger Wilco (RW) and Advanced Voice Client (AVC) "push to talk" facilities (see earlier). These work with Squawkbox4 too.

- Controls to operate the "PVT" private voice push-to-talk facility in Squawkbox 4.

- Controls to operate Squawkbox 4's transponder standby/mode c signal, and the transponder ident.

- A control for switching Comms transmission between COM1 and COM2.

- Three controls to handle FSUIPC's "button flags", used in conditional button programming. *Button flag clear, set and toggle* are programmable both here and in the **Keys** programming page, but they can only be used in complex button programs by editing the FSUIPC7.INI file. This is described fully in the Advanced User's Guide.

- A control for sending **KeySend** messages to WideFS clients running on other PCs. The specific KeySend number is set as a parameter (1–255) and is equated to a key press directed to a client program in the relevant WideClient.ini file. Just be sure to set the parameter for the KeySend control to the same number ("N" in

KeySendN=…) used in the WideClient.ini file, as it is this which related the two. (Please see the WideFS documentation for more details of its KeySend facilities).

• A "**Nothing**" control which can be assigned in aircraft-specific or profile-specific cases to prevent operation of a generally assigned key which is needed for purposes outside of FSUIPC for those specific aircraft.

• Optionally many extra controls for Project Magenta modules. To see the PM controls in the drop-down lists check the option box on the page. All the Project Magenta entries start with "Pm" and the Airbus-specific ones "Pm Ab". A list of those supported is provided in the FSUIPC Advanced Users Guide.

• An **Autobrake Set** control, to directly set the autobrake (0=RTO, 1=off, 2–5=settings to Max).

• A set of "fast" MSFS autopilot value increment/decrement controls. These add to the default MSFS "inc" and "dec" controls with ten times the increment. Values handled are Altitude, Speed, Mach, V/S, Heading, and both OBIs (Courses).

• A set of controls to adjust the in-use radio frequencies directly, as well as controls to set the transponder using only two dials instead of four.

• **Spoiler inc** and **Spoiler dec** controls, which raise or lower the spoilers (speed brakes) by a small amount on each use. The increment is set, by default, to give 32 steps in the whole range, but you can change that in the FSUIPC.INI file—see the Advanced User's document.

• **Freeze pos** controls (on, off and toggle) to freeze the aircraft's geographical position (latitude/longitude) but allowing it to otherwise fly normally, including altitude changes. [*However, please note that MSFS actually comes with its own control for this, known as 'Active Pause'*].

• **Engine N Autostart:** Separate engine auto-start controls.

• **Throttles on, off, toggle:** Controls to disconnect and re-connect throttle inputs.

• An assortment of controls to handle Lua plug-ins, including starting and 'killing' them, and setting flags for them to test and parameters for them to use.

• **Offset** controls, which allow values to be written to FSUIPC's offsets, for all sorts of actions and effects. There are controls for byte (8 bit), word (16-bit), double word (32-bit) and both 32-bit and 64-bit floating point values, and facilities for bit setting, clearing, toggling, incrementing and decrementing (the latter two only for byte and word values). When using these, another input field appears on screen for the entry of the specific offset, which is best entered in hexadecimal (precede the value with the letter 'x'). The value is provided as a parameter and this too can be entered in hexadecimal, preceded by 'x'.

For more details of the **Offset Increment/Decrement** controls, please see the Boxed section under **Buttons**, earlier.

For some help in untangling the real Flight Sim controls you can refer to the "Controls List for MSFS" document, which is automatically generated and installed for you into the FSUIPC7 Documents folder. It is in txt form (for reading in a standard editor such as Notepad++). For FSUIPC added controls refer instead to the Advanced User's Guide.

The best way to work them out, really, is to try them—pick one that seems to indicate that it may do what you want, and see if it does. I know for sure that some of them do not work, or at least do not work the way you might hope or expect

Most if not all of those with the word "set" in their name are controls which take a parameter. Axis controls are like this of course, but so are, for example, the controls "MAGNETO SET" and the equivalents for each of four separate engines. Note that these do not necessarily work—if fact I don't think the MAGNETO controls work as they should.

Naturally, there is no way FSUIPC can dynamically provide different parameters for a single button. You can have a single keypress combination programmed to send a control with a fixed parameter—you specify this in the parameter field. A good example is a key to set the "standard barometer" setting (1013.2 hPa or 29.92"), for flying Flight Levels. For this, select:

Kohlsman Set	as the control, and
16211	for the parameter (1013.2 mb x 16 = 16211)

Another useful example is a key to exactly centre the elevator trim:

Elevator Trim Set	as the control, and
0	is the parameter

If you want, you can make FSUIPC send a sequence of MSFS controls when you use a key press, but for this you will need to edit the FSUIPC7.INI file. This is described in the Advanced User's Guide. You can only edit or program the first such control in the dialogue.

The review and deletion facility forms the right-hand part of the options page. Here you can simply scan through all your FSUIPC assignments (*not* those made in MSFS's dialogues). The list is shown in the order they appear in your FSUIPC7.INI file, and cycles back to the beginning at the end and vice versa. You can delete any assignment here by using the **Delete this entry** button.

Finally, if you want to clear ALL the FSUIPC key press assignments, without pressing "Delete this entry" for every one, just delete the complete [Keys] section (for global keys) and [Keys.<aircraft name>] sections (for profile-specific keys) in FSUIPC7.INI. Try to do this when MSFS is not running. Most changes to the INI file are only noticed by FSUIPC when MSFS is started, though both the Keys and Buttons sections are reloaded when you change aircraft and you can force reloads using the Reload button on each of those option screens.

Axis assignments

Assignment of joystick buttons, joystick levers (called axes here) and key presses can all be done, quite normally, in MSFS via its Assignments dialogues. We have already seen that FSUIPC can expand upon the assignment of Buttons and Key presses, allowing a much wider range of controls to be assigned, including special ones added by itself, and, quite usefully, allowing different assignments for different aircraft, automatically changing when aircraft are loaded.

In this section we shall see that the same flexibility can be achieved with the assignment of joystick axes too. Axes are those devices which provide a variable parameter rather than just an 'on' and/or 'off' event like a button, keypress, or switch. Most axes are controlled by the joystick or yoke itself (for aileron and elevator control), rudder pedals, analogue brakes, and those levers or push-pull rods associated with throttle, mixture and propeller pitch controls.

For most normal uses it is perfectly acceptable, and easier, to assign as normally in MSFS itself. The only strong reason for using FSUIPC assignments is when you want to be rather more ambitious with axes, possibly having different assignments for different aircraft or aircraft types ("profiles"), or having multiple assignments to the same controls, or using separate reverser or tiller axes. If you want things to remain simple, please skip this chapter as well as the Joystick Calibration.

FSUIPC offers a greater range of controls for axis assignment, and can make them aircraft-specific too. Further, it can cause non-axis controls to be sent as the axis parameter passes into, out of, or through specific calibrated areas of the axis range. It can even produce multiple MSFS control inputs for one axis input—up to four axis results into MSFS, thus allowing single lever control of multiple engines, in different configurations according to the aircraft loaded. It even offers a facility to insert a delay in the sending of axis values to MSFS—apparently this is needed for accurate simulation of some helicopter models. This latter facility, however, can only be set by editing the FSUIPC7.INI file, and is described in the Advanced User's document, not here.

One additional axis of note is the **Steering Tiller**. Note that MSFS does actually provide a Steering Axis which can also be assigned and calibrated in FSUIPC. The MSFS offering may be better with some aircraft. You'll need to choose.

The FSUIPC version uses the MSFS Rudder control, but it can be calibrated separately (e.g. to be more responsive—use the inverse S-shaped slope options). If the steering tiller axis is assigned, you must then calibrate it in FSUIPC's Joysticks section (this is on the same page as the PAN controls … sorry!). You need to calibrate the rudder axis in FSUIPC too. Then the two are used together as follows:

When on the ground and at any ground speed less than 60 knots (default—adjustable by the **MaxSteerSpeed** parameter in the INI file), the actual MSFS rudder action is controlled by a blend of the tiller and rudder axis inputs. At low speed it is predominantly tiller (with a minimum contribution from the rudder input of 10% of its deflection), and as speed increases the tiller becomes gradually less effective and (above 6 knots) the rudder input more so. Above the **MaxSteerSpeed**, or in the air, the tiller has no effect.

As with all of the axis and joystick facilities, the calibrations, assignments and parameters such as **MaxSteerSpeed**, can be different for different aircraft. The **MaxSteerSpeed** parameter actually has several options which can control the rate at which the blending occurs, so it doesn't have to be linear. Please refer to the Advanced Users guide if you want to know more.


IMPORTANT: Before making any assignments in FSUIPC, you should be sure that the same axes are not being assigned in MSFS itself.

However, if you just want to use FSUIPC to program the odd axis, but leave the rest to MSFS, then you need to go into MSFS's assignment dialogues and de-assign the axis you want FSUIPC to handle. **But be sure to check this on your next load of MSFS, just in case it gets reassigned automatically. It does happen—*especially* if you are in the habit of unplugging your USB devices!**

The only usual reason you may want MSFS to handle some axes rather than FSUIPC is for Force Feedback. FSUIPC does not use the same methods to access joysticks as MSFS and does not support Force Feedback.

If Force Feedback isn't a requirement, and you want the maximum flexibility (and most efficiency—I believe FSUIPC's joystick handling is *still* actually faster than MSFS's), then the best way forward is certainly to disable MSFS's joystick facilities altogether (upi can create an empty profile in MSFS), then program all buttons and axes in FSUIPC.

Okay. Now load MSFS, start FSUIPC7 and select the **Assignments → Axes...** menu entry, and you will see this:
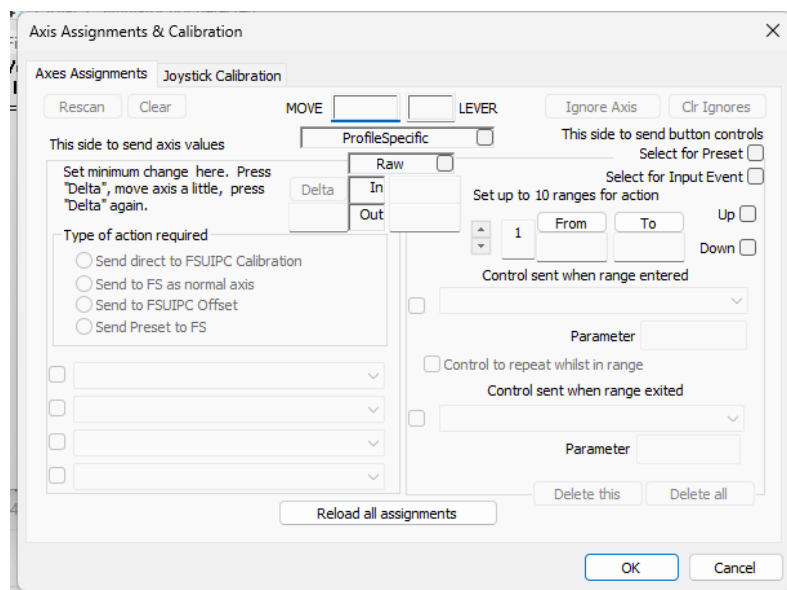
This does look a bit daunting, and maybe messy too. But please bear with me. It is very functional, I have made things look familiar where the facilities are similar to those in other sections.

The main reason there is a cluttered look to this section is that there's a lot going on: the facilities are very flexible and powerful. To make things easier to follow I will include rather more pictures here than I usually do. First, let's start at the top with this little section:
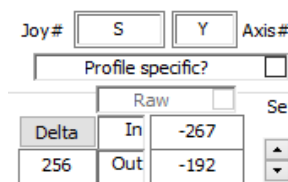
Here I have a device attached and I have moved one of its axes—the "Y" axis, or elevator (usually). Note that, because axes are providing data all the time, quite often on entry to this section of FSUIPC options, one of those axes will have already been detected as moving and will therefore already be displayed.

FSUIPC handles up to 16 joystick devices, numbered 0 to 15, each with up to 12 "axes". Here, the Joystick has been assigned a letter, 'S', as the "JoyLetters" facility is being used (now the default).

The possible axis identifications are X, Y, Z, R, U, V, S, T, P, Q, M and N. The first six (XYZRUV) are the original non-DirectInput axes FSUIPC has long supported, whilst S and T are Sliders, and P, Q, M and N are all "Point-of-View" (POV) controllers which cannot be used like genuine analogue axes.

Of these it is common for X and Y to be assigned to aileron and elevator, respectively, Z or S to throttle, R to rudder, leaving U and V for additional uses such as Mixture and Prop, or analogue toe brakes, and the inevitable POV (P) for view selection or panning.

If the displayed axis is not the one you want to set, try clicking on the "Rescan" button (top left), then moving the axis you want to set. If the wrong one comes up again it will be because that one is apparently changing more than the one you want—FSUIPC selects the one with the greatest changes. To get over that you can choose, temporarily, to ignore the axis that is interfering with your efforts. To do this, click on the "Ignore axis" button, over there on the right. This doesn't ignore it forever, only during the scanning. You can ignore as many as you like, and clear the whole list of ignored axes by clicking on the "Clr ignores" button at any time.

The other parts of this centre part of the display are:

- Profile specific checkbox: this is where you can tell FSUIPC that the assignments for this axis are specific to the current aircraft or profile. If this checkbox is not checked then the assignment applies to all aircraft that do not have any of their own specific assignments and are not assigned to a profile. NOTE that this is different to the way it

31

operates in Button and Key assignments. Your generic (non-specific) assignments do *not* apply to any aircraft for which you make specific assignments. This difference is necessary due to the active nature of axes compared to unpressed buttons or keys.

FSUIPC only retains one set of assignments in memory and therefore in action. This will be the general [Axes] set if the currently loaded aircraft has no specific assignments, otherwise it will be the latter. Also, the short aircraft name option (needing INI file editing) applies just as for Buttons, Keys and Calibrations.

When the profile specific option is selected for an aircraft you are presented with an option to retain (copy) all the general ones, to work from as a basis, or to start from a clean sheet. For minor variations you will probably opt for the former, but if you have completely separate controls for, say, a Boeing and an Airbus, or a prop and a helicopter, then you might find it easier to start again for each one.

Once a profile has been created and an aircraft assigned to that profile, the profile specific checkbox becomes inactive. To remove an aircraft from a profile, you'd have to manually remove the aircraft from the relevant Profile list in the FSUIPC7.ini file. You can also (optionally) delete the profile specific sections in the INI file if they are no longer used or needed.

At various stages you may be given the option to save any changes made so far. If you don't do this then axis assignment changes since entering the Options system may be lost.

- Raw checkbox: this is rather technical, so skip this explanation if you like and come back later. For now leave it unchecked.

  FSUIPC uses DirectX. This provides two ways to read joystick axis values—'raw' and 'calibrated'. In Raw mode, none of the calibration performed in "Game Controllers" is applied. The value passed through to the program is the actual value read from the device (or rather its driver).

  FSUIPC can use either. Whilst you will usually find that raw values are more honest in showing the true resolution of your device, you need to consider their end use. Where calibrated values will seem to vary enormously between large numbers like –32767 and +32767, the raw values are often just 0 to 255, or even 0 to 127. There are some higher resolution devices about which provide larger ranges, but not many. The main exception is the EPIC card which can, via its "soft" (programmed) axis facilities provide full 16 bit values.

  When FSUIPC is asked to apply RAW input to a normal analogue control, it scales it by a factor of 256 or 512 to bring it up from its 7 or 8 bit range to a full 16 bit value. However, a problem can then arise. Many USB devices, especially those which come with their own specific drivers, supply values in the higher ranges even though their true resolution is really no more than those with the smaller numbers. Whether it is the hardware or the software which is doing this pre-scaling is unclear, and it probably varies in any case.

  Unless you are sure (and, in fact, see) that the values from your device never exceed 255, the only real use of the Raw input facilities is when you are using an axis to set a precise value, such as a heading, altitude, speed, or even a radio frequency. For this you will almost inevitably be using a precise programmable input device, such as EPIC as previously mentioned. For all 'normal' analogue input needs, you should avoid "RAW" mode, so leave the setting to its default, and do a preliminary calibration in Game Controllers.

  Note that, in the illustration the 'Raw' option is disabled (greyed out). This is because I have other axes on the same joystick already set using the normal calibrated input. This is a restriction on the way the Windows API works when polling joystick axes. All 6 axes on a specific joystick must be read in the one mode, calibrated or raw. You cannot mix the modes on a single joystick. If you need to change the Raw/Calibrated option once an axis has been assigned, you will first have to clear all axis assignments on that joystick.

- Delta value (256 here): this value is the minimum change in the axis input that will be used by FSUIPC to cause any event (axis or other control to MSFS). The idea of this is to prevent minor jitters or very small changes from resulting in a massive queue of events being sent to MSFS for its attention. In the main, calibrated axis resolutions are much lower than folks may imagine—possibly only 30–60 different values in reality for the average hardware device. And the resolution of MSFS's reaction to different values is not designed to be that much better, for the obvious reason is that it need not be, with such low resolution inputs. Therefore, bombarding MSFS with many changes that make absolutely no difference is a waste of processing time.

  Except for the POVs, FSUIPC sets a default Delta value of 256 for calibrated input, but only 1 for Raw input—the latter on the basis that often the reason for the choice of 'raw' in the first place is to set a specific exact value. It is in your interest to set the highest Delta value that still gives you all the sensitivity and resolution you need. To change the Delta, click the Delta button, move the joystick lever the minimum fraction you want to detect, then press the button again. If you don't press it again within a few seconds, the delta value will be set to whatever change it has seen in that time, so take care.

POV axes (i.e. those designated as P, Q, M or N) have an assumed Delta value of zero, and this is not user adjustable. This means that they auto-repeat even when providing the same value. The repetition rate is throttled to around 18 per second, even if the Button polling rate has been set faster.

- The IN and OUT values here show the actual input value from the joystick ("IN") and the value actually used after the Delta change check has been applied ("OUT"). With the default Delta value the IN and OUT values will almost always be the same. Only if you set the Delta to a value greater than the resolution of the axis will you often see differences.

Okay. Now let's progress to the main points of this section: the actual assignments. You will see immediately that there are two halves for this. The left half deals with the normal assignment of axis controls to the currently selected axis. We'll deal with that first, as it is the one you'll be using most. The right half allows you do send button or switch type controls (even keypresses using an FSUIPC control for this) when certain parts of the axis range are entered, exited or passed through. And, although they look like two discrete parts, you can actually program the one axis to do both types of things.

First, then, the axis control assignment:
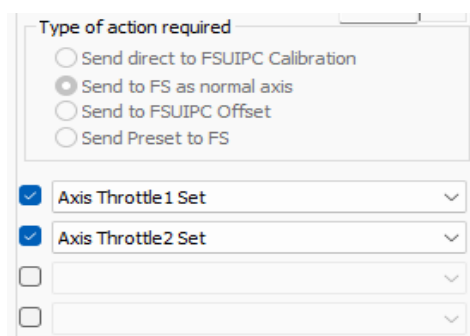
First, note that there are *four* ways you can assign axis controls. The normal (and default) method is to assign them to the available MSFS axis (or analogue parameter) controls. This is the "Send to FS as normal axis" selection. The drop down list contains not only all of those controls which MSFS's own axis assignments provide, but all those which FSUIPC knows about too—including, for example, the older 'THROTTLEn SET' controls which FSUIPC uses to provide reverse thrust zones on the same axis. Added to MSFS's own full list are those FSUIPC-specific controls which can take parameters, like Autobrake Set, though some of these would really need the Raw input setting to be of use (on this side of the dialogue anyway).

In the illustration you will see that you can assign up to four different controls to the one axis. You enable each of the four by checking it to the left. This provides full flexibility for "mapping" your levers to selected engines and so on. Together with the aircraft specific option and the automatic loading of the correct settings (and calibrations, if you calibrate in FSUIPC), you have the utmost flexibility. In this example I've assigned an axis to the throttles for engines 1 and 4, the outer two on a four-engined aircraft. I would likewise assign another lever to control the throttles of engines 2 and 3, the inner engines. Such variations are possible here, which just aren't easily supported elsewhere.

Turning to the other assignment method, the one labelled "Send direct to FSUIPC calibration" actually bypasses MSFS altogether until the axis input has been through FSUIPC's own Joystick Calibration section. In this case the drop-down shows only those controls which can be calibrated in FSUIPC, but this includes some which you would otherwise have to assign special numbers to in FSUIPC7.INI as described in the Advanced User's guide. For example these include Aileron Trim, Rudder Trim, and the four Cowl Flaps controls. (Part of the drop-down list for this option is shown in the illustration to the right).

Among the special controls there are three "combination" controls, which allow one axis to be treated differently depending upon MSFS's mode. These are:

Aileron/SlewSide	controlling ailerons in flight mode, lateral slewing in Slew mode
Elev/SlewAhead	controlling elevator in flight mode, longitudinal slewing in Slew mode
Rudder/SlewHdg	controlling rudder in flight mode, yaw slewing in Slew mode
Throttle/SlewAlt	controlling throttle in flight mode, vertical slewing in Slew mode

[Note however, that slew control is currently not working with MSFS. I have kept these for the time being and hope to re-enable this functionality when suitable controls for slewing are provided by the MSFS SDK.]

Whichever you choose, you can still use the FSUIPC calibration section as normal - in fact you have to in the case of the "direct to FSUIPC" controls. In other words these all need selecting and calibrating in the Joystick Calibrations section, which Flight Mode settings set and calibrated in flight mode, and the Slew mode settings in slew mode.

NOTE, however, that using the Direct to FSUIPC option will make the selected axis controls bypass any advanced cockpit panel coding you may be using. Try things first, and if they do not work with your panel you will probably need to revert to the normal methods.

ALSO NOTE that for those of you with multiple controls (e.g. for pilot/co-pilot) there is one special advantage in assigning axes "direct to FSUIPC calibration". As well as being able to assign several different axes to the *same* FSUIPC controls, FSUIPC will actually arbitrate between them, the maximum value (away from zero) being the one which "wins". This arbitration only takes place when an axis is moved, however.

The third option, "Send to FSUIPC Offset" is a much-less used facility to allow axis values to be sent direct to FSUIPC offsets. This is by assignment to one of five Offset controls -- Byte (8-bit), Word (16-bit), Dword (32-bit) Float32 (32-bit floating point) or Float64 ("double" floating point, 64-bit). Up to two offsets can be selected, with different Offset controls. Note that with the Byte assignment, any value exceeding the 8 bit capacity will be simply truncated, only the lower 8 bits surviving.

Care must be taken using this facility not to overwrite critical offsets. Normally the offsets chosen will be one of the user-assigned batch (66C0 - 66FF) or some specifically assigned to an FSUIPC application program. There are no ordinary user reasons for selecting this option.

Finally, the fourth option, "Send Preset to FS", allows an axis to be assigned to a *preset,* which are defined in either the *events.txt* file (provided by MobiFlight) or in the *myevents.txt* file (for your own presets). You would usually only use this option to assign to a preset that uses a parameter placeholder (either **$Param** or the **@** symbol) to accept the axis value, and you would also need to calibrate this axis value (in the preset calculator code) to accept values. For example, here is a preset definition which uses an axis with a usual range to control an lvar for the condition levers in the ATR, which accepts a value of 0-3:

```
MSATR CONDITION1 LEVER SET#@ 16384 + 10923 / near 0 max 3 min (>L:MSATR_CONDITION1_POS)

MSATR CONDITION2 LEVER SET#@ 16384 + 10923 / near 0 max 3 min (>L:MSATR_CONDITION2_POS)
```

To change between the four methods you will have to de-assign the axis controls first. Do this by removing the check mark to the left of each assignment.

Now, let's look at the rather more complex right-hand side of the Axis assignments section. Here are two pictures that I'll explain, as examples:
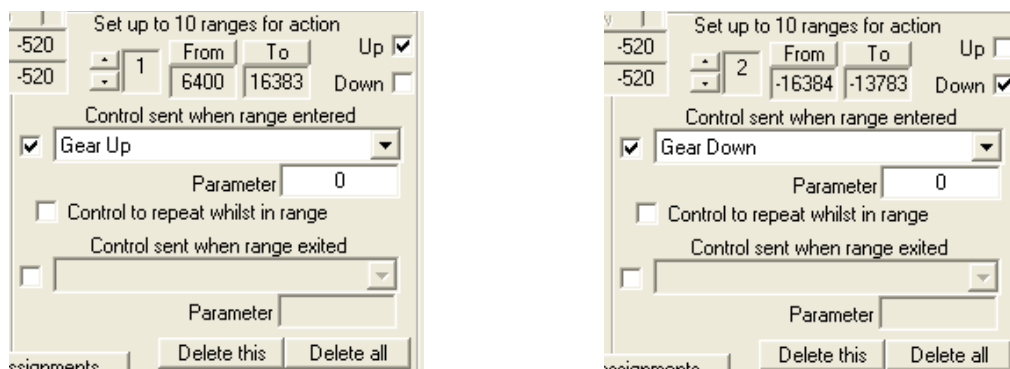
In this section you can set up two 10 zones (ranges of values) on the current axis that will each, separately, trigger a control or preset. The controls are any of the button/switch controls that you can assign in FSUIPC's button section—all are listed in the drop down according to the same rules as on the Buttons + Switches tab. To trigger a preset, first check the "Select for Preset" checkbox in the top-right, and the presets in your *events.txt* and *myevents.txt* files will be available for assignment. To trigger an Input Event, first check the "Select for Input Event" checkbox in the top right – these are only available when you have an aircraft loaded and not in the MSFS menu system. Those controls, input events and presets which take a parameter can do so here, and you can enter an offset too for the FSUIPC offset controls. Everything is exactly comparable to assigning buttons, except that the trigger is related to the values arriving from the axis.

First you set the range of values in which this action will occur. The little spin control on the left selects which of the 10 ranges you are now setting (you can go back and change or adjust them using that spin control). Then you simply move the lever to the two extremes of the range, clicking the "From" button at one end (the lower number) and "To" at the other (the higher number). If you get these the wrong way round, no matter—FSUIPC will still understand. The values should be different, though.

The check boxes 'Up' and 'Down'' allow you to select whether the action is to occur when the axis is moving from low values to higher ones ("Up"), or the other way ("Down"), or both. This allows you do have different things happen one way than the other.

You can assign one control to be sent when the specified zone is entered, and a separate one to be sent when the specified zone is exited—both or either, your choice. Check the box to the left of the relevant drop-down to enable it.

You can also assign to a Preset or Input Event, by checking the appropriate checkbox (top-right). By default, with no checkboxes checked, you can assign to standard FS controls/events.

In the example above, I have programmed a joystick lever to operate the Gear. When the lever is pushed away/up, into the higher range (6400 to 16383) I get a Gear Up, when it is pulled to me/down, into the lower range (–16384 to –13783) I get a Gear Down.

Other points to note here are:

- You can have the "entry" control repeat, at about 10 or so per second, whilst the axis remains within the range—just check the "control to repeat …" option.
- To delete any one of the 10 separate ranges assigned here, just find it and use the "Delete This" button. To delete all the ranges (but leave any axis assignments on the left-hand side) use the "Delete All" button.
- The "Clear" button at the top left of the page clears all axis assignments for the current axis, both left and right.

Finally, one final question should occur to you: in the Buttons+Switches section you can assign P3 controls or Key Presses. The Key Press assignments come in useful for those add-on aircraft panels which don't provide controls for everything. Suppose I want to send keypresses based on an axis position?

This *is* possible using added FSUIPC controls to send Key Presses. Scroll down the drop down list of controls and you will see the three options: Key Press and Release, Key Press/Hold, and Key Release. These take a parameter that defines the key press to be used. Here's a rather unlikely fictitious example:

When my selected axis zone is entered, I am sending Key Press 81+0. Here the '81' is a KeyCode and '0' are the Shift keys I need. KeyCodes are listed in the Advanced User's guide—81 is 'Q'. The shift key combinations you can have are also listed in that document. Here I just want a plain Q (for MSFS's sound toggle—for which I should really have used an MSFS control of course), and I could use '8' or '0' for that.


When the zone is exited I am sending 190+2. The 190 Key code is the .> key, the one normally allocated in MSFS to brakes. The +2 sets the Control shift, so this is "Ctrl+.", which will operate the parking brake toggle by default.

In both cases I would be better off using an MSFS control from the drop-down, but with an add-on panel, or even an external program looking for a hot key, you can see that the key press control will be useful.

Joystick Calibration

Selecting the "Joystick Calibration" tab in the FSUIPC **Axes Assignments & Calibration** panel, accessible from the **Assignments → Axes...** menu option, and you will see something similar to this:
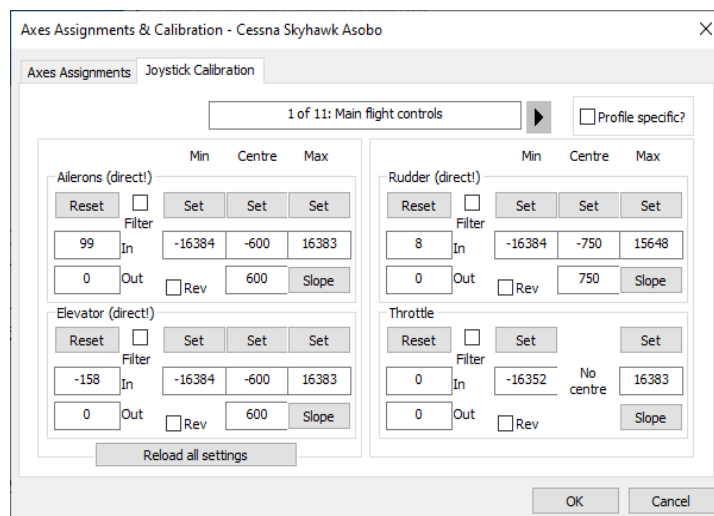
FSUIPC does not interfere with any joystick axis at all by default. You have to assign axes in the Axes Assignment pages first. In this picture all four axes have been assigned and will be handled by FSUIPC.

The numbers for IN and OUT under the left-hand Set/Reset buttons are actual values arriving at MSFS's simulation control in real time, as you move the controls. When FSUIPC is not processing an axis, the OUT value will be the same as the IN value, otherwise it will show the results of FSUIPC's processing. For those axes where FSUIPC is supplying a function not actually provided in MSFS (as in the case of the Reverser axis), you normally won't see any values unless you elect to process it by pressing the 'Set' button—pressing 'Set' changes it to 'Reset' and this indicates that FSUIPC is doing something with that axis.

When you set an axis by pressing 'Set', two or three new "Set" buttons for the axis become available and the main one becomes the "Reset" button (used then to stop FSUIPC's processing), so, as an example, page 1 with some things being calibrated may look like this (*Note that this picture is actually from FSUIPC3, but it illustrates the subject identically*):
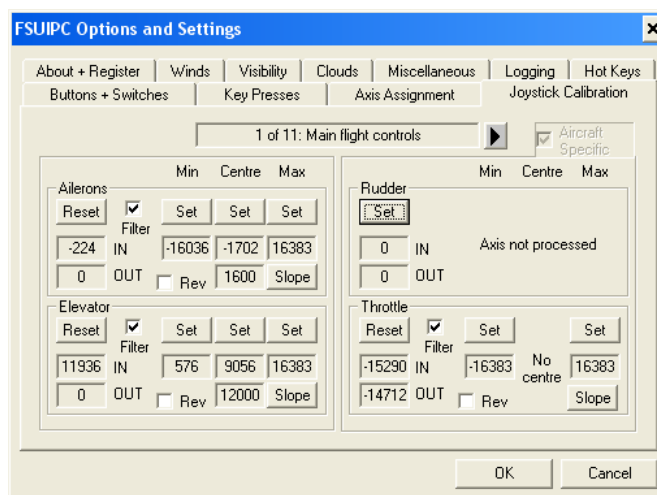
The numbers shown under the 'Set' buttons are those which will be used to scale the inputs, stretching or compressing them to fit the specific needs of the control, and also to set dead or null zones. You don't need to worry about the numbers particularly, as you will calibrate by simply moving the axis and clicking the buttons.

The default settings which first appear when you set an axis give full range (no scaling) appropriate to the axis type, with no limit dead zones and only a nominal (about 3%) central dead zone where this is applicable.

Unless you are using FSUIPC's Axis Assignment and other facilities for complete control of your joystick inputs, then before starting to use FSUIPC for joystick handling, to get maximum benefit you need to go to the MSFS joystick

options and make sure you have the **sensitivity** at maximum and the **null zone** at minimum. Any other setting will reduce the range of your joystick and make FSUIPC's more accurate calibration less effective.

Of course, if you are using FSUIPC for everything to do with joysticks, be sure instead to remove any assignments in MSFS's own settings menu). You won't want to do this if you are using a Force Feedback device, unless you don't like that option, as FSUIPC does not support Force Feedback.

<div style="border: 1px solid black; padding: 10px;">

Note for PFC throttle quadrant users:

If you are using a PFC throttle quadrant, with my PFCcom64 driver for MSFS, then it provides a facility to suppress "Game Port" throttles so that they do not interfere with its own. This actually refers to any Windows-recognised throttles, so it will apply to USB ones too.

If you use that facility you will see the comment "but using PFC" against the affected axes in the FSUIPC calibration pages. Normally you are expected to use the assignment and calibration facilities in the PFC driver, but if you have alternative throttle devices for specific aircraft you may want to override the PFC setting just for those aircraft. To do this you need to edit the FSUIPC7.INI file. Find the relevant [JoystickCalibration ...] section for the aircraft (assuming you are making aircraft or profile-specific assignments) and add this line:

AllowSuppressForPFCQuad=No

You should also consider preventing interference from the PFC throttles by assigning a blank user configuration for the same aircraft, though this shouldn't be necessary if you park the PFC throttles in a non-jittery position.

</div>

<div style="border: 1px solid black; padding: 10px;">

IMPORTANT: Throttles for Wilco Airbus users (and possibly other Airbus add-ons)

Normally, the 4-Throttles calibrations result in an output with either a range which includes the reverse zone, or, if the "no reverse zone" option is checked, a range from 0 (idle) to 16383 (max). These are sent to MSFS using the older THROTTLEn_SET controls, since these are the ones providing the reverse zone below zero.

The Wilco Arbus add-ons, however, appear to only like seeing the more modern AXIS_THROTTLEn_SET controls, which have no reverse zone, running from -16384 idle to +16383 full thrust.

Therefore you need to take special precautions when calibrating throttles for these aircraft. First you need to edit the FSUIPC7.INI file. Add this line (or change it) in the [JoystickCalibration] section applicable to your Wilco Airbus aircraft:

UseAxisControlsForNRZ=Yes

This will be Aircraft or Profile-specific if you set it in the appropriate calibration section of the INI file.

Then check the "**NRZ**" ("No reverse zone") box on the 4 throttles calibration tab in FSUIPC options. If you've already calibrated you should now re-calibrate because the values used will be different.

</div>

THE EASY STEP-by-STEP WAY TO CALIBRATE YOUR CONTROLS

To calibrate any axis, just do this:

1. For most axes, you need to be in normal flight mode (i.e. not **Slew**). However, if you wish to calibrate your joystick for slew movements, you *must* enter slew mode (e.g. press Y) before opening the FSUIPC assignments and calibration panel. FSUIPC like MSFS itself only sees slew axes when in slew mode, and only sees normal flight axes when not in slew mode.

2. Move the control for this axis and verify that the values for IN and OUT are changing. If they aren't, then you've either got the wrong control or your configuration is wrong (in MSFS's Assignments).

3. Set the axis, to obtain the individual set buttons, if these aren't already showing. Do this by pressing the Set button on the left, turning it into the 'Reset' button.

4. Move the control in either direction and verify that the value is changing up or down as you'd expect. Aileron and rudder controls increase (more positive numbers) when turned left (=**MAX**), decrease (more negative numbers) when turned right (=**MIN**). Elevator controls increase when pushed (nose down) (=**MAX**), decrease when pulled (nose up) (=**MIN**). All others are pretty logical—more is higher, less is lower. If your axis is changing in the wrong direction you will need reverse the axis, either in MSFS itself, or here in FSUIPC (see the little 'Rev' checkbox?).

5. Now move the control to its lowest (right-most/rear-most) position. If you want a dead area, relax it the amount you want to stay fixed, then press the "Set" button in the "min" column (for 'minimum'). The currently read value is recorded in the box below the button.

6. For controls which have centres or intermediate 'idle' positions (aileron, elevator, rudder, spoiler, trim *and* the separate throttle and propeller pitch controls where centre (idle) is between forward and reverse), position the control in its centre, détente (e.g. "arm" for spoiler) or default position. Pressing the centre "Set" button will enter the value in one or other of the two boxes beneath. Each time you press Set the alternate value is recorded, and the boxes show the two most recent values in order (lower above higher). If the values are the same you will have no dead zone around the centre.

    [*Note that the Flaps control uses the "centre" calibration mechanism to set positions for every flap détente— please refer t the separate section, later, for the way to deal with flap levers with détentes for specific settings*]

    If you want to be able to simply take your hands and feet off the controls and expect them to centre well automatically, you will almost certainly need a central dead zone. You will also need one on a throttle or other control with a reverse range, otherwise finding idle will certainly be a problem.

    You can *either*:

    •   simply push/pull/swivel the controls in different directions and let go, so that they centre with their normal variation, and press the centre Set for the two most extreme values you get for the supposedly centred control, *or*

    •   more precisely, and the only way with throttles and other levers, move the control one way a little, press the centre Set, then the other way a little, and press Set again. This way you get to choose the size of the dead zone with more precision.

7. Move the control to its highest (left-most/forward-most) position. Again, if you want a dead area relax it an appropriate amount, then press the Set button in the "max" column.

8. You can select a response curve ("**slope**"). There are 31 different slopes to choose from, ranging in one direction from linear (default) to one with a very flat centre (for reduced sensitivity—i.e. more control movement needed for same effect) and steep ends to reach the maximum deflections, and in the other direction to the reverse extreme: steep centres and flat extremes.

    To select the slope desired, press the **Slope** button. You will see a window like these:
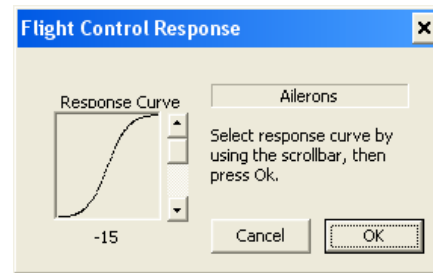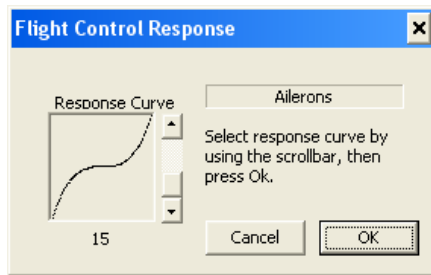
Here the one on the left is slope 15, the one with the flattest centre response (i.e. more stick movement, less effect) and steepest extremes, and the one on the right is slope –15 which is the other extreme, a very steep and therefore over-sensitive centre, and flat extremes. A flat response, the default, is slope 0.

For axes with no centres you only get to right-hand part of the slope, but the same variety is available. For axes with "off-centred" centres, such as the separate throttles with a small reverse zone below an off-centred idle position, the left hand part is kept linear in order to be sure that the very extreme left position can be reached. The slope changes apply only to the right-hand or positive part of the lever movement.

Just move the slider in the scroll bar down to the position for the sort of response you'd like. Press OK when you are happy, or Cancel to leave it as it was. Note that the response curves are only applied when the axes are calibrated in FSUIPC, though the slope settings are remembered anyway, even if no calibration is set.

9. There are two other options you may be wondering about. **Filter** allows you to switch on a simple digital filter for the axis. This tries to suppress fast changes such as jitter. Because it is kept simple (for performance reasons) it isn't perfect, but if you do suffer from jitter on an axis, try enabling it. It won't do any harm and it may help. [*Note that enabling the filter action on some axes, particularly Brakes, can apparently adversely affect the way some advanced cockpits operate. If you have any doubt do **not** check the Filter option*].

10. The other option is **Rev**. If you check that, FSUIPC reverses the axis completely, just before supplying the calibrated value to MSFS. All the calibrations and so on are done exactly as above—if you select Reverse you will simply see the Output value change.

11. On the four throttles page there is an option (top left) to calibrate the separate throttles with no reverse thrust zone. Check this if you are fusing an aircraft without reverse, or if you are either using separate reverser levers, or engaging reverse via button operation.

Similar options are provided on the 4 Propeller Pitch page and the 4 Mixtures page. The three options operate independently of each other.

12. On the four throttles page there is an option (lower centre) to exclude THROTTLEn_SET axis controls. This is provided to handle a specific problem when calibrating the throttles in some advanced add-on aircraft, and it is enabled by default . Only disable this option if you have anything which is feeding MSFS the older FS2000 and FS98 controls which you need calibrating or otherwise directing.

Similar options are provided on the 4 Propeller Pitch page and the 4 Mixtures page.

13. The "Sync Pos" button on the 4Throttles, 4 Prop Pitch and 4 Mixture pages invokes a special calibration facility designed to allow multiple levers to line up correctly for the same input to MSFS. This subject is dealt with in its own section, below.

14. That's it. Do this for each axis you feel needs it. Scroll through the 6 pages of Joystick axis settings and choose the ones you have connected. Note that, in *all* cases, calibration values increase left-to-right. That is, the boxes under the assorted "Set" buttons used in calibration can only be set such that the value in the right-hand one ("Max" usually) is greater than that in the left-hand one ("Min" perhaps), and the centre pair, if any, are in between these extremes.

NOTES and EXCEPTIONS

There are several special cases you may want to deal with:

- If you have a single brake lever or pedal assigned to "BRAKES" in MSFS, be sure to use that single axis to calibrate *both* the left and right brake exactly the same (right-hand side of page 2 of 6), otherwise every time you brake you will swerve to one side.

- If you have a single throttle control and you'd like to have an idle and reverse zone on this, then in the throttle part of page 1 select the "Map to 4 throttles" option. Then turn to page 3 of 6 and calibrate your throttle using the throttle 1 axis. The other three will match exactly. Choose the minimum (full reverse) and maximum (full forward) in the usual way, but make a centre 'idle' zone wherever you want it on your lever's movement—close to a détente or other stop if you have one.

- Similarly, you can use a single propeller pitch control lever to provide reverse pitch control (useful on the KingAir, for example). In the prop pitch section on page 2 of 6 select the "Map to 4 props" option, then turn to page 5 of 6 and calibrate your propeller control using the prop pitch 1 axis. The other three will match exactly. Choose the minimum (full reverse) and maximum (full forward) in the usual way, but then make a centre 'idle' zone wherever you want it on your lever's movement—close to a détente or other stop if you have one.

- Again, you can use a single mixture (or 'condition') control lever to provide a specific idle zone above the full cut-off minimum position—also useful on the KingAir. In the mixture section on page 2 of 6 select the "Map to 4 mixt" option, then turn to page 4 of 6 and calibrate your mixture/condition control using the mixture 1 axis. The other three will match exactly. Choose the minimum (cut-off) and maximum (full rich) in the usual way, but then make a centre 'idle' zone wherever you want it on your lever's movement—close to a détente or other stop if you have one.

- If you have twin throttle levers and you'd like to control left wing engines and right wing engines separately on both 2 and 4-engined places, then make sure both throttles are operating correctly with twin planes then go to the third page of the Joystick controls in FSUIPC, the one showing 4 throttles. Calibrate throttles 1 and 2 then check the option "Map 1->12, 2->34". This will do the job. The mapping only occurs when a 4-engined aircraft is loaded. This facility also applies to the mixture and propeller pitch levers.

- A similar facility is available for flying 3-engined aircraft with two levers. In this case the first lever controls both engines 1 and 2, and the second one controls engine 3. Follow the same calibrations as in the previous instance, but then check the option "Map 1->12, 2->3". This mapping only occurs when a 3-engined aircraft is loaded. This facility also applies to the mixture and propeller pitch levers.

> **IMPORTANT:** If you have more than one throttle, and calibrate them separately on Page 3 of the FSUIPC Joystick options, you need to take care with any aircraft panels equipped with an "Engine Control Unit" (ECU) or "Electronic Engine Control" (EEC). The aircraft panels equipped with these are altering the values you send from your throttles in order to keep the engine within set bounds. But they do this by sending the same controls as your real throttles, so FSUIPC applies calibration to those as well as yours. If you calibrate with the idle position well below centre (i.e. negative), which is quite likely, as you will want to use a smaller portion of the travel for reverse than for forward thrust, then the "idle" value of 0 sent by the panel will equate to some positive forward thrust.
>
> There is no easy answer to this. You will either have to calibrate taking care to keep the idle area centred on the value 0, or disable the Engine Control Unit in the panel.

- MSFS does not operate its "elevator trim" facilities on the pitch control of helicopter models. If you want a pitch trim control for these, FSUIPC can provide it for you. You must calibrate the elevator axis itself (i.e. the Y axis of the cyclic) in FSUIPC, and edit the FSUIPC7.INI file, adding '**ApplyHeloTrim=Yes**' to the appropriate [JoystickCalibration …] section in the FSUIPC7.INI file. FSUIPC will then apply both MSFS elevator trim inc/dec and axis controls to the aircraft's pitch axis. For more details see the **ApplyHeloTrim** parameter in the Advanced User's document.

- Similar considerations apply to the trimming of bank control on helicopter models. This can be handled, using MSFS's aileron trim, in addition to the pitch trim by calibrating the aileron via FSUIPC and setting the **ApplyHeloTrim** parameter to **Both** instead of **Yes**. Again, for more details see the Advanced User's document.

- The Spoiler (speed brake) control is a little special. It has no "centre" as such, but the facilities for setting a centre in FSUIPC are used to calibrate a zone on your lever in which the spoiler should be "armed". You don't need to do this, it is options: if the two centre calibration values are the same (i.e. there is no centre "zone"), then there is no arming action taken by FSUIPC. If, however, you do calibrate with a centre zone, then the values from "minimum"

(the left most value), or lower, up to the lowest of the Centre values will all select spoilers down (i.e. parked). The centre zone will arm the spoilers, and the range from the higher of the two centre values up to the maximum (the right-most value) will actually operate the full range from 0% to 100% deployed.

Note that if you arm the spoilers when on the ground, MSFS will deploy them to 100%, as if you have just touched down. However, they should return to the "armed" position when the throttles are advanced.

• The Flaps axis control can be calibrated simply, just like the other axes. If the simple approach is taken then FSUIPC will compute zones along the calibrated range for selection of each of the flap settings available with the current aircraft. However, for those of you who are building a flaps lever with détentes for the flap selections specific to your chosen aircraft, facilities exist to calibrate the axis values to match your détentes precisely. This is described in the following section.

CALIBRATING MULTIPLE THROTTLE, PROP & MIXTURE AXES TO "LINE UP"

The 4 throttle, 4 prop pitch and 4 mixtures pages all feature a button, centre bottom of the display, labelled "**Sync Pos**". This is used to help you calibrate multiple axes, for 2–4 engines, so that they are close to lining up, in similar positions, when providing the same input values (thrust or whatever) to Flight Sim. This is the procedure to follow when you want to do this:

1. Before starting on this process, first make sure you set whatever options you need. Choose the "no reverse zone" option before doing anything if this is what you want, and also the "Rev" options where needed to alter direction. The Slope and Filter facilities shouldn't be used.

2. Calibrate the first axis (for Engine #1) following the correct procedures as outlined above. It is important to get this right first. Calibrate the others too—but try to set the minimum, maximum (and centre if used) values such that the levers all line up at each of those two (or four) places. This gets you half way there already.

3. Now, line up all the levers in several intermediate places—especially those where you know they didn't give equal outputs before. For each lined up position, press the "**Sync Pos**" button. The positions of all the calibrated levers will be remembered. Do at least 4 such positions, spread over the lever range (between your minimum and maximum calibrations). The more positions you do, the smoother will be the calibration between each. FSUIPC will record up to 63 such positions, but will overwrite close ones with later settings. You will never need that many!

4. You don't need to set a large number of positions all in one go. Do a few, then OK out of the options and check the results. Bring up the quadrant levers display in MSFS so you can see how you are doing. If you need to do more, go back and simply use the **Sync Pos** button again. FSUIPC will ask you if you want to replace the settings you already did. If you think you ought to start again, say Yes. Else your new settings will be added to the previous ones, or replace them when very close.

For axes which have had a Synchronisation list made in this way, the "Slope" option does not apply. Instead, pressing the Slope button will show you the 'curve' representing how your sync pos recording has affected how the calibration will be performed. You cannot amend this there, but if you want to delete this feature for any of these axes, you can do so there. Bear in mind that if you do delete it there, then OK out of the Options, you will lose those settings forever. But if you are simply experimenting and want to go back to normal calibration, that's the way to do it—easier than editing the INI file.

The synchronisation positions are saved in the INI file, in the specific [JoystickCalibration] section currently applicable (eg. specific for the aircraft or profile when this is applied), in lines beginning "SyncSlope...". The numbers shown in pairs represent the x and y positions, respectively, of the (up to 63) points on the "slope" being used to calibrate the named axis.

CALIBRATING FLAPS WITH SPECIFIC DÉTENTES

If you have a flaps lever with détentes (fixed places for specific flap settings), then you can calibrate the axis specifically so that the region of those détentes calibrates exactly to the desired flap setting. Your calibration is then specifically suited to that particular range of flaps settings.

Do not even entertain the idea of doing this for an ordinary free-moving flaps lever. It simply isn't worth it—you'll have more difficulty selecting the correct flaps than by simply making small movements and observing the gauge.

If you really wish to proceed, go to page 6 in the Joystick Calibration tab, the one with the Flaps details shown. After pressing the "Set" button, to indicate you want to calibrate the flaps, you will see something like this (the 737 is loaded here):

As it stands you can go ahead and calibrate the flaps axis just like any other. The secret for setting specific détente zones is all behind the button labelled "Dt", for Détentes. It's a toggle. Click on it once and you'll see changes, as shown in the second picture, below:
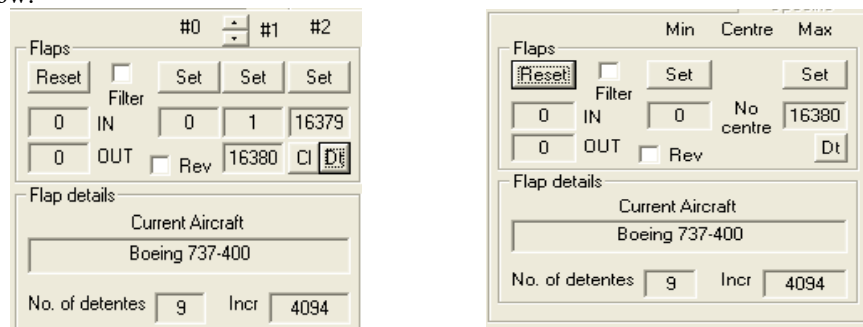
There are now two 'centre' values showing, a "Cl" button (for Clear), and the column headings showing #0 #1 and #2, with a little spin control next to the centre #1. Note that the spin control and the #1 are above the two newly appeared centre values. THIS IS IMPORTANT!

The left heading number (#0) indicates that the 'Minimum' (left-most) calibration is the limit for 0 flaps—that is no flaps, or flaps completely up. The zone for this no flap setting goes from –16384 up to the value calibrated in this column. You set this value by using the leftmost of the three Set buttons, and the value calibrated here remains set for Flaps 0.

The right heading number (#2 at present) is the highest flap number so far entered—because we must have a 'flaps down' and a 'full flaps', and at least one 'centre' (intermediate) flap value. Thus, the initial default is for flaps #0, #1 and #2. Note that these are flap numbers, always sequential, not flap angles!

The process to calibrate your détentes is as follows:

1. Set the flaps lever to a position a bit inside the Flaps Up position/notch (i.e. towards your Flap #1), and make this the minimum (i.e. click the left-hand 'Set'). The zone for Flaps 0 is now set. You don't change the minimum again (unless you need to adjust Flaps 0 zone on the axis).

2. Set the lever to a position a bit before the Flaps Full position/notch (i.e. at the other end, but not quite 'full'), and make this the maximum (i.e. click the right-hand 'Set'). The zone for Flaps Full (currently #2, but that will change as we add more détentes) is now set. You don't change the maximum again (unless you need to adjust the Flaps Full zone on the axis).

    So far it has been like a normal axis calibration. In fact, if you'd already calibrated the axis in this way before selecting 'Dt', the correct values would likely have already been set in Min and Max. All that now remains to do is set all the intermediate positions, to match zones around your own lever's détentes or notches.

    ALL this is done in the centre column, alone.

    The first flap to calibrate is #1. Use the spin control up and down keys (near top centre) to select flap #1 if #1 isn't already shown there.

3. Move the lever to just before the first flap down position/détente, i.e. close to the Flaps #0 zone, but above it. Then click the centre Set button. The value will record as one of the centre column values, and defines one end of the zone that will be considered setting Flap #1.

4. Move the lever to just after the same flap notch, and click the centre Set again. This second, higher, value will become the second centre column value.

    Congratulations, you have just defined the zone for Flap #1. The centre column heading is still #1, showing this as the selected flap number.

5. Now use the little spin control (top arrow) to increment the centre flap number (the one just to the right of the spin control) to the next value.

    Note that flap numbers can only be increased until the TOTAL number of flaps détentes recorded is the same as the number for the currently loaded aircraft (as shown in the Joysticks section below the flaps calibration section). So, for a 737, the headings end up as #0, #7, #8 (for a total of 9 positions 0–8).

6. Move the lever to just before the next flap position/détente. Then click the centre Set button. The value will record as one of the centre column values, defining one end of the zone which will be considered setting the selected Flap number.

7. Move the lever to just after the same flap notch, and click the centre Set again. This second, higher, value will become the second centre column value. And that's the next détente zone defined.

   Here's an example for a 737 fully set with the 9 needed positions (7 'centre' values, plus the minimum #0 and maximum #8). Here I've actually been reviewing the settings, and although all 9 are set (as shown by the #8 as the right column heading), I'm looking at the range for #5:

8. Repeat steps 5 to 7 until all the intermediate détentes are set—ALL with two values, all showing in the centre column when the relevant flaps number is shown above, next to the spin control.

That's it. If you make a mess, just press the 'Cl' button ("Clear") and start again. The values can be reviewed and re-adjusted at any time, individually (that's why the spin control can go up or down), but take care not to have overlapping zones else things may go a little odd. I you hear a 'ding' when pressing a Set then you have a value out of order. That is a no-no. They must all increase, left to right, flaps #0 to flaps #n.

All the détente calibration values are maintained independently of the simpler, analogue, calibration values. Pressing the 'Dt' button will swap between the two methods. You don't lose any détente numbers unless you press "Cl" to Clear them.

When you press Ok, the values are saved in the INI. For example:

   FlapStarts=–16384, –16219, –15450, –12975, –6980, –573,4606,15600,16217

   FlapEnds=–16256, –16200, –15417, –12950, –6950, –550,4644,15650,16384

The –16384 and +16384 extremes are fixed—you can change them in the INI but you really shouldn't, and there is no need to.

The 'Dt' button toggles this parameter:

   FlapDetents=Yes

If this is 'No' the détentes are still loaded, just not active.

Of course, as with all things in the Joystick Calibration, Buttons and Keys sections, you can have different sets for different aircraft. Otherwise the détentes you specify will be applied as well as possible to an aircraft with a different number of flap positions.

DETAILS OF MSFS's AXIS CONTROLS

Each axis that you can assign and calibrate is known to MSFS by a control name. Here are the names and brief descriptions of those that can currently be handled by FSUIPC:

Axis name in MSFS	Function	Notes
AXIS_AILERONS_SET	Joystick/yoke ailerons, also known as 'X' axis	Values used in MSFS run from –16384 (extreme right) to +16384 (extreme left), with 0 as centre
AXIS_ELEV_TRIM_SET	Elevator trim adjustment	–16384 to +16384, with 0 neutral trim
AXIS_ELEVATOR_SET	Joystick/yoke elevator, also known as 'Y' axis	–16384 (pull back/nose up) to +16384 (push forward/nose down), with 0 as centre
AXIS_MIXTURE_SET	Fuel mixture control	–16384 (lean) to +16384 (rich), but see also MIXTURE_SET below
AXIS_PROPELLER_SET	Propeller pitch control	–16384 (feathered) to +16384 (full), with no reverse. See also PROP_PITCH_SET below. FSUIPC allows you to map this onto the 4 separate propeller pitch controls (PROP_PITCHn_SET) if you need reverse pitch control on the same lever.
AXIS_RUDDER_SET	Rudder input, normally from pedals but also from twist function on some joysticks. Also known as the 'R' axis	–16384 (extreme right) to +16384 (extreme left), with 0 as centre. Note that if you fly with the rudder "spike" eliminator enabled (see the **Miscellaneous** section, earlier), then you will probably need to first calibrate your rudder, making sure you have adequate 'dead' zones at either extreme. If you don't do this then attaining maximum deflection may not be possible
AXIS_SPOILER_SET	Spoiler input, allowing precise spoiler positioning for excellent descent control.	–16384 (retracted) to +16384 (fully deployed). If enabled, FSUIPC maps this control to SPOILERS_SET, so it really matters little which you choose.
AXIS_THROTTLE_SET	Throttle input, whether from a throttle level, push/pull rod, or wheel. Also known as the 'Z' axis	Running from –16384 to +16384 for idle up to full thrust, this is a single throttle which operates all selected engines (the selection being by E on the keyboard followed by 1, 2, 3 and/or 4). No reverse is normally available on this control, but FSUIPC allows you to map this onto the 4 separate propeller pitch controls (AXIS_THROTTLEn_SET) if you want reverse thrust control on the same lever.
AXIS_LEFT_BRAKE_SET AXIS_RIGHT_BRAKE_SET	Separate brakes normally operated by toe pressing actions on the rudders.	–16384 (off) to +16384 (full on). MSFS seems to use an exponential growth function on these – a flattened initial response but steep at the fuller braking end.
AXIS_FLAPS_SET	Selection of flap setting by relative position	Flaps are set in 'notches' or 'détentes', not continuously. FSUIPC interprets the incoming axis value and sets the resulting output to the nearest exact notch position for the currently loaded aircraft. The range is –16384 (flaps up) to +16384 (full flaps)
MIXTURE_SET AXIS_MIXTURE_SET (FS2002/4)	Fuel mixture control	0 (lean) to +16384 (rich). If enabled, FSUIPC maps this control to AXIS_MIXTURE_SET, so it really matters little which you choose.
MIXTURE1_SET * MIXTURE2_SET * MIXTURE3_SET * MIXTURE4_SET * AXIS_MIXTURE1_SET (FS2002/4) AXIS_MIXTURE2_SET AXIS_MIXTURE3_SET AXIS_MIXTURE4_SET	Separate fuel mixture controls for each of up to 4 engines * These are the older ones which are normally ignored by FSUIPC's calibrations (via the Exclude option).	0 (lean or cut-off) via 8192 (idle) to +16384 (rich) Centering is provided for the 8192 value so that turbo controls with détentes on the mixture levers can be calibrated with a usable 'dead' zone. By default, however, this dead zone is eliminated by having both upper and lower input values for it set the same.
PROP_PITCH_SET AXIS_PROPELLER_SET (FS2002/4)	Propeller pitch control	–4096 (reverse) via 0 (feathered) to +16384 (full). But if enabled, FSUIPC maps this control to AXIS_PROPELLER_SET, so it really matters little

		which you choose.
PROP_PITCH1_SET * PROP_PITCH2_SET * PROP_PITCH3_SET * PROP_PITCH4_SET * AXIS_PROPELLER1_SET AXIS_PROPELLER2_SET AXIS_PROPELLER3_SET AXIS_PROPELLER4_SET	Separate fuel mixture controls for each of up to 4 engines * These are the older ones which are normally ignored by FSUIPC's calibrations (via the Exclude option).	–4096 (reverse) via 0 (feathered) to +16384 (full). All four are operated together if you map a single axis assigned to AXIS_PROPELLER_SET or PROP_PITCH_SET to these. FSUIPC handles these identically.
SPOILERS_SET AXIS_SPOILER_SET	Spoiler input, allowing precise spoiler positioning for excellent descent control.	0 (retracted) to +16384 (fully deployed).
THROTTLE1_SET * THROTTLE2_SET * THROTTLE3_SET * THROTTLE4_SET * AXIS_THROTTLE1_SET AXIS_THROTTLE2_SET AXIS_THROTTLE3_SET AXIS_THROTTLE4_SET	Separate throttle controls for each of up to 4 engines, with reverse thrust capability * These are the older ones which are normally ignored by FSUIPC's calibrations (via the Exclude option).	–4096 (full reverse**) via 0 (idle) to +16384 (full forward). All four are operated together if you map a single axis assigned to AXIS_THROTTLE_SET to these. FSUIPC handles these identically ** The actual negative value for "full reverse" is defined in the aircraft parameters (the .AIR file). FSUIPC calibrates reverse to match the aircraft, so you will see different values here, not just the nominal '–4096'.
~~AXIS_PAN_HEADING~~ ~~AXIS_PAN_PITCH~~ ~~AXIS_PAN_TILT~~	~~Used in MSFS virtual cockpit mode.~~	~~Range –16384 via neutral 0 to +16384~~
~~AXIS_SLEW_AHEAD_SET~~ ~~AXIS_SLEW_ALT_SET~~ ~~AXIS_SLEW_HEADING_SET~~ ~~AXIS_SLEW_SIDEWAYS_SET~~	~~Slew mode controls~~	~~Range –16384 via neutral 0 to +16384~~ ~~Note that these are only used in Slew mode. FSUIPC cannot see them in flight mode.~~
COWLFLAP1_SET COWLFLAP2_SET COWLFLAP3_SET COWLFLAP4_SET		Assign via FSUIPC Axis assignments, calibrate in the FSUIPC joystick calibration pages.
RUDDER_TRIM_SET AILERON_TRIM_SET		Assign via FSUIPC Axis assignments, calibrate in the FSUIPC joystick calibration pages.

RESETTING DEFAULTS

Please note that resetting the FSUIPC defaults (using the menu entry in the "Options" page) does not change the joystick calibration system. This is deliberate, as it could be very annoying having spent a while carefully setting up the joysticks perfectly just to have them obliterated through selecting this item once. If you want to switch any of this off, either de-select them individually (pressing the Reset button disables the FSUIPC axis processing actions and restores defaults), or, to do this for all at once, edit the FSUIPC7.INI file after closing FSUIPC and delete the entire section called [*JoystickCalibration*].