

General description of FSUIPC main functions

FSUIPC cannot and does not convert everything that may ever have been known in FS98 to work in the same way in FS2000, and it gets harder in FS2002. It does its best, and will improve as it is developed and as more is found out about FS2002. Many actions in FS98 that were simply triggered by writing into the global data area simply will not work that way on FS2000 or FS2002. For each specific action which may be needed and which doesn't simply map to a location, FSUIPC has to trap the changes and call routines in different parts of FS to cause the action to occur. So far this works with throttles and many of the other analogue inputs, and with some other things, but by no means everything (yet).

That understood, FSUIPC provides the following additional facilities when used with FS2000/FS2002:

- Converts FS98 offsets, for data in the FS globals area, into appropriate offsets for the same data in FS2000. This applies to those that are known to have moved but which are still available and located successfully in FS2000, or even FS2002. Note that sometimes the changes vary even in FS2000: for example the N1% and N2% values for Jets are in the FS98 positions for transposed FS98 aircraft but swapped over for FS2000 native aircraft. FSUIPC deals with this particular difference (but maybe not for FS2002 – it is not expected that FS98 aircraft will work well in FS2002).
- Obtains some information such as ambient wind details, jet EPR, Fuel Flow, and other engine related data which is not otherwise available in the FS2000/2002 globals at all, or available for some aircraft types and not others.
- Provides operating analogue inputs to FS2000/2002 from locations for throttles, propeller pitch, fuel mixture and many other aspects which otherwise don't occur in FS2000/2002 because the data there is not acted upon when changed. Amongst the controls here are gear, brakes, spoilers, flaps and of course the primary flight controls.
- Constructs weather data structures for FS2000 and FS2002's weather engines from FS98-style weather details placed into the old FS98 global weather positions. This allows programs such as Real Weather, FS_Meteo, Flight Director and SquawkBox to control the FS2000/FS2002 weather as they did in FS98, or better.
- Detects when FS2000 or FS2002 downloaded 'real weather' is in operation locally, and decodes this for application programs to use. If a weather control program wants to change the weather, FSUIPC automatically clears the local weather so that the external control can be implemented.
- Optionally patches the FS2000/2002 adventure interpreter so that the FS98 weather variables again contain relevant weather data (assuming the other weather features are also left enabled). See the section later, entitled "Weather Data for Adventures".
- Provides additional joystick calibration and centring facilities and enables fully proportional analogue toe brakes. To be used.

None of these functions are performed by WideServer (part of the WideFS package). Because of the complications surrounding the operation of throttles, flight controls and other inputs, and of course the weather system, WideServer depends upon FSUIPC to perform ALL accesses to FS's innards. It is therefore important, when both modules are used, to make sure they are compatible.

Options in the FSUIPC.INI file (*for FS2000 and FS2002 primarily*)

All of the interesting options can be controlled through the Settings window obtained by selecting the FS2000/2002 Modules menu, then FSUIPC (ALT, M, then F). This is the recommended way, and allows changes 'on the fly'. Changes made in that dialogue are recorded in a file so that they are retained for the next re-load.

These options are all recorded in the [General] section of FSUIPC.INI, which is an editable text file initially created for you in the Modules folder. They are mostly weather processing options and also mostly only applicable to FS2000 and FS2002 (and possibly CFS2). Those marked (*) can be controlled (i.e. overridden) by external programs interfacing to FSUIPC, unless this is prevented by the **ExternalOptionsControl** parameter.

Only those parameters shown underlined are not adjustable within the Settings window.

General weather options

PatchWeatherToADV: This parameter controls the facility, specific to FS2000/2002 only, to patch the variable table in ADVDRV.DLL (the adventure interpreter) so that the weather variables report the same sort of values for the same sort of weather as did FS98. Only set this to “No” if you do not want such FS98 compatibility for your adventures. Note that successful patching is Logged (if logging is enabled at all), as is an unsuccessful attempt. The latter may occur if Microsoft releases any changes to ADVDRV.DLL.

AutoClearWeather: FSUIPC will, by default, automatically operate the "Clear All Weather" function in FS2000/2002 if local weather is in force and:

- (a) An FS98 weather control program changes the weather, or
- (b) The "Force Weather" hot key is used (see next parameter), or
- (c) The "Clear All" command is received on the Advanced Weather Interface.

If this automatic action is not required, set this parameter to “No”.

ForceWeatherKey: This allows you to assign a key press which, when used, will force-update the weather. When you use this keystroke it will re-form the FS2000/2002 weather from the last weather received from the external application. It will even remember the last weather received from the application after the latter is terminated. The second time it is used *with no intervening Weather change* it clears the weather completely, just as if FS2000's “Clear All Weather” button had been pressed.

Note that if the “AutoClearWeather” option is disabled you may need to clear the weather manually in the FS2000/2002 dialogues before the hotkey will restore the external weather.

The keystroke is defined as in Flight Simulator's own controls, and documented in my FS98 and FS2000 Controls documents. For example, I use (and recommend) "CTRL+SHIFT+W" which would be

ForceWeatherKey=87,11

The same control codes are used in FS2002.

SendWeatherInterval: In FS2002 (only), the weather provided by the built-in ATIS reports does not necessarily abide by the current weather prevailing. To get these reports updated, FSUIPC has to send weather change signals to other parts of FS2002. Unfortunately this results in an update of the ATIS identifier, which in real life only occurs at hourly intervals. So FSUIPC offers two options: this one, to set the *minimum* interval between weather updates for ATIS (in seconds), and the next one, to avoid sending any such updates unless the weather has changed noticeably.

The default interval is set for 60 seconds. You shouldn't set it too short, but you may want to set it to a larger value. But be aware that this interval is imposed even if you transit to another METAR station area, but FSUIPC does recognise when you've loaded a new flight, so the interval is not imposed at this time.

SendWeatherAlways: By default the ATIS weather is only sent when there are noticeable changes in the weather (the criteria are listed below). By setting this parameter to ‘Yes’ you can make FSUIPC send the updates at the specified intervals whether there's been any change or not.

The criteria for deciding on weather changes are as follows. You should note that all these refer to *surface* weather, not necessarily the weather at the aircraft. Furthermore, if you are using FSMeteo and have set the destination weather, *and* FSMeteo has now supplied this for ATIS reports, it will be this weather which is used in the comparison even though FS's own ATIS will be reacting to current weather, not the station weather.

- Surface temperature altitude same (this is used for METAR station elevation)
- Temperature changed by more than 3C
- Wind speed changed by more than 5 knots
- Wind direction changed by more than 5 degrees
- Cloud base changed by more than 500 feet
- Cloud cover (lowest layer) changed by more than 2 oktas
- Visibility changed by more than 50% of lower value
- Barometric Pressure changed by more than 5 mb

Winds

WindTransitions (*): If you enable this option FSUIPC will operate all FS2000/2002 winds other than those in ‘local weather’ mode (provided by the downloaded ‘real weather’ feature) in such a way that the transition across wind layers is reasonably smooth. It does this by setting only one wind layer into FS2000/2002—a very deep surface

wind layer. The wind speed and direction is then programmed into this layer on a second-by-second basis according to the actual requested wind layer prescribed by the weather control program and the current plane altitude. At altitudes nearer to a layer boundary than 250 metres the actual speed and direction is computed proportionally.

Note that if 250 metres is more than 10% of the current layer's thickness, then that 10% is used instead. This allows some amount of wind shear to be set if required by defining a very narrow upper wind layer.

WindSmoothness: This facility only operates when WindTransitions are enabled, but unlike WindTransitions, it also works for downloaded 'local weather'. It allows the wind changes to be restricted to a maximum of so many knots and so many degrees per second—with the default set to 5 (knots or degrees), which seems to work quite well. It is designed to prevent sudden wind changes when the weather control program selects a new METAR station, or the user loads a new METAR report. The feature does not operate when the aircraft is on the ground (or being slewed in the air from a ground start). To switch the smoothing off, set this parameter to 0.

ExtendTopwind: This option extends the highest current wind layer to operate all the way up to 100000 feet. This is really intended as a stop-gap for downloaded real weather, which only supplies a thin surface wind layer and no upper winds.

MaxSurfaceWind: This allows the surface wind to be limited to a specified maximum wind speed, in knots. This facility is disabled if the value assigned here is 0. It applies to winds from any source.

WindDiscardLevel: This parameter sets a wind speed above which inputs from an external weather control program, using the FS98 interface (*not* the Advanced Weather Interface) are ignored. The default for this value is 400 knots. If a weather control program tries to set a wind speed above this, it is ignored and the previously set speed for this wind layer is retained. (This parameter is provided specifically to prevent problems occurring with programs using corrupted data from an Internet download or other problems). Set this parameter to 0 to disable this check altogether.

WindLimitLevel: This parameter sets a limit on the wind speed which can be accepted from an external weather control program, using the FS98 interface (*not* the Advanced Weather Interface). The default for this value is 200 knots. If a weather control program tries to set a wind speed above this (but below the "WindDiscardLevel" above), it is ignored and 200 knots is set instead. Set this parameter to 0 to disable this check altogether.

WindShearSharp: Set to 'No' to make FSUIPC set the Wind Shear to the default (minimum) setting. FSUIPC normally sets this to "Sharp" to avoid horrible spurious winds occurring during the transition, apparently an FS2000 bug (which may or may not be fixed in FS2002). [Note: if "WindTransitions=Yes" there are no wind layer transitions seen by FS2000/2002, so this parameter then does nothing].

UpperWindGusts (*): Set to 'Yes' to make FSUIPC copy the upper wind gust information provided by the weather control program. These are normally suppressed by FSUIPC because upper winds aren't gusty, and FS2000's gusts seem pretty wild anyway. *Note: this parameter is not operational if SuppressAllgusts has been enabled.*

SuppressAllGusts: Set this to 'Yes' if you feel that FS2000's simulation of wind gusts is unrealistic, and cannot be corrected by the adjustments in FS2000.CFG suggested elsewhere. If this is set to "Yes" then the **UpperWindGusts** parameter is ineffective.

GustsRelative: Set to 'No' to set gust velocities as the maximum gust speed, which is what FS2000 *should* be using according to the way the Winds dialogue works. The default setting (Yes) makes FSUIPC set gust speeds to the *difference* between the upper gust speed and the normal wind speed. This is to get around an apparent bug in FS2000 where it seems to add gust speeds to the wind speed rather than treat them as a maximum. (*See the hints in the main User Guide on getting FS2000 gusts working better*). The correct setting for FS2002 has not yet been determined.

WindTurbulence: Set this to 'Yes' to make FSUIPC generate some random turbulence in all wind levels. This will range from none to extreme, but it will normally stay fairly mild. If this is set it overrides any other settings, from FS2000/2002 "real weather" or from an external weather program. It will vary over time as well.

ToggleTaxiWindKey: This allows you to assign a keypress which, when used, will swap the current surface layer wind speed and gust setting with a wind speed of 1 knot and no gusts. Using the same hot key again will restore the original speed and gust setting. If it is used when the current wind is *not* related to the requested surface wind layer then nothing is changed (but a 'beep' may be heard as a warning). *Note that this is inoperative if AutoTaxiWind is enabled*

This feature can be useful to avoid the excessive weather-vaning whilst taxiing. It works with any type of weather applied to FS2000/2002, but only on the lowest wind layer. However, the **AutoTaxiWind** is probably more suitable for most uses.

The keystroke is defined as in Flight Simulator's own controls, and documented in my FS98 and FS2000 Controls documents. For example, I use (and recommend) "CTRL+SHIFT+T" which would be

ToggleTaxiWindKey=84,11

The same control codes are used in FS2002.

AutoTaxiWind: This feature operates the 1 knot wind automatically, setting it to 1 knot when the plane is on the ground, and allowing the correct wind to transition in (according to the WindSmoothing setting above) after take-off. If this option is enabled the manual taxi wind on/off switching is disabled.

WindAjustAltitude: (Apologies for mis-spelling). Set to 'Yes' if FSUIPC should add the value specified in **WindAjustAltitudeBy** to all the wind layer boundaries specified by an external weather control program using the FS98 interface (*not* the Advanced Weather Interface, as used by FS_Meteo since version 4.77).

WindAjustAltitudeBy: See previous parameter. This is in feet and defaults to 2000.

WindSetVariance: This option when enabled makes FSUIPC convert any wind turbulence into wind "variance". This latter feature seems to at least make FS2000 do something (actually it introduces random variations in the wind direction), whilst the turbulence options seem ineffective. Whether this is needed in FS2002 as well hasn't been determined, but it seems likely. There are few changes in the weather engine.

WindVarFactor: This sets a value from 1 to 20 (default 7) which effectively controls the percentsge effect of wind turbulence on wind variance, when The **WindSetVariance** option is enabled. The default of 7 is rather less that the value I felt was realistic, which would be 10 (equating to 100%).

MagWindsToFST: If this is set 'Yes' then the wind data supplied to FSTraffic gives the wind direction in degrees Magnetic. Otherwise this is in degrees True, as it would be without FSUIPC running.

UpperWindsToFST: Set this to tell FSUIPC to send a fixed surface wind direction to FSTraffic when the aircraft is above a specified altitude. This is needed by some tracks for airways. As an example:

UpperWindsToFST=270,18000

Will cause all surface winds reported to FSTraffic to be from 270 degrees (Mag), once the aircraft is above 18000 feet.

Visibility

MinimumVisibility: This parameter, which defaults to 0 (meaning it is inactive), is used to prevent *any* weather source setting a visibility below a specified minimum. The value is set in hundredths of a statute mile (i.e. 100 = 1 mile). Note that there may be a short delay (possibly a second) after a new low visibility has been applied before it is detected and corrected by FSUIPC.

MaximumVisibility: This parameter, which defaults to 2000 (20 miles), is used to prevent *any* weather source setting a surface visibility above a specified maximum when there is any cloud layer with more than 2/8ths cover. The value is set in hundredths of a statute mile (i.e. 100 = 1 mile). Note that there may be a short delay after a new high visibility has been applied before it is detected and corrected by FSUIPC. The parameter is only effective if the value is greater than the MinimumVisibility parameter.

MaximumVisibilityFewClouds: This is the same as the previous parameter, except that it gives the maximum to be used when there are no cloud layers of more than 2/8ths cover. It defaults to 6000 (60 miles) The idea is that the extended visibility gives bluer skies by day and more stars by night (but lower frame rates. Sorry, you can't win every way <G>).

LowerVisAltitude: When the visibility is set from an FS2000/2002 source, such as its downloaded 'real weather', or by the user setting it through the weather dialogues, there is already an upper altitude, above which global visibility values take over (unless influenced by FSUIPC's graduated visibility facility, described below). However, for visibility controlled by external programs, via the FS98-compatible interface, there is no such altitude so one has to be inserted by FSUIPC. This is specified by LowerVisAltitude in feet, which defaults to 6000.

GraduatedVisibility (*): With this enabled FSUIPC provides a smooth change in visibility from the upper altitude of the surface level visibility to a specified upper visibility at another, specified, upper altitude. The two parameters, UpperVisibility and UpperVisAltitude, control this. The surface visibility extends up to the LowerVisAltitude (above) for visibility controlled by external programs using the FS98 interface, but is controlled by FS2000/2002 for its "real weather" or for visibilities set through the FS2000/2002 dialogues.

UpperVisibility: This parameter, which defaults to 6000 (60 statute miles), is used to prevent *any* weather source setting a visibility above a specified maximum. The value is set in hundredths of a statute mile (i.e. 100 = 1 mile). If GraduatedVisibility is enabled, it is used in conjunction with the next parameter.

UpperVisAltitude: This is only used when GraduatedVisibility is enabled, and sets the altitude by which the UpperVisibility should be attained. Above this altitude the visibility stays fixed at this value. The default UpperVisAltitude is 25000 feet.

ExtendMetarMaxVis (*): This checks the visibility being set and adjusts it in three specific circumstances, as follows:

1. If the FS98 program sets it to a value between 99.95 and 100.04 miles, it is reset to 6.20 miles. This is in order to rectify the results from any programs that take the 9999 metre maximum METAR visibility and transmit it literally as a number of 1/100ths of statute miles.
2. If the value is then in the range 6.15 to 6.24 miles (i.e. close to the 9999 metres maximum of a metric METAR), it is adjusted to a random value between 6.20 miles and the current maximum value (which will either be the MaximumVisibility parameter value, or 150 miles).
3. If the value is between 9.95 and 10.05 miles (i.e. close to the 10 statute mile maximum of a U.S. METAR), then it is adjusted to a random value from 10 miles to the current maximum (which will either be the MaximumVisibility parameter value, or 150 miles).

Note that the random addition is computed only once every five minutes, to avoid constant changes in visibility should the weather control program re-write the value from time to time.

Clouds and Precipitation

GenerateRain (*): Set this option to 'No' to stop FSUIPC doing any semi-random rain/snow generation, assuming the external weather program is not controlling this in any case.

GenerateCirrus (*): Set to 'No' to stop the occasional extra cirrus layer being added automatically by FSUIPC. Set to 'Force' to make FSUIPC add the occasional cirrus layer even if an external weather control program turns the option off.

CloudforJetTrails: Set to 'Yes' to make FSUIPC often add a 1/8th cover cumulus layer, high (but below the added cirrus). This is in order to allow Jet Trails to be produced by FSClouds 2000. (Hopefully this will also work for FS2002).

OneCloudLayer: This defaults to 'No'. Set it to 'Yes' to prevent there ever being more than one layer of clouds. This may help get better performance on slower machines. It won't help much on faster machines.

ThinClouds: This defaults to 'No'. Set it to 'Yes' to prevent any single cloud layer being thicker than 1000 feet (or whatever is set in the CloudThinness parameter, below), from the nominal cloud base to its top (not including any variations which may be set). This may help get better performance on slower machines. It won't help much on faster machines.

CloudThinness: This allows the limit applied by the ThinClouds option to be changed. It defaults to 1000 feet. The range accepted is 100–9999 (feet).

StormsAutomatic (*): Leave as 'No' to allow suitably programmed weather control programs to use all three FS98 cloud layers for any types of cloud. With this option set to 'Yes' the "thunderstorm" layer can only be used for storms, as it was in FS98.

StormProbability: A value from 0 to 100 representing a percentage probability of a storm. For a storm to be generated the winds and clouds must also be adequate—as defined in **StormParameters** below. This is checked every two minutes. The same probability is used to determine when a storm dissipates, after its minimum duration.

StormParameters: This should be used with care. The value provided is used to determine what conditions must prevail before a 'random' storm is even considered. It is used as follows:

StormParameters=WWCBBHD

All 7 characters are decimal,

WW	Minimum surface wind speed needed (in knots). Default is 10.
C	Minimum cloud cover (1–8, default 3).
BB	Maximum cloud base AGL, in thousands of feet (default 05, i.e. 5000 feet).

- H Minimum cloud thickness, in thousands of feet (default 3, i.e. 3000 feet).
- D Minimum duration of storm, in minutes, with 0 meaning 10 (default 0, i.e. 10 minutes)

Regardless of the value of D, the duration may be extended at random, with a probability then of it ending being the same as that of it starting. Of course if the cloud or wind conditions change the storm may end much earlier than the specified minimum.

The default parameter (applicable even if it not shown in the .ini file) is therefore:

StormParameters=1030530

StormMinTemp: This is an additional Storm Parameter, and sets the minimum *surface* air temperature at which the random storms will be allowed to occur. Default is 10 (Celsius), range –99 to 99.

CloudTurbulence: Set this to ‘Yes’ to make FSUIPC generate some random turbulence in all cloud layers. This will range from none to extreme, but it will normally stay fairly mild. If this is set it overrides any other settings, from FS2000 “real weather” or from an external weather program. It will vary over time as well.

CloudIcing: Set this to ‘Yes’ to make FSUIPC generate some random icing in clouds. This will range from none to extreme, but it will normally stay fairly mild. If this is set it overrides any other settings, from FS2000 “real weather” or from an external weather program. It will vary over time as well.

ApplyVisFix: By default FSUIPC will attempt to stop the “stuck low visibility” or white-out problem, apparently due to a bug in FS2000’s Weather.dll. It does this by checking the effective visibility once a second and trying to progressively correct it if it is lower than it should be when the aircraft is not inside a cloud layer. Note that the detection of cloud layers is not 100% reliably done at present (I’m working on it!), but at least the only bad symptom should be an occasional higher visibility than you’d expect. Note that it isn’t proven that this ‘fix’ works in 100% of cases where a whiteout may occur, but it certainly reduces their frequency by a large amount!

When Microsoft fix the bug causing the problem, simply disable this work-around by setting ApplyVisFix=No. Whether the bug still exists in FS2002 has not been determined at the time of writing.

Temperature

CopyDewPtToDayNightVar: When there are multiple temperature layers, the ATIS reports of FS2002 (and perhaps FS2000) get the Dew Point wrong: they report the Day/Night variation as the Dew Point. To get around this, FSUIPC normally copies the Dew Point to the Variation. The latter is not actually used in FS2000 or FS2002 in any case. If you want to stop this occurring, just set this parameter to ‘No’.

Other options

ExternalOptionControl: Set this to ‘No’ if you want to retain control over all the settings for FSUIPC. Normally some of the original options are available for an external weather control program to set according to its needs.

PatchSimApAlt: This option provides two ‘improvements’ to the FS2000 autopilot. First, it patches SIM1.SIM (the main aircraft simulating part of FS) to correct some inaccuracy in the autopilot’s altitude holding capability. The inaccuracy occurs when flying Flight Levels and increases with the difference between the altimeter setting (e.g. the standard pressure setting of 29.92” or 1013mb for Flight Levels) and the actual barometric pressure at sea level (QNH). This action is not applied to FS2002 as it seems to have been fixed in the new simulation engine.

Second, the same option corrects the vertical speed setting if it is set to climb when the aircraft would need to descend, or vice versa. It does this by inverting the sign of the vertical speed setting. It only does this when altitude acquire/hold is enabled, so that vertical speed control by itself is not affected. The correction is also not applied if the target altitude is set to a value over 65000 feet—a trick being used by some panels to provide V/S controlled ascents and descents.

If neither of these functions are required, set this parameter to ‘No’.

PlanLoadNoPosition: This option changes the behaviour of the FS2000/2002 facility to load flight plans into the GPS. If you set “PlanLoadNoPosition” to Yes, the FS2000 plan loader will *not* position the aircraft for you. Many folks prefer this as they like to start the flight with the aircraft parked on the ramp, and taxi to the correct runway themselves. This option is defaulted off (=No) to avoid confusing users already used to and happy with FS2000 working as it does now. Set it to “Yes” if you would prefer the facility not to move the aircraft.

MagicBattery: This allows electric power to be provided even when the FS2000/2002 battery is completely flat, so getting over the apparent error in the airliners, which makes it discharge far too quickly before engine start.

MagicBatteryForced: The Magic Battery facility normally operates only when both Battery *and* Avionics switches are **on**. This is a change for the way the original facility operated. If you want it to operate solely on the Battery switch, set this parameter to “Yes” (or check the option in FSUIPC’s Technical page).

ExtendedJoyCalib: This simply enabled the extra three Joystick calibration pages in the Settings and Options display. The actions carried out are not affected, only whether the settings are shown or not.

N1N2asFS98: Set this option (add the line if it isn’t there) to “yes” if (and only if) you want FS2000 to start up with an FS98 jet aircraft with Engines off. It makes FSUIPC assume that the N1% and N2% values are provided as they are in FS98 (i.e. reversed), rather than as they have been ‘corrected’ in FS2000 for FS2000 aircraft. FSUIPC does do this automatically, but it cannot differentiate the two cases until the engines are running.

Whether this is of any use in FS2002 isn’t known at present.

AutoTuneADF: This controls an option to ‘auto-tune’ the ADF radio. If this is enabled, when FSUIPC detects no NDB signal being received it alternates the fractional part of the ADF frequency between .0 and .5 every seven seconds or so. This allows external cockpits built with only whole-number ADF radio facilities to be used in areas like the U.K. which have many NDB frequencies ending in .5.

AxisCalibration: This facility deals with inputs to the rudder, aileron and elevator axis offsets, via the FS98 offsets to the IPC interface. These values are subject to a range check, and always scaled down if this range is exceeded. The correct limits are –16383 to +16383.

Additionally, axis inputs can be scaled *upwards* to meet this extent, if required. To do this set:

AxisCalibration=Yes

By default with this option selected some flattening is applied to the values so that the response is not so vigorous near the centre (0). To calibrate the axes you must move all three controls to their maximum extends on each fresh load of FS2000.

Alternatively, you can set "AxisCalibration=Set". This operates as above, but adds a new section to the .ini file, thus:

```
[AxisCalibration]
Rudder=<max>,<slope>
Elevator=<max>,<slope>
Aileron=<max>,<slope>
```

The <max> values are those which are scaled to 16383, whilst the <slope> values control the amount of flattening in the centre: from 0 (no flattening) to 100 (maximum flattening. note that the flatter the centre, the steeper the sides, so it is always a compromise.

The default “slope” values are 50, 40, 40 respectively, for the three axes.

Once this calibration has been done and the section in the ini file produced (or added manually), there is no need to re-calibrate on each new FS reload. The "AxisCalibration" parameter resets automatically to "Yes".

Note that the "AxisCalibration=No" setting is equivalent to setting "Yes" and adding the section:

```
[AxisCalibration]
Rudder=16383,0
Elevator=16383,0
Aileron=16383,0
```

However, if these values are exceeded during an FS2000 session, the new maxima will replace any values in the ini file.

MainMenu=&Modules: This parameter controls which main (top-level) menu entry in Flight Simulator is used to access the FSUIPC Settings screen. The default, as shown here, is the Modules menu. Note the “&” character, which tells Windows which letter in the name is used for the keyboard accelerator (as in “Alt+M” here).

If you prefer to have FSUIPC accessed through, say, the Flights menu, then you can change this to &Flights. Note that the spelling and “&” characters *must* match whichever menu you are adding to, else a new one will be created instead. Foreign language versions of FS will have differences too, remember.

You can have FSUIPC Settings accessed *directly* from the top level menu if you like. To do this simply choose a unique menu name and add “...” to the end. FSUIPC will take this to mean that you want direct access. For example:

MainMenu=FS&UIPC ...

Will create a top-level menu entry “FSUIPC ...” which will lead directly to the Settings window. The accelerator here is U, because the F is already taken (for “Flights”)—the Settings window can then be obtained very quickly by just Alt+U.

SubMenu=&FSUIPC ...: This supplies the name and the keyboard accelerator character (the one following the “&”) which will appear in the selected ‘Main Menu’ entry (see previous item) and which leads directly to the FSUIPC Settings window. If the Main Menu itself is made to lead directly to the Settings this entry is ignored.

FixWindows: set to ‘Yes’ prevents cockpit windows resizing and moving. This facility can also be used in FS98 but as there’s no in-program options for FS98 you must make sure the panel and scenery windows are exactly as you want them to be before setting this parameter.

SmoothPressure: set to ‘Yes’ to smooth barometric pressure changes by limiting the changes from external programs to 1 millibar ever so many seconds. The number of seconds is given by **PressureSmoothness** which defaults to 5 and can have any value from 1 to 30, inclusive.

SetStdBaroKey: This allows you to assign a keypress which, when used, will set the ‘Kollsman’ window on the Altimeter to the standard pressure, 29.92” or 1013.2mb. This is used when flying ‘flight Levels’.

The keystroke is defined as in Flight Simulator’s own controls, and documented in my FS98 and FS2000 Controls documents. For example, I use (and recommend) "CTRL+SHIFT+B" which would be

SetStdBaroKey=66,11

The same control codes are used in FS2002.

TCASid: FSUIPC supplies data on the FS2002 additional “Artificially Intelligent” (A.I.) aircraft flying in the neighbourhood, for external TCAS or mapping programs to display. Normally the aircraft is identified by its Airline and Flight number, if there is one, otherwise by the Tail number.

However, other types of identification string can be chosen instead. In particular, the optional labels placed on the aircraft by FS in the scenery view only shows tail numbers, so if you want to match them up you’d want to set this parameter to “Tail”. The full list of options here is:

Flight	for airline+flight, or tail number, as available (default)
Tail	for tail numbers only
Type	for the “ATC type”, generally only the Make
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

The utility “TrafficLook” is supplied—you can see the difference in its display.

SetSimSpeedX1: optionally sets a Hot Key which when used resets the simulation rate to x1 (i.e. normal). The keystroke is defined as in Flight Simulator’s own controls, and documented in my FS98 and FS2000 Controls documents. The same control codes are used in FS2002. As an example, for "CTRL+SHIFT+S" this would be

SetSimSpeedX1=83,11

Logging facilities (FS98, FS2000 and FS2002)

These options work with FS98, FS2000 and FS2002. On FS2000 and FS2002 they can be controlled ‘on the fly’ from the FSUIPC dialogue window (select the Modules menu then FSUIPC, ALT, M then F). With any logging option enabled FSUIPC produces a text file called FSUIPC.LOG in the Modules folder. Entries in the log are timed, from the start of the FS session. The time is in milliseconds and appears on the extreme left of each line.

Please use the logging facilities to check things before reporting problems or omissions in FSUIPC, and supply an appropriate log file (or extract) properly zipped up with such reports.

Note that log files can get very large if all the options are turned on. Keep test flights short. You can read log files whilst flying provided you use a reader which shares access (like recent Notepad programs), or use the ‘NewLogKey’ described below to close logs and start new ones.

All Log control parameters go into the [General] section of FSUIPC.INI. None are included by default.

LogWeather=Yes: Logs weather data. This will log incoming data, set by a weather control program, on FS98 as well as FS2000 and FS2002. On FS2000/2002 you will also get the actual weather data constructed by FSUIPC in FS2000/2002 terms. Then you get the weather read out by FSUIPC and lastly placed back into the globals for applications to read. Incoming weather control data on the Advanced Weather interface for FS2000/2002 is also logged in full.

LogWrites=Yes: Logs the global 'writes' received from applications, with global offset address and data size, plus all bytes of data. The offsets shown are the ones used by the application. [Take care: the Log file may get very large!]

LogReads=Yes: Logs the global 'reads' received from applications, with global offset address and data size, plus all bytes of data. The offsets shown are the ones used by the application. [Take care: the Log file may get very large!]

LogExtras=Yes: This logs additional technical data about the inner workings of FSUIPC, the nature of which will vary from time to time according to needs. There is nothing here that would be of interest to the user, but when investigating problems users may be asked to enable it so that the logs returned can be more meaningful in solving them. Do not fly extensively with this option enabled or you will fill up your disk and probably compromise the simulator's performance!

NewLogKey, StopLogKey: These allow you to assign keypresses to close the current Log file (if logging was enabled), and (in the case of 'NewLogKey') start a new one. The 'NewLogKey' will start a new log even if one wasn't enabled by default, so between them these two keys give complete control over the logging. (Note that in FS2000 both actions are also available in the FSUIPC dialogue window).

The current log file is always called FSUIPC.LOG. The others are named in numerical order FSUIPC.1.LOG, ... 2.LOG, ... etc.

If 'NewLogKey' is used when logging was not originally enabled (i.e. none of the Log options above were selected), this key will not only start logging, but will also assume the 'LogWeather=Yes' option (but none of the others).

The keystrokes are defined as in Flight Simulator's own controls, and documented in my FS98 and FS2000 Controls documents. For example, I use "Shft+Ctrl+L" and "Shft+Ctrl+O" (for "Log" and "Off" respectively) which would be

NewLogKey=76,11
StopLogKey=79,11

The same control codes are used in FS2002.

You don't need to define both keys, but note that if you stop the logging but have no "NewLogKey" defined you can only restart logging by terminating and reloading Flight Simulator (except with FS2000/2002, where you can use the button in the FSUIPC dialogue window).

Assignment of FLAPS_SET control (for FS2002)

The Flaps calibration facility in FSUIPC cannot be used directly in FS2002, because the convenient FLAPS_SET control can no longer be assigned to an Axis in FS2002.CFG. To get around this, you can select any one of the following FS2002 Axis controls (obviously one you are not otherwise using!), assign it (by name) to your Axis in FS2002.CFG, tell FSUIPC to use this by declaring its numeric value, as about to be explained, *then* calibrating it in FSUIPC's Joystick section (as the FLAPS control, on page 6 of 6).

The AXIS controls at your disposal are listed below. Use the chosen name in FS2002.CFG and the relevant number in a new parameter in the [JoystickCalibration] section of FSUIPC.INI, thus:

FlapsSetControl=<control number>

This is set to 0 to disable the Flaps Set interception.

Valid Axis Controls (N.B. Not all tested. Please advise if you find any which don't work in FS2002):

AXIS_AILERONS_SET	65763
AXIS_ELEV_TRIM_SET	65766
AXIS_ELEVATOR_SET	65762
AXIS_LEFT_BRAKE_SET	66387
AXIS_MIXTURE_SET	66292
AXIS_MIXTURE1_SET	66422
AXIS_MIXTURE2_SET	66425
AXIS_MIXTURE3_SET	66428
AXIS_MIXTURE4_SET	66431

AXIS_PAN_HEADING	66504
AXIS_PAN_PITCH	66503
AXIS_PAN_TILT	66505
AXIS_PROPELLER_SET	66291
AXIS_PROPELLER1_SET	66421
AXIS_PROPELLER2_SET	66424
AXIS_PROPELLER3_SET	66427
AXIS_PROPELLER4_SET	66430
AXIS_RIGHT_BRAKE_SET	66388
AXIS_RUDDER_SET	65764
AXIS_SLEW_AHEAD_SET	65867
AXIS_SLEW_ALT_SET	65870
AXIS_SLEW_BANK_SET	65871
AXIS_SLEW_HEADING_SET	65869
AXIS_SLEW_PITCH_SET	65872
AXIS_SLEW_SIDEWAYS_SET	65868
AXIS_SPOILER_SET	66382
AXIS_THROTTLE_SET	65765
AXIS_THROTTLE1_SET	66420
AXIS_THROTTLE2_SET	66423
AXIS_THROTTLE3_SET	66426
AXIS_THROTTLE4_SET	66429

Multiple Joysticks

On FS2000/2002, using the Joystick sections of the FSUIPC dialogue to calibrate the main flight controls, FSUIPC can also accept up to four different control inputs for each main flight control, treating them equally. You can have up to 4 aileron, elevator, rudder, throttle, left and right brake controls. FSUIPC takes the value from the input giving *maximum* deflection from ‘neutral’ or ‘idle’. There’s no averaging, or other types of conflict resolution, taking place.

You have to somehow *connect* up your multiple joystick axes, whether by using an EPIC card, multiple Game Ports, or multiple USB devices. FSUIPC cannot help there. Having done that, you need to find ‘spare’ FS controls which you will not otherwise be using from joystick inputs (see the lists in my FS2000 Controls documents)—it doesn’t matter if you will be using those controls from the keyboard. FSUIPC only pinches the joystick inputs. You have to assign the additional joystick axes, wherever they may be, to these “spare” controls. Again you should find enough explanation for this in my FS2000 Controls document.

Now add to the FSUIPC.INI file’s JoystickCalibration section (add the section if necessary) a list of declarations which define the additional controls you have assigned. You define these by *number*.

For example, suppose I have a second joystick and rudder set up for a CoPilot on JOYSTICK_01. The section in FS2000.CFG might look like this (*FS2002 sections are similar but **not** the same*):

```
[JOYSTICK_01]
LOCKED=1
TYPE=0
AXIS_FLAGS=7
AXIS_EVENT_00=EGT
AXIS_SCALE_00=64
AXIS_NULL_00=0
AXIS_EVENT_01=EGT_DEC
AXIS_SCALE_01=64
AXIS_NULL_01=0
AXIS_EVENT_02=EGT_INC
AXIS_SCALE_02=64
AXIS_NULL_02=0
```

Here I’ve decided that these three EGT controls aren’t going to be useful on any joystick or button inputs, so I’ve ‘pinched’ them for my copilot controls. From my FS2000 controls document you find that these are numbered 65558, 66033 and 66032 respectively. The parameters now to be added to the FSUIPC file are:

```
[JoystickCalibration]
AileronB=65558
ElevatorB=66033
```

RudderB=66032

Other parameters here can define LeftBrakeB, RightBrakeB, ThrottleB, and also C and D versions of all 6 controls, so providing up to 4 copies of each one.

Note that you will need to calibrate all controls so that the ones controlling the same values are as close as possible in range and response. Do this first in Windows Control Panel, then, after making the above adjustments and assignments, in FSUIPC. Calibrate dead zones at the ends (and in the centre for aileron, elevator and rudder) to “cover up” any discrepancies—in other words, calibrate for the worst of each.

Facility for multiple INI installations (FS2000 and FS2002)

Different FSUIPC.ini files can be used for differing FS2000/2002 requirements, even loading from the same FS2000/2002 installation. This involves using multiple FS2000.CFG or FS2002.CFG files with different filenames, with the following section added in each one:

```
[FSUIPC]
ControlName=<name>
```

Then you load FS2000 for each configuration with the command line parameter specifying the CFG file, thus:

```
FS2000.exe /CFG:<filename>.CFG or
FS2002.exe /CFG:<filename>.CFG
```

And this will allow FSUIPC to identify its correct .INI file, <name>.ini

Note that the Log files, when logging is enabled, will also use this <name>, not just FSUIPC.log etc.

The main use of this feature is so that a PC can be used in two or more modes with *one* FS installation, for example:

- As a WidevieW “slave” with the appropriate default Flight loaded by FS (to place it into slew mode with the correct view) and the correct FSUIPC options set for allowing WidevieW to copy the weather correctly, and:
- With different FS cfg and FSUIPC ini files to run FS in normal ‘local control’ mode with all normal options.

More about FS2000/2002 ‘Real Weather’

FSUIPC needs to detect when downloaded real weather is being used, and needs to identify exactly what that weather is. There are three reasons it needs this data:

1. So that correct weather details can be supplied in the FS98-compatible areas and in the Advanced Weather Interface, for external programs and some third party gauges.
2. So that correct weather details can be provided to Adventures.
3. So that it can correctly detect whether the aircraft is inside or outside a cloud layer and apply the ‘stuck white-out’ fix mentioned elsewhere.

Unfortunately finding the *correct* local weather has been the biggest single problem in FSUIPC’s rather short development history to date. Determining whether local or global weather is in effect was easy enough, and recent investigations have revealed the ICAO table listing METAR reporting station ICAO codes along with their geographical location and elevation. However, the problem has been in identifying *reliably* exactly which of these reporting stations is the one actually currently being used.

Before version 1.56 I thought I’d cracked it. There was one variable in the Weather module that seemed to always provide an index into a table of ‘weather pointers’ which revealed the correct current weather structures. Most of the testing seemed to confirm this. But, alas, it wasn’t tested enough, and some inconsistent reports from Adventures giving bad ATIS reports showed that this index only pointed to one of three possible reporting stations.

That was the situation until version 2.11. Thanks to a little outside help, since version 2.11 FSUIPC can actually determine fairly accurately and, I think, reliably, all the current local weather details and make them available for applications to read correctly. It turns out that the actual local weather is an interpolation between the two or three nearest weather reporting stations, so the values you see will be continuously changing.