FSUIPC4 and FSUIPC5 Status of IPC Offsets for FSX and P3D

Applicable to FSUIPC4 version 4.963 and FSUIPC5 version 5.14 (and later)

Key for status indications:

	Items in blue are new to FSX, those in purple are specific to **P3D4**
Ok-SimC	works okay using SimVars
Ok?-SimC	was reported as working via SimVars but this was contradicted by other reports
Ok-SimC*	more or less works using SimVars, but there are difficulties *(explained in Notes)*
Ok-SimE	for write only, works okay, but resorting to Sim Events via SimC, not SimVar reads or writes
Ok-Lvar	Okay using gauge local variables ("L:variables").
Ok-Intl	works okay, is internal to FSUIPC in any case
Ok-Intl*	more or less works using internal derivation, but there are difficulties *(explained in Notes)*
Ok-Hack	works by hooks or patches or other devilish means
?-Intl	May work, untested, but FSUIPC internal in any case
?-SimC	Mapped to SimConnect variables, but validity unknown. Needs checking and feedback please
?-SimE	Mostly for write only, mapped to Sim Event, but operation unknown. Needs checking and feedback please
??	situation unknown – try it or wait for next issue
No-SimC	Not working, awaiting fix in SimConnect
No-SimE	Not working, Sim Event seems broken, needs fix in FSX?
No-SimC+	Not working, hoping for additions to SimConnect
No	Not supported. (Appeals to Pete Dowson, with reasons, please)
Not yet	Maybe can do okay, but not yet got around to it!
No info	Data unknown, not listed for SimConnect. Not yet followed through
Not tested	Maybe already okay, not tested yet
Maybe	Question mark, see italic text in "use" section
Problem	See italic text in "use" section
N/A	Not applicable

Offset	Size	Use	FS Read	FS Write
0000	32	*Reserved for diagnostics*		
0020	4	Ground altitude in Metres x 256. (see also offset 0B4C)	Ok-SimC	No
0024	256	READ: Zero terminated string giving the Start-Up situation or flight name, including the path (complete, or from the FS folder)	Ok-Intl	No

		WRITE: Operates a facility to "spoof" Flight Sim read-out values as supplied to all FSUIPC and WideFS client applications.
		
		To use this, write the following to offset 0024, as one structure (i.e. one FSUIPC_Write call):
		
		struct {
		WORD offset; // base offset of data to be overridden
		WORD length; // length of data (max 252)
		BYTE[ ] data; // Up to 252 bytes of data
		
		};
		
		The override is established the first time you do this for a specific offset + length combination. Any overlapping setting replaces the previous one, but re-writing the same one with different data is fast as it merely writes to the relevant offsets -- the FS data is already being diverted.
		
		Cancel the overrides by writing the same with no data and a length of 0. If you don't cancel, but don't update, the override will be cancelled after about 12 seconds (not counting Menu time). Avoid letting this happen, though -- always explicitly cancel when finishing.
		
		Note that not all FSX values can be overridden in this way, and none of the normal FSUIPC values can be permanently overridden. However, this facility does provide direct access to ALL offsets, and you can easily wreck things and ruin someone's day! Those which are normally write-protected are not so protected using this facility.
		
		The "Liar.lua" plug-in supplied with the Lua additions demonstrates this facility, and also shows the only way provided of reading the un-spoofed values: a privilege afforded only to the Lua ipc.readStruct facility in the Lua program which actually applies the overrides too.
			No	Ok-Intl

012C	1	The name of the current Log book—not available in FSX	No	No
0130	256	The current flight Plan path & file name (in UNC format if WideFS is in use).	Ok-SimC	Ok-SimC
0230	8	"Absolute Time", in seconds, double float. This is unchecked, but is said to be the time since 12 noon on January 1st, Year 0000 (?).	Ok-SimC	No
0238	1	Hour of local time in FS (0–23)	Ok-simC	?-SimE
0239	1	Minute of local time in FS (0–59)	Ok-simC	?-SimE
023A	1	Second of time in FS (0–59) *For setting, FSX provides "KEY_CLOCK_SECONDS_ZERO" only. No way to directly set a number of seconds.*	Ok-simC	Ok-SimE but *Only setting zero when close – see Notes*
023B	1	Hour of Zulu time in FS (also known at UTC or GMT)	Ok-simC	Ok-simE
023C	1	Minute of Zulu time in FS2	Ok-simC	Ok-simE
023D	1	Zulu day of month in FS (counting from 1)	Ok-simC	No
023E	2	Day number in year in FS (counting from 1)	Ok-simC	?-SimE
0240	2	Zulu year in FS	Ok-simC	?-SimE
0242	1	Zulu month of year in FS	Ok-simC	No
0243	1	Zulu day of week in FS	Ok-simC	No
0244	1	Local month of year in FS	Ok-simC	No
0245	1	Local day of month in FS	Ok-simC	No
0246	2	Local time offset from Zulu (minutes). +ve = behind Zulu, –ve = ahead	Ok-simC	No
0248	2	Season: 0=Winter, 1=Spring, 2=Summer, 3=Fall	Ok-Intl	No
024A	2	Local year in FS	Ok-simC	No
024C	4	Available FS memory in kilobytes (updated every 10 seconds) See also offsets 0258 and 0290.	Ok-Intl	No
0250	1	AI Airline Traffic Density % (0–100). If you increase this you	Ok-Intl	Ok-Intl

		will normally see an FS progress bar as it reloads traffic	*(Hack)*	*(Hack)*
0251	1	AI General Aviation Traffic Density % (0–100). If you increase this you will normally see an FS progress bar as it reloads traffic	Ok-Intl *(Hack)*	Ok-Intl *(Hack)*
0252	1	AI Ships & Ferries Traffic Density % (0–100). If you increase this you will normally see an FS progress bar as it reloads traffic	Ok-Intl *(Hack)*	Ok-Intl *(Hack)*
0254	1	Cloud cover density: 5=LOW to 8=MAX. *This can be written to, and it does change the slider position, but whether it directly affects the cloud drawing isn't known at present.*	Ok-Intl *(Hack)*	?-Intl *(Hack)*
0255	1	Cloud simple/complex flag: 0=Simple, 1=Complex. *This can be written to, and it does change the setting, but whether it directly affects the cloud drawing isn't known at present.*	Ok-Intl *(Hack)*	?-Intl *(Hack)*
0256	1	Thermal visualisation setting: 0=None, 1=Natural, 2=Schematic	Ok-Intl *(Hack)*	No
0258	4	Memory currently assigned to FSUIPC4 (including WideServer) See also offset 024C. This is in Bytes.	Ok-Intl	No
025C	4	The current total number of AI Trsffic aircraft	Ok-Intl	No
0262	2	Pause control (write 1 to pause, 0 to un-pause).	N/A	Ok-SimE
0264	2	Pause indicator (0=Not paused, 1=Paused)	Ok-simE	N/A
0266	2	Centre (nose or tail) wheel RPM, as a 16-bit integer	Ok-SimC	N/A
0268	2	Left wheel RPM, as a 16-bit integer	Ok-SimC	N/A
026A	2	Right wheel RPM, as a 16-bit integer	Ok-SimC	N/A
0274	2	Frame rate is given by 32768/this value	Ok-SimE	N/A
0278	2	Auto-co-ordination ("auto-rudder"), 1=on, 0=off Different to FS9 and before: this setting cannot be changed via any of the usual controls, or the documented as "settable" SimVar. It is broken, an FSX/ESP bug!	Ok-SimC	No
0280	1	Lights: this operates the NAV, TAXI, PANEL and WING lights. For separate switches see offset 0D0C	Ok-Intl (via 0D0C)	Ok-Intl (via 0D0C)
0281	1	Beacon and Strobe lights. For separate switches see offset 0D0C	Ok-Intl (via 0D0C)	Ok-Intl (via 0D0C)
0284	2	ADF1 Standby Frequency: main 3 digits, in Binary Coded Decimal. A frequency of 1234.5 will have 0x0234 here and 0x0105 in offset 0286.	Ok-SimC	Not possible to write!
0286	2	Extended ADF1 Standby Frequency: the high byte contains the 1000's digit and the low byte the fraction, so, for a frequency of 1234.5 this offset will contain 0x0105.	Ok-SimC	Not possible to write!
0288	2	ADF2 Standby Frequency: main 3 digits, in Binary Coded Decimal. A frequency of 1234.5 will have 0x0234 here and 0x0105 in offset 0286.	Ok-SimC	Not possible to write!
028A	2	Extended ADF2 Standby Frequency: the high byte contains the 1000's digit and the low byte the fraction, so, for a frequency of 1234.5 this offset will contain 0x0105.	Ok-SimC	Not possible to write!
028C	1	Landing lights. (See also offset 0D0C).	Ok-Intl (via 0D0C)	Ok-Intl (via 0D0C)
0290	4	Maximum available FS contiguous memory block in kilobytes (updated every 10 seconds). See also offset 024C	Ok-Intl	No
029B	1	Alternate static air source (0=off, 1=on)	Ok-SimC	Ok-SimE
029C	1	Pitot Heat switch (0=off, 1=on)	Ok-SimC	Ok-SimE
02A0	2	Magnetic variation (signed, –ve = West). For degrees *360/65536. Convert True headings to Magnetic by *subtracting* this value, Magnetic headings to True by *adding* this value.	Ok-SimC	N/A
02B2	2	Zoom factor: 64=x1, 128=x2 et cetera	No-SimC+	Ok-SimE
02B4	4	GS: Ground Speed, as 65536*metres/sec. Not updated in Slew mode!	Ok-SimC	No
02B8	4	TAS: True Air Speed, as knots * 128	Ok-SimC	?-SimC
02BC	4	IAS: Indicated Air Speed, as knots * 128	Ok-SimC	?-SimC
02C4	4	Barber pole airspeed, as knots * 128	Ok-SimC	No
02C8	4	Vertical speed, signed, as 256 * metres/sec. For the more usual ft/min you need to apply the conversion *60*3.28084/256	Ok-SimC	?-SimC
02CC	8	Whiskey Compass, degrees in 'double' floating point format (FLOAT64)	Ok-SimC	?-SimC
02D4	2	ADF2 Frequency: main 3 digits, in Binary Coded Decimal. See also offset 02D6. A frequency of 1234.5 will have 0x0234 here and 0x0105 in offset 02D6.	Ok-SimC	Ok-SimE

02D6	2	Extended ADF2 frequency. The high byte contains the 1000's digit and the low byte the fraction, so, for a frequency of 1234.5 this offset will contain 0x0105.	Ok-SimC	Ok-SimE
02D8	2	ADF2: relative bearing to NDB (*360/65536 for degrees, –ve left, +ve right)	?-SimC	No
02DC	6	ADF2 IDENTITY (string supplied: 6 bytes including zero terminator)	Ok-SimC	No
02E2	25	ADF2 name (string supplied: 25 bytes including zero terminator)	Ok-SimC	No
02FB	1	ADF2 morse ID sound (1 = on, 0 = off), read for state, write to control	?-SimC	?-SimE
0300	2	VOR1 DME distance, 16-bit integer, nm * 10	Ok-SimC	No
0302	2	VOR1 DME speed, 16-bit integer, kts * 10	Ok-SimC	No
0304	2	VOR1 DME time to station, 16-bit integer, secs * 10	Ok-Intl	No
0306	2	VOR2 DME distance, 16-bit integer, nm * 10	Ok-SimC	No
0308	2	VOR2 DME speed, 16-bit integer, kts * 10	Ok-SimC	No
030A	2	VOR2 DME time to station, 16-bit integer, secs * 10	Ok-Intl	No
030C	4	Vertical speed, copy of offset 02C8 whilst airborne, not updated whilst the "on ground" flag (0366) is set. Can be used to check hardness of touchdown (but watch out for bounces which may change this).	Ok-Intl	N/A
0310	8	Timer (double float, elapsed seconds including fractions, adjusted each 'tick' – i.e. 1/18th sec). See also 0368	Ok-Intl	No
0318	4	Pressurisation cabin altitude at present (feet, 32-bit integer)	?-SimC	No
031C	4	Pressurisation cabin altitude set goal (feet, 32-bit integer)	?-SimC	No
0320	4	Pressurisation cabin altitude set change rate (feet/sec, 32-bit floating point)	?-SimC	No
0324	4	Pressurisation cabin pressure differential (lbs/sq.ft, 32-bit floating point): set – actual.	?-SimC	No
0328	4	Pressurisation dump switch (1 = open, 0 = closed)	?-SimC	?-SimE
032C	2	"Plane is in fuel box" flag (same as Scenery BGL variable 0288)	No	No
032E	2	*Reserved (used internally)*	N/A	N/A
0330	2	Altimeter pressure setting ("Kollsman" window). As millibars (hectoPascals) * 16	Ok-SimC	Ok-SimE
0332	2	Altimeter pressure secondary setting ("Kollsman" window). As millibars (hectoPascals) * 16. This is the one used in the G1000 gauge.	Ok-SimC	Ok-SimE
0334	4	Pushback angle, radians, as a 32-bit Float.	Ok-SimC	No
0338	4	Pushback X contact, feet, as a 32-bit Float.	Ok-SimC	No
033C	4	Pushback Y contact, feet, as a 32-bit Float.	Ok-SimC	No
0340	4	Pushback Z contact, feet, as a 32-bit Float.	Ok-SimC	No
0344	2	Pushback wait flag, 16-bit integer (probably only 0 or 1)	Ok-SimC	No
0346	1	Surface condition: 0=normal, 1=wet, 2=icy, 3=snow	Ok-SimC	No
0347	1	Surface info valid flag. [*not working -- ignore*]	No	No
0348	2	Structural ice formation quantity, 0 – 16384	Ok-SimC	No
034A	2	Pitot ice formation quantity, 0 – 16384	Ok-SimC	No
034C	2	ADF1 Frequency: main 3 digits, in Binary Coded Decimal. See also offset 0356. A frequency of 1234.5 will have 0x0234 here and 0x0105 in offset 0356.	Ok-SimC	Ok-SimE
034E	2	COM1 frequency, 4 digits in BCD format. A frequency of 123.45 is represented by 0x2345. The leading 1 is assumed.	Ok-SimC	Ok-SimE
0350	2	NAV1 frequency, 4 digits in BCD format. A frequency of 113.45 is represented by 0x1345. The leading 1 is assumed.	Ok-SimC	Ok-SimE
0352	2	NAV2 frequency, 4 digits in BCD format. A frequency of 113.45 is represented by 0x1345. The leading 1 is assumed.	Ok-SimC	Ok-SimE
0354	2	Transponder setting, 4 digits in BCD format: 0x1200 means 1200 on the dials.	Ok-SimC	Ok-SimE
0356	2	Extended ADF1 frequency. The high byte contains the 1000's digit and the low byte the fraction, so, for a frequency of 1234.5 this offset will contain 0x0105.	Ok-SimC	Ok-SimE
0366	2	Aircraft on ground flag (0=airborne, 1=on ground). Not updated in Slew mode.	Ok-SimC	N/A
0368	4	Control timer 2 (see also 0310), a 32-bit 'float'.	Ok-Intl	No

036C	1	Stall warning (0=no, 1=stall)	**Ok-SimC**	**No**
036D	1	Overspeed warning (0=no, 1=overspeed)	**Ok-SimC**	**No**
036E	1	Turn co-ordinator ball position (slip and skid). –128 is extreme left, +127 is extreme right, 0 is balanced. (See 0374 for more accuracy)	**Ok-SimC**	**No**
0371	1	*Reserved for ASE weather control flags*	**No**	**No**
0372	2	Reliability % (0–100)	**No**	**No**
0374	2	NAV1 or NAV2 select [*Not used for several FS releases?*]	**No**	**No**
0378	2	DME1 or DME2 select (1=DME1, 2=DME2)	**Ok-SimC**	**Ok-SimE**
037C	2	Turn Rate (for turn coordinator). 0=level, –512=2min Left, +512=2min Right (See 0378 for more accuracy)	**Ok-SimC**	**?-SimC**
0380	4	32-bit floating point turn coordinator ball position, –1.0 to +1.0	**Ok-SimC**	**No**
0384	4	32-bit floating point turn rate, degrees per second (–3.0 to +3.0 is equivalent to the 2 mins left/right range)	**Ok-SimC**	**No**
03A0	4	1st FSUIPC monitored value (right-hand side of Logging tab), if numeric. Provided in 32-bit floating point format	**Ok-Intl**	**n/a**
03A4	4	2nd FSUIPC monitored value (right-hand side of Logging tab), if numeric. Provided in 32-bit floating point format	**Ok-Intl**	**n/a**
03A8	4	3rd FSUIPC monitored value (right-hand side of Logging tab), if numeric. Provided in 32-bit floating point format	**Ok-Intl**	**n/a**
03AC	4	4th FSUIPC monitored value (right-hand side of Logging tab), if numeric. Provided in 32-bit floating point format	**Ok-Intl**	**n/a**
03B0	8	Left aileron deflection, in radians, as a double floating point value	**Ok-SimC**	**No**
03B8	8	Right aileron deflection, in radians, as a double floating point value	**Ok-SimC**	**No**
03C0	64	The current state of the buttons on actively scanned joysticks (local ones, 0 to 15). Each of the 16 DWORDS contain the 32-bit state of the joystick 0-15, in order. Button 0 is the least significant bit (bit 0) in each DWORD.	**Ok-Intl**	**No**
0400	128	The filename of the last flight (or situation) saved, as an ASCII string with a zero terminator. The filetype (.flt or .stn) is not included. Use the counter at 3BD2 to determine when this has changed.	**Ok-SimC**	**N/A**
0480	8	Aileron trim axis input, 64-bit floating point (double), read-only	**?-Intl**	**N/A**
0488	8	Rudder trim axis input, 64-bit floating point (double), read-only	**?-Intl**	**N/A**
0490	8	Aileron trim axis required value, 64-bit floating point (double). If 2^0 is set in the byte at 04A0, then, when written, this value is copied to the FS trim (2EB0) instead of the value in 0480	**N/A**	**?-Intl**
0498	8	Rudder trim axis required value, 64-bit floating point (double). If 2^1 is set in the byte at 04A0, then, when written, this value is copied to the FS trim (2EC0) instead of the value in 0488	**N/A**	**?-Intl**
04A0	1	Aileron and rudder trim connection control. See offsets 480–0498 above. 2^0 = 1 to disconnect aileron trim (2EB0) from FS 2^1 = 1 to disconnect rudder trim (2EC0) from FS This byte will be cleared and the connection restored (together with the most recent axis values) within about 10 seconds of it being written non-zero, so you need to write this every few seconds.	**?-Intl**	**?-Intl**
04A8	8	Elapsed seconds value, as a double. Accurate to fractions of a second but only updated frame by frame. This value counts simulated time, stopping in paused and menu modes, speeding up and slowing down according to the actual sim rate.	**Ok (from Gauge Token)**	**No**
04B0	48	Area reserved by FSUIPC.	**N/A**	**N/A**
04B4	2	ADVENTURE WEATHER: This provides the TEMPERATURE_SURFACE_ALT in metres. This is used to provide the METAR reporting station altitude so that the cloud bases can be converted to AGL.	**?-Intl**	**No**
04BA	2	ADVENTURE WEATHER: This provides the WIND_SURF_TURB which is used to provide the surface wind's upper gust speed in knots, with zero indicating no gusts.	**?-Intl**	**No**
04BC	2	ADVENTURE WEATHER: This provides the BAROMETRIC_DRIFT	**?-Intl**	**No**

		variable, which is used to provide the *difference* between the current aircraft position QNH (which may be in transition), and the METAR reported QNH as set by the weather control program. Adding this 'drift' value to the pressure will give the correct value for ATIS reports		
04C0	2	ADVENTURE WEATHER: This provides the FSUIPC_VISIBILITY in statute miles * 100	**?-Intl**	**No**
04C2	2	ADVENTURE WEATHER: This provides the CLOUD_THUNDER_BASE in metres AMSL	**?-Intl**	**No**
04C4	2	ADVENTURE WEATHER: This provides the CLOUD_LOW_BASE in metres AMSL	**?-Intl**	**No**
04C6	2	ADVENTURE WEATHER: This provides the CLOUD_HIGH_BASE in metres AMSL	**?-Intl**	**No**
04C8	2	Dew point as degrees C *256, for the surface temperature layer, read only	**?-Intl**	**No**
04CB	1	Precipitation rate, 0–5, read only.	**?-Intl**	**No**
04CC	1	Precipitation type, 0=none, 1=rain, 2=snow, read only.	**?-Intl**	**No**
04CD	1	ADVENTURE WEATHER: This provides the CLOUD_THUNDER_COVER 0–8	**?-Intl**	**No**
04CE	1	ADVENTURE WEATHER: This provides the CLOUD_LOW_COVER 0–8	**?-Intl**	**No**
04CF	1	ADVENTURE WEATHER: This provides the CLOUD_HIGH_COVER 0–8	**?-Intl**	**No**
04D2	2	Precipitation control: write hi-byte=type 0–2, low byte=rate 0–5. Write 0xFFFF to release control back to FS.	**N/A**	**?-Intl**
04D4	2	Dew point control: degrees C * 256. Sets surface layer dewpoint only, FSUIPC does rest. Write 0x8000 to release control back to FS.	**N/A**	**?-Intl**
04D6	2	Set to 0xFADE if FSUIPC's weather interface has initialised.	**Ok-Intl**	**No**
04D8	2	Surface layer wind speed, in knots. This may be different to the current wind speed at the aircraft—see offset 0E90. This also provides WIND_SURF_VEL for Adventures.	**?-Intl**	**No**
04DA	2	Surface layer wind direction, *360/65536 to get degrees MAGNETIC. This may be different to the current wind direction at the aircraft—see offset 0E92. This also provides WIND_SURF_DIR for Adventures.	**?-Intl**	**No**
04DE	2	Weather option control: not supported	**No**	**No**
04E0	88	Area reserved for Project Magenta	**N/A**	**N/A**
0538	8	Design speed VS0 (stall speed full flaps), ft/sec, as a double (64-bit floating point).	**Ok-SimC**	**No**
0540	8	Design speed VS1 (stall speed clean), ft/sec, as a double (64-bit floating point).	**Ok-SimC**	**No**
0548	8	Design speed VC (cruise speed), ft/sec, as a double (64-bit floating point).	**Ok-SimC**	**No**
0550	8	Minimum drag velocity, ft/sec, as a double (64-bit floating point).	**Ok-SimC**	**No**
0558	4	INITIAL POSITION: Airspeed setting. Write the desired airspeed here (in knots), along with, *in the same IPC write*, those of the following fields (on-ground, LLAPBH – Lat/Lon/Alt/Pitch/Bank/Hdg) which you need to set. FSUIPC4 will use the *INITIAL POSITION* facility in FSX to place your aircraft and set the speed. To set the speed at the current position (but not on ground), just write this offset and FSUIPC4 will use the following values as they currently stand.	**N/A**	**Ok-SimC**
055C	4	INITIAL POSITION: On-ground setting. Write 0 for in-flight or 1 for on-ground here, along with, *in the same IPC write*, those of the following fields (LLAPBH – Lat/Lon/Alt/Pitch/Bank/Hdg) which you need to set. FSUIPC4 will use the *INITIAL POSITION* facility in FSX to place your	**N/A**	**Ok-SimC**

		aircraft. It will set the speed to 0 if the on-ground value is non-zero, but otherwise it will use the current airspeed from 02BC.		
0560	8	Latitude of aircraft in FS units. *(Read offset 6010 for easier conversion!)* To convert to Degrees: *If your compiler supports long long (64-bit) integers* then use such a variable to simply copy this 64-bit value into a double floating point variable and multiply by 90.0/(10001750.0 * 65536.0 * 65536.0). *Otherwise* you will have to handle the high 32-bits and the low 32-bits separately, combining them into one double floating point value (say dHi). To do, copy the high part (the 32-bit int at 0564) to one double and the low part (the 32-bit unsigned int at 0560) to another (say dLo). Remember that the low part is only *part* of a bigger number, so doesn't have a sign of its own. Divide dLo by (65536.0 * 65536.0) to give it its proper magnitude compared to the high part, then either add it to or subtract it from dHi according to whether dHi is positive or negative. This preserves the integrity of the original positive or negative number. Finally multiply the result by 90.0/10001750.0 to get degrees. Either way, a negative result is South, positive North. [Can be written to move aircraft]	**Ok-SimC**	**Ok-SimC**
0568	8	Longitude of aircraft in FS format. *(Read offset 6018 for easier conversion!)* To convert to Degrees: *If your compiler supports long long (64-bit) integers* then use such a variable to simply copy this 64-bit value into a double floating point variable and multiply by 360.0/(65536.0 * 65536.0 * 65536.0 * 65536.0). *Otherwise* you will have to handle the high 32-bits and the low 32-bits separately, combining them into one double floating point value (say dHi). To do, copy the high part (the 32-bit int at 056C) to one double and the low part (the 32-bit unsigned int at 0568) to another (say dLo). Remember that the low part is only *part* of a bigger number, so doesn't have a sign of its own. Divide dLo by (65536.0 * 65536.0) to give it its proper magnitude compared to the high part, then either add it to or subtract it from dHi according to whether dHi is positive or negative. This preserves the integrity of the original positive or negative number. Finally multiply the result by 360.0/(65536.0 * 65536.0) to get degrees. Either way, a negative result is West, positive East. If you did it all unsigned then values over 180.0 represent West longitudes of (360.0 – the value). [Can be written to move aircraft]	**Ok-SimC**	**Ok-SimC**
0570	8	Altitude, in metres and fractional metres. The units are in the high 32-bit integer (at 0574) and the fractional part is in the low 32-bit integer (at 0570). [Can be written to move aircraft] *(Read offset 6020 for easier conversion!)*	**Ok-SimC**	**Ok-SimC**
0578	4	Pitch, *360/(65536*65536) for degrees. 0=level, –ve=pitch up, +ve=pitch down	**Ok-SimC**	**Ok-SimC**
057C	4	Bank, *360/(65536*65536) for degrees. 0=level, –ve=bank right, +ve=bank left	**Ok-SimC**	**Ok-SimC**
0580	4	Heading, *360/(65536*65536) for degrees TRUE.	**Ok-SimC**	**Ok-SimC**
0584	4	Bits here mark which of the aircraft situation variables (LLAPBH, Lat Lon alt Pitch Bank Heading) in offsets 0560-0580 were updated by FS at the time provided in offset 0588. The bits are (bit 0 = least significant): 0 = Lat, 2 = Lon, 4 = Alt, 6 = Pitch, 7 = Bank, 8 = Heading	**Ok-Intl**	**N/A**
0588	8	Double floating point value giving the elapsed real time, in seconds, at the last time any of the aircraft situation variables (LLAPBH, Lat Lon alt Pitch Bank Heading) in offsets 0560-	**Ok-Intl**	**N/A**

		0580 were updated by FS.		
05C4	4	Active COM1 frequency in Hz (32 bit int)	Ok-SimC	Ok-SimC
05C8	4	Active COM2 frequency in Hz (32 bit int)	Ok-SimC	Ok-SimC
05CC	4	Standby active COM1 frequency in Hz (32 bit int):	Ok-SimC	Ok-SimC
05D0	4	Standby COM2 frequency in Hz (32 bit int)	Ok-SimC	Ok-SimC
05D4	2	Smoke system available if True	?-SimC	No
05D8	2	Smoke system enable: write 1 to switch on, 0 to switch off (see also 05D4)	?-SimC	?-SimE
05DC	2	Slew mode (indicator and control), 0=off, 1=on. (See 05DE also).	Ok-SimC	Ok-SimE
05E4	2	Slew roll rate: 0=static, –ve = right roll, +ve=left roll, rate is such that 192 gives a complete 360 roll in about one minute.	No	Ok-SimE
05E6	2	Slew yaw rate: 0=heading constant, –ve = right, +ve=left, rate is such that 24 gives a complete 360 turn in about one minute.	No	Ok-SimE
05E8	2	Slew vertical rate: 16384=no change, 16385–32767 increasing rate down, 16383–0 increasing rate up. One keypress on Q (up) or A (down) makes a change of 512 units.	No	Ok-SimE
05EB	1	Slew forward/backward movement: +ve=backward, –ve=forward. Values 1–127 give slow to fast slewing (–128 is the fastest forward slew).	No	Ok-SimE
05ED	1	Slew left/right movement: +ve=right, –ve=left. Values 1–127 give slow to fast sideways slewing (–128 is the fastest leftward slew).	No	Ok-SimE
05EE	2	Slew pitch rate: 16384=no change, <16384=pitch up, >16384 pitch down, range 0–32767.	No	Ok-SimE
05F4	2	Slew mode display: 0=off, 1=coords/hdg/spd, 2=fps, 3=all	No	No
05FC	2	Flight mode display: 0=off, 1=coords/hdg/spd, 2=fps, 3=all	No	No
0609	1	Engine type: 0=Piston (and some Helo models like the Robinson) 1=Jet 2=Sailplane, or anything with no engines 3=Helo (Bell) Turbine 4=Rocket *(unsupported)* 5=Turboprop	Ok-SimC	No
060C	1	Gear is retractable (1 = retractable, 0 = fixed)	Ok-SimC	No
0614	2	Retractable left float extension. 0=fully retracted, 16384=fully extended	Ok-SimC	No
0616	2	Retractable right float extension. 0=fully retracted, 16384=fully extended	Ok-SimC	No
0628	4	Instant replay flag & control, 1=on, 0=off. Can write to turn on and off whilst there is still time to play (see offset 062C)	No	No
062C	4	Instant replay: time left to run, in seconds. Whilst this is non-zero, the flag in offset 0628 controls the playback.	No	No
0658	120	This area provides a table of data about the 6 nearest airports to the user aircraft, in order nearest to furthest. This works properly in FSX with Acceleration, and should work okay in SP2 versions. It should also work in P3D. 1.4. Each entry is 20 bytes long, consisting of 4 fields as follows: 0 4 bytes ICAO ID of the airport. If this is 3 characters only the 4th will be zero. 4 4 bytes Latitude of the airport, in degrees, as a 32-bit Float (N positive, S negative). 8 4 bytes Longitude of the airport, in degrees, as a 32-bit Float (E positive, W negative). 12 4 bytes Altitude of the airport, in feet, as a 32-bit Float. 16 4 bytes Distance from the user aircraft, in nm. It is only updated when the user aircraft's Latitude or Longitude change by one minute or more. If there are less than 6 airports within the current "reality bubble" the unused entries will be all	Intl	N/A

		zero.		
06D0	144	Area used for operating, controlling and configuring the facilities in FSUIPC for feedback flight control (bank, pitch, speed, yaw). For full details of this please see the separate TXT documentation in the SDK.	**Ok-Intl**	**Ok-Intl**
0760	4?	Video recording flag, 1=on, 0=off	**No**	**No**
0764	4	Autopilot available	**Ok-SimC**	**N/A**
0778	4	Flaps available	**Ok-SimC**	**N/A**
077C	4	Stall horn available	**Ok-SimC**	**N/A**
0780	4	Engine mixture available	**Ok-SimC**	**N/A**
0784	4	Carb heat available	**Ok-SimC**	**N/A**
078C	4	Spoiler available	**Ok-SimC**	**N/A**
0790	4	Aircraft is tail dragger	**Ok-SimC**	**N/A**
0794	4	Strobes available	**Ok-SimC**	**N/A**
079C	4	Toe brakes available	**Ok-SimC**	**N/A**
07A0	4	NAV1 available	**Ok-SimC**	**N/A**
07A4	4	NAV2 available	**Ok-SimC**	**N/A**
07B6	1	Fly by wire ELAC switch	**?-SimC**	**?-SimE**
07B7	1	Fly by wire ELAC computer failed flag	**?-SimC**	**No**
07B8	1	Fly by wire FAC switch	**?-SimC**	**?-SimE**
07B9	1	Fly by wire FAC computer failed flag	**?-SimC**	**No**
07BA	1	Fly by wire SEC switch	**?-SimC**	**?-SimE**
07BB	1	Fly by wire SEC computer failed flag	**?-SimC**	**No**
07BC	4	Autopilot Master switch	**Ok-SimC**	**Ok-SimE**
07C0	4	Autopilot wing leveller	**Ok-SimC**	**Ok-SimE**
07C4	4	Autopilot NAV1 lock	**Ok-SimC**	**Ok-SimE**
07C8	4	Autopilot heading lock	**Ok-SimC**	**Ok-SimE**
07CC	2	Autopilot heading value, as degrees*65536/360	**Ok-SimC**	**Ok-SimE**
07D0	4	Autopilot altitude lock	**Ok-SimC**	**Ok-SimE**
07D4	4	Autopilot altitude value, as metres*65536	**Ok-SimC**	**Ok-SimE**
07D8	4	Autopilot attitude hold	**Ok-SimC**	**?-SimE**
07DC	4	Autopilot airspeed hold	**Ok-SimC**	**Ok-SimE**
07E2	2	Autopilot airspeed value, in knots	**Ok-SimC**	**Ok-SimE**
07E4	4	Autopilot mach hold	**Ok-SimC**	**Ok-SimE**
07E8	4	Autopilot mach value, as Mach*65536	**Ok-SimC**	**Ok-SimE**
07EC	4	Autopilot vertical speed hold	**Ok-SimC**	**?-simE**
07F2	2	Autopilot vertical speed value, as ft/min	**Ok-SimC**	**Ok-SimE**
07F4	4	Autopilot RPM (N1) hold	**Ok-SimC**	**Ok-SimE**
07FA	2	Autopilot RPM (N1) hold value, 16384 = 100% N1. Writing rounds to the nearest whole %	**Ok-SimC**	**Ok-SimE** *(but see note)*
07FC	4	Autopilot GlideSlope hold N.B. setting this also sets 0800, approach hold. To clear both you need to write 0 to them in the same FSUIPC process call, as if they are separated by an FS frame, an interlock stops them clearing.	**Ok-SimC**	**Ok-SimE plus Intl operations**
0800	4	Autopilot Approach hold. See the note above, for offset 07FC.	**Ok-SimC**	**Ok-SimE plus Intl operations**
0804	4	Autopilot Back course hold. The note for offset 07FC may also apply here.	**Ok-SimC**	**Ok-SimE**
0808	4	Yaw damper	**Ok-SimC**	**Ok-SimE**
080C	4	Autothrottle TOGA (take off power)	**Ok-SimC**	**Ok-SimE**
0810	4	Autothrottle Arm	**Ok-SimC**	**Ok-SimE**
0814	4	Flight analysis mode (0=0ff, 1=Landing, 2=Course tracking, 3=Manoevres)	**No**	**No**
081E	1	Rotor Brake Active (0 = off, 1 = on). Applicable to Robinson model helicopter only	**Ok-SimC**	**No**
081F	1	Rotor Clutch Active (0 = off, 1 = on). Applicable to Robinson model helicopter only	**Ok-SimC**	**No**
0820	1	Rotor Chip Detected (0 = off, 1 = on). Applicable to Robinson model helicopter only	**Ok-SimC**	**No**
0821	1	Rotor Gov Active (0 = off, 1 = on). Applicable to Robinson	**Ok-SimC**	**No**

		model helicopter only		
0822	2	Rotor brake application (0 to 16384). Applicable to Robinson model helicopter only. Writing: there appears no way to set the level of braking directly. The only way to influence it is to send Rotor Brake controls. In an attempt to achieve the written value, FSUIPC4 send Rotor Brake controls to FSX on every FS frame whilst the read-out for the rotor braking value is less than that last written to 0822. There is an exception—if the read-out remains zero for 4 such attempts, the written value is reset to zero too. This is to infallibly cope with aircraft with no implemented rotor brake, avoiding continuous useless control applications.. This was intended to achieve the result of a sustained brake pressure oscillating close to the value being written, but unfortunately the Rotor Brake control imposes immediate maximum brake pressure but with a fast reduction. The result, therefore, is an oscillation between maximum and just under the requested value.	**Ok-SimC**	**Ok-SimE** *(but see notes)*
0824	2	Rotor lateral trim (0 to 16384). Applicable to Robinson model helicopter only	**?-SimC**	**?-SimE**
0826	1	Rotor Gov switch (0 = off, 1 = on). Applicable to Robinson model helicopter only	**Ok-SimC**	**Ok-SimE**
0828	8	Rotor transmission temperature (64-bit double float, in degrees Rankine). Possibly only applicable to Robinson model helicopter, but no success in seeing this!	**?-SimC**	**No**
0830	2	Action on crash (not working). For FS2004 and before this was a 4-byte value. Now the two high bytes are used for flags as shown in the next two entries.	**No**	**No**
0832	1	Crash detection: 1=Crash detection is on, 0 = off	**?-SimC**	**No**
0833	1	Crash detection: 1=Crash with other aircraft is on, 0 = off	**?-SimC**	**No**
0834	4	DME2 Latitude when available separately. Same units as in 085C above.	**Ok-SimC**	**N/A**
0838	4	DME2 Longitude when available separately. Same units as in 0864 above.	**Ok-SimC**	**N/A**
083C	4	DME2 elevation in metres when available separately.	**Ok-SimC**	**N/A**
0840	2	Crashed flag.	**Ok-SimE**	**N/A**
0842	2	Vertical speed in metres per minute, but with –ve for UP, +ve for DOWN. Multiply by 3.28084 and reverse the sign for the normal fpm measure.	**?-SimC**	**N/A**
0844	2	NAV2 ILS localiser inverse runway heading if VOR2 is ILS. Convert to degrees by *360/65536. This is 180 degrees different to the direction of flight to follow the localiser.	**Ok-SimC**	**N/A**
0846	2	NAV2 ILS glideslope inclination if VOR2 is ILS. Convert to degrees by *360/65536.	**Ok-SimC**	**N/A**
084C	4	VOR2 Latitude, as in 085C below, except when NAV2 is tuned to an ILS, in which case this gives the localiser Latitude.	**Ok-SimC**	**N/A**
0850	4	VOR2 Longitude, as in 0864 below, except when NAV2 is tuned to an ILS, in which case this gives the localiser Longitude.	**Ok-SimC**	**N/A**
0854	4	VOR2 Elevation, in metres, except when NAV2 is tuned to an ILS, in which case this gives the localiser Elevation.	**Ok-SimC**	**N/A**
0858	4	VOR2 Latitude in FS form. Convert to degrees by *90/10001750. If NAV2 is tuned to an ILS this gives the glideslope transmitter Latitude.	**Ok-SimC**	**N/A**
085C	4	VOR1 Latitude in FS form. Convert to degrees by *90/10001750. If NAV1 is tuned to an ILS this gives the glideslope transmitter Latitude.	**Ok-SimC**	**N/A**
0860	4	VOR2 Longitude in FS form. Convert to degrees by *360/(65536*65536). If NAV2 is tuned to an ILS this gives the glideslope transmitter Longitude.	**Ok-SimC**	**N/A**
0864	4	VOR1 Longitude in FS form. Convert to degrees by *360/(65536*65536). If NAV1 is tuned to an ILS this gives the glideslope transmitter Longitude.	**Ok-SimC**	**N/A**

0868	4	VOR2 Elevation in metres. If NAV2 is tuned to an ILS this gives the glideslope transmitter Elevation.	**Ok-SimC**	**N/A**
086C	4	VOR1 Elevation in metres. If NAV1 is tuned to an ILS this gives the glideslope transmitter Elevation.	**Ok-SimC**	**N/A**
0870	2	NAV1 ILS localiser inverse runway heading if VOR1 is ILS. Convert to degrees by *360/65536. This is 180 degrees different to the direction of flight to follow the localiser.	**Ok-SimC**	**N/A**
0872	2	NAV1 ILS glideslope inclination if VOR1 is ILS. Convert to degrees by *360/65536	**Ok-SimC**	**N/A**
0874	4	VOR1 Latitude, as in 085C above, except when NAV1 is tuned to an ILS, in which case this gives the localiser Latitude.	**Ok-SimC**	**N/A**
0878	4	VOR1 Longitude, as in 0864 above, except when NAV1 is tuned to an ILS, in which case this gives the localiser Longitude.	**Ok-SimC**	**N/A**
087C	4	VOR1 Elevation, as in 086C above, except when NAV1 is tuned to an ILS, in which case this gives the localiser Elevation.	**Ok-SimC**	**N/A**
0880	4	DME1 Latitude when available separately. Same units as in 085C above.	**Ok-SimC**	**N/A**
0884	4	DME1 Longitude when available separately. Same units as in 0864 above.	**Ok-SimC**	**N/A**
0888	1	Active engine (select) flags. Bit 0 = Engine 1 selected … Bit 3 = Engine 4 selected. See notes against offset 0892.	**Ok-SimC**	**Ok-SimC**
0889	1	Rotor clutch switch, when applicable. 1=On, 0=Off. Can be read and written.	**?-SimC**	**?-SimE**
088A	2	DME1 Elevation in metres, when available separately.	**Ok-SimC**	**N/A**
088C	152	ENGINE 1 values, as detailed below		
088C	2	Engine 1 Throttle lever, –4096 to +16384 [Programs controlling throttle directly from user inputs should write to 089A instead if the input should be disconnectable via offset 310A (e.g. for auto-throttle management)]	**Ok-SimC**	**Ok-SimC**
088E	2	Engine 1 Prop lever, –4096 to +16384	**Ok-SimC**	**Ok-SimC**
0890	2	Engine 1 Mixture lever, 0 – 16384	**Ok-SimC**	**Ok-SimC**
0892	2	Engine 1 Starter switch position (Magnetos), Jet/turbojet: 0=Off, 1=Start, 2=Gen/Alt Prop: 0=Off, 1=right, 2=Left, 3=Both, 4=Start Don't forget to switch fuel on to start (mixture to max).	**Ok-SimC/Intl**	**Ok-SimE/Intl**
0894	2	Engine 1 combustion flag (TRUE if engine firing)	**Ok-SimC**	**?-SimC**
0896	2	Engine 1 Jet N2 as 0 – 16384 (100%). This also appears to be the Turbine RPM % for proper helo models (and now also for the FS2004 Robinson model and derivatives)	**Ok-SimC**	**?-SimC**
0898	2	Engine 1 Jet N1 as 0 – 16384 (100%), or Prop RPM (derive RPM by multiplying this value by the RPM Scaler (see 08C8) and dividing by 65536). Note that Prop RPM is signed and negative for counter-rotating propellers. In FS2004 this also now gives the Robinson model's RPM, when scaled by the RPM scaler.	**Ok-SimC**	**?-SimC**
089A	2	Engine 1 Throttle lever, –4096 to +16384, same as 088C above except that values written here are treated like axis inputs and are disconnectable via offset 310A, and have the last written value obtainable from offset 3330	**N/A**	**Ok-Intl**
08A0	2	Engine 1 Fuel Flow PPH SSL (pounds per hour, standardised to sea level). Don't know units, but it seems to match some gauges if divided by 128. Not maintained in all cases.	**Ok-SimC**	**?-SimC**
08B2	2	Engine 1 Anti-Ice or Carb Heat switch (1=On)	**Ok-SimC**	**Ok-SimE**
08B8	2	Engine 1 Oil temperature, 16384 = 140 C.	**Ok-SimC**	**?-SimC**
08BA	2	Engine 1 Oil pressure, 16384 = 55 psi. Note that in some aircraft (eg the B777) this can exceed the 16-bit capacity of this location. FSUIPC limits it to fit, i.e.65535 = 220 psi	**Ok-SimC**	**?-SimC**
08BC	2	Engine 1 Pressure Ratio (where calculated): 16384 = 1.60	**?-SimC**	**?-SimC**
08BE	2	Engine 1 EGT, 16384 = 860 C. [*Note that for Props this value is not actually correct. You will get the correct value from 3B70. The value here has been derived by FSUIPC to be compatible with FS2004, FS2002 et cetera*]	**Ok-SimC**	**?-SimC**
08C0	2	Engine 1 Manifold Pressure: Inches Hg * 1024	**Ok-SimC**	**?-SimC**

08C8	2	Engine 1 RPM Scaler: For Props, use this to calculate RPM – see offset 0898 *(On turboprops this will give the shaft RPM, since there is currently no Gear Reduction Ratio available to fix values on such aircraft. I will fix this when I can)*	**Ok-Intl*** *(see note)*	**N/A**
08D0	4	Engine 1 Oil Quantity: 16384 = 100%	**Ok-SimC**	**?-SimC**
08D4	4	Engine 1 Vibration: 16384 = 5.0. This is a relative measure of amplitude from the sensors on the engine which when too high is an indication of a problem. The value at which you should be concerned varies according to aircraft and engine.	**Ok-SimC**	**No**
08D8	4	Engine 1 Hydraulic pressure: appears to be 4*psi	**Ok-SimC**	**No**
08DC	4	Engine 1 Hydraulic quantity: 16384 = 100%	**Ok-SimC**	**No**
08E8	8	Engine 1 CHT, degrees F in double floating point (FLOAT64)	**?-SimC**	**?-SimC**
08F0	4	Engine 1 Turbine temperature: degree C *16384 (Helos?) *(Turbine engine ITT)*	**?-SimC**	**?-SimC**
08F4	4	Engine 1 Torque % (16384 = 100%). This is correct for true Helo models like the Bell. Other prop-based models have this computed by FSUIPC4 from the actual torque in 0920, assuming a maximum of 600 ft-lbs.	**?-SimC**	**?-SimC**
08F8	4	Engine 1 Fuel pressure, psf (i.e. psi*144): not all aircraft files provide this, valid for helo models?	**?-SimC**	**?-SimC**
08FC	4	Engine 1 Electrical Load. (some sort of percentage as a proportion of 16k or 64k?). True helo models only I think.	**?-SimC**	**No**
0900	4	Engine 1 Transmission oil pressure (psi * 16384): for true helos	**?-SimC**	**No**
0904	4	Engine 1 Transmission oil temperature (degrees C * 16384): for true helos	**?-SimC**	**No**
0908	4	Engine 1 Rotor RPM % (16384=100%): for true helos	**?-SimC**	**No**
090C	4	Engine 1 fuel used since start (in pounds, 32-bit float)	**Ok-SimC**	**No**
0910	4	Engine 1 elapsed time (in hours, 32-bit float)	**Ok-SimC**	**No**
0918	8	Engine 1 Fuel Flow Pounds per Hour, as floating point double (FLOAT64)	**Ok-SimC**	**?-SimC**
0920	4	Engine 1 Torque in foot-pounds, as a 32-bit Float. (Not jets)	**Ok-SimC**	**No**
0924	152	ENGINE 2 values, as detailed below **SEE STATUS FOR ENGINE 1**		
0924	2	Engine 2 Throttle lever, –4096 to +16384 [Programs controlling throttle directly from user inputs should write to 0932 instead if the input should be disconnectable via offset 310A (e.g. for auto-throttle management)]		
0926	2	Engine 2 Prop lever, –4096 to +16384		
0928	2	Engine 2 Mixture lever, 0 – 16384		
092A	2	Engine 2 Starter switch position (Magnetos), Jet/turbo: 0=Off, 1=Start, 2=Gen; Prop: 0=Off, 1=right, 2=Left, 3=Both, 4=Start (See Notes in Engine 1 entry)		
092C	2	Engine 2 combustion flag (TRUE if engine firing)		
092E	2	Engine 2 Jet N2 as 0 – 16384 (100%)		
0930	2	Engine 2 Jet N1 as 0 – 16384 (100%), or Prop RPM (derive RPM by multiplying this value by the RPM Scaler (see 08C8) and dividing by 65536). Note that Prop RPM is signed and negative for counter-rotating propellers.		
0932	2	Engine 2 Throttle lever, –4096 to +16384, same as 088C above except that values written here are treated like axis inputs and are disconnectable via offset 310A, and have the last written value obtainable from offset 3332		
0938	2	Engine 2 Fuel Flow PPH SSL (pounds per hour, standardised to sea level). Don't know units, but it seems to match some gauges if divided by 128. Not maintained in all cases.		
094A	2	Engine 2 Anti-Ice or Carb Heat switch (1=On)		
0950	2	Engine 2 Oil temperature, 16384 = 140 C.		
0952	2	Engine 2 Oil pressure, 16384 = 55 psi. Note that in some aircraft (e.g. the B777) this can exceed the 16-bit capacity of this location. FSUIPC limits it to fit, i.e.65535 = 220 psi		
0954	2	Engine 2 Pressure Ratio (where calculated): 16384 = 1.60		
0956	2	Engine 2 EGT, 16384 = 860 C. [*Note that for Props this value is*		

		not actually correct. You will get the correct value from 3AB0. The value here has been derived by FSUIPC to be compatible with FS2004, FS2002 et cetera]		
0958	2	Engine 2 Manifold Pressure: Inches Hg * 1024		
0960	2	Engine 2 RPM Scaler: For Props, use this to calculate RPM – see offset 0930 *(On turboprops this will give the shaft RPM, since there is currently no Gear Reduction Ratio available to fix values on such aircraft. I will fix this when I can)*		
0968	4	Engine 2 Oil Quantity: 16384 = 100%		
096C	4	Engine 2 Vibration: 16384 = 5.0. This is a relative measure of amplitude from the sensors on the engine which when too high is an indication of a problem. The value at which you should be concerned varies according to aircraft and engine.		
0970	4	Engine 2 Hydraulic pressure: appears to be 4*psi		
0974	4	Engine 2 Hydraulic quantity: 16384 = 100%		
0980	8	Engine 2 CHT, degrees F in double floating point (FLOAT64)		
0988	4	Engine 2 Turbine temperature: degree C *16384		
098C	4	Engine 2 Torque % (16384 = 100%)		
0990	4	Engine 2 Fuel pressure, psf (i.e. psi*144): not all aircraft files provide this.		
09A4	4	Engine 2 fuel used since start (in pounds, 32-bit float)		
09A8	4	Engine 2 elapsed time (in hours, 32-bit float)		
09B0	8	Engine 2 Fuel Flow Pounds per Hour, as floating point double (FLOAT64)		
09B8	4	Engine 2 Torque in foot-pounds, as a 32-bit Float. (Not jets)		
09BC	152	ENGINE 3 values, as detailed below **SEE STATUS FOR ENGINE 1**		
09BC	2	Engine 3 Throttle lever, –4096 to +16384 [Programs controlling throttle directly from user inputs should write to 09CA instead if the input should be disconnectable via offset 310A/B (e.g. for auto-throttle management)]		
09BE	2	Engine 3 Prop lever, –4096 to +16384		
09C0	2	Engine 3 Mixture lever, 0 – 16384		
09C2	2	Engine 3 Starter switch position (Magnetos), Jet/turbo: 0=Off, 1=Start, 2=Gen; Prop: 0=Off, 1=right, 2=Left, 3=Both, 4=Start (see Notes in Engine 1 entry)		
09C4	2	Engine 3 combustion flag (TRUE if engine firing)		
09C6	2	Engine 3 Jet N2 as 0 – 16384 (100%)		
09C8	2	Engine 3 Jet N1 as 0 – 16384 (100%), or Prop RPM (derive RPM by multiplying this value by the RPM Scaler (see 08C8) and dividing by 65536). Note that Prop RPM is signed and negative for counter-rotating propellers.		
09CA	2	Engine 3 Throttle lever, –4096 to +16384, same as 088C above except that values written here are treated like axis inputs and are disconnectable via offset 310A/B, and have the last written value obtainable from offset 3334		
09D0	2	Engine 3 Fuel Flow PPH SSL (pounds per hour, standardised to sea level). Don't know units, but it seems to match some gauges if divided by 128. Not maintained in all cases.		
09E2	2	Engine 3 Anti-Ice or Carb Heat switch (1=On)		
09E8	2	Engine 3 Oil temperature, 16384 = 140 C.		
09EA	2	Engine 3 Oil pressure, 16384 = 55 psi. Note that in some aircraft (eg the B777) this can exceed the 16-bit capacity of this location. FSUIPC limits it to fit, i.e.65535 = 220 psi		
09EC	2	Engine 3 Pressure Ratio (where calculated): 16384 = 1.60		
09EE	2	Engine 3 EGT, 16384 = 860 C. [*Note that for Props this value is not actually correct. You will get the correct value from 39F0. The value here has been derived by FSUIPC to be compatible with FS2004, FS2002 et cetera*]		
09F0	2	Engine 3 Manifold Pressure: Inches Hg * 1024		
09F8	2	Engine 3 RPM Scaler: For Props, use this to calculate RPM – see offset 09C8		

		(On turboprops this will give the shaft RPM, since there is currently no Gear Reduction Ratio available to fix values on such aircraft. I will fix this when I can)		
0A00	4	Engine 3 Oil Quantity: 16384 = 100%		
0A04	4	Engine 3 Vibration: 16384 = 5.0. This is a relative measure of amplitude from the sensors on the engine which when too high is an indication of a problem. The value at which you should be concerned varies according to aircraft and engine.		
0A08	4	Engine 3 Hydraulic pressure: appears to be 4*psi		
0A0C	4	Engine 3 Hydraulic quantity: 16384 = 100%		
0A18	8	Engine 3 CHT, degrees F in double floating point (FLOAT64)		
0A20	4	Engine 3 Turbine temperature: degree C *16384		
0A24	4	Engine 3 Torque % (16384 = 100%)		
0A28	4	Engine 3 Fuel pressure, psf (i.e. psi*144): not all aircraft files provide this.		
0A3C	4	Engine 3 fuel used since start (in pounds, 32-bit float)		
0A40	4	Engine 3 elapsed time (in hours, 32-bit float)		
0A48	8	Engine 3 Fuel Flow Pounds per Hour, as floating point double (FLOAT64)		
0A50	4	Engine 3 Torque in foot-pounds, as a 32-bit Float. (Not jets)		
0A54	152	ENGINE 4 values, as detailed below **SEE STATUS FOR ENGINE 1**		
0A54	2	Engine 4 Throttle lever, –4096 to +16384 [Programs controlling throttle directly from user inputs should write to 0A62 instead if the input should be disconnectable via offset 310A/B (e.g. for auto-throttle management)]		
0A56	2	Engine 4 Prop lever, –4096 to +16384		
0A58	2	Engine 4 Mixture lever, 0 – 16384		
0A5A	2	Engine 4 Starter switch position (Magnetos), Jet/turbo: 0=Off, 1=Start, 2=Gen; Prop: 0=Off, 1=right, 2=Left, 3=Both, 4=Start (see Notes in Engine 1 entry)		
0A5C	2	Engine 4 combustion flag (TRUE if engine firing)		
0A5E	2	Engine 4 Jet N2 as 0 – 16384 (100%)		
0A60	2	Engine 4 Jet N1 as 0 – 16384 (100%), or Prop RPM (derive RPM by multiplying this value by the RPM Scaler (see 08C8) and dividing by 65536). Note that Prop RPM is signed and negative for counter-rotating propellers.		
0A62	2	Engine 4 Throttle lever, –4096 to +16384, same as 088C above except that values written here are treated like axis inputs and are disconnectable via offset 310A/B, and have the last written value obtainable from offset 3336		
0A68	2	Engine 4 Fuel Flow PPH SSL (pounds per hour, standardised to sea level). Don't know units, but it seems to match some gauges if divided by 128. Not maintained in all cases.		
0A7A	2	Engine 4 Anti-Ice or Carb Heat switch (1=On)		
0A80	2	Engine 4 Oil temperature, 16384 = 140 C.		
0A82	2	Engine 4 Oil pressure, 16384 = 55 psi. Note that in some aircraft (eg the B777) this can exceed the 16-bit capacity of this location. FSUIPC limits it to fit, i.e.65535 = 220 psi		
0A84	2	Engine 4 Pressure Ratio (where calculated): 16384 = 1.60		
0A86	2	Engine 4 EGT, 16384 = 860 C. [*Note that for Props this value is not actually correct. You will get the correct value from 3930. The value here has been derived by FSUIPC to be compatible with FS2004, FS2002 et cetera*]		
0A88	2	Engine 4 Manifold Pressure: Inches Hg * 1024		
0A90	2	Engine 4 RPM Scaler: For Props, use this to calculate RPM – see offset 0A60 *(On turboprops this will give the shaft RPM, since there is currently no Gear Reduction Ratio available to fix values on such aircraft. I will fix this when I can)*		
0A98	4	Engine 4 Oil Quantity: 16384 = 100%		
0A9C	4	Engine 4 Vibration: 16384 = 5.0. This is a relative measure of amplitude from the sensors on the engine which when too high is an indication of a problem. The value at which you should be		

		concerned varies according to aircraft and engine.		
0AA0	4	Engine 4 Hydraulic pressure: appears to be 4*psi		
0AA4	4	Engine 4 Hydraulic quantity: 16384 = 100%		
0AB0	8	Engine 4 CHT, degrees F in double floating point (FLOAT64)		
0AB8	4	Engine 4 Turbine temperature: degree C *16384		
0ABC	4	Engine 4 Torque % (16384 = 100%)		
0AC0	4	Engine 4 Fuel pressure, psf (i.e. psi*144): not all aircraft files provide this.		
0AD4	4	Engine 4 fuel used since start (in pounds, 32-bit float)		
0AD8	4	Engine 4 elapsed time (in hours, 32-bit float)		
0AE0	8	Engine 4 Fuel Flow Pounds per Hour, as floating point double (FLOAT64)		
0AE8	4	Engine 4 Torque in foot-pounds, as a 32-bit Float. (Not jets)		
0AEC	2	Number of Engines	**Ok-SimC**	**N/A**
0AF0	2	Propeller pitch control: 0=Fixed, 1=Auto, 2=Manual, but on FS2004 it was 0=fixed pitch, 1=constant speed, no differentiation between auto and manual.	**No**	**No**
0AF4	2	Fuel weight as pounds per gallon * 256	**Ok-SimC**	**No**
0AF8	2	Fuel tank selector: 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main, 20=Right Mai**n** *(Engine 1 only—see also separate Engine selectors)*	**Ok-SimC**	**Ok-SimE**
0B00	2	Throttle lower limit, 16384=100%. (e.g. for aircraft with reverse thrust this is normally –4096 indicating 25% in reverse)	**Ok-SimC**	**No**
0B0C	4	Mach Max Operating speed *20480	**Ok-SimC**	**No**
0B18	8	Gyro suction in inches of mercury (Hg), floating point double (FLOAT64)	**Ok-SimC**	**?-SimC**
0B20	2	Sound control: 0 to switch off, 1 to switch on	**N/A**	**Ok-SimE**
0B24	2	Sound flag: reads 0 if off, 1 if on	**Ok-SimE**	**N/A**
0B4C	2	Ground altitude (metres). See 0020 for more accuracy.	**Ok-SimC**	**N/A**
0B50	1	Bleed air source control. *Documented as 0=Min, 1=auto, 2=Off, 3=APU, 4=Engines* *But in the FSX A321 these work:* *0=Auto, 1=Shut (off), 2=APU, 3=Engines*	**Ok-SimC**	**Ok-SimE**
0B51	1	APU generator switch	**Ok-SimC**	**Ok-SimE**
0B52	1	APU generator active flag	**Ok-SimC**	**No**
0B53	1	APU on fire flag	**?-SimC**	**No**
0B54	4	APU RPM as percentage of maximum, 32-bit float	**Ok-SimC**	**No**
0B58	4	APU Starter as percentage (of what?), 32-bit float. FSUIPC4 interprets writes here as start /stop APU requests. Just write any Non-Zero value to start, or all zero to stop.	**Ok-SimC**	**Ok-SimE**
0B5C	4	APU generator voltage level, 32-bit float	**Ok-SimC**	**No**
0B60	2	Scenery complexity level, 0 – 5	**No**	**No**
0B62	1	Fail mode, 0 ok, Hydraulics failure = 1	**No-SimC+**	**?-SimE**
0B63	1	Fail mode, 0 ok, Brakes failures: Bit 0 = Left brake Bit 1 = Right brake Bit 2 = Total brake failure	**No-SimC+**	**?-SimE**
0B64	1	Fail mode: 0 ok, ADF gauge inoperable = 1 (both ADFs)	**Ok?-SimC**	**Ok-SimC**
0B65	1	Fail mode: 0 ok, ASI gauge inoperable = 1	**Ok-SimC**	**Ok-SimC**
0B66	1	Fail mode: 0 ok, Altimeter gauge inoperable = 1	**Ok-SimC**	**Ok-SimC**
0B67	1	Fail mode: 0 ok, Attitude Indicator gauge inoperable = 1	**Ok-SimC**	**Ok-SimC**
0B68	1	Fail mode: 0 ok, COM radio gauges inoperable = 1 See also 3BD6	**?-SimC**	**No-SimC+**
0B69	1	Fail mode: 0 ok, Mag Compass inoperable = 1	**SimC**	**SimC**
0B6A	1	Fail mode: 0 ok, Electrics inoperable = 1	**?-SimC**	**?-SimE**
0B6B	1	Fail mode: 0 ok, Engine inoperable = 1, extended for up to 4 individual engines: bit 0 =Engine 1 … bit 3= Engine 4.	**?-SimC**	**?-SimE**
0B6C	1	Fail mode: 0 ok, Fuel indicators inoperable = 1	**?-SimC**	**No-SimC+**
0B6D	1	Fail mode: 0 ok, Direction Indicator gauge inoperable = 1	**Ok-SimC**	**Ok-SimC**

0B6E	1	Fail mode: 0 ok, VSI gauge inoperable = 1	Ok-SimC	Ok-SimC
0B6F	1	Fail mode: 0 ok, Transponder gauge inoperable = 1	?-SimC	?-SimC
0B70	1	Fail mode: 0 ok, NAV radio gauges inoperable = 1 See also 3BD6	?-SimC	No-SimC+
0B71	1	Fail mode: 0 ok, Pitot inoperable = 1	?-SimC	?-SimC
0B72	1	Fail mode: 0 ok, Turn coordinator gauge inoperable = 1	?-SimC	No-SimC+
0B73	1	Fail mode: 0 ok, Vacuum gauge inoperable = 1	?-SimC	No-SimC+
0B74	4	Fuel: centre tank level, % * 128 * 65536	Ok-SimC	Ok-SimC
0B78	4	Fuel: centre tank capacity: US Gallons (see also offsets 1244– for extra fuel tanks)	Ok-SimC	No
0B7C	4	Fuel: left main tank level, % * 128 * 65536	Ok-SimC	Ok-SimC
0B80	4	Fuel: left main tank capacity: US Gallons	Ok-SimC	No
0B84	4	Fuel: left aux tank level, % * 128 * 65536	Ok-SimC	Ok-SimC
0B88	4	Fuel: left aux tank capacity: US Gallons	Ok-SimC	No
0B8C	4	Fuel: left tip tank level, % * 128 * 65536	Ok-SimC	Ok-SimC
0B90	4	Fuel: left tip tank capacity: US Gallons	Ok-SimC	No
0B94	4	Fuel: right main tank level, % * 128 * 65536	Ok-SimC	Ok-SimC
0B98	4	Fuel: right main tank capacity: US Gallons	Ok-SimC	No
0B9C	4	Fuel: right aux tank level, % * 128 * 65536	Ok-SimC	Ok-SimC
0BA0	4	Fuel: right aux tank capacity: US Gallons	Ok-SimC	No
0BA4	4	Fuel: right tip tank level, % * 128 * 65536	Ok-SimC	Ok-SimC
0BA8	4	Fuel: right tip tank capacity: US Gallons	Ok-SimC	No
0BAC	2	Inner Marker: activated when TRUE	Ok-SimC	No
0BAE	2	Middle Marker: activated when TRUE	Ok-SimC	No
0BB0	2	Outer Marker: activated when TRUE	Ok-SimC	No
0BB2	2	Elevator control input: –16383 to +16383	Ok-SimC	Ok-SimC
0BB4	2	Elevator position indicator (maybe adjusted from input!)	Ok-SimC	No
0BB6	2	Aileron control input: –16383 to +16383	Ok-SimC	Ok-SimC
0BB8	2	Aileron position indicator (maybe adjusted from input!) *(Note that FSX provides left and right values. Only the left is used here)*	Ok-SimC* *(see note)*	No
0BBA	2	Rudder control input: –16383 to +16383	Ok-SimC	Ok-SimC
0BBC	2	Rudder position indicator (maybe adjusted from input!)	Ok-SimC	No
0BBE	2	Helo pitch (elevator) trim control: –16383 to +16383, but only when “ApplyHeloTrim” set.	Ok-Intl	Ok-Intl
0BC0	2	Elevator trim control input: –16383 to +16383	Ok-SimC	Ok-SimC
0BC2	2	Elevator trim indicator (follows input)	Ok-SimC	No
0BC4	2	Left brake application read-out (0 off, 16383 full: parking brake=16383). You can also apply a fixed brake pressure here, or else use the byte at 0C01 to apply brakes emulating the keypress. *Note that the values READ here run from 0 to 16384, but will not match exactly the values written. They seem to follow an exponential curve, being much lower at the low end (e.g. only 33% of what is written), gradually catching up to meet at the top.*	Ok-SimC	Ok-SimC
0BC6	2	Right brake application read-out (0 off, 16383 full: parking brake=16383). You can apply a fixed brake pressure here, or else use the byte at 0C00 to apply brakes emulating the keypress. *Note that the values READ here run from 0 to 16384, but will not match exactly the values written. They seem to follow an exponential curve, being much lower at the low end (e.g. only 33% of what is written), gradually catching up to meet at the top.*	Ok-SimC	Ok-SimC
0BC8	2	Parking brake: 0=off, 32767=on	Ok-SimC	Ok-SimE
0BCA	2	Braking indicator: brake applied if non-zero (1=Left, 2=Right, 3=both	Ok-SimC	N/A
0BCC	4	Spoilers arm (0=off, 1=arm for auto deployment)	Ok-SimC	Ok-SimE
0BD0	4	Spoilers control, 0 off, 4800 arm, then 5620 (7%) to 16383 (100% fully deployed).	Ok-SimC	Ok-SimC

		The 4800 value is set by arming. Values from 0 to somewhere close to, but below, 4800 do nothing. The percentage extension is the proportion of the distance in the range 4800 to 16383, even though values 4800 to 5619 cannot be used—7% seems to be the minimum.		
0BD4	4	Spoiler Left position indicator (0-16383)	**Ok-SimC**	**No**
0BD8	4	Spoiler Right position indicator (0-16383)	**Ok-SimC**	**No**
0BDC	4	Flaps control, 0=up, 16383=full. The "notches" for different aircraft are spaced equally across this range: calculate the increment by 16383/(number of positions-1), ignoring fractions. See also offset 3BFA below. N.B. Do not expect to read this and see 100% accurate values.	**Ok-SimC**	**Ok-SimE**
0BE0	4	Flaps position indicator (left). This gives the proportional amount, with 16383=full deflection. It doesn't correspond to the equally spaced notches used for the control lever. If you know the maximum deflection angle you can derive the current angle by ((max * position indicator) / 16383). This only gives the (inboard?) trailing edge flaps. Please see offsets 30E0–30FF for greater details where needed.	**Ok-SimC*** *(see note)*	**No**
0BE4	4	Flaps position indicator (right). This gives the correct proportional amount, with 16384=full deflection. It doesn't correspond to the equally spaced notches used for the control lever. This only gives the inboard trailing edge flaps. Please see offsets 30E0–30FF for greater details where needed.	**Ok-SimC*** *(see note)*	**No**
0BE8	4	Gear control: 0=Up, 16383=Down	**Ok-SimC**	**Ok-SimC**
0BEC	4	Gear position (nose): 0=full up, 16383=full down	**Ok-SimC**	**Ok-SimC**
0BF0	4	Gear position (right): 0=full up, 16383=full down	**Ok-SimC**	**Ok-SimC**
0BF4	4	Gear position (left): 0=full up, 16383=full down	**Ok-SimC**	**Ok-Sim**
0BF8	4	Unlimited visibility value, as 1600* statute miles. This is the value set in the Display Quality Settings.	**No-SimC+**	**No**
0BFC	1	Flaps handle index (0 full up)	**Ok-SimC**	**Ok-SimC**
0BFD	1	Anti-skid Brake active indicator, non-zero when active	**OK-SimE**	**Ok-SimC**
0C00	1	Right toe brake control: 0 – 200, proportional braking with timed decay	**N/A**	**Ok-Intl**
0C01	1	Left toe brake control: 0 –200, proportional braking with timed decay	**N/A**	**Ok-Intl**
0C02	2	Aileron trim value/control: –16383 to +16383 [NEW!]	**Ok-SimC**	**?-SimC**
0C04	2	Rudder trim value/control: –16383 to +16383 [NEW!]	**Ok-SimC**	**?-SimC**
0C06	2	Helo bank (aileron) trim control: –16383 to +16383, but only when "ApplyHeloTrim" set to 'Both'.	**Ok-Intl**	**Ok-Intl**
0C08	2	Steering tiller input value (FSUIPC optional axis), -16384 to +16383, if calibrated	**Ok-Intl**	**N/A**
0C0A	2	Rudder input value, -16384 to +16383, if calibrated	**Ok-Intl**	**N/A**
0C14	4	ADF2 signal strength	**Ok-SimC**	**No**
0C18	2	International units: 0=US, 1=Metric+feet, 2=Metric+metres	**?-SimC**	**No**
0C1A	2	Simulation rate *256 (i.e. 256=1x). *(The Sim Rate values can't be written to directly, and the SIM_RATE_SET control does nothing. At present, FSUIPC4 tries to accommodate writes to this value by using INCR and DECR. This gives powers of two values, range 64 to 32768 – i.e. 1/4X to 128X. If you use intermediate values you will get the next one up or down).*	**Ok-SimE**	**No-SimE** *(see note)*
0C1C	4	ADF1 signal strength	**Ok-SimC**	**No**
0C20	9	Local time in character format: "hh:mm:ss" (with zero terminator)	**Ok-Intl**	**No**
0C29	5	DME1 distance as character string, either "nn.n" or "nnn." (when > 99.9 nm). The 5th character may be a zero or a space. Don't rely on it.	**Ok-Intl**	**N/A**
0C2E	5	DME1 speed as character string, "nnn" followed by either space then zero or just zero.	**Ok-Intl**	**N/A**

0C33	5	DME2 distance as character string, either "nn.n" or "nnn." (when > 99.9 nm). The 5th character may be a zero or a space. Don't rely on it.	**Ok-Intl**	**N/A**
0C38	5	DME2 speed as character string, "nnn" followed by either space then zero or just zero.	**Ok-Intl**	**N/A**
0C3E	2	Gyro drift amount (*360/65536 for degrees). Note that whilst it may appear that the value is accurate to fractions of a degree, the actual setting capability (via an event) is based on whole degrees, just like the INC/DEC controls. Any value written here will normally be read back slightly differently, based upon this granularity.	**Ok-SimC**	**Ok-SimE**
0C40	2	NAV1 Mag Var (*360/65536 for degrees) *(Note that there are two different data sources for MagVars, and this may not agree with the airport MagVar for airport-based VORs)*	**Ok-SimC** *(but see note)*	**No**
0C42	2	NAV2 Mag Var (*360/65536 for degrees) *(Note that there are two different data sources for MagVars, and this may not agree with the airport MagVar for airport-based VORs)*	**Ok-SimC** *(but see note)*	**No**
0C44	2	Realism setting, 0 – 100	**Ok-SimC**	**No**
0C48	1	NAV1 Localiser Needle: –127 left to +127 right	**Ok-SimC**	**No**
0C49	1	NAV1 Glideslope Needle: –119 up to +119 down	**Ok-SimC**	**No**
0C4A	1	NAV1 Back Course flags: 0 BC available 1 Localiser tuned in 2 On Back Course *(Not found for FSX)* 7 Station active (even if no BC)	**Ok-SimC** *(see note)*	**No**
0C4B	1	NAV1 To/From flag: 0=not active, 1=To, 2=From	**Ok-SimC**	**No**
0C4C	1	NAV1 GS flag: TRUE if GS alive	**Ok-SimC**	**No**
0C4D	1	NAV1 code flags, bits used as follows: 0 DME available 1 TACAN *(Not found for FSX)* 2 Voice available *(Not found for FSX)* 3 No signal available 4 DME/GS co-located *(Not found for FSX)* 5 No back course 6 GS available 7 This is a localiser (else it's a VOR)	**Ok-SimC** *(see notes)*	**No**
0C4E	2	NAV1 OBS setting (degrees, 0–359)	**Ok-SimC**	**Ok-SimE**
0C50	2	NAV1 radial (*360/65536 for degrees). Note that this is in degrees Magnetic for a VOR, but TRUE for an ILS LOC.	**Ok-SimC**	**No**
0C52	4	NAV1 signal strength: For Localisers, seems to be either 0 or 256 For VORs varies from 0 to over 1,000,000 when really close!	**Ok-SimC**	**No**
0C56	2	NAV1: relative bearing to VOR1, in degrees (0–359)	**Ok-SimC**	**No**
0C59	1	NAV2 Localiser Needle: –127 left to +127 right	**Ok-SimC**	**No**
0C5A	1	NAV2 Back Course flags: 0 BC available 1 Localiser tuned in 2 On Back Course *(Not found for FSX)* 7 Station active (even if no BC)	**Ok-SimC** *(but see note)*	**No**
0C5B	1	NAV2 To/From flag: 0=not active, 1=To, 2=From	**Ok-SimC**	**No**
0C5C	2	NAV2: relative bearing to VOR2, in degrees (0–359)	**Ok-SimC**	**No**
0C5E	2	NAV2 OBS setting (degrees, 0–359)	**Ok-SimC**	**Ok-SimE**
0C60	2	NAV2 radial (*360/65536 for degrees). Note that this is in degrees Magnetic for a VOR, but TRUE for an ILS LOC.	**Ok-SimC**	**No**
0C62	4	NAV2 signal strength: For Localisers, seems to be either 0 or 256 For VORs varies from 0 to over 1,000,000 when really close!	**Ok-SimC**	**No**
0C6A	2	ADF1: relative bearing to NDB (*360/65536 for degrees, –ve left, +ve right)	**Ok-SimC**	**No**
0C6C	2	ADF1: dial bearing, where adjustable (in degrees, 1–360)	**?-SimC**	**?-SimE**
0C6E	1	NAV2 Glideslope Needle: –127 up to +127 down	**?-SimC**	**No**

0C6F	1	NAV2 GS flag: TRUE if GS alive	**?-SimC**	**No**
0C70	1	NAV2 code flags, bits used as follows: 0 DME available 1 TACAN *(Not found for FSX)* 2 Voice available *(Not found for FSX)* 3 No signal available 4 DME/GS co-located *(Not found for FSX)* 5 No back course 6 GS available 7 This is a localiser (else it's a VOR)	**Ok-SimC** *(see notes)*	**No**
0C92	2	Texture quality, 0–3, as on slider in Display Quality	**No**	**No**
0D0C	2	Lights, a switch for each one (bits from lo to hi): 0 Navigation 1 Beacon 2 Landing 3 Taxi 4 Strobes 5 Instruments 6 Recognition 7 Wing 8 Logo 9 Cabin	**Ok-SimC**	**Ok-SimE** *(Intl decode)*
0D50	24	The Tower Latitude (8 bytes), Longitude (8 bytes) and Altitude (8 bytes) in the same format as 0560–0577 above.	**No-SimC+**	**No-SimC+**
0D6C	4	Parameter associated with any Macro, Lua or L:Var request sent to the following offset (0D70)	**N/A**	**Ok-Intl**
0D70	40	**Macros and Lua requests** Write here the complete identity string of a Macro control or Lua program control in order to have FSUIPC execute it. For a Macro, the string should begin with up to 16 characters giving the .MCRO file name (just the name part, not the type), and then, separated by a ':' character, the macro name within that file—again, up to 16 characters. Spaces either side of the ':' are optional. For a Lua program operation, the actual Lua control should be provided, followed (with one space or ':' separator) by the Lua program name (without the .Lua suffix). The valid Lua controls are: Lua, LuaDebug, LuaKill, LuaSet, LuaClear, LuaToggle Note that a parameter should always be written first for the Set, Clear and Toggle controls as this specifies the flag to be changed (0–31). A parameter is never used with "Lua Kill". If a parameter is to be supplied, it should first be written to offset 0D6C, above. Otherwise whatever was last written there will be supplied. **L:Var read and write requests** First write the offset address to which the resulting value (an 8-byte double or FLT64) will be written (for a Read) or the value to be written can be found (for a Write). This MUST be one of the user offsets, i.e. in the range 0x66C0 to 0x66F8 (or up to 0x66FF depending on the next setting). This offset value only occupies the low 16-bits (LOWORD) of the 32-bit value. The high part specifies the value format. Assuming the offset is 'nnnn', the options are: 0x0nnnn for 64-bit double (as before) 0x1nnnn for 32-bit float (FLT) 0x2nnnn for 32-bit signed integer (SD)	**N/A**	**Ok-Intl**

		0x3nnnn for 32-bit unsigned integer (UD) 0x4nnnn for 16-bit signed integer (SW) 0x5nnnn for 16-bit unsigned integer (UW) 0x6nnnn for 8-bit signed integer (SB) 0x7nnnn for 8-bit unsigned integer (UB) With reads into a fixed point value (the last 6 above), the floating point value provided from the Gauge system is rounded to the nearest integer (up for positive numbers, down for negative). Then write to 0D70 the name of the LVar, preceded by just one : (colon) character for a read or :: (two colons) for a write, and terminated by a zero byte. The reason for the use of user offsets is to avoid corruption when more than one application is running which reads L:Vars in this way. It is a matter for the programs, probably with user cooperation, to avoid clashes. Both 0D6C and 0D70 can be written together or at least in one Process call, and the result of a read can be read immediately, even in the same Process call. For a write the value to be written can be placed in the stated offset in the same Process call too, provided it is before the writes to 0D6C and 0D70. If the Lvar does not currently exist the result of a read will be 0.0. There's no way to detect if a write succeeded other than to read the L:var afterwards. If the offset provided is invalid the request is just ignored and the offset value unchanged.		
0D98	2	International N/S setting: 2=North, 3=South	**No-SimC+**	**No**
0D9C	2	International E/W setting: 0=East, 1=West	**No-SimC+**	**No**
0DD6	2	Scenery BGL variable "usrvar" (originally 0312h in BGL)	**No-SimC+**	**No-SimC+**
0DD8	2	Scenery BGL variable "usrvr2" (originally 0314h in BGL)	**No-SimC+**	**No-SimC+**
0DDA	2	Scenery BGL variable "usrvr3" (originally 0316h in BGL)	**No-SimC+**	**No-SimC+**
0DDC	2	Scenery BGL variable "usrvr4" (originally 0318h in BGL)	**No-SimC+**	**No-SimC+**
0DDE	2	Scenery BGL variable "usrvr5" (originally 031Ah in BGL)	**No-SimC+**	**No-SimC+**
0E00	2	Default 738 and A321 EFIS: ND scale: 738: 0=5nm up to 7=640nm A321: 0=10nm up to 5=320nm	**Ok-Lvar**	**Ok-Lvar**
0E02	2	Default 738 EFIS: ND mode: 0=APP, 1=VOR, 2=MAP	**Ok-Lvar**	**Ok-Lvar**
0E04	2	Default 738 and A321 EFIS: ND map items shown: 738: 0=WPT, 1=APT, 2=NDB, 3=VOR A321: 0=WPT, 1=VOR, 2=NDB, 3=APT	**Ok-Lvar**	**Ok-Lvar**
0E06	2	Default 738 EFIS: ND VOR/ADF1 switch: 0=VOR, 1=OFF, 2=ADF	**Ok-Lvar**	**Ok-Lvar**
0E08	2	Default 738 EFIS: ND VOR/ADF2 switch: 0=VOR, 1=OFF, 2=ADF	**Ok-Lvar**	**Ok-Lvar**
0E0A	2	Default 738 EFIS: ND arc=0, centred=1	**Ok-Lvar**	**Ok-Lvar**
0E0C	2	Default 738 EFIS: AP speed/mach C/O button (pressed if 1, not pressed if 0). Only useful reading. Write has no effect except graphical.	**Ok-Lvar**	**No**
0E0E	2	Default A321 EFIS: ND mode: 0=ILS, 1=VOR, 2=NAV, 3=ARC	**Ok-Lvar**	**Ok-Lvar**
0E10	2	Default A321 EFIS: ND VOR/ADF1 switch: 0=VOR, 1=OFF, 2=ADF	**Ok-Lvar**	**Ok-Lvar**
0E12	2	Default A321 EFIS: ND VOR/ADF2 switch: 0=VOR, 1=OFF, 2=ADF	**Ok-Lvar**	**Ok-Lvar**
0E14	2	Default A321 EFIS: ND InHg/hPA switch, 0=InHg, 1=hPA	**Ok-Lvar**	**Ok-Lvar**
0E16	2	Default A321 EFIS: ND ILS mode button, 0 = off, 1=on	**Ok-Lvar**	**Ok-Lvar**
0E18	2	Default A321 EFIS: AP speed/mach C/O button (pressed if 1, not pressed if 0). Only useful reading. Write has no effect except graphical.	**Ok-Lvar**	**No**

0E1A	2	Default A321 EFIS: Altitude change rate switch (0 = 100, 1=1000)	**Ok-Lvar**	**Ok-Lvar**
0E80	4	ICAO id of nearest weather station, if FSUIPC4 is reading weather. This is 4 ASCII characters, no zero terminator.	**Ok-Intl**	**No**
0E84	1	At aircraft altitude: cloud type, 1–10, if the aircraft is in a cloud layer. Otherwise 0	**Ok-Intl**	**No**
0E85	1	At aircraft altitude: cloud coverage in Oktas (0-8)	**Ok-Intl**	**No**
0E86	2	At aircraft altitude: cloud icing lelel, 0-4	**Ok-Intl**	**No**
0E88	2	At aircraft altitude: cloud turbulence level 0-255 (see 0EFC). (Actual values 0, 72, 144, 216, 252)	**Ok-Intl**	**No**
0E8A	2	Current visibility (Statue miles * 100) *("Ambient visibility")*	**Ok-SimC**	**No-SimC**
0E8C	2	Outside Air Temperature (OAT), degrees C * 256 *("Ambient Temperature")*	**Ok-SimC**	**No**
0E8E	2	Dew point, degrees C * 256. This is the interpolated value for the aircraft altitude, as supplied by FSX.	**Ok-SimC**	**No**
0E90	2	Ambient wind speed (at aircraft) in knots	**Ok-SimC**	**No-SimC**
0E92	2	Ambient wind direction (at aircraft), *360/65536 to get degrees True.	**Ok-SimC**	**No-SimC**
0E94	2	At aircraft altitude: wind gusting value: max speed in knots, or 0 if no gusts	**Ok-Intl**	**No**
0E96	2	At aircraft altitude: Wind directional variation—degrees in the same units as wind directions	**Ok-Intl**	**No**
0E98	2	At aircraft altitude: Wind turbulence value, 0–255, just like offset 0ED2, etc (Actual values 0, 64, 128, 192, 255)	**Ok-Intl**	**No**
0E9A	112	FS98 style Current Aircraft Weather* as Set: details follow. [See 0F1C for Global weather *setting* area] N.B. See also 0E8A above, which is the "current" visibility equivalent of the global setting at 0F8C. * FSX supplies interpolated weather for the aircraft position, including altitude. Hence for layered weather aspects the only accurate values are for the altitude of the aircraft. This applies to temperature and wind layers. The other layers are populated by FSUIPC4 from the weather reported by the *nearest* Weather Station.	**Ok-SimC** *(but see notes)*	***See 0F1A***
0E9A	2	Upper cloud layer ceiling in metres AMSL		
0E9C	2	Upper cloud layer base in metres AMSL		
0E9E	2	Upper cloud layer coverage, 65535 = 8 oktas, … 32768= 4 oktas … 0 = clear		
0EA0	2	Upper cloud layer, cloud altitude variation (metres)		
0EA2	2	Lower cloud layer ceiling in metres AMSL		
0EA4	2	Lower cloud layer base in metres AMSL		
0EA6	2	Lower cloud layer coverage, 65535 = 8 oktas, … 32768= 4 oktas … 0 = clear		
0EA8	2	Lower cloud layer, cloud altitude variation (metres)		
0EAA	2	Storm layer ceiling in metres AMSL		
0EAC	2	Storm layer base in metres AMSL (if a Storm layer is present, it must be the lowest, below "Lower Cloud").		
0EAE	2	Storm cloud layer coverage, 65535 = 8 oktas, … 32768= 4 oktas … 0 = clear		
0EB0	2	Storm cloud layer, cloud altitude variation (metres)		
0EB2	2	Upper Temperature level, metres AMSL		
0EB4	2	Upper Temperature in degrees C * 256		
0EB6	2	Middle Temperature level, metres AMSL		
0EB8	2	Middle Temperature in degrees C * 256		
0EBA	2	Lower Temperature level, metres AMSL		
0EBC	2	Lower Temperature in degrees C * 256		
0EBE	2	Surface Temperature level, metres AMSL (best to be the ground elevation)		
0EC0	2	Surface Temperature in degrees C * 256		
0EC2	2	Temperature drift, degrees C *256 (not used?)		
0EC4	2	Temperature day/night variation, degrees C *256		

0EC6	2	Pressure (QNH) as millibars (hectoPascals) *16.		
0EC8	2	Pressure drift as millibars *16 (not used?)		
0ECA	2	Upper wind ceiling, metres AMSL		
0ECC	2	Upper wind base, metres AMSL		
0ECE	2	Upper wind speed, knots		
0ED0	2	Upper wind direction, *360/65536 gives degrees True		
0ED2	2	Upper wind turbulence setting, 0 none, 64, 128, 192, 255 worst		
0ED4	2	Upper wind gusts, enabled if True.		
0ED6	2	Middle wind ceiling, metres AMSL		
0ED8	2	Middle wind base, metres AMSL		
0EDA	2	Middle wind speed, knots		
0EDC	2	Middle wind direction, *360/65536 gives degrees True		
0EDE	2	Middle wind turbulence setting, 0 none, 64, 128, 192, 255 worst		
0EE0	2	Middle wind gusts, enabled if True.		
0EE2	2	Lower wind ceiling, metres AMSL		
0EE4	2	Lower wind base, metres AMSL		
0EE6	2	Lower wind speed, knots		
0EE8	2	Lower wind direction, *360/65536 gives degrees True		
0EEA	2	Lower wind turbulence setting, 0 none, 64, 128, 192, 255 worst		
0EEC	2	Lower wind gusts, enabled if True.		
0EEE	2	Surface wind ceiling, metres AGL		
0EF0	2	Surface wind speed, knots. [See also 04D8]		
0EF2	2	Surface wind direction, *360/65536 gives degrees Magnetic (!). [See also 04DA]		
0EF4	2	Surface wind turbulence setting, 0 none, 64, 128, 192, 255 worst		
0EF6	2	Surface wind gusts, enabled if True.		
0EF8	2	Upper cloud layer type: 0=user-defined, 1=cirrus, 8=stratus, 9=cumulus		
0EFA	2	Upper cloud layer icing: enabled if True		
0EFC	2	Upper cloud layer turbulence (0 to 255). Divided into steps by FSUIPC: 0, 72, 144, 216, 252.		
0EFE	2	Lower cloud layer type: 0=user-defined, 1=cirrus, 8=stratus, 9=cumulus		
0F00	2	Lower cloud layer icing: enabled if True		
0F02	2	Lower cloud layer turbulence (0 to 255. Divided into steps by FSUIPC: 0, 72, 144, 216, 252		
0F04	2	Storm layer type: 10=storm. [FSUIPC allows this to be a third and lowest layer of any type, so then: 0=user-defined, 1=cirrus, 8=stratus, 9=cumulus]		
0F06	2	Storm layer icing: enabled if True		
0F08	2	Storm layer turbulence (0 to 255. Divided into steps by FSUIPC: 0, 72, 144, 216, 252		
0F1C	114	FS98 style Global Weather setting area: details follow.	**As 0E9A**	**Ok-Intl (sets Global weather mode)**
0F1C	2	Upper cloud layer ceiling in metres AMSL		
0F1E	2	Upper cloud layer base in metres AMSL		
0F20	2	Upper cloud layer coverage, 65535 = 8 oktas, … 32768= 4 oktas … 0 = clear		
0F22	2	Upper cloud layer, cloud altitude variation (metres)		
0F24	2	Lower cloud layer ceiling in metres AMSL		
0F26	2	Lower cloud layer base in metres AMSL		
0F28	2	Lower cloud layer coverage, 65535 = 8 oktas, … 32768= 4 oktas … 0 = clear		
0F2A	2	Lower cloud layer, cloud altitude variation (metres)		
0F2C	2	Storm layer ceiling in metres AMSL		
0F2E	2	Storm layer base in metres AMSL (if a Storm layer is present, it must be the lowest, below “Lower Cloud”).		
0F30	2	Storm cloud layer coverage, 65535 = 8 oktas, … 32768= 4 oktas … 0 = clear		
0F32	2	Storm cloud layer, cloud altitude variation (metres)		
0F34	2	Upper Temperature level, metres AMSL		

0F36	2	Upper Temperature in degrees C * 256		
0F38	2	Middle Temperature level, metres AMSL		
0F3A	2	Middle Temperature in degrees C * 256		
0F3C	2	Lower Temperature level, metres AMSL		
0F3E	2	Lower Temperature in degrees C * 256		
0F40	2	Surface Temperature level, metres AMSL (set this to the ground elevation of the weather reporting station)		
0F42	2	Surface Temperature in degrees C * 256		
0F44	2	Temperature drift, degrees C *256 (not used?)		
0F46	2	Temperature day/night variation, degrees C *256		
0F48	2	Pressure (QNH) as millibars (hectoPascals) *16.	**Ok-SimC**	
0F4A	2	Pressure drift as millibars *16 (not used?)		
0F4C	2	Upper wind ceiling, metres AMSL		
0F4E	2	Upper wind base, metres AMSL		
0F50	2	Upper wind speed, knots		
0F52	2	Upper wind direction, *360/65536 gives degrees True		
0F54	2	Upper wind turbulence setting, 0 none, 64, 128, 192, 255 worst		
0F56	2	Upper wind gusts, enabled if True.		
0F58	2	Middle wind ceiling, metres AMSL		
0F5A	2	Middle wind base, metres AMSL		
0F5C	2	Middle wind speed, knots		
0F5E	2	Middle wind direction, *360/65536 gives degrees True		
0F60	2	Middle wind turbulence setting, 0 none, 64, 128, 192, 255 worst		
0F62	2	Middle wind gusts, enabled if True.		
0F64	2	Lower wind ceiling, metres AMSL		
0F66	2	Lower wind base, metres AMSL		
0F68	2	Lower wind speed, knots		
0F6A	2	Lower wind direction, *360/65536 gives degrees True		
0F6C	2	Lower wind turbulence setting, 0 none, 64, 128, 192, 255 worst		
0F6E	2	Lower wind gusts, enabled if True.		
0F70	2	Surface wind ceiling, metres AGL		
0F72	2	Surface wind speed, knots. [See also 04D8]		
0F74	2	Surface wind direction, *360/65536 gives degrees Magnetic (!). [See also 04DA]		
0F76	2	Surface wind turbulence setting, 0 none, 64, 128, 192, 255 worst		
0F78	2	Surface wind gusts, enabled if True.		
0F7A	2	Upper cloud layer type: 0=user-defined, 1=cirrus, 8=stratus, 9=cumulus		
0F7C	2	Upper cloud layer icing: enabled if True		
0F7E	2	Upper cloud layer turbulence (0 to 255). Divided into steps by FSUIPC: 0, 72, 144, 216, 252.		
0F80	2	Lower cloud layer type: 0=user-defined, 1=cirrus, 8=stratus, 9=cumulus		
0F82	2	Lower cloud layer icing: enabled if True		
0F84	2	Lower cloud layer turbulence (0 to 255). Divided into steps by FSUIPC: 0, 72, 144, 216, 252.		
0F86	2	Storm layer type: 10=storm. [FSUIPC allows this to be a third and lowest layer of any type, so then: 0=user-defined, 1=cirrus, 8=stratus, 9=cumulus]		
0F88	2	Storm layer icing: enabled if True		
0F8A	2	Storm layer turbulence (0 to 255). Divided into steps by FSUIPC: 0, 72, 144, 216, 252.		
0F8C	2	Visibility setting as 100 * statute miles		
0FF0	16	This was previously the Path and Filename reading facility, as follows, for reading into offset 1000 one of:: 1. The default Flight path 2. The AI traffic pathname for a specified AI aircraft (see parameter) [FS2004 only] 3. The filename (no path) of the last saved Flight (FLT) file.	**See text**	**Not used**

		However, since version 3.47 of FSUIPC, the filename of the last saved flight has been readable directly at offset 0400. So it really isn't needed here with a complex protocol, and at present there are no plans to support the AI traffic pathname option in FSX or beyond (though if it requested I would look at placing it elsewhere). So, there's only one use for the area at 1000 now and that is as shown below. Consequently, for compatibility, FSUIPC will now always set 0FF0 to zero and continually change the timestamp at 0FFC		
1000	256	The full path to the folder where FS will save flights, in UNC format (i.e. \\pcname\ …) if possible and WideFS is in use, otherwise local PC format (drive:\ …).	**Ok-Intl**	**N/A**
1100	4	Inner Marker Latitude in FS form. Convert to degrees by *90/10001750.	**?-SimC**	**No**
1104	4	Inner Marker Longitude in FS form. Convert to degrees by *360/(65536*65536).	**?-SimC**	**No**
1108	4	Inner Marker Altitude in metres	**?-SimC**	**No**
110C	4	Middle Marker Latitude in FS form. Convert to degrees by *90/10001750.	**?-SimC**	**No**
1110	4	Middle Marker Longitude in FS form. Convert to degrees by *360/(65536*65536).	**?-SimC**	**No**
1114	4	Middle Marker Altitude in metres	**?-SimC**	**No**
1118	4	Outer Marker Latitude in FS form. Convert to degrees by *90/10001750.	**?-SimC**	**No**
111C	4	Outer Marker Longitude in FS form. Convert to degrees by *360/(65536*65536).	**?-SimC**	**No**
1120	4	Outer Marker Altitude in metres	**?-SimC**	**No**
1124	4	ADF1 Latitude in FS form. Convert to degrees by *90/10001750.	**?-SimC**	**No**
1128	4	ADF1 Longitude in FS form. Convert to degrees by *360/(65536*65536).	**?-SimC**	**No**
112C	4	ADF1 Altitude in metres	**?-SimC**	**No**
1130	4	ADF2 Latitude in FS form. Convert to degrees by *90/10001750.	**?-SimC**	**No**
1134	4	ADF2 Longitude in FS form. Convert to degrees by *360/(65536*65536).	**?-SimC**	**No**
1138	4	ADF2 Altitude in metres	**?-SimC**	**No**
1140	8	G-Force: the full 'raw' value from FS's SimConnect	**SimC**	**?-SimC**
115E	1	Time of day indicator, 0=Dawn, 1=Day, 2=Dusk, 3=Night. Set according to the local time, read for lighting effects and so on in BGLs. *(Note change from FS9: both dawn and dusk were 2, and night was 4, not 3)*	**Ok-SimC**	**No**
11A2	1	Ground scenery shadows on/off (1=On, 2=Off).	**No**	**No**
11A4	2	Aircraft shadows on/off. Can write to this to control them (1= On, 0=Off).	**No**	**No**
11B6	1	Aircraft reflections on/off. (2=On, 1=Off).	**No**	**No**
11B8	2	G Force: copy of 11BA on touchdown.	**SimC**	**No**
11BA	2	G Force: units unknown, but /624 seems to give quite sensible values. See also offset 1140	**SimC**	**?-SimC**
11BE	2	Angle of Attack Indicator angle, with 360 degrees = 65536. The value 32767 is 180 degrees Angle of Attack. The angle is expressed in the usual FS 16-bit angle units (360 degrees = 65536), with 180 degrees pointing to the 0.0 position (right and down about 35 degrees in a Boeing type AofA indicator). Note that the indicator angle actually decreases as the wing AofA increases. The FS9 and earlier interpretation was documented as a relative value, giving in %*32767 the difference between the current AofA and the maximum angle of attack for the current aircraft,	**Ok-SimC**	**No**

		Really this revised understanding does not conflict with this, as the indicator would presumably vary from aircraft ot aircraft in any case.		
11C6	2	Mach speed *20480.	**Ok-SimC**	**No**
11D0	2	Total Air Temperature (TAT), degrees Celsius * 256	**Ok-SimC**	**No**
123E	1	Fuel: number of fuel selectors available in this aircraft	**Ok-SimC**	**No**
123F	1	Fuel: unlimited fuel is set in "realism" if this is non-zero	**Ok-SimC**	**No**
1240	4	Fuel: total capacity in gallons (32-bit integer)	**Ok-SimC**	**No**
1244	4	Fuel: centre 2 tank level, % * 128 * 65536	**Ok-SimC**	**Ok-SimC**
1248	4	Fuel: centre 2 tank capacity: US Gallons	**Ok-SimC**	**No**
124C	4	Fuel: centre 3 tank level, % * 128 * 65536	**Ok-SimC**	**Ok-SimC**
1250	4	Fuel: centre 3 tank capacity: US Gallons	**Ok-SimC**	**No**
1254	4	Fuel: external 1 tank level, % * 128 * 65536	**Ok-SimC**	**Ok-SimC**
1258	4	Fuel: external 1 tank capacity: US Gallons	**Ok-SimC**	**No**
125C	4	Fuel: external 2 tank level, % * 128 * 65536	**Ok-SimC**	**Ok-SimC**
1260	4	Fuel: external 2 tank capacity: US Gallons	**Ok-SimC**	**No**
1264	4	Fuel: total quantity in gallons (32-bit integer)	**Ok-SimC**	**No**
1268	4	Fuel: selected quantity in gallons (32-bit integer)	**Ok-SimC**	**No**
126C	4	Fuel: total quantity weight in pounds (32-bit integer)	**Ok-SimC**	**No**
1270	4	Estimated fuel flow at cruise, in pounds per hour (32-bit integer)	**Ok-SimC**	**No**
1274	2	Text display mode (eg for ATIS): =0 static, =1 scrolling	**No**	**No**
132C	4	NAV/GPS switch. 0=NAV, 1=GPS	**Ok-SimC**	**Ok-SimE**
1330	4	Empty weight, lbs * 256. This is the aircraft weight without the payload and fuel.	**?-SimC**	**No**
1334	4	Max Gross weight, lbs * 256. This is the maximum aircraft weight including payload and fuel.	**?-SimC**	**No**
13FC	4	Count of Payload Stations	**Ok-SimC**	**No**
1400	48 x n	A set of Payload Station data, 48 bytes for each payload station (the count is in 13FC above). Each 48 byte entry contains: 0 double weight (lbs) *(Okay in FSX)* 8 double, lat dist from datum (ft) *(not FSX)* 16 double vert dist from datum (ft) *(not FSX)* 24 double longl dist from datum (ft) *(not FSX)* 32 char Name[16], zero at end *(Okay in FSX)* There's room for up to 61 such stations here. If there are more you can't access them this way. These loadings can be changed, and this does have some effect, but are changes are being promulgated to the overall weights (offsets 30C0, 30C8, 3BFC) and balance (2EF8)? Needs checking *in FSX*.	**Ok-SimC** **Missing parts: ?-simC+**	**?-SimC** *(weight values only)*
1F80	40	Write-only area for a TCAS_DATA structure, used to add entries to the TCAS data tables (but NOT to create AI aircraft, please note!). The 40-byte format is as for the TCAS_DATA structure (see offset F080). You need to write it all as one FSUIPC_Write block. You cannot read back what you have written here. You can add more writes to the same (or other) offsets before actually sending them (e.g. via FSUIPC_Process). The only important thing is that the whole TCAS_DATA structure is written in one block, with the length obviously set to 40. The data this structure should contain is as follows: id Any id number UNIQUE to all aircraft you supply. It does not have to be unique to the AI aircraft. FSUIPC keeps an internal flag to distinguish the two types. [*Note that if in the future this field is re-used for other indications, FSUIPC may have to adjust the value supplied*]. lat, lon, alt, hdg, gs, vs, com1 As possible: all would be good, but obviously a	**N/A**	**Ok-Intl**

		minimum of lat/lon/alt. idATC Any string of up to 14, plus a zero terminator, to identify the aircraft. This doesn't need to be unique but it could be rather confusing to the user if it isn't. To erase an aircraft provide the specific id for that entry, and set the idATC field to null (i.e. zero length string, just a zero). In any case, FSUIPC will automatically erase any externally supplied aircraft after about 8–12 seconds if it receives no further updates in that time. Even if the aircraft is static you'll need to supply updates for it regularly. Apart from the user-adjustable range, which is applied, FSUIPC is not performing any filtering for these aircraft—i.e. you can include aircraft on the ground if required. However, once the airborne TCAS table is full (current capacity 96) whether with AI aircraft, MP aircraft, or a mixture, no others will be accepted until slots become free. So in this sense slot management is up to you.		
2000	8	Turbine Engine 1 N1 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.	**Ok-SimC**	**?-SimC**
2008	8	Turbine Engine 1 N2 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.	**Ok-SimC**	**?-SimC**
2010	8	Turbine Engine 1 corrected N1 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.	**Ok-SimC**	**?-SimC**
2018	8	Turbine Engine 1 corrected N2 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.	**Ok-SimC**	**?-SimC**
2020	8	Turbine Engine 1 corrected fuel flow (pounds per hour) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.	**Ok-SimC**	**?-SimC**
2028	8	Turbine Engine 1 max torque fraction (range 0.0–1.0) as a double (FLOAT64).	**?-SimC**	**?-SimC**
2030	8	Turbine Engine 1 EPR as a double (FLOAT64). This is for jets and turboprops.	**Ok-SimC**	**?-SimC**
2038	8	Turbine Engine 1 ITT (interstage turbine temperature) in degrees Rankine, as a double (FLOAT64). This is for jets and turboprops.	**Ok-SimC**	**?-SimC**
2048	4	Turbine Engine 1 Afterburner switch (1 = on, 0 = off)	**Ok-SimC**	**?-SimE**
204C	8	Turbine Engine 1 jet thrust, in pounds, as a double (FLOAT64). This is the jet thrust. See 2410 for propeller thrust (turboprops have both).	**Ok-SimC**	**No**
2054	4	Turbine Engine 1 Tank Selector: 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main, 20=Right Main	**Ok-SimC**	**Ok-SimE**
2058	4	Turbine Engine 1 Tanks Used, a bit mask: 0 Center 1 1 Center 2 2 Center 3 3 Left Main 4 Left Aux 5 Left Tip 6 Right Main 7 Right Aux 8 Right Tip 9 External 1 10 External 2	**Ok-SimC**	**No**

205C	4	Turbine Engine 1, number of fuel tanks available	Ok-SimC	No
2060	8	Turbine Engine 1 fuel flow (pounds per hour) as a double (FLOAT64). This is for jets and turboprops.	Ok-SimC	No
2068	4	Turbine Engine 1 Fuel Available flag	?-SimC	No
206C	8	Turbine Engine 1 bleed air pressure (pounds per square inch) as a double (FLOAT64). This is for jets and turboprops.	Ok-SimC	No
207C	8	Turbine Engine 1 reverser fraction, a double (FLOAT64), in the range 0.0–1.0, providing the reverse as a proportion of the maximum reverse throttle position.	Ok-SimC	No
2084	8	Turbine Engine 1 Vibration	?-SimC	No
208C	4	Turbine Engine 1 Ignition Switch	Ok-SimC	Ok-SimE
2100	8	Turbine Engine 2 N1 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2108	8	Turbine Engine 2 N2 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2110	8	Turbine Engine 2 corrected N1 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2118	8	Turbine Engine 2 corrected N2 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2120	8	Turbine Engine 2 corrected fuel flow (pounds per hour) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2128	8	Turbine Engine 2 max torque fraction (range 0.0–1.0) as a double (FLOAT64).		
2130	8	Turbine Engine 2 EPR as a double (FLOAT64). This is for jets and turboprops.		
2138	8	Turbine Engine 2 ITT (interstage turbine temperature) in degrees Rankine, as a double (FLOAT64). This is for jets and turboprops.		
2148	4	Turbine Engine 2 Afterburner switch (1 = on, 0 = off)		
214C	8	Turbine Engine 2 jet thrust, in pounds, as a double (FLOAT64). This is the jet thrust. See 2510 for propeller thrust (turboprops have both).		
2154	4	Turbine Engine 2 tank selector: 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main, 20=Right Main		
2158	4	Turbine Engine 2 tanks used, a bit mask: 0 Center 1 1 Center 2 2 Center 3 3 Left Main 4 Left Aux 5 Left Tip 6 Right Main 7 Right Aux 8 Right Tip 9 External 1 10 External 2		
215C	4	Turbine Engine 2, number of fuel tanks available		
2160	8	Turbine Engine 2 fuel flow (pounds per hour) as a double (FLOAT64). This is for jets and turboprops.		
2168	4	Turbine Engine 2 fuel available flag		
216C	8	Turbine Engine 2 bleed air pressure (pounds per square inch) as a double (FLOAT64). This is for jets and turboprops.		

217C	8	Turbine Engine 2 reverser fraction, a double (FLOAT64), in the range 0.0–1.0, providing the reverse as a proportion of the maximum reverse throttle position.		
2184	8	Turbine Engine 2 vibration		
218C	4	Turbine Engine 2 Ignition Switch	Ok-SimC	Ok-SimE
2200	8	Turbine Engine 3 N1 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2208	8	Turbine Engine 3 N2 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2210	8	Turbine Engine 3 corrected N1 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2218	8	Turbine Engine 3 corrected N2 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2220	8	Turbine Engine 3 corrected fuel flow (pounds per hour) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2228	8	Turbine Engine 3 max torque fraction (range 0.0–1.0) as a double (FLOAT64).		
2230	8	Turbine Engine 3 EPR as a double (FLOAT64). This is for jets and turboprops.		
2238	8	Turbine Engine 3 ITT (interstage turbine temperature) in degrees Rankine, as a double (FLOAT64). This is for jets and turboprops.		
2248	4	Turbine Engine 3 Afterburner switch (1 = on, 0 = off)		
224C	8	Turbine Engine 3 jet thrust, in pounds, as a double (FLOAT64). This is the jet thrust. See 2610 for propeller thrust (turboprops have both).		
2254	4	Turbine Engine 3 tank selector: : 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main, 20=Right Main		
2258	4	Turbine Engine 3 tanks used, a bit mask: 0 Center 1 1 Center 2 2 Center 3 3 Left Main 4 Left Aux 5 Left Tip 6 Right Main 7 Right Aux 8 Right Tip 9 External 1 10 External 2		
225C	4	Turbine Engine 3, number of fuel tanks available		
2260	8	Turbine Engine 3 fuel flow (pounds per hour) as a double (FLOAT64). This is for jets and turboprops.		
2268	4	Turbine Engine 3 fuel available flag		
226C	8	Turbine Engine 3 bleed air pressure (pounds per square inch) as a double (FLOAT64). This is for jets and turboprops.		
227C	8	Turbine Engine 3 reverser fraction, a double (FLOAT64), in the range 0.0–1.0, providing the reverse as a proportion of the maximum reverse throttle position.		
2284	8	Turbine Engine 3 vibration		
228C	4	Turbine Engine 3 Ignition Switch	Ok-SimC	Ok-SimE
2300	8	Turbine Engine 4 N1 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop		

		aircraft.		
2308	8	Turbine Engine 4 N2 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2310	8	Turbine Engine 4 corrected N1 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2318	8	Turbine Engine 4 corrected N2 value (%) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2320	8	Turbine Engine 4 corrected fuel flow (pounds per hour) as a double (FLOAT64). This is for jets and turboprops—it has no meaning on reciprocating prop aircraft.		
2328	8	Turbine Engine 4 max torque fraction (range 0.0–1.0) as a double (FLOAT64).		
2330	8	Turbine Engine 4 EPR as a double (FLOAT64). This is for jets and turboprops.		
2338	8	Turbine Engine 4 ITT (interstage turbine temperature) in degrees Rankine, as a double (FLOAT64). This is for jets and turboprops.		
2348	4	Turbine Engine 4 Afterburner switch (1 = on, 0 = off)		
234C	8	Turbine Engine 4 jet thrust, in pounds, as a double (FLOAT64). This is the jet thrust. See 2710 for propeller thrust (turboprops have both).		
2354	4	Turbine Engine 4 tank selector: 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main, 20=Right Main		
2358	4	Turbine Engine 4 tanks used, a bit mask: 0 Center 1 1 Center 2 2 Center 3 3 Left Main 4 Left Aux 5 Left Tip 6 Right Main 7 Right Aux 8 Right Tip 9 External 1 10 External 2		
235C	4	Turbine Engine 4, number of fuel tanks available		
2360	8	Turbine Engine 4 fuel flow (pounds per hour) as a double (FLOAT64). This is for jets and turboprops.		
2368	4	Turbine Engine 4 fuel available flag		
236C	8	Turbine Engine 4 bleed air pressure (pounds per square inch) as a double (FLOAT64). This is for jets and turboprops.		
237C	8	Turbine Engine 4 reverser fraction, a double (FLOAT64), in the range 0.0–1.0, providing the reverse as a proportion of the maximum reverse throttle position.		
2384	8	Turbine Engine 4 vibration		
238C	4	Turbine Engine 4 Ignition Switch	**Ok-SimC**	**Ok-SimE**
2400	8	Propeller 1 RPM as a double (FLOAT64). This value is for props and turboprops and is negative for counter-rotating propellers. *(On turboprops this will give the shaft RPM, since there is currently no Gear Reduction Ratio available to fix values on such aircraft. I will fix this when I can)*	**?-SimC**	**?-SimC**
2408	8	Propeller 1 RPM as a fraction of the maximum RPM. (double)	**?-SimC**	**No**
2410	8	Propeller 1 thrust in pounds, as a double (FLOAT64). This is for props and turboprops.	**?-SimC**	**No**
2418	8	Propeller 1 Beta blade angle in radians, as a double (FLOAT64).	**?-SimC**	**No**

		This is for props and turboprops.		
2420	4	Propeller 1 feathering inhibit	?-SimC	No
2424	4	Propeller 1 feathered flag	?-SimC	No
2428	8	Propeller 1 sync delta lever	?-SimC	No
2430	4	Propeller 1 autofeather armed flag	?-SimC	No
2434	4	Propeller 1 feather switch	?-SimC	?-SimE
2438	4	Propeller 1 panel auto-feather switch *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*	?-SimC	?-SimE
243C	4	Propeller 1 sync active *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*	?-SimC	?-SimE
2440	4	Propeller 1 de-ice switch *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*	?-SimC	?-SimE
2500	8	Propeller 2 RPM as a double (FLOAT64). This value is for props and turboprops and is negative for counter-rotating propellers. *(On turboprops this will give the shaft RPM, since there is currently no Gear Reduction Ratio available to fix values on such aircraft. I will fix this when I can)*		
2508	8	Propeller 2 RPM as a fraction of the maximum RPM. (double)		
2510	8	Propeller 2 thrust in pounds, as a double (FLOAT64). This is for props and turboprops.		
2518	8	Propeller 2 Beta blade angle in radians, as a double (FLOAT64). This is for props and turboprops.		
2520	4	Propeller 2 feathering inhibit		
2524	4	Propeller 2 feathered flag		
2528	8	Propeller 2 sync delta lever		
2530	4	Propeller 2 autofeather armed flag		
2534	4	Propeller 2 feather switch		
2538	4	Propeller 2 panel auto-feather switch *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*		
253C	4	Propeller 2 sync active *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*		
2540	4	Propeller 2 de-ice switch *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*		
2600	8	Propeller 3 RPM as a double (FLOAT64). This value is for props and turboprops and is negative for counter-rotating propellers. *(On turboprops this will give the shaft RPM, since there is currently no Gear Reduction Ratio available to fix values on such aircraft. I will fix this when I can)*		
2608	8	Propeller 3 RPM as a fraction of the maximum RPM. (double)		
2610	8	Propeller 3 thrust in pounds, as a double (FLOAT64). This is for props and turboprops.		
2618	8	Propeller 3 Beta blade angle in radians, as a double (FLOAT64). This is for props and turboprops.		
2620	4	Propeller 3 feathering inhibit		
2624	4	Propeller 3 feathered flag		
2628	8	Propeller 3 sync delta lever		
2630	4	Propeller 3 autofeather armed flag		
2634	4	Propeller 3 feather switch		
2638	4	Propeller 3 panel auto-feather switch *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*		
263C	4	Propeller 3 sync active *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*		
2640	4	Propeller 3 de-ice switch *(There appears to be only one control, not one for each prop, so*		

		changing any of these 4 changes all 4)		
2700	8	Propeller 4 RPM as a double (FLOAT64). This value is for props and turboprops and is negative for counter-rotating propellers. *(On turboprops this will give the shaft RPM, since there is currently no Gear Reduction Ratio available to fix values on such aircraft. I will fix this when I can)*		
2708	8	Propeller 4 RPM as a fraction of the maximum RPM. (double)		
2710	8	Propeller 4 thrust in pounds, as a double (FLOAT64). This is for props and turboprops.		
2718	8	Propeller 4 Beta blade angle in radians, as a double (FLOAT64). This is for props and turboprops.		
2720	4	Propeller 4 feathering inhibit		
2724	4	Propeller 4 feathered flag		
2728	8	Propeller 4 sync delta lever		
2730	4	Propeller 4 autofeather armed flag		
2734	4	Propeller 4 feather switch		
2738	4	Propeller 4 panel auto-feather switch *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*		
273C	4	Propeller 4 sync active *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*		
2740	4	Propeller 4 de-ice switch *(There appears to be only one control, not one for each prop, so changing any of these 4 changes all 4)*		
281C	4	Master battery switch (1=On, 0=Off)	**Ok-SimC**	**Ok-SimC**
2824	8	Total load amps	**?-SimC**	**?-SimC**
282C	8	Battery load	**?-SimC**	**?-SimC**
2834	8	Battery voltage	**Ok-SimC**	**Ok-SimC**
2840	8	Main bus voltage	**Ok-SimC**	**?-SimC**
2848	8	Main bus amps	**?-SimC**	**?-SimC**
2850	8	Avionics bus voltage	**?-SimC**	**?-SimC**
2858	8	Avionics bus amps	**?-SimC**	**?-SimC**
2860	8	Hot battery bus voltage	**?-SimC**	**?-SimC**
2868	8	Hot battery bus amps	**?-SimC**	**?-SimC**
2870	8	Battery bus voltage	**?-SimC**	**?-SimC**
2878	8	Battery bus amps	**?-SimC**	**?-SimC**
2880	8	Generator alternator 1 bus voltage	**Ok-SimC**	**?-SimC**
2888	8	Generator alternator 1 bus amps	**?-SimC**	**?-SimC**
2890	8	Generator alternator 2 bus voltage	**Ok-SimC**	**?-SimC**
2898	8	Generator alternator 2 bus amps	**?-SimC**	**?-SimC**
28A0	8	Generator alternator 3 bus voltage	**Ok-SimC**	**?-SimC**
28A8	8	Generator alternator 3 bus amps	**?-SimC**	**?-SimC**
28B0	8	Generator alternator 4 bus voltage	**Ok-SimC**	**?-SimC**
28B8	8	Generator alternator 4 bus amps	**?-SimC**	**?-SimC**
28C0	8	Ambient air density, in slugs per cubic foot, double floating point.	**Ok-SimC**	**No**
28C8	8	Ambient air pressure, in lbs per square foot, double floating point.	**Ok-SimC**	**No**
28D0	8	Static air temperature, in degrees Fahrenheit, double floating point.	**Ok-SimC**	**No**
28D8	8	Static air temperature, in degrees Rankine, double floating point.	**Ok-SimC**	**No**
28E0	8	"Theta", or standard temperature ratio (i.e ambient air temperature divided by the ISO standard sea level air temperature), double floating point. *(In FSX this is currently calculated by FSUIPC)*	**Ok-Intl**	**No**
28E8	8	"Delta", or standard pressure ratio (ambient pressure divided by the ISO standard sea level pressure), double floating point. *(In FSX this is currently calculated by FSUIPC)*	**Ok-Intl**	**No**
28F0	8	"Sigma", or standard density ratio (ambient density divided by the ISO standard sea level density), double floating point. *(In FSX this is currently calculated by FSUIPC)*	**Ok-Intl**	**No**
28F8	8	Obsolete AI traffic control, only retained for compatibility. *Do*	**--**	**--**

		not use! See 2900 following:		
2900	12	A.I. traffic control. Write all 3 32-bit values (i.e. 12 bytes) together to send an FS control to a specific AI aircraft. The values needed are: Bytes 0–3: Aircraft Id (from the TCAS table) Bytes 4–7: The FS Control (see published lists) Bytes 8–11: A parameter for the control, if needed Note that most of the many hundreds of FS controls will have no noticeable affect on the AI aircraft. Experimentation is needed. If folks find out what does what, please let me know and I'll try to publish a collated guide as an appendix later. Note that you can write these values in separate FSUIPC Writes, but if you do the ID must be last, as it is only when this is written that the control is activated. The special control value 0xFFFF (65535) is supported as a request to delete the specified aircraft. ***(This currently uses a hack into the FS code)***.	**N/A**	**Ok-SimE** *(part hacked)*
290C	4	Number of Hot Joystick Button slots available for Application Programs to use. Currently this is fixed at 56, representing the 56 DWORDs available in the following offsets:	**Ok-Intl**	**N/A**
2910	224	56 DWORDs containing zero (when free for use), or a Hot Joystick Button specification. See also 32FF below. This "HOT BUTTON" facility allows programs to detect selected joystick button presses. This facility is very similar to the Hot Key system described for offset 3210. Up to 56 such hot buttons can be specified, but this number is shared by all running applications. The facility operates using these offsets: 56 32-bit values ("DWORDs") from 0ffset 0x2910 onwards (i.e. 0x2910, 0x2914 ...) are 'slots' for Applications to specify Hot Keys. These will be zero initially, and zero if free. The application must search through to find an empty slot, then set this into it: Byte 0 (bits 0-7): Joystick number (0-15) + 128. In other words 128 for Joystick 0, 129 for joystick 1, etc. Joysticks are numbered from 0. (Note that Windows 'Game Controllers' numbers from 1). Byte 1 (bits 8-15): Button number (0-39) Again buttons are numbered here from 0. Buttons 0–31 are the normal buttons, numbers 32–39 are a representation of the 8 "Points of View" at 45 degree angles supported by some joystick drivers for the POV Hats. Byte 2 (bits 16-23): Flags from application. This byte indicates which change is to be notified: = 0 for Off to On = 1 for On to Off = 2 for both Off to On and On to Off = 3 for Off to On but repeating about 6 times per second whilst it is on. Byte 3 (bits 24-31): Flags from FSUIPC. Bit 0 (value 1) is set when the specified Hot Button change occurs. Needs to be cleared by Application when seen so it can detect another. (No queuing). Bit 1 (value 2) is set when bit 0 is set only if the button is still pressed. This can be used to differentiate the two events when Byte 2 is given as "2" for both off–on and on–off events. **Note**: If the same Hot button is listed more than once (for instance by several applications), every copy for the same Hot button will get the flag set.	**Ok-Intl**	**Ok-Intl**

		Use: Having found an empty slot, write the above value into it, then monitor the highest byte of that same slot for Non-Zero. That's the button event. Clear that byte to detect it again. If you register several HotKey Buttons it will be more efficient to only scan the slots themselves when a hot button actually occurs. To detect this, just monitor the one byte at offset 32FF. (This can be paired with 32FE to scan for buttons and keys). When it changes, read and check the flags in your slots. (The count at 32FF may change without any of your buttons occurring, of course, if other applications are trapping other hot buttons). When finished, and certainly before exit, be sure to clear the whole DWORD to zero so other applications can use it. If you only want to use joystick buttons for a certain part of the operation of your program, only set the entries there and clear them when done. Note that if several applications want the same button, they will all get it. Of course, your application can check through the whole list to make sure there are no clashes/duplicates and warn the user if so. You might have to do that at intervals in case a clashing application is loaded after yours. This system will work through WideFS with no problems too.		
29F0	4	This DWORD provides a facility to set, clear or toggle any of the virtual buttons at offset 3340 without needing to read anything first. To do this, write to offset 29F0 a 32-bit value (4 bytes) made up as follows: Byte 0: Button Number on Joystick (0 - 31) Byte 1: Virtual Joystick Number (64 - 72) Byte 2: Action: 0 = Toggle 1 = Set (Press/On) 2 = Clear (Release/Off). Byte 3: 0 (Reserved)		
2A00	8	Elevon 1 deflection	?-SimC	No-SimC
2A08	8	Elevon 2 deflection	?-SimC	No-SimC
2A10	8	Elevon 3 deflection	?-SimC	No-SimC
2A18	8	Elevon 4 deflection	?-SimC	No-SimC
2A20	8	Elevon 5 deflection	?-SimC	No-SimC
2A28	8	Elevon 6 deflection	?-SimC	No-SimC
2A30	8	Elevon 7 deflection	?-SimC	No-SimC
2A38	8	Elevon 8 deflection	?-SimC	No-SimC
2A48	8	Folding wing (for reading), left percent, as double float.	?-SimC	?-SimC
2A50	8	Folding wing (for reading), right percent, as double float.	?-SimC	?-SimC
2A70	8	Canopy open, as double float.	?-SimC	?-SimC
2A78	8	Water left rudder extended (double float)	?-SimC	No
2A80	8	Water right rudder extended (double float)	?-SimC	No
2A88	4	Water rudder handle position (100% = 16384)	Ok-SimC	Ok-SimE
2A90	4	Tail wheel lock (BOOLEAN, 1= locked, 0= unlocked)	Ok-SimC	Ok-SimE
2AAC	4	NAV1 course deviation needle (CDI), 32-bit float value, –127.0 left to +127.0 right	Ok-SimC	No
2AB0	4	NAV1 glideslope needle (GSI), 32-bit float value, –127.0 up to +127.0 down	Ok-SimC	No
2AB4	4	NAV2 course deviation needle (CDI), 32-bit float value, –127.0 left to +127.0 right	Ok-SimC	No
2AB8	4	NAV2 glideslope needle (GSI), 32-bit float value, –127.0 up to +127.0 down	Ok-SimC	No
2B00	8	Gyro compass heading (magnetic), including any drift. 64-bit floating point.	Ok-SimC	No
2B08	8	Hydraulics1 pressure psf	?-SimC	No
2B1C	8	Hydraulics1 reservoir pct	?-SimC	No
2C08	8	Hydraulics2 pressure psf	?-SimC	No

2C1C	8	Hydraulics2 reservoir pct	**?-SimC**	**No**
2D08	8	Hydraulics3 pressure psf	**?-SimC**	**No**
2D1C	8	Hydraulics3 reservoir pct	**?-SimC**	**No**
2DC6	2	Helicopter "beep" (whatever that is—something to do with the governor). This value is also controlled by the *Increase Heli Beep* and *Decrease Heli Beep* FS controls. It appears to change from 0 to 16313 then more slowly to 16368.	**No info**	**No info**
2DC8	8	The wind at the aircraft in the lateral (X) axis—relative to the aircraft orientation, in feet per second, as a 64-bit double. (+ve Right Crosswind, -ve Left)	**Ok-SimC**	**No-SimC+**
2DD0	8	The wind at the aircraft in the vertical (Y) axis—relative to the aircraft orientation, in feet per second, as a 64-bit double. (+ve pushing on aircraft's under surfaces, -ve over surfaces)	**Ok-SimC**	**No-SimC+**
2DD8	8	The wind at the aircraft in the longitudinal (Z) axis—relative to the aircraft orientation, in feet per second, as a 64-bit double. (+ve Headwind, -ve Tailwind)	**Ok-SimC**	**No-SimC+**
2DE0	8	Wind direction at the aircraft, in degrees True, as a 64-bit double floating point – for writing, not reading. See 3490 for reading. This can be written to directly affect the wind direction at the aircraft.	**Ok-SimC**	**Ok-Hack**
2DE8	8	Wind speed at the aircraft, in knots, as a 64-bit double floating point – for writing, not reading. See 3488 for reading. This can be written to directly affect the wind direction at the aircraft.	**Ok-SimC**	**Ok-Hack**
2DF0	8	Visibility at the aircraft, in metres, as a 64-bit double floating point – for reading.	**Ok-SimC**	**No-SimC+**
2DF8	4	Ambient in cloud … BOOLEAN *new value found for FSX. Not sure what it is yet – it should be TRUE when the user aircraft is in cloud, but it doesn't appear to work like that.*	**?-SimC** *(see note)*	**No**
2E00	4	Ambient precip state … *new value found for FSX. Not sure what it is yet.*	**?-SimC** *(see note)*	**No**
2E04	4	Autopilot max bank … *degrees. Works for the default FSX 737. (Writing here uses the AP MAX BANK INC and DEC controls to try to approximate to the angle written.)*	**OK-SimC**	**Partly (SimE)** *(see note)*
2E08	8	Hydraulics4 pressure psf	**?-SimC**	**No**
2E1C	8	Hydraulics4 reservoir pct	**?-SimC**	**No**
2E78	8	CG percent *laterally*, as a double (FLOAT64). This is the position of the actual CoG as a fraction (%/100) of MAC (Mean Aerodynamic Chord).	**Ok-SimC**	**No**
2E80	4	Master avionics switch (0=Off, 1=On)	**Ok-SimC**	**Ok-SimE**
2E88	4	Panel auto-feather arm switch (0=Off, 1=On) *(This is for #1 propeller, not all?)*	**?-SimC** *(see note)*	**No** *(see 2438)*
2E90	4	Standby vacuum circuit on	**?-SimC**	**No**
2E98	8	Elevator deflection, in radians, as a double (FLOAT64). Up positive, down negative.	**Ok-SimC**	**No**
2EA0	8	Elevator trim deflection, in radians, as a double (FLOAT64). Up positive, down negative.	**Ok-SimC**	**?-SimC**
2EA8	8	Aileron deflection, in radians, as a double (FLOAT64). Right turn positive, left turn negative. *(This is the average of left and right)*	**Ok-SimC**	**No**
2EB0	8	Aileron trim deflection, in radians, as a double (FLOAT64). Right turn positive, left turn negative. *(for write, converted to proportion assuming max .2 and written via 0C02)*	**Ok-SimC**	**?-SimC** *(see note)*
2EB8	8	Rudder deflection, in radians, as a double (FLOAT64).	**Ok-SimC**	**No**
2EC0	8	Rudder trim deflection, in radians, as a double (FLOAT64). *(for write, converted to proportion assuming max .2 and written via 0C04)*	**Ok-SimC**	**?-SimC** *(see note)*
2EC8	4	Prop sync active (1=Active, 0=Inactive)	**Ok-SimC**	**Ok-SimE**
2ED0	8	Incidence "alpha", in radians, as a double (FLOAT64). This is the aircraft *body* angle of attack (AoA) not the *wing* AoA.	**Ok-SimC**	**No**

		Note that it has been found that that FS disregards wing incidence and twist effects (in the Aircraft.CFG file), so this value is actually the wing AofA as well.		
2ED8	8	Incidence "beta", in radians, as a double (FLOAT64). This is the side slip angle.	**Ok-SimC**	**No**
2EE0	4	Flight Director Active, control and indicator. 1=active, 0=inactive.	**Ok-SimC**	**Ok-SimE**
2EE8	8	Flight director pitch value, in degrees. Double floating point format, only when FD is active.	**Ok-SimC**	**No**
2EF0	8	Flight director bank value, in degrees. Double floating point format, right is negative, left positive.	**Ok-SimC**	**No**
2EF8	8	CG percent, as a double (FLOAT64). This is the position of the actual CoG as a fraction (%/100) of MAC (Mean Aerodynamic Chord).	**Ok-SimC**	**No**
2F00	8	CG aft limit *(%/100)*	**?-SimC**	**No**
2F08	8	CG fwd limit *(%/100)*	**?-SimC**	**No**
2F10	8	CG max mach	**Ok-SimC**	**No**
2F18	8	CG min mach	**?-SimC**	**No**
2F20	8	Concorde visor nose handle (%)	**?-SimC**	**?-SimC**
2F28	8	Concorde visor pos pct (%)	**?-SimC**	**No**
2F30	8	Concorde nose angle (Rads)	**?-SimC**	**No**
2F38	8	Gear pos tail	**?-SimC**	**?-SimC**
2F40	8	Autopilot max speed (hold?)	**?-SimC**	**?-SimC**
2F48	8	Autopilot cruise speed (hold?)	**?-SimC**	**?-SimC**
2F50	8	Barber pole mach	**?-SimC**	**No**
2F58	4	Selected fuel transfer mode: 0=Off, 1=Auto, 2=Fwd, 3=Aft	**Ok-SimC**	**Ok-SimE**
2F60	8	Hydraulic system integrity (%)	**?-SimC**	**?-SimC**
2F68	4	Attitude cage button	**?-SimC**	**?-SimC**
2F70	8	Attitude indicator pitch value, in degrees. Double floating point format.	**Ok-SimC**	**?-SimC**
2F78	8	Attitude indicator bank value, in degrees. Double floating point format.	**Ok-SimC**	**?-SimC**
2F80	1	Panel autobrake switch Read to check setting, write to change it. 0=RTO, 1=Off, 2=brake1, 3=brake2, 4=brake3, 5=max	**Ok-SimC**	**Ok-SimE**
2F88	8	HSI CDI needle position, -127.0 to +127.0 double floating point. Full range represents –10 to +10 degrees for a VOR, -2.5 to +2.5 degrees fr a LOC	**Ok-SimC**	**No**
2F90	8	HSI GSI needle position, -119.0 to +119.0 double floating point. Full range represents –0.7 to +0.7 degrees	**Ok-SimC**	**No**
2F98	8	HSI speed, as a double floating point. I think it should be in metres/sec, but it doesn't look right – feedback please!	**?-SimC**	**No**
2FA0	8	HSI distance, as a double floating point. In metres.	**Ok-SimC**	**No**
2FA8	2	HSI bearing. In degrees? Doesn't seem to work. Feedback?	**?-SimC**	**No**
2FAA	1	HSI CDI valid flag. Doesn't appear to work?	**?-SimC**	**No**
2FAB	1	HSI GSI valid flag.	**Ok-SimC**	**No**
2FAC	1	HSI bearing valid flag. (Not seen this set yet – see 2FA8)	**?-SimC**	**No**
2FAD	1	HSI To/From flag: 0=off, 1=To, 2=From	**Ok-SimC**	**No**
2FAE	1	HSI has localiser flag	**Ok-SimC**	**No**
2FB0	6	HSI ident string	**Ok-SimC**	**No**
2FE0	32	**FS "Add-Ons" menu access for Applications:** This facility allows an application to add an entry to the Add-Ons menu. The Application finds a free Hot Key slot, then sets it up to receive notification on menu access, and writes the text needed for the menu item to another location. When the menu item is selected, the flag in the hot key slot is set just as when a hot key is used. This way of accessing the menu has the advantage that it will also work when the application is running on another PC, via WideFS. Of course, any response to that menu selection will occur on whichever PC the application is running.	**N/A**	**Ok-SimC** *(see notes)*

To avoid having menu items relating to applications that have crashed or terminated without tidying up correctly, each menu item added is subjected to a time-out. Applications have to refresh a count in the Hot Key slot at regular intervals (10 seconds or less) otherwise the menu item is deleted and the Hot Key slot freed. The time-out is suspended when FS is paused, and there is an option to have FS pause automatically when the menu entry is selected.

Note that FS subjects the nuber of entries in the Add-Ons menu to a maximum of 16. FSUIPC is already using one for itself. If the maximum is already reached your entry will simply *not* appear. There is no error indication of this provided back to the Application, though a SimConnect exception may appear in the FSUIPC Log file if exception logging is enabled.

This is the way this facility is used:

1. Find a free Hot Key slot (i.e. search the 56 DWORDs at offset 0x3210 for a zero value). Say slot ***I*** is the one found.
2. Write 0x0000FFFF to the slot (i.e to the DWORD at offset 0x3210 + 4****I***). If you want FS to pause when the menu item is selected, write 0x0002FFFF instead. The 02 part is the flag indicating that a pause is required.
3. Write the text for the menu entry required to offset 0x2FE0, with the first byte set to the slot number (***I***). For example, for an entry "UIPC Hello" (H being the shortcut) you would set the string to be written to 0x2FE0 as follows:

   static chMenuEntry[] = "?UIPC &Hello";
   chMenuEntry[0] = ***I***;

4. The '&' in the string tells Windows which character to underscore, and this denotes the shortcut key, but this is optional.
5. The string is limited to 31 characters, including the slot number at the beginning, plus a zero terminator. In other words the offset range is 0x2FE0–0x2FFF inclusive. This area is "write only". Don't expect to be able to read back what you write here.
6. The write to 0x2FE0 triggers FSUIPC into asking FS to add the menu entry to the Add-Ons main menu item, but this is dependent upon the slot it references being set with 0xFF in its first (least significant) byte. From the moment the *slot* is set with 0xFF there it is changed every 55 mSecs or so, unless FS is paused or in a dialogue. The change is a decrement of the next byte in the slot—the other one you also set to 0xFF. When this reaches zero, the menu entry is removed and the slot is cleared. This gives a maximum timeout of 255 x 55mSecs, or about 14 seconds. You can make it less, of course, by initialising that byte to a lower value than 0xFF (255), but I'd recommend sticking to the maximum.

   This means that if you want the menu entry to stay available you must write 0xFF (or whatever) to that byte (i.e. the slot offset + 1) at regular intervals, say every 10 seconds. The 4 second leeway allows some safety, but you may want more—very little FS overhead is caused by writing that one byte every 1 second if you need to, but this is really over the top. More overhead is caused by writes when running on another PC using WideFS, so I would suggest 5 seconds as a minimum.

		7. When the user selects your menu entry, FSUIPC will set the 2^0 (0x01) bit in the top byte (offset+3) in your slot. Just as with Hot Keys, you need to be looking for this at regular intervals, perhaps every 200 milliseconds or so. Frequent reads pose little overhead for WideFS use, but very frequent ones should really be avoided when you are running on the FS PC. 8. After processing the user request, whatever it is, don't forget to clear the indicator so you can detect the next one—writing zero to the byte at the offset+3 is all that is needed. 9. Finally, if you opted for FS to pause when the menu item is selected you need to unpause FS so that it can continue. Write zero to the 16-bit value at offset 0x262. When you no longer need the menu entry, or just before terminating your program, you should write zero to the DWORD Hot Key slot. This will make FSUIPC remove the menu entry immediately. If your program does not tidy up the entry will be removed on the timeout. **Adding submenu entries to your menu entry:** [*Not available in FSUIPC3*] Having already setup the main menu, as above, write this, in one write, to 0x2FE0: Byte 0: 0x80 + slot number of main entry, as before (i.e. 0 for 3210, 1 for 3214 etc. Remember the max is 55, there being 56 slots). Byte 1: Response value (any non-zero value 1 - 255). This is merely a value for you to test so you know which submenu was selected. Bytes 2-31 The zero-terminated string for the submenu entry. There's a limit of 16 submenus per menu entry (imposed by SimConnect), and there are no further sub-levels. When the user selects the submenu FSUIPC will fill in byte 3 of the slot with the "Response value" provided. Naturally you don't get notified when the main menu entry is selected when there are submenus. You can remove a submenu by doing the same as above but with a null string for the submenu entry (i.e. a single zero byte).		
3000	6	VOR1 IDENTITY (string supplied: 6 bytes including zero terminator)	**Ok-SimC**	**N/A**
3006	25	VOR1 name (string supplied: 25 bytes including zero terminator)	**Ok-SimC**	**N/A**
301F	6	VOR2 IDENTITY (string supplied: 6 bytes including zero terminator)	**Ok-SimC**	**N/A**
3025	25	VOR2 name (string supplied: 25 bytes needed including zero terminator)	**Ok-SimC**	**N/A**
303E	6	ADF1 IDENTITY (string supplied: 6 bytes including zero terminator)	**Ok-SimC**	**N/A**
3044	25	ADF1 name (string supplied: 25 bytes including zero terminator)	**Ok-SimC**	**N/A**
305D	1	Count of "Toggle aircraft name display" controls seen, 0-255, wrapping back to 0 after 255.	**Ok-Intl**	**No**
3060	8	X (lateral, or left/right) acceleration in ft/sec/sec relative to the body axes in double floating point format.	**Ok-SimC**	**?-SimC**
3068	8	Y (vertical, or up/down) acceleration in ft/sec/sec relative to the body axes in double floating point format.	**Ok-SimC**	**?-SimC**
3070	8	Z (longitudinal, or forward/backward) acceleration in ft/sec/sec	**Ok-SimC**	**Ok-SimC**

		relative to the body axes in double floating point format.		
3078	8	Pitch acceleration in radians/sec/sec relative to the body axes in double floating point format.	Ok-SimC	No
3080	8	Roll acceleration in radians/sec/sec relative to the body in double floating point format.	Ok-SimC	No
3088	8	Yaw acceleration in radians/sec/sec relative to the body in double floating point format.	Ok-SimC	No
3090	8	Z (longitudinal, or forward/backward) GS-velocity in ft/sec relative to the body axes in double floating point format.	?-SimC	?-SimC
3098	8	X (lateral, or left/right) GS-velocity in ft/sec relative to the body axes in double floating point format.	?-SimC	?-SimC
30A0	8	Y (vertical, or up/down) GS-velocity in ft/sec relative to the body axes in double floating point format.	?-SimC	?-SimC
30A8	8	Pitch velocity in rads/sec relative to the body axes in double floating point format.	Ok-SimC	?-SimC
30B0	8	Roll velocity in rads/sec relative to the body axes in double floating point format.	Ok-SimC	?-SimC
30B8	8	Yaw velocity in rads/sec relative to the body axes in double floating point format.	Ok-SimC	?-SimC
30C0	8	Current loaded weight in lbs in double floating point format.	Ok-SimC	No
30C8	8	Plane's current mass, in slugs (1 slug = 1lb*G = 32.174049 lbs) mass. This is in double floating point format (FLOAT64). The current mass = current loaded weight (as in 30C0) * G, where G is 32.174049.	?-SimC	No
30D0	8	Vertical acceleration in G's. This is in double floating point format (FLOAT64).	No-SimC+	No
30D8	8	Dynamic pressure (lbs/sqft). [FS2k/CFS2/FS2002 only]	Ok-SimC	No
30E0	2	Trailing edge left inboard flap extension as a percentage of its maximum, with 16383 = 100%	Ok-SimC	?-SimC
30E2	2	Trailing edge left outboard flap extension as a percentage of its maximum, with 16383 = 100%	Ok-SimC	?-SimC
30E4	2	Trailing edge right inboard flap extension as a percentage of its maximum, with 16383 = 100%	Ok-SimC	?-SimC
30E6	2	Trailing edge right outboard flap extension as a percentage of its maximum, with 16383 = 100%	Ok-SimC	?-SimC
30E8	2	Leading edge left inboard flap extension as a percentage of its maximum, with 16383 = 100%	Ok-SimC	?-SimC
30EA	2	Leading edge left outboard flap extension as a percentage of its maximum, with 16383 = 100%	Ok-SimC	?-SimC
30EC	2	Leading edge right inboard flap extension as a percentage of its maximum, with 16383 = 100%	Ok-SimC	?-SimC
30EE	2	Leading edge right outboard flap extension as a percentage of its maximum, with 16383 = 100%	Ok-SimC	?-SimC
30F0	2	Trailing edge left inboard flap extension in degrees * 256	Ok-SimC	No
30F2	2	Trailing edge left outboard flap extension in degrees * 256	Ok-SimC	No
30F4	2	Trailing edge right inboard flap extension in degrees * 256	Ok-SimC	No
30F6	2	Trailing edge right outboard flap extension in degrees * 256	Ok-SimC	No
30F8	2	Leading edge left inboard flap extension in degrees * 256	Ok-SimC	No
30FA	2	Leading edge left outboard flap extension in degrees * 256	Ok-SimC	No
30FC	2	Leading edge right inboard flap extension in degrees * 256	Ok-SimC	No
30FE	2	Leading edge right outboard flap extension in degrees * 256	Ok-SimC	No
3100	1	Engine primer (just write a non-zero byte to operate the primer. This is a one-shot and reading it is meaningless)	?-SimC	?-SimC
3101	1	Alternator (1 = on, 0 = off), read for state, write to control *(This is for Alternator 1)*	?-SimC	?-SimE
3102	1	Battery (1 = on, 0 = off), read for state, write to control	?-SimC	?-SimC
3103	1	Avionics (1 = on, 0 = off), read for state, write to control	?-SimC	?-SimE
3104	1	Fuel pump (1 = on, 0 = off), read for state, write to control. For separate switches for separate fuel pumps see offset 3125. *(This is for Pump 1)*	Ok-SimC	Ok-SimE
3105	1	VOR1 morse ID sound (1 = on, 0 = off), read for state, write to	?-SimC	?-SimC

		control (see also 3122)		
3106	1	VOR2 morse ID sound (1 = on, 0 = off), read for state, write to control (see also 3122)	?-SimC	?-SimC
3107	1	ADF1 morse ID sound (1 = on, 0 = off), read for state, write to control (see also 3122)	?-SimC	?-SimC
3108	1	Write 1 here to disable FSUIPC's "AutoTune ADF1" facility, if this has been enabled by the user in FSUIPC.INI.	N/A	?-Intl
3109	1	This is a bit-oriented control flag byte. These bits are allocated so far: 2^0 (1) = 1 to disable AxisCalibration even if enabled in FSUIPC.INI. Note that this "AxisCalibration" is the one specifically concerned with direct offset values—see the Advanced User's guide for the description of the INI parameter for more details. 2^1 (2) = 1 to allow the older (FS98-compatible) axis controls to remain connected even when the main axis controls are disconnected via bits in 310A and 310B below. These are AILERON_SET, ELEVATOR_SET, ELEVATOR_TRIM_SET, RUDDER_SET, THROTTLE_SET and the four THROTTLEn_SET controls. Allowing these through will let autopilot of FBW programs control the relevant values without writing direct to the appropriate offsets, but take care also that the THROTTLEn_SET controls aren't being claibrated in the user's 4-throttle option (page 3 in FSUIPC options). 2^7(128) is reserved for external applications to use as they wish. In order to protect the user from a broken or crashed application, the 2^1 flag is cleared 10 seconds after it has been set, so applications will need to repeat the setting every few seconds.	N/A	?-Intl
310A	1	Controls the joystick connection to the main flight controls. Normally all zero, set the following bits to actually disconnect the specific joystick axes (from least significant bit = 0): 0 Elevator 1 Aileron 2 Rudder 3 Throttles (all). 4 *See below (throttle sync control)* 5 Elevator trim 6 Throttle #1 7 Throttle #2 (see next byte for others) This feature is intended for use in protecting autopilot flight from interference from axis flutter. In order to protect the user from a broken or crashed application, all the flags are cleared 10 seconds after they have been set, so applications will need to repeat the setting every few seconds. If the user option is set to automatically disconnect the trim axis in FS A/P vertical modes, the disconnection of Elevator inputs via bit 0 above also disconnects Trim even if bit 5 is not also set. This allows existing A/P or fly-by-wire applications to work with those user implementations using a trim axis. Additionally, bit 2^4 is available to switch "throttle sync" on. In this mode all throttles are driven from the main throttle or throttle 1 inputs, and other throttle inputs are discarded. (The same option can also be used from an optional Hot Key). See also offset 3109 above, and also offsets 3328–3339, which	N/A	?-Intl

		provide the live axis values, post calibration. These would have been applied to FS if not prevented by the flags above. Applications can use these facilities to provide a responsive "fly-by-wire" control.		
310B	1	Controls the joystick connection to the slewing controls, and the other two separate throttle controls. Normally all zero, set the following bits to actually disconnect the specific axes (from least significant bit = 0): 0 Slew Ahead 1 Slew Side 2 Slew Heading 3 Slew Altitude 4 Slew Bank 5 Slew Pitch 6 Throttle #3 (see previous byte for #1, #2) 7 Throttle #4 In order to protect the user from a broken or crashed application, all the flags are cleared 10 seconds after they have been set, so applications will need to repeat the setting every few seconds. See also offset 3109 above.	**N/A**	**?-Intl**
310C	4	*Reserved*		
3110	8	Operates a facility to send any 'controls' to Flight simulator. This works with ***all*** versions of FS & CFS. Write all 8 bytes for controls which use a value (axes and all _SET controls), but just 4 will do for 'button' types. This is really two 32-bit integers. The first contains the Control number (normally 65536 upwards), as seen in my FS Controls lists. The second integer is used for the parameter, such as the scaled axis value, where this is appropriate. Always write all 8 bytes in one IPC block if a parameter is used, as FSUIPC will fire the control when you write to 3110. Since version 3.40, FSUIPC-added controls (other than the offset ones) can be used via these offsets too. See the Advanced User's Guide for a current list.	**N/A**	**Ok-Intl**
3118	2	COM2 frequency, 4 digits in BCD format. A frequency of 123.45 is represented by 0x2345. The leading 1 is assumed.	**Ok-SimC**	**Ok-SimE**
311A	2	COM1 standby frequency, 4 digits in BCD format. A frequency of 123.45 is represented by 0x2345. The leading 1 is assumed.	**Ok-SimC**	**Ok-SimE**
311C	2	COM2 standby frequency, 4 digits in BCD format. A frequency of 123.45 is represented by 0x2345. The leading 1 is assumed.	**Ok-SimC**	**Ok-SimE**
311E	2	NAV1 standby frequency, 4 digits in BCD format. A frequency of 113.45 is represented by 0x1345. The leading 1 is assumed.	**Ok-SimC**	**Ok-SimE**
3120	2	NAV2 standby frequency, 4 digits in BCD format. A frequency of 113.45 is represented by 0x1345. The leading 1 is assumed.	**Ok-SimC**	**Ok-SimE**
3122	1	Radio audio switches. Read/write bit settings as follows: 2^7 COM1 transmit 2^6 COM2 transmit 2^5 COM receive both 2^4 NAV1 sound 2^3 NAV2 sound 2^2 Marker sound 2^1 DME sound 2^0 ADF1 sound For ADF2 sound, on FS2004, see offset 02FB.	**?-SimC**	**?-SimE**
3123	1	Radio Use/Standby swap toggles, Write bits to operate toggles. Don't bother to read it, there's no meaning to anything read. 2^3 COM1 swap 2^2 COM2 swap 2^1 NAV1 swap 2^0 NAV2 swap	**N/A**	**Ok-SimE**

3124	1	The specific version of FSX or P3D being used. The values are: **FSX**: 1 to 4 for the RTM, SP1, SP2 and Acc versions, respectively **FSX-SE**: 101 to 109 (etc) for builds 62607 to 62615 (etc) **P3Dv1**: 10 to 14 for versions 1.0 to 1.4 (but versions before 1.4 not supported). **P3Dv2**: 20 to 25 for versions 2.0 to 2.5 **P3Dv3:** 30 to 32 (etc) for versions 3.0 to 3.2 (etc)	**Intl**	**No**
3125	1	Separate switches for up to 4 Fuel Pumps (one for each engine). Bit 2^0=Pump1, 2^1=Pump2, 2^2=Pump3, 2^4=Pump4. (*see also offset 3104*)	**Ok-SimC**	**Ok-SimE**
3126	1	Set view direction (write only, current view not detected). 0 = FORWARD 1–7 = FORWARD RIGHT and 45 degree views, clockwise 8 = DOWN 9 = UP 10-17 = FORWARD UP then 45 degree UP views, clockwise all other values = RESET	**N/A**	**Ok-SimE**
3127	9	FSUIPC weather option control area: not planned for FSX		
3130	12	ATC flight number string for currently loaded user aircraft, as declared in the AIRCRAFT.CFG file. This is limited to a maximum of 12 characters, including a zero terminator. *SimConnect allows this SimVar to be written, but this may not change the Flight Number being used by ATC unless a flight plan has been loaded too (see offset 0130).*	**Ok-SimC**	**Ok-SimC** *(but see note)*
313C	12	ATC identifier (tail number) string for currently loaded user aircraft, as declared in the AIRCRAFT.CFG file. This is limited to a maximum of 12 characters, including a zero terminator. *(SimConnect seems to allow this SimVar to be written, but whether this does actually change the Tail Number being used I by ATC, I don't yet know)*	**Ok-SimC**	**?-SimC** *(see note)*
3148	24	ATC airline name string for currently loaded user aircraft, as declared in the AIRCRAFT.CFG file. This is limited to a maximum of 24 characters, including a zero terminator. *(SimConnect seems to allow this SimVar to be written, but whether this does actually change the Airline Name being used by ATC, I don't yet know)*	**Ok-SimC**	**?-SimC** *(see note)*
3160	24	ATC aircraft type string for currently loaded user aircraft, as declared in the AIRCRAFT.CFG file. This is limited to a maximum of 24 characters, including a zero terminator.	**Ok-SimC**	**No**
3178	8	Z (longitudinal, or forward/backward) TAS-velocity in ft/sec relative to the body axes. This is in double floating point format (FLOAT64).	**Ok-SimC**	**?-SimC**
3180	8	X (lateral, or left/right) TAS-velocity in ft/sec relative to the body axes in double floating point format (FLOAT64).	**Ok-SimC**	**?-SimC**
3188	8	Y (vertical, or up/down) TAS-velocity in ft/sec relative to the body axes in double floating point format (FLOAT64).	**Ok-SimC**	**?-SimC**
3190	8	Z (longitudinal, or forward/backward) GS-velocity in ft/sec relative to world axes in double floating point format (FLOAT64).	**?-SimC**	**?-SimC**
3198	8	X (lateral, or left/right) GS-velocity in ft/sec relative to world axes in double floating point format (FLOAT64).	**?-SimC**	**?-SimC**
31A0	8	Y (vertical, or up/down) GS-velocity in ft/sec relative to world axes in double floating point format (FLOAT64).	**?-SimC**	**?-SimC**
31A8	8	Pitch velocity in rads/sec relative to world axes in double floating point format (FLOAT64).	**Ok-SimC**	**No**
31B0	8	Roll velocity in rads/sec relative to world axes in double floating point format (FLOAT64).	**Ok-SimC**	**No**

31B8	8	Yaw velocity in rads/sec relative to world axes in double floating point format (FLOAT64).	Ok-SimC	No
31C0	8	X (lateral, or left/right) acceleration in ft/sec/sec relative to the world axes in double floating point format (FLOAT64).	?-SimC	?-SimC
31C8	8	Y (vertical, or up/down) acceleration in ft/sec/sec relative to the world axes in double floating point format (FLOAT64).	?-SimC	?-SimC
31D0	8	Z (longitudinal, or forward/backward) acceleration in ft/sec/sec relative to the world axes in double floating point format (FLOAT64).	?-SimC	?-SimC
31D8	2	Slew mode longitudinal axis (i.e. forward/backward) input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310B)	?-Intl	N/A
31DA	2	Slew mode lateral axis (i.e. left/right) input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310B)	?-Intl	N/A
31DC	2	Slew mode yaw axis (i.e. heading) input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310B)	?-Intl	N/A
31DE	2	Slew mode vertical axis (i.e. altitude) input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310B)	?-Intl	N/A
31E0	2	Slew mode roll axis (i.e. bank) input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310B)	?-Intl	N/A
31E2	2	Slew mode pitch axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310B)	?-Intl	N/A
31E4	4	Radio altitude in metres * 65536	Ok-SimC	No
31E8	4	Surface type as a 32-bit integer. I think this only applies when the aircraft is on the ground. The values probably correspond to the surface encoding in the scenery files, thus: CONCRETE 0 GRASS 1 SOFT, BUMPY GROUND (LANDABLE) WATER 2 GRASS BUMPY 3 VERY BUMPY GRASS & MUD (CRASHABLE) ASPHALT 4 SHORT GRASS 5 LONG GRASS 6 HARD TURF 7 SNOW 8 ICE 9 URBAN 10 FOREST 11 DIRT 12 CORAL 13 GRAVEL 14 OIL TREATED 15 TAR & CHIP STEEL MATS 16 STEEL MESH TEMPORARY RUNWAYS BITUMINUS 17 BRICK 18 MACADAM 19 PLANKS 20 SAND 21 SHALE 22 TARMAC 23 UNKNOWN 254	?-SimC	No
31EC	4	Surface condition as a 32-bit integer, probably as follows: NORMAL 0 WET 1 ICY 2 SNOW 3 SNOW ON A NON-SNOW SURFACE	No-SimC+	No
31F0	4	Pushback status 3=off, 0=pushing back, 1=pushing back, tail to swing to left (port), 2=pushing back, tail to swing to right (starboard)	OK-SimC	N/A
31F4	4	Pushback control. Write 0–3 here to set pushback operation, as described for the status, above.	N/A	OK-SimE
31F8	4	Tug Heading control, for gliders I assume. [*write only*]. The units appear to be the same as the aircraft heading units (see offset 0580).	N/A	?-SimE

31FC	4	Tug Speed control, for gliders I assume. [*write-only*]. Units not confirmed, but possible ft/sec.	**N/A**	**?-SimE**
3200	12	These locations operate the FSUIPC facility to send keystrokes to FS. For this to operate correctly the PC must be using Windows 98, ME, 2000, XP or Vista. The facilities used just do not exist in Windows 95 or NT. 3200 message (WM_KEYDOWN or WM_KEYUP) 3204 wParam for the message 3208 lParam for the message All 12 bytes must be written in one IPC write.	**N/A**	**?-Intl**
320C	4	Number of Hot Key slots available for Application Programs to use. Currently this is fixed at 56, representing the 56 DWORDs available in the following offsets:	**Ok-Intl**	**N/A**
3210	224	56 DWORDs containing zero (when free for use), or a Hot Key specification. See also 32FE below. Note that although up to 56 such hot keys can be specified, but this number is shared by all running applications. However, an extra key pressed before the main hotkey is released can be requested and supplied, multiplying the number of possibilities immensely without needing many slots. The facility operates using 56 32-bit values ("DWORDs") from offset 0x3210 onwards (i.e. 0x3210, 0x3214 ...). Each of these is a 'slot' for Applications to specify Hot Keys. These will be zero initially, and zero if free. The application must search through to find an empty slot, then set this into it: Byte 0 (bits 0-7): Virtual Keycode (see the list in my FS Controls documents or the FSUIPC Advanced Users Guide). Byte 1 (bits 8-15): Shift state indicator Bit 0, the least significant, = shift Bit 1= ctrl Bit 2= alt (but use of alt strongly discouraged, see Note 1) Bit 3= "expect another keypress". If this bit is set then when the Hot Key is detected FSUIPC waits for the KEYUP *or* another key press first. The virtual keycode for that keypress is then returned in Byte 3, below. Bit 4= tab (provided as an extra "shift", for more key press flexibility) Byte 2 (bits 16-23): Flags from application. Bit 0 (1)=*reserved.* This was originally used to control the next option, but it was implemented incorrectly in FSUIPC, so now, to avoid problems, the bit is deliberately ignored. Bit 1 (2)= set if Hot Key should be passed through to FS, else it will be trapped. See Notes 1 & 2. Byte 3 (bits 24-31): Flags or results from FSUIPC. This byte needs to be cleared by the application so that it can detect when the Hot Key occurs. There is no queuing. If the Hot Key alone is seen, this byte is set to 1. If bit 3 was set in Byte 1 above *and* another key was pressed before the hotkey was released, then the virtual keycode for the extra key (2–255) is provided here. **Note 1**: ALT key combinations are not a good idea, and cannot be stopped from passing to FS. You can get them, but FS will open the menu in any case. **Note 2**: If the same Hot key is listed more than once (for instance by several applications), every copy for the same Hot Key will get the flag set, irrespective of the pass-through option. The option only applies to finally passing it to FS. If any one Hot Key user says that the key is *not* to be passed to FS (i.e. by	**Ok-Intl**	**Ok-Intl**

		leaving Flag Bit 1 unset), then it isn't passed through. **Note 3**: FSUIPC hotkeys, allocated in its "HotKeys" page, take precedence and are not passed through to applications or FS. **Use:** Having found an empty slot, write the above value into it, then monitor the highest byte of that same slot for Non-Zero. That's the keystroke. Clear that byte to detect it again. If you register several Hot Keys it will be more efficient to only scan the slots themselves when a hot key actually occurs. To detect this, simply monitor the one byte at offset 32FE (this can be paired with 32FF to scan for keys and buttons together). When it changes, read and check the flags in your slots. (The count at 32FE may change without any of your keys occurring, of course, if other applications are trapping other hot keys). When finished, and certainly before exit, be sure to clear the whole DWORD to zero so other applications can use it. If you only want to use keystrokes for a certain part of the operation of your program, only set the entries there and clear them when done. Note that if several applications want the same keystroke, they will all get it. Of course, your application can check through the whole list to make sure there are no clashes/duplicates and warn the user if so. You might have to do that at intervals in case a clashing application is loaded after yours. This system will work through WideFS with no problems too. --- **Add-Ons menu access for Applications:** See offset 2FE0.		
32F0	4	This DWORD controls some protected mode facilities in FSUIPC, designed to set known conditions in FSUIPC and prevent access to specific menus, whilst an application is running. Support in FSX not planned yet, and not assured.	**Not yet**	**Not yet**
32F4	2	The 16-bit ID of the last menu command item accessed in FS. Not planned for FSX.	**No**	**No**
32F6	2	FSUIPC selected technical option inhibits. Set bits here to turn *off* specific options and prevent the user turning them back on, for a limited time (max 14 seconds). To keep options turned off you need to write this mask at regular intervals (e.g. every 5 seconds). Note that this is not obeyed if the user has selected to option to disallow all external control of his options. If he has done this, you can detect it by reading this location back within the time limit. If it is zero, not the value written, then the user is preventing your control over his settings. Bits allocated are as follows (bit 0 = 2^0 bit), but support for most of these isn't planned for FSX at present in any case. 0 Reverse elevator trim sense 1 Fix control accelerations 2 Rudder spike elimination 3 Elevator spike elimination 4 Aileron spike elimination 5 Autopilot altitude fix (enable V/S sign corrn.) 6 Extend battery life 7 FS clock seconds sync	**Not yet**	**Not yet**
32F8	1	This provides options to inhibit certain aircraft operations, for	**Ok-Intl**	**Ok-Intl**

		use in breakdown or precise control implementations. Set individual bits for individual subsystems. Currently the following are available, all related to hydraulic power: 2^0 Set to inhibit flap operation 2^1 Set to inhibit spoiler operation 2^2 Set to inhibit gear operation 2^3 *reserved* 2^4 Set to inhibit Engine #1 reverser 2^5 Set to inhibit Engine #2 reverser 2^6 Set to inhibit Engine #3 reverser 2^7 Set to inhibit Engine #4 reverser Note that these stop operation from axis and button controls very well, and also from key presses and mouse clicks—but in these latter two cases it is done by detecting a change in the system and changing it back. This works, but the device will sometimes try to move, and this can be noticeable, especially for some reason with the flaps—the indicator gives a little jump and the noise briefly starts.		
32F9	1	Brakes being used flag. This is non-zero if the user has pressed the brakes (left, right or both) recently. It stays non-zero for a second after the last brake control or significant axis increase seen. It does *not* stay set for continued constant brake pressure via the axis inputs. It operates also for increasing values written to offset 0C00 or 0C01.	**Ok-Intl**	**N/A**
32FA	2	Text display control word. You can display messages from an external program just like an Adventure. Write the message as a zero-terminated string to offset 3380 (see below), subject to the maximum of 128 characters *including* the zero terminator, then write a number to this offset, 32FA, as follows: 0 display till replaced +n display for n seconds, or until replaced –1 display and scroll, or until replaced –n display and scroll, or for n seconds, or until replaced In the last two cases, whether the message scrolls or not depends upon the setting of the "Options—Settings—General—Text Display" option (?). The time limit only applies when scrolling is off, otherwise the message simply expires when fully scrolled off the screen.	**N/A**	**Ok-SimC** *(multiline window still Internal, via hack)*
32FC	2	AIR file change counter (incremented by FSUIPC whenever the AIR file as defined at offset 3C00 changes). This is also incremented when the FS control to "reload user aircraft" is detected—assign it to a joystick button or to a Key in FSUIPC for this.	**Ok-Intl**	**N/A**
32FE	1	Hot Key change counter, incremented by FSUIPC whenever any of the Hot Keys defined in the table at offset 3210 occurs and therefore has its flag set by FSUIPC.	**Ok-Intl**	**N/A**
32FF	1	Hot Button change counter, incremented by FSUIPC whenever any of the Hot Buttons defined in the table at offset 2910 changes state in the right way, and therefore has its flag set by FSUIPC.	**Ok-Intl**	**N/A**
3300	2	Additional radio and autopilot status indicators (read only access). Allocation by bits which are set when true. Bit 0 = least significant (value 1): 0 = reserved 1 = good NAV1 2 = good NAV2 3 = good ADF1 4 = NAV1 has DME	**Ok-Intl**	**N/A**

		5 = NAV2 has DME 6 = NAV1 is ILS 7 = AP NAV1 radial acquired 8 = AP ILS LOC acquired (incl BC—see 10) 9 = AP ILS GS acquired 10=AP ILS LOC is BC 11=good ADF2 12=NAV2 is ILS 13–15 reserved		
3302	2	Assorted FSUIPC options, set by user parameters: read-only via the IPC. None yet applicable for FSX.	**Not yet**	**N/A**
3304	4	FSUIPC version number: The HIWORD (i.e. bytes 3306-7) gives the main version as BCD x 1000: e.g. 0x1998 for 1.998 The LOWORD (bytes 3304-5) gives the Interim build letter: 0=none, 1-26=a-z: e.g. 0x0005 = 'e'	**Ok-Intl**	**N/A**
3308	2	FS version, as determined by FSUIPC: Currently only one of these: 1 = FS98 2 = FS2000 3 = CFS2 4 = CFS1 5 = reserved 6 = FS2002 7 = FS2004 "A Century of Flight" 8 = FSX 9 = ESP 10=P3D	**Ok-Intl**	**N/A**
330A	2	Fixed *read-only* pattern, set to 0xFADE. Use this to check that the values in 3304-3308 are valid (Note: the supplied LIB writes its version number here, but this has no effect and is only for assistance when viewing LOG files).	**Ok-Intl**	**N/A**
330C	2	Assorted status flags, the only ones which are of use to applications being: 2^1 When set this indicates that programs have full access to the IPC not. This can be read without triggering the message box to users which tells them of an unaccredited access attempt. Note that on WideClient it will always be set, assuming WideServer is registered on the FS PC. (*should always be 1 in FSUIPC4*) 2^2 Set if the user has fully registered FSUIPC 2^4 Set when the user Throttle Sync option (in the Hot Keys page of FSUIPC options) is enabled.	**?-Intl**	**N/A**
330E	1	Count of external IPC applications seen connecting since the session began. Keeps increasing till it gets to 255 then stays at that value.	**No**	**No**
330F	17	Reserved area for WideFS KeySend facility		
3320	2	This word is used to activate a facility supported by WideFS to automatically shut down the PCs running WideServer (i.e. this one) and WideClient. The .ini files of each WideFS component which is to activate the shutdown needs the "AllowShutdown=Yes" parameter included. The application performing the shut down action must write 0xABCD to this offset. WideServer automatically resets this word to zero 5 seconds afterwards, before it initiates its own PC's shutdown if specified. This delay is to ensure the Clients get the message before the host dies, and the clearing to zero is done so that the survivors	**Ok-Intl**	**Ok-Intl**

		can continue. WideFS also provides the lesser option “AllowShutdown=App” which only closes down the WideClient or, in the case of WideServer, the FS session. Later still the “AppOnly” variation was added, which keeps WideClient running, ready to reload the applications when FS restarts. A hot key facility to invoke this WideFS shutdown from the FS keyboard is provided via WideServer’s INI parameters. The pattern 0xDCBA written here invokes a “close application” action. On all WideFS PCs with any form of shutdown allowed, this pattern closes only those applications loaded by WideFS and leaves WideClient running ready to reload them. On the Server, if it is allowed, it closes FS itself. A hot key facility is provided for this variant, too.		
3322	2	WideServer version number, if enabled. Otherwise this is zero. This is a BCD value giving the version number x 1000, for example 0x5110 means version 5.110. See also offset 333C.	**Ok-Intl**	**Ok-Intl**
3324	4	This is the altimeter reading in feet (or metres, if the user is running with the preference for altitudes in metres), as a 32-bit signed integer. Please check offset 0C18 to determine when metres are used (0C18 contains ‘2’). The same value can be calculated from the actual altitude and the difference between the QNH and the altimeter “Kollsman” pressure setting, but this value ensures agreement.	**Ok-SimC**	**?-SimC**
3328	2	Elevator Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A).	**?-Intl**	**N/A**
332A	2	Aileron Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A).	**?-Intl**	**N/A**
332C	2	Rudder Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A).	**?-Intl**	**N/A**
332E	2	Throttle Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A). This is the single throttle, applied to whichever engines are denoted by the bits in offset 0888.	**?-Intl**	**N/A**
3330	2	Throttle 1 Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A).	**?-Intl**	**N/A**
3332	2	Throttle 2 Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A).	**?-Intl**	**N/A**
3334	2	Throttle 3 Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A).	**?-Intl**	**N/A**
3336	2	Throttle 4 Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A).	**?-Intl**	**N/A**
3338	2	Elevator Trim Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 310A).	**?-Intl**	**N/A**
333A	2	Throttle lower limit. This is normally 0 if no reverse is available, otherwise gives the reverse limit such as –4096 (for 25%).	**?-SimC**	**No**
333C	2	WideFS flags: those used so far are: 2^0 1 =if TCP is being used, 0 if SPX		

		2^1 1 if connected at all, 0 is waiting for connections See offset 3322 for WideFS version number, which also confirms that WideServer is registered and running.		
333E	2	Weather clear count: This is incremented every time FS's "clear weather" routine is called, for whatever reason.	**No**	**No**
3340	36	This area is used for externally signalled "joystick button" control, a set of 288 "virtual buttons". Each DWORD or 32 bits represents one "joystick" with 32 buttons. If an external program sets or clears a bit in any of these 9 DWORDS the "Buttons" page in FSUIPC will register the change as a button operation on one of Joystick numbers 64 to 73 (corresponding to the 9 DWORDs). So, FSUIPC can be used to program whatever actions the user wants. See also offset 29F0	**Ok-Intl**	**Ok-Intl**
3364	1	FS2004 "Ready to Fly" indicator. This is non-zero when FS is loading, or reloading a flight or aircraft or scenery, and becomes zero when flight mode is enabled (even if the simulator is paused or in Slew mode). *(Note that in FSX it tends to only be set during initial loading. Use together with 3365)*	**Ok-SimE** *(See note)*	**N/A**
3365	1	"In Menu or Dialog" flag. This byte is non-zero when FS is effectively paused because the user accessed the Menu, or is in a dialogue resulting from menu or other selection activity. The non-zero values are: 1 = FS in a menu (simulation stopped) 2 = FS in a dialogue (simulation probably stopped) Both bits may be set in dialogues accessed through the menu. Note that the 2 bit may flicker a little on exit from the dialogue, due to the way it is detected. *(In FSX/P3D these two states may be a little confused. Not also that FSX/P3D does* ***not*** *freeze whilst navigating menus – it only does so in the dialogues themselves, and then not all of them)*	**Ok-SimE** *(See note)*	**N/A**
3366	1	This byte reflects the FS2004 "Engine on Fire" flags. I'm not sure if FS actually simulates such events, but it appears to have allocated Gauge-accessible variables to indicate them. This byte uses bits 2^0–2^3 as flags for fires in Engines 1 to 4, respectively.	**Ok-SimC**	**Ok-SimC**
3367	1	This byte shows doors that are open, one bit per door: 2^0 = Exit1 … 2^3 = Exit 4. *N.B. FSUIPC4 does handle up to 8 doors, one for each bit 0-7. Whether FSX can actually process Exits 5-8 is unknown however.*	**Ok-SimC**	**Ok-SimE**
3368	4	*Reserved* for PFC.DLL events.		
336C	2	Frame rate calling counter. This is simply a number that is incremented each time FSUIPC is entered from FS using the entry related to frame rates.	**Ok-Intl**	**N/A**
336E	2	Toe brake axes have been selected as "Set" in FSUIPC's joystick pages if this is non-zero. Byte 336E is non-zero for Left Brake, byte 336F for Right Brake. Note that this only means that the user has told FSUIPC to handle the toe braking, by pressing "Set". It will only actually do so if it sees brake messages.	**Ok-Intl**	**N/A**
3370	4	Four single byte PFC driver "alive" counters: 3370 = COM port read thread alive and running 3371 = Elevator trim motor action (0=off, 1=up, 2=dn) 3372 = COM port write thread alive 3373 = Main FS chain alive N.B. without the main FS chain running the other three aren't		

		maintained in any case, so mean nothing.		
3374	4	This is the "live" millisecond count as used in the FSUIPC Log. It is updated on each FS chained call to FSUIPC.	**Ok-Intl**	**N/A**
3378	4	This is the millisecond timestamp value of the most recent line in the current FSUIPC Log. It is updated when each line is logged.	**Ok-Intl**	**N/A**
337C	1	Propeller de-ice switches, (1 = on, 0 = off), read for state, write to control: one bit for each prop, bits 0—3 = Props 1--4	**?-SimC**	**?-SimC**
337D	1	Structural de-ice switch, (1 = on, 0 = off), read for state, write to control.	**Ok-SimC**	**Ok-SimC**
337E	2	FSUIPC activity count. Simply a number that is incremented every time FSUIPC receives a call or message from Flight Simulator. This can be used through WideFS to check if FS is still active, for example. Note that when FS is loading aircraft or scenery/textures, this value may not change for many seconds as FSUIPC is then not getting any processor time at all.	**Ok-Intl**	**N/A**
3380	128	Message text area: The text is truncated if longer than 127 characters, there always being a zero terminator provided. You can *write* messages to this area, always zero terminated, for display on the FS windshield *or* via ShowText or other applications. After placing the message text, you must write the 16-bit timer value to offset 32FA to make FSUIPC send the message (see 32FA above).	**Ok-Intl**	**Ok-Intl** *(see 32FA)*
3400	2	FSUIPC logging options, reading and setting, bit-oriented with bits used as follows (bit numbers from bit 0 = least significant): 0 = logging enabled (ignored, as logging is always enabled nowadays) 1 = weather 2 = IPC writes 3 = IPC reads 4 = Extras 5 = Extended, technical button and key logging 6 = VRI comms 7 = com HID 8 = L:Vars 9 = Axes 10 = Events 11 = Button/key standard logging 12 = Lua logging separate Note that FSUIPC revents writes to this offset ***unless*** the User allows it to happen. This is via the [General] section parameter: LogOptionsProtect =Yes If you need to allow programs to change the logging, change this to 'No' before running FS.	**Ok-Intl**	**Ok-Intl** **But see Note!**
3410	2	Assorted indicator flags. These are the only ones currently set (bit numbers, bit 0 = 2^0): 4 Engine 1 Reverser is set but inhibited* 5 Engine 2 Reverser is set but inhibited* 6 Engine 3 Reverser is set but inhibited* 7 Engine 4 Reverser is set but inhibited* * Reverser inhibits are set in offset 32F8. Note that these flags will be cleared only when the inhibit is removed *or* the relevant throttle input goes positive (i.e. not just to idle).	**Ok-Intl**	**N/A**
3412	2	Spoiler Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 341A). Copy this to 0BD0 for normal spoiler action.	**Ok-Intl**	**N/A**
3414	2	Flaps Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 341A). Copy this to 0BDC for normal flaps action.	**Ok-Intl**	**N/A**
3416	2	Left Brake Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 341A). Copy this to 0BC4 for normal left brake action.	**Ok-Intl**	**N/A**
3418	2	Right Brake Axis input value, post calibration, just before being applied to the simulation (if allowed to by the byte at offset 341A). Copy this to 0BC6 for normal right brake action.	**Ok-Intl**	**N/A**

341A	1	Controls the joystick connection for ancillary axis controls, currently Left and Right brake, flaps and spoiler axes. Normally all zero, set the following bits to actually disconnect the specific joystick axes (from least significant bit = 0): 0 Left brake ("Axis Left Brake Set") 1 Right Brake ("Axis Right Brake Set") 2 Flaps 3 Spoilers This feature is intended for use in simulating relevant subsystem failures or partial failures. Programs can read the input axis values from offsets 3412–3418 above, and apply them, after appropriate modification, to the relevant FS axis offsets (at 0BC4 and 0BC6 for Brakes, 0BDC for Flaps or 0BD0 for Spoiler. In order to protect the user from a broken or crashed application, the flags are cleared 10 seconds after they have been set, so applications will need to repeat the setting every few seconds. Note that this byte is effectively "write only". Upon reading it will always appear to contain zero.	N/A	OK-Intl
341C	1	No smoking alert switch (1 = on, 0= off)	Ok-SimC	Ok-SimE
341D	1	Seat belts alert switch (1 = on, 0 = off)	Ok-SimC	Ok-SimE
341E	1	Hydraulic switches, one bit for each: 2^0=pump1 ... 2^3=pump3	Ok-SimC	Ok-SimE
341F	1	Fuel cross feed switch	Ok-SimC	Ok-SimE
3420	4	Rad ins switch	?-SimC	No
3424	4	Low height warning	No info	No info
3428	8	Decision height in metres (64-bit floating point double	?-SimC	No
3438	8	Engine 1 fuelflow bug position	?-SimC	No-SimC+
3440	8	Engine 2 fuelflow bug position	?-SimC	No-SimC+
3448	8	Engine 3 fuelflow bug position	?-SimC	No-SimC+
3450	8	Engine 4 fuelflow bug position	?-SimC	No-SimC+
3458	8	Panel autopilot speed setting (But see preferred offset 07E2)	?-SimC	No
3460	8	LINEAR CL ALPHA, Float64, per radian	SimC	No
3468	8	ZERO LIFT ALPHA, Float64, radians	SimC	No
3470	8	Ambient wind X component, double float, m/sec (+ve West, -ve East)	Ok-SimC	No-SimC+
3478	8	Ambient wind Y component, double float, m/sec (+ve Up, -ve Down)	Ok-SimC	No-SimC+
3480	8	Ambient wind Z component, double float, m/sec (+ve South, -ve North)	Ok-SimC	No-SimC+
3488	8	Ambient wind velocity, double float, m/sec	Ok-SimC	No-SimC+
3490	8	Ambient wind direction, double float, True	Ok-SimC	No-SimC+
3498	8	Ambient pressure, double float.	Ok-SimC	No
34A0	8	Sea level pressure (QNH), double float	Ok-SimC	No
34A8	8	Ambient temperature, double float	Ok-SimC	No
34B0	8	Pressure Altitude (metres), double float. This is the indicated altitude when the altimeter Kollsman setting is 1013.2 hPa (29.92").	Ok-SimC	No
34B8	8	Standard ATM Temperature, degrees Rankine, double float. This is the expected temperature at the actual AMSL in the International Standard Atmosphere model.	Ok-SimC	No
34C0	8	Sigma Sqare Root, double float. This is actually the square root of the Sigma value as provided at offset 28F0.	Ok-SimC	No
34C8	8	Total velocity, ft/sec, double float. This is the resultant velocity of the three X,Y,Z orthogonal velocities given in offsets 3178, 3180 and 3188.	Ok-SimC	No
34D0	8	G force maximum	Ok-SimC	No
34D8	8	G force minimum	Ok-SimC	No
34E8	4	Engine1 max rpm (*Appears to mean max RPM actually reached*)	Ok-SimC	No
34EC	4	Engine2 max rpm (*Appears to mean max RPM actually reached*)	Ok-SimC	No

34F0	4	Engine3 max rpm (*Appears to mean max RPM actually reached*)	**Ok-SimC**	**No**
34F4	4	Engine4 max rpm (*Appears to mean max RPM actually reached*)	**Ok-SimC**	**No**
34F8	2	PFCFSX left brake application (0 - 16383)		
34FA	2	PFCFSX right brake application (0 - 16383)		
3500	24	ATC aircraft model string for currently loaded user aircraft, as declared in the AIRCRAFT.CFG file. This is limited to a maximum of 24 characters, including a zero terminator.	**Ok-SimC**	**No**
3518	8	This double provides the FS-set "Ambient Wind Y" value within about one second of offset 3478 being written by an application, to control up and down drafts. This allows such a program to monitor FS/scenery arranged updrafts and adjust its actions accordingly.	**Not yet**	**N/A**
3520	2	Earliest version number of connected WideClients (or clients which have been connected). Zero if no connections have been made, or if all connected clients have been version 6.441 or before.	**Ok-Intl**	**N/A**
3541	1	This operates the FSUIPC "freeze flight position" facility. This keeps the aircraft at the same latitude and longitude for as long as it is engaged. The altitude and attitude of the aircraft is free to change, and, in fact, the aircraft flies as normal except for not changing its position over the ground. This is apparently a very useful facility for training environments. For program control, write a non-zero values to this one byte offset. This acts as a timer. The freeze will last for as long as this byte is non-zero. It is used as a time, counting down 1 every timer tick of 55 mSecs or so. To retain the freeze for a good time, write 255 here and do so every 5–10 seconds. Allow for WideFS delays. Note that if FS is paused, then the freeze lasts until the pause is released and re-engaged.	**N/A**	**Ok-Intl/SimC**
3542	2	Standby altimeter pressure setting ("Kollsman" window). As millibars (hectoPascals) * 16. [*This is used by FSUIPC to maintain offset 3544. It is not used by FS at all*]	**Ok-Intl**	**Ok-Intl**
3544	4	This is the standby altimeter reading in feet (or metres, if the user is running with the preference for altitudes in metres), as a 32-bit signed integer. Please check offset 0C18 to determine when metres are used (0C18 contains '2'). This value is maintained by FSUIPC using the pressure setting supplied in offset 3542. It isn't used in FS itself, but is supplied for additional gauges and external altimeters so that the standby can be kept at the correct (or last notified) QNH whilst the main altimeter is used for Standard settings (for airliners flying Flight Levels).	**Ok-Intl**	**Ok-Intl**
3548	8	Horizon bars offset, as a percentage of maximum, in floating point double format. (–100.0 down to +100.0 up). On the default Cessnas the maximum offset is 10 degrees.	**?-SimC**	**No-SimC+**
3550	56	*Reserved for FSUIPC diagnostics related to Gauge Mousing*		
3590	4	Engine 1 Fuel Valve, 1 = open, 0 = closed.	**Ok-SimC**	**Ok-SimE**
3594	4	Engine 2 Fuel Valve, 1 = open, 0 = closed.	**Ok-SimC**	**Ok-SimE**
3598	4	Engine 3 Fuel Valve, 1 = open, 0 = closed.	**Ok-SimC**	**Ok-SimE**
359C	4	Engine 4 Fuel Valve, 1 = open, 0 = closed.	**Ok-SimC**	**Ok-SimE**
35A0	8	Airspeed Mach value, double float.	**Ok-SimC**	**No**
		RECIPROCATING ENGINE 4 DATA		
35A8	8	Reciprocating engine 4 manifold pressure, in lbs/sqft, as a double (FLOAT64). Divide by 70.7262 for inches Hg.	**Ok-SimC**	**Ok-SimC**
35B0	8	Engine 4 cowl flap position, as a double float: 0.0=fully closed, 1.0=fully open. Can be used to handle position and set it.	**Ok-SimC**	**Ok-SimC**
35B8	8	Reciprocating engine 4 carb heat pos *("alternate air" instead?)*	**No-SimC?**	**No-SimC?**
35C0	8	Reciprocating engine 4 alternate air pos	**?-SimC**	**?-SimC**
35C8	8	Reciprocating engine 4 coolant reservoir percent	**?-SimC**	**?-SimC**

35D0	4	Reciprocating engine 4, left magneto select (1 = on, 0 = off)	Ok-SimC	No
35D4	4	Reciprocating engine 4, right magneto select (1 = on, 0 = off)	Ok-SimC	No
35D8	8	Reciprocating engine 4 fuel/air mass ratio, as a double (FLOAT64).	?-SimC	?-SimC
35E0	8	Reciprocating engine 4 brake power in ft-lbs, as a double (FLOAT64). Divide by 550 for HP.	?-SimC	?-SimC
35E8	8	Reciprocating engine 4 carburettor temperature, in degrees Rankine, as a double (FLOAT64).	?-SimC	?-SimC
35F0	8	Reciprocating engine 4 starter torque	?-SimC	?-SimC
35F8	4	Reciprocating engine 4 turbocharger failed	?-SimC	?-SimC
35FC	4	Reciprocating engine 4 emergency boost active flag (32-bit BOOLEAN). On some aircraft this controls whether the supercharger is active or not.	?-SimC	?-SimC
3600	8	Reciprocating engine 4 emergency boost elapsed time in seconds, as a double (FLOAT64). This counts how long the boost has been engaged, when it is made active by an FS control. FS turns it off when reaching 312. You can keep it going by occasionally writing 0 here.	?-SimC	?-SimC
3608	8	Reciprocating engine 4 wastegate position (read-only, effectively)	?-SimC	?-SimC
3610	8	Reciprocating engine 4 TIT degrees Rankine	?-SimC	?-SimC
3618	8	Reciprocating engine 4 CHT degrees Rankine, FLOAT64	Ok-SimC	Ok-SimC
3620	8	Reciprocating engine 4 Radiator temperature degrees Rankine	?-SimC	?-SimC
3628	8	Reciprocating engine 4 fuel pressure (double or FLOAT64)	?-SimC	?-SimC
3640	4	Reciprocating engine 4 tank selector: : 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main, 20=Right Main	Ok-SimC	Ok-SimE
3644	4	Reciprocating engine 4 tanks used, a bit mask: 0 Center 1 1 Center 2 2 Center 3 3 Left Main 4 Left Aux 5 Left Tip 6 Right Main 7 Right Aux 8 Right Tip 9 External 1 10 External 2	Ok-SimC	No
3648	4	Reciprocating engine 4, number of fuel tanks supplying fuel.	Ok-SimC	No
3654	4	Reciprocating engine 4 fuel available flag (0 or 1).	?-SimC	?-SimE
		RECIPROCATING ENGINE 3 DATA		
3668	8	Reciprocating engine 3 manifold pressure, in lbs/sqft, as a double (FLOAT64). Divide by 70.7262 for inches Hg.	Ok-SimC	Ok-SimC
3670	8	Engine 3 cowl flap position, as a double float: 0.0=fully closed, 1.0=fully open. Can be used to handle position and set it.	Ok-SimC	Ok-SimC
3678	8	Reciprocating engine 3 carb heat pos		
3680	8	Reciprocating engine 3 alternate air pos		
3688	8	Reciprocating engine 3 coolant reservoir percent		
3690	4	Reciprocating engine 3, left magneto select (1 = on, 0 = off)	Ok-SimC	No
3694	4	Reciprocating engine 3, right magneto select (1 = on, 0 = off)	Ok-SimC	No
3698	8	Reciprocating engine 3 fuel/air mass ratio, as a double (FLOAT64).		
36A0	8	Reciprocating engine 3 brake power in ft-lbs, as a double (FLOAT64). Divide by 550 for HP.		
36A8	8	Reciprocating engine 3 carburettor temperature, in degrees Rankine, as a double (FLOAT64).		
36B0	8	Reciprocating engine 3 starter torque		
36B8	4	Reciprocating engine 3 turbocharger failed		

36BC	4	Reciprocating engine 3 emergency boost active flag (32-bit BOOLEAN). On some aircraft this controls whether the supercharger is active or not.		
36C0	8	Reciprocating engine 3 emergency boost elapsed time in seconds, as a double (FLOAT64). This counts how long the boost has been engaged, when it is made active by an FS control. FS turns it off when reaching 312. You can keep it going by occasionally writing 0 here.		
36C8	8	Reciprocating engine 3 wastegate position (read-only, effectively)		
36D0	8	Reciprocating engine 3 TIT degrees Rankine		
36D8	8	Reciprocating engine 3 CHT degrees Rankine, FLOAT64	**Ok-SimC**	**Ok-SimC**
36E0	8	Reciprocating engine 3 Radiator temperature degrees Rankine		
36E8	8	Reciprocating engine 3 fuel pressure (double or FLOAT64)		
3700	4	Reciprocating engine 3 tank selector: 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main, 20=Right Main	**Ok-SimC**	**Ok-SimE**
3704	4	Reciprocating engine 3 tanks used, a bit mask: 0 Center 1 1 Center 2 2 Center 3 3 Left Main 4 Left Aux 5 Left Tip 6 Right Main 7 Right Aux 8 Right Tip 9 External 1 10 External 2	**Ok-SimC**	**No**
3708	4	Reciprocating engine 3, number of fuel tanks supplying fuel.	**Ok-SimC**	**No**
3714	4	Reciprocating engine 3, fuel available flag (0 or 1).		
		RECIPROCATING ENGINE 2 DATA		
3728	8	Reciprocating engine 2 manifold pressure, in lbs/sqft, as a double (FLOAT64). Divide by 70.7262 for inches Hg.	**Ok-SimC**	**Ok-SimC**
3730	8	Engine 2 cowl flap position, as a double float: 0.0=fully closed, 1.0=fully open. Can be used to handle position and set it.	**Ok-SimC**	**Ok-SimC**
3738	8	Reciprocating engine 2 carb heat pos		
3740	8	Reciprocating engine 2 alternate air pos		
3748	8	Reciprocating engine 2 coolant reservoir percent		
3750	4	Reciprocating engine 2, left magneto select (1 = on, 0 = off)	**Ok-SimC**	**No**
3754	4	Reciprocating engine 2, right magneto select (1 = on, 0 = off)	**Ok-SimC**	**No**
3758	8	Reciprocating engine 2 fuel/air mass ratio, as a double (FLOAT64).		
3760	8	Reciprocating engine 2 brake power in ft-lbs, as a double (FLOAT64). Divide by 550 for HP.		
3768	8	Reciprocating engine 2 carburettor temperature, in degrees Rankine, as a double (FLOAT64).		
3770	8	Reciprocating engine 2 starter torque		
3778	4	Reciprocating engine 2 turbocharger failed		
377C	4	Reciprocating engine 2 emergency boost active flag (32-bit BOOLEAN). On some aircraft this controls whether the supercharger is active or not.		
3780	8	Reciprocating engine 2 emergency boost elapsed time in seconds, as a double (FLOAT64). This counts how long the boost has been engaged, when it is made active by an FS control. FS turns it off when reaching 312. You can keep it going by occasionally writing 0 here.		
3788	8	Reciprocating engine 2 wastegate position (read-only, effectively)		

3790	8	Reciprocating engine 2 TIT degrees Rankine		
3798	8	Reciprocating engine 2 CHT degrees Rankine, FLOAT64	**Ok-SimC**	**Ok-SimC**
37A0	8	Reciprocating engine 2 Radiator temperature degrees Rankine		
37A8	8	Reciprocating engine 2 fuel pressure (double or FLOAT64)		
37C0	4	Reciprocating engine 2 tank selector: 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main, 20=Right Main	**Ok-SimC**	**Ok-SimE**
37C4	4	Reciprocating engine 2 tanks used, a bit mask: 0 Center 1 1 Center 2 2 Center 3 3 Left Main 4 Left Aux 5 Left Tip 6 Right Main 7 Right Aux 8 Right Tip 9 External 1 10 External 2	**Ok-SimC**	**No**
37C8	4	Reciprocating engine 2, number of fuel tanks supplying fuel.	**Ok-SimC**	**No**
37D4	4	Reciprocating engine 2, fuel available flag (0 or 1).		
		RECIPROCATING ENGINE 1 DATA		
37E8	8	Reciprocating engine 1 manifold pressure, in lbs/sqft, as a double (FLOAT64). Divide by 70.7262 for inches Hg.	**Ok-SimC**	**Ok-SimC**
37F0	8	Engine 1 cowl flap position, as a double float: 0.0=fully closed, 1.0=fully open. Can be used to handle position and set it.	**Ok-SimC**	**Ok-SimC**
37F8	8	Reciprocating engine 1 carb heat pos		
3800	8	Reciprocating engine 1 alternate air pos		
3808	8	Reciprocating engine 1 coolant reservoir percent		
3810	4	Reciprocating engine 1, left magneto select (1 = on, 0 = off)	**Ok-SimC**	**No**
3814	4	Reciprocating engine 1, right magneto select (1 = on, 0 = off)	**Ok-SimC**	**No**
3818	8	Reciprocating engine 1 fuel/air mass ratio, as a double (FLOAT64).		
3820	8	Reciprocating engine 1 brake power in ft-lbs, as a double (FLOAT64). Divide by 550 for HP.		
3828	8	Reciprocating engine 1 carburettor temperature, in degrees Rankine, as a double (FLOAT64).		
3830	8	Reciprocating engine 1 starter torque		
3838	4	Reciprocating engine 1 turbocharger failed		
383C	4	Reciprocating engine 1 emergency boost active flag (32-bit BOOLEAN). On some aircraft this controls whether the supercharger is active or not.		
3840	8	Reciprocating engine 1 emergency boost elapsed time in seconds, as a double (FLOAT64). This counts how long the boost has been engaged, when it is made active by an FS control. FS turns it off when reaching 312. You can keep it going by occasionally writing 0 here.		
3848	8	Reciprocating engine 1 wastegate position (read-only, effectively)		
3850	8	Reciprocating engine 1 TIT degrees Rankine		
3858	8	Reciprocating engine 1 CHT degrees Rankine, FLOAT64	**Ok-SimC**	**Ok-SimC**
3660	8	Reciprocating engine 1 Radiator temperature degrees Rankine		
3868	8	Reciprocating engine 1 fuel pressure (double or FLOAT64)		
3870	8	Engine 1 primer		
3880	4	Reciprocating engine 1 tank selector: 0=None, 1=All, 2=Left, 3=Right, 4=LeftAux, 5=RightAux, 6=Centre, 7=Centre2, 8=Centre3, 9=External1, 10=External2, 11=Right Tip, 12=Left Tip, 13=Crossfeed, 14=Crossfeed LtoR, 15=Crossfeed RtoL, 16=Crossfeed both, 17=External, 18=Isolate, 19=Left Main,	**Ok-SimC**	**Ok-SimE**

		20=Right Main		
3884	4	Reciprocating engine 1 tanks used, a bit mask: 0 Center 1 1 Center 2 2 Center 3 3 Left Main 4 Left Aux 5 Left Tip 6 Right Main 7 Right Aux 8 Right Tip 9 External 1 10 External 2	**Ok-SimC**	**No**
3888	4	Reciprocating engine 1, number of fuel tanks supplying fuel.	**Ok-SimC**	**No**
3894	4	Reciprocating engine 1, fuel available flag (0 or 1).		
		GENERAL ENGINE 4 DATA		
38A0	4	General engine 4 failure (0=none, 1=full)	**Ok-SimC**	**No**
38A4	4	General engine 4 combustion	**Ok-SimC**	**Ok-SimC**
38A8	8	General engine 4 throttle lever position, as a double (FLOAT64). 0.0=idle, 1.0=max	**Ok-SimC**	**No**
38B0	8	General engine 4 mixture lever position, as a double (FLOAT64). 0.0=cutoff, 1.0=full rich	**Ok-SimC**	**No**
38B8	8	General engine 4 propeller lever position, as a double (FLOAT64). 0–1	**Ok-SimC**	**No**
38C0	4	General Engine 4 Starter	**Ok-SimC**	**No**
3918	8	General engine 4 oil temperature in degrees Rankine, as a double (FLOAT64).	**Ok-SimC**	**Ok-SimC**
3920	8	General engine 4 oil pressure in lbs/sqft, as a double (FLOAT64). Divide by 144 for PSI.	**Ok-SimC**	**No**
3928	8	Reciprocating engine 4 oil leak percent, as a double (FLOAT64)	**Ok-SimC**	**No**
3930	8	General engine 4 EGT in degrees Rankine, as a double (FLOAT64). Convert to Fahrenheit by Rankine – 459.67. FS default gauges show Centigrade.	**OK-SimC**	**OK-SimC**
3938	4	Engine 4 generator switch, a 32-bit BOOL (0 = off, 1= on)	**OK-SimC**	**OK-SimC**
393C	4	Engine 4 generator active, a 32-bit BOOL (0 = off, 1= on), Goes to 0 when engine stops.	**OK-SimC**	**OK-SimC**
3940	8	Reciprocating engine 4 damage percent, 64-bit floating point.	**OK-SimC**	**No**
3948	8	Reciprocating engine 4 combustion sound percent, 64-bit floating point.	**OK-SimC**	**No**
3958	4	Engine 4 fuel pump switch, a 32-bit BOOL (0 = off, 1= on)	**Ok-SimC**	**Ok-SimE**
		GENERAL ENGINE 3 DATA		
3960	4	General engine 3 failure (0=none, 1=full)	**Ok-SimC**	**No**
3964	4	General engine 3 combustion	**Ok-SimC**	**Ok-SimC**
3968	8	General engine 3 throttle lever position, as a double (FLOAT64). 0.0=idle, 1.0=max	**Ok-SimC**	**No**
3970	8	General engine 3 mixture lever position, as a double (FLOAT64). 0.0=cutoff, 1.0=full rich	**Ok-SimC**	**No**
3978	8	General engine 3 propeller lever position, as a double (FLOAT64). 0–1	**Ok-SimC**	**No**
3980	4	General Engine 3 Starter	**Ok-SimC**	**No**
39D8	8	General engine 3 oil temperature in degrees Rankine, as a double (FLOAT64).	**Ok-SimC**	**Ok-SimC**
39E0	8	General engine 3 oil pressure in lbs/sqft, as a double (FLOAT64). Divide by 144 for PSI.	**Ok-SimC**	**No**
39E8	8	Reciprocating engine 3 oil leak percent, as a double (FLOAT64)	**Ok-SimC**	**No**
39F0	8	General engine 3 EGT in degrees Rankine, as a double (FLOAT64). Convert to Fahrenheit by Rankine – 459.67. FS default gauges show Centigrade.	**OK-SimC**	**OK-SimC**
39F8	4	Engine 3 generator switch, a 32-bit BOOL (0 = off, 1= on)	**OK-SimC**	**OK-SimC**
39FC	4	Engine 3 generator active, a 32-bit BOOL (0 = off, 1= on), Goes to 0 when engine stops.	**OK-SimC**	**OK-SimC**
3A00	8	Reciprocating engine 3 damage percent, 64-bit floating point.	**OK-SimC**	**No**

3A08	8	Reciprocating engine 3 combustion sound percent, 64-bit floating point.	**OK-SimC**	**No**
3A18	4	Engine 3 fuel pump switch, a 32-bit BOOL (0 = off, 1= on)	**Ok-SimC**	**Ok-SimE**
		GENERAL ENGINE 2 DATA		
3A20	4	General engine 2 failure (0=none, 1=full)	**Ok-SimC**	**No**
3A24	4	Reciprocating engine 2 combustion	**Ok-SimC**	**Ok-SimC**
3A28	8	General engine 2 throttle lever position, as a double (FLOAT64). 0.0=idle, 1.0=max	**Ok-SimC**	**No**
3A30	8	General engine 2 mixture lever position, as a double (FLOAT64). 0.0=cutoff, 1.0=full rich	**Ok-SimC**	**No**
3A38	8	General engine 2 propeller lever position, as a double (FLOAT64). 0–1	**Ok-SimC**	**No**
3A40	4	General Engine 2 Starter	**Ok-SimC**	**No**
3A98	8	General engine 2 oil temperature in degrees Rankine, as a double (FLOAT64).	**Ok-SimC**	**Ok-SimC**
3AA0	8	General engine 2 oil pressure in lbs/sqft, as a double (FLOAT64). Divide by 144 for PSI.	**Ok-SimC**	**No**
3AA8	8	Reciprocating engine 2 oil leak percent, as a double (FLOAT64)	**Ok-SimC**	**No**
3AB0	8	General engine 2 EGT in degrees Rankine, as a double (FLOAT64). Convert to Fahrenheit by Rankine – 459.67. FS default gauges show Centigrade.	**OK-SimC**	**OK-SimC**
3AB8	4	Engine 2 generator switch, a 32-bit BOOL (0 = off, 1= on)	**OK-SimC**	**OK-SimC**
3ABC	4	Engine 2 generator active, a 32-bit BOOL (0 = off, 1= on), Goes to 0 when engine stops.	**OK-SimC**	**OK-SimC**
3AC0	8	Reciprocating engine 2 damage percent, 64-bit floating point.	**OK-SimC**	**No**
3AC8	8	Reciprocating engine 2 combustion sound percent, 64-bit floating point.	**OK-SimC**	**No**
3AD8	4	Engine 2 fuel pump switch, a 32-bit BOOL (0 = off, 1= on) (*Note that it only copes with off-lo on the Baron*)	**Ok-SimC**	**Ok-SimE**
		GENERAL ENGINE 1 DATA		
3AE0	4	General engine 1 failure (0=none, 1=full)	**Ok-SimC**	**No**
3AE4	4	Reciprocating engine 1 combustion	**Ok-SimC**	**Ok-SimC**
3AE8	8	General engine 1 throttle lever position, as a double (FLOAT64). 0.0=idle, 1.0=max	**Ok-SimC**	**No**
3AF0	8	General engine 1 mixture lever position, as a double (FLOAT64). 0.0=cutoff, 1.0=full rich	**Ok-SimC**	**No**
3AF8	8	General engine 1 propeller lever position, as a double (FLOAT64). 0–1	**Ok-SimC**	**No**
3B00	4	General Engine 1 Starter	**Ok-SimC**	**No**
3B58	8	General engine 1 oil temperature in degrees Rankine, as a double (FLOAT64).	**Ok-SimC**	**Ok-SimC**
3B60	8	General engine 1 oil pressure in lbs/sqft, as a double (FLOAT64). Divide by 144 for PSI.	**Ok-SimC**	**No**
3B68	8	Reciprocating engine 1 oil leak percent, as a double (FLOAT64)	**Ok-SimC**	**No**
3B70	8	General engine 1 EGT in degrees Rankine, as a double (FLOAT64). Convert to Fahrenheit by Rankine – 459.67. FS default gauges show Centigrade.	**OK-SimC**	**OK-SimC**
3B78	4	Engine 1 generator switch, a 32-bit BOOL (0 = off, 1= on)	**OK-SimC**	**OK-SimC**
3B7C	4	Engine 1 generator active, a 32-bit BOOL (0 = off, 1= on), Goes to 0 when engine stops.	**OK-SimC**	**OK-SimC**
3B80	8	Reciprocating engine 1 damage percent, 64-bit floating point.	**OK-SimC**	**No**
3B88	8	Reciprocating engine 1 combustion sound percent, 64-bit floating point.	**OK-SimC**	**No**
3B98	4	Engine 1 fuel pump switch, a 32-bit BOOL (0 = off, 1= on) (*Note that it only copes with off-lo on the Baron*)	**Ok-SimC**	**Ok-SimE**
3BA0	8	The tailhook position, as a double floating point value (0.0=fully retracted, 1.0=fully lowered).	**?-SimC**	**?-SimC**
3BA8	40	Area used by PFCFSX.DLL for axis input, for optional assignment and calibration in FSUIPC. When the PFC driver is not being used, other programs can make use of these offsets to input axis values directly to FSUIPC, which also can then be assigned in FSUIPC and thence	**Ok-Intl**	**Ok-Intl**

		calibrated. Note that by default FSUIPC assumes that the normal input here is in the range 0–127, and scales it accordingly. For applications supplying a greater range, possibly up to the maximum allowed for joysticks (–16383 to +16383) you can either use the "RAW" option, or, better, let FSUIPC adjust its scaling to suit the range being input. It will adjust this automatically upon seeing the extreme values, or you can preset the scaling using a parameter added to the axis assignments line in the INI file. When the PFC driver is running, application programs or modules can access the raw PFC axis values at these offsets, which are assigned to the hardware as listed below. One 16-bit word is allowed for each (although the PFC axes have a maximum range of 0 to 127). The axes are: 3BA8 0 Aileron 3BAA 1 Elevator 3BAC 2 Rudder 3BAE 3 Quadrant axis 5 3BB0 4 Quadrant axis 3 3BB2 5 Quadrant axis 1 3BB4 6 Left toe brake 3BB6 7 Quadrant axis 6 3BB8 8 Quadrant axis 4 3BBA 9 Quadrant axis 2 3BBC 10 Right toe brake 3BBE 11 Elevator trim 3BC0 12 Aileron trim 3BC2 13 Rudder trim 3BC4 14 Steering tiller 3BC6 15 *not used* There are control flags (to disconnect these axes) at offset 3BC8. Each bit, 2^0 to 2^15 can be set to disconnect the equivalent numbered axis above.		
3BD0	1	*Reserved*		
3BD2	2	This is a 16-bit counter that is incremented each time a FLT file is saved in FS. This applies to flights saved through FS Flights menu, the shortcut key (;), AutoSave, and via the FSUIPC flight saving facilities. The filenames of the saved flights can be read at offset 0400, or (historically) by using the path reading facility at offset 0FF0 and following.	**Ok-SimE**	**N/A**
3BD6	18	Panel failure modes (FS2002 and FS2004 only): one byte flag/control for each of the following "partial panel" gauge modes: 3BD6 ADF (both on FS2004) 3BD7 ASI 3BD8 Altimeter 3BD9 Attitude Indicator 3BDA COM *(both COM1/2 in FSX)* *(Not writable – SimC?)* 3BDB AVIONICS *(was COM2 pre-FSX)* *(Not writable – SimC?)* 3BDC Compass 3BDD Electrical *(new in FSX)* 3BDE Engine (see 0B6B for separate engines) 3BDF Fuel Indicator *(Not writable – SimC?)* 3BE0 Heading Indicator 3BE1 NAV *(both NAV1/2 in FSX)* *(Not writable – SimC?)* 3BE2 NAV *(ditto)* *(Not writable – SimC?)*	**?-SimC** *(See differences)*	**?-SimC & No-SimC+** *(See exceptions)*

		3BE3 Pitot heat 3BE4 Transponder 3BE5 Turn Co-ordinator *(Not writable – SimC?)* 3BE6 Vacuum *(Not writable – SimC?)* 3BE7 VSI		
3BF6	2	SimConnect re-connection count. This is incremented each time FSUIPC4 succeeds in connecting or re-connecting to SimConnect. Re-connection is sometimes needed if SimConnect starves FSUIPC4 of information for longer than the timeout (set by the INI parameter **SimConnectStallTime**, defaulting to 1 second), other than during normal flight loading or menu stoppage times (i.e. between Stop and Start notifications).	**Ok-Intl**	**N/A**
3BF8	2	Number of flap positions not including flaps full up.	**Ok-SimC**	**No**
3BFA	2	Flaps détente increment. The full range of flap movement is 0–0x3FFF (16383). Each détente position or "notch" is spaced equally over this range, no matter what flap angle is represented—a table in the AIR file gives those. To obtain the number of détentes, divide this increment value into 16383 and add 1. For example 2047 (0x7FF) would be the increment for 9 positions.	**Ok-Intl/SimC**	**No**
3BFC	4	Zero Fuel Weight, lbs * 256. This is the aircraft weight plus the payload weight, minus fuel. This changes as the payload is adjusted. Note that this value fluctuates slightly. It is not clear whether this a bug, or an artefact of the physics simulation, but the empty weight (1330) and the payload data (1400) may be used to get a static value.	**?-SimC**	**No**
3C00	256	Full pathname of the current AIR file (in UNC form when applicable *). This is zero padded to fill the 256 bytes available. When this changes the 16-bit counter at 32FC is incremented, so interested programs don't have to keep on reading the whole 256 bytes to check. Note: If you are accessing this from a Gauge, it has been reported that it will not contain the correct aircraft path until FSX loads the gauges completely and begins the update sequence PANEL_SERVICE_PRE_UPDATE ... PANEL_SERVICE_POST_UPDATE * UNC paths are only used if WideFS is in use	**Ok-SimE** *(small difference, see description)*	**No**
3D00	256	Name of the current aircraft (from the "title" parameter in the AIRCRAFT.CFG file).	**Ok-SimC**	**No**
3E00	256	Path of the Flight Simulator installation, down to and including the FS main folder and a following \ character. If the PC is on a Network and WideFS is in use, then if possible the full UNC (universal naming convention) path is given. Examples are: D:\FS2000\ (non-Network) \\MyMainPC\drived\FS2000\ (Network, named PC and named shared drive))	**Ok-Intl**	**No**
3F00	2	To load or save a Flight (.FLT) you first set up the pathname (and optional description) at offset 3F04 below, then write here. Write one of these values: 0 to simply load the specified flight/situation. 1 to save the flight/situation with no description 257 to save the flight/situation with a description Flights are saved in the "My Documents" FS folder. Flights are loaded by default from there too – you don't have to specify a path. If you are Loading a file, please allow time for the file to load before expecting any further meaningful response across the	**N/A**	**Ok-SimC**

		FSUIPC interface. FSUIPC will probably not be able to respond for several seconds even on the fastest machines.		
3F02	2	FLT/STN file loading counter (incremented by FSUIPC whenever the FLT file, as defined at offset 3F04 changes *or* the same FLT is reloaded). If FSUIPC4 re-initialises the SimConnect link at any time (e.g. because of timeout), the flight name provided may change at the same time, due to the way SimConnect operates and FSUIPC obtains the flight names. The value in offset **3BF6** also updates when SimConnect is re-initialised, so this may help distinguish the cause of the change.	**Ok-Intl/SimC**	**N/A**
3F04	252	READ: Pathname of the currently loaded FLT file, excluding the FS main path (see 3E00) if applicable, else the full path, in UNC format if WideFS is in use. This is zero padded to fill the 252 bytes available, or truncated if longer. When this changes (or simply reloaded) the 16-bit counter at 3F02 is incremented, so interested programs don't have to keep on reading the whole 252 bytes to check. WRITE: Write the file name for the FLT+WX file you wish to Load or Save. The name can include the final ".flt" but this will be discarded in any case. You can specify a folder (existing within FS's main folder) for Loading, but files can only be saved to your "My Documents" FS folder. If you give a path for saving, it is discarded. There must be a zero terminator. If you are writing the file, a description can also be specified, following the pathname and its zero terminator. Obviously this is limited by the space available. It must also be terminated by a zero byte, and indicated in the value written to 3F00 above. See 3F00 above for details of actually Loading or Saving the Flight or Situation so identified.	**Ok-SimC**	**Ok-SimC**
4000	512	*Reserved*		
4200	256	FSUIPC's sound playing interface: see the section on this in the main "FSUIPC for Programmers" document.		
4300	7424	*Reserved*		
6000	512	GPS data area—only known offsets listed below:		
6004	4	GPS flags (bits numbered from least significant): 0 not used 1 Active Plan 2 Active Way point 3 Arrived 4 not used 5 Direct To 6 not used 7 Active way point locked 8 Approach loaded 9 Approach Active	**Ok-SimC**	**No**
6010	8	GPS: aircraft latitude, floating point double, in degrees (+ve = N, –ve = S).	**Ok-SimC**	**No**
6018	8	GPS: aircraft longitude, floating point double, in degrees (+ve = E, –ve = W).	**Ok-SimC**	**No**
6020	8	GPS: aircraft altitude, floating point double, in metres.	**Ok-SimC**	**No**
6028	8	GPS: magnetic variation at aircraft, floating point double, in radians (add to magnetic for true, subtract from true for magnetic).	**Ok-SimC**	**No**
6030	8	GPS: aircraft ground speed, floating point double, metres per second.	**Ok-SimC**	**No**

6038	8	GPS: aircraft true heading, floating point double, in radians.	Ok-SimC	No
6040	8	GPS: aircraft magnetic track, floating point double, in radians.	Ok-SimC	No
6048	8	GPS: distance to next waypoint, floating point double, in metres.	Ok-SimC	No
6050	8	GPS: magnetic bearing to next waypoint, floating point double, in radians.	?-SimC	No
6058	8	GPS: cross track error, floating point double, in metres.	Ok-SimC	No
6060	8	GPS: required true heading, floating point double, in radians.	?-SimC	No
6068	8	GPS: track error, floating point double, in radians.	?-SimC	No
6078	8	GPS: aircraft vertical speed	?-SimC	No
6080	1	GPS: previous waypoint valid flag (=0 if not valid)	?-SimC	No
6081	6	GPS: string ID of previous way point, zero terminated	?-SimC	No
608C	8	GPS: previous waypoint latitude, floating point double, in degrees (+ve = N, –ve = S).	?-SimC	No
6094	8	GPS: previous waypoint longitude, floating point double, in degrees (+ve = E, –ve = W).	?-SimC	No
609C	8	GPS: previous waypoint aircraft altitude, floating point double, in metres.	?-SimC	No
60A4	6	GPS: string ID of next waypoint, zero terminated	Ok-SimC	No
60AC	8	GPS: next way point latitude, floating point double, in degrees (+ve = N, –ve = S).	?-SimC	No
60B4	8	GPS: next waypoint longitude, floating point double, in degrees (+ve = E, –ve = W).	?-SimC	No
60BC	8	GPS: next waypoint aircraft altitude, floating point double, in metres.	?-SimC	No
60E4	4	GPS: Next waypoint ETE as 32-bit integer, in seconds	Ok-SimC	No
60E8	4	GPS: Next waypoint ETA as 32-bit integer in seconds, local time	Ok-SimC	No
60EC	8	GPS: Distance to next waypoint, floating point double, in metres	?-SimC	No
60F4	8	GPS: Distance between previous and next waypoints, floating point double, in metres	No-SimC+	No
60FC	4	GPS: Approach mode, as 32-bit integer	?-SimC	No
6100	4	GPS: Approach way point type, as 32-bit integer	?-SimC	No
6104	4	GPS: Approach segment type, as 32-bit integer	?-SimC	No
6108	1	GPS: Approach mode, flag indicating approach waypoint is the runway	?-SimC	No
610C	8	GPS: Course to set (CTS), floating point double, in radians	?-SimC	No
6120	4	GPS: Flight Plan, total number of waypoints, as 32-bit integer	?-SimC	No
6128	4	GPS: Approach way point count, as 32-bit integer	?-SimC	No
6137	5	GPS: Flight plan destination airport ID	?-SimC	No
613C	4	GPS: Approach way point index, as 32-bit integer	?-SimC	No
6140	8	GPS: Approach name	?-SimC	No
6150	4	GPS: Approach transition index, as 32-bit integer. –1 means not valid.	?-SimC	No
6154	8	GPS: Approach transition name	?-SimC	No
615C	1	GPS: Approach is missed flag	?-SimC	No
6160	4	GPS: Approach type	?-SimC	No
6168	4	GPS: Approach time zone deviation, as 32-bit integer	?-SimC	No
616C	4	GPS: Current way point index, starting at 1, as 32-bit integer	Ok-SimC	No
6170	4	GPS: Approach current way point index, as 32-bit integer	?-SimC	No
6190	4	GPS: Time last waypoint was crossed, seconds since Zulu midnight	No-SimC+	No
6198	4	GPS: Destination ETE as 32-bit integer, in seconds	No-SimC+	No
619C	4	GPS: Destination ETA as 32-bit integer, in seconds, local time	No-SimC+	No
61A0	8	GPS: Route total distance, double floating point, in metres	No-SimC+	No
61A8	8	GPS: Estimated fuel burn, double floating point, in gallons	No-SimC+	No
61B0	4	GPS: Time of last update to 61B8 (seconds since Zulu midnight)	No	No
61B8	4	GPS: Count updated every 5 seconds.	No	No
6200	1216	*Reserved*		
66C0	64	**Free for general use**, for example in button or keys programming.		
6700	1632	*Reserved*		
6D60	32	FSUIPC message window title—up to 32 characters including a zero terminator.	N/A	Ok-Intl *(via hack at present)*

		The message window title can be set by the program using it, but as only one such Window is supported only one title is available. The first program writing it ***and then*** a multiline message wins! This only needs doing once, immediately before any multiline messages are sent to 3380.		
6D80	1504	*Reserved*		
7360	12	This are provides an offset method of setting friction values, similar to that offered by the ipc.SetFriction function for Lua plug-ins. This uses the 12 bytes as follows 7360 4 bytes 32-bit float value (FLT) to be written* 7364 1 byte The "class" value, 0-6 7365 1 byte The surface type, 0-24 7366 1 byte The direction, 0 or 1 7367 1 byte The condition, 0-3 7368 4 bytes 32-bit float value (FLT) giving the previous value (read only) The complete 8 bytes, 7360-7367 must either be written at once, i.e. as a single structure, or the FLT value must be written to 7360 last. It is that write which triggers the action. The saved original frictions can be restored by setting the Class value in 7364 to 255 (0xFF) then just writing anything to 7360. The 4 single byte values are as follows: Class: 0 BRAKE 1 WHEEL 2 SCRAPE 3 SKID 4 FLOAT 5 WRUDDER 6 SKI Surface: 0 CONCRETE 1 GRASS 2 WATER 3 GRASS_BUMPY 4 ASPHALT 5 SHORT_GRASS 6 LONG_GRASS 7 HARD_TURF 8 SNOW 9 ICE 10 URBAN 11 FOREST 12 DIRT 13 CORAL 14 GRAVEL 15 OIL_TREATED 16 STEEL_MATS 17 BITUMINUS 18 BRICK 19 MACADAM 20 PLANKS 21 SAND 22 SHALE 23 TARMAC 24 WRIGHT_FLYER_TRA Direction: 0 ROLLING 1 SLIDING Condition: 0 DRY 1 RAIN 2 ICE 3 SNOW		
736C	2	Unsigned 16-bit word giving the distance in nm to the		

		nearest ground AI aircraft		
736E	2	Unsigned 16-bit word giving the distance in nm to the nearest airborne AI aircraft		
7370	3216	*Reserved*		
8000	768	*Reserved for FSUIPC and WideFS internals*		
8300	256	Area in FS2002 and FS2004 reporting and controlling assorted views. Details of those values known follow. This information has been supplied by Matthias Neusinger.		
8320	1	Byte value, the view mode: *In FSX this appears to refer to the last view in which the view mode was changed. It does not necessarily refer to the currently selected view, i.e. the one with focus. The only values provided (referring to standad camera views only) are:* 1=cockpit, 2=virtual cockpit, 4=external, 5=top down	**OK-SimC*** ***(see note)***	**No-SimC+**
832C	2	Zoom setting for selected window in cockpit mode (64 = 1x), read/write	**No-SimC+**	**No-SimC+**
832E	2	Zoom setting for selected window in virtual cockpit mode (64 = 1x), read/write	**No-SimC+**	**No-SimC+**
8330	2	Zoom setting for selected window in tower mode (64 = 1x), read/write	**No-SimC+**	**No-SimC+**
8334	2	Zoom setting for selected window in spot plane mode (64 = 1x), read/write	**No-SimC+**	**No-SimC+**
8336	2	Zoom setting for selected window in top down mode (64 = 1x), read/write	**No-SimC+**	**No-SimC+**
833C	2	Relative direction of spot plane from user aircraft, read/write (in degrees in usual 360 = 65536 format).	**No-SimC+**	**No-SimC+**
8340	4	Distance of spot plane from user aircraft, read/write (in metres * 256).	**No-SimC+**	**No-SimC+**
8345	1	Spot plane transition: gradual is 0, instant if 1. (read/write)	**No-SimC+**	**No-SimC+**
8348	4	Relative altitude of spot plane from user aircraft, read/write (in metres * 256).	**No-SimC+**	**No-SimC+**
83BC	24	View point latitude/longitude/altitude, exactly as at offset 05B0. Read only, FS2004 only.	**No-SimC+**	**No**
83D4	12	View point pitch, bank and heading, in same format as that for the user's aircraft at offset 0578. Read only, FS2004 only.	**No-SimC+**	**No**
8638	4	ActiveSky needed: Ambient turbulence at aircraft (0-1000), 32-bit float		
863C	4	ActiveSky needed: Exported ambient visibility (metres), 32-bit float (-ve if not supported)		
8640	4	ActiveSky needed: Exported precipitation type (0 none, 1 rain, 2 snow, 3 hail), 8-bit integer (Byte)		
8641	4	ActiveSky needed: Exported precipitation rate (0-4), 8-bit integer (Byte)		
8642	4	ActiveSky needed: In cloud flag (non-zero if aircraft is in cloud), Byte		
8670	16	Surface detail request area (P3D4 only): the area is used as follows: 8670 int 32bit Resulting altitude AMSL in metres. Also the "trigger" (see below) 8674 float 32 Latitude, in degrees 8678 float 32 Longitude, in degrees 867C byte Surface type: value (same encoding as in offset 31E8) 867D byte Surface condition value (same encoding as in offset 31EC) 867E word 16bit Flags: 2^15 = 1 when valid result is supplied 2^0 = 1 if surface is a platform 2^1 = 1 if platform is moving		

		To use this, first write the Lat/Lon to the assigned locations, and zero to 867E, then write anything to offset 8670 (the act of writing to it triggers the query to P3D4). The result is available in offsets 8670 and 867E when 867E is non-zero. The action should only take a few milliseconds, so if 867E remains zero for, say, half a second, then the request has failed.		
8680	32	Camera name for use with the following facility (Prepar3D version 2 or later, only).	**N/A**	**OK-Intl**
86A0	24	This provides a way to use the **SimConnect_CameraSetRelative6DOF** function to manipulate the current camera. Please refer to SimConnect SDK documentation for details of this function. Offsets 86A0, 86A4, 86A8, 86AC, 86B0 and 86B4 (ie 6 consecutive 32-bit values) should be written with the 6 **float** parameters for **SimConnect_CameraSetRelative6DOF.** The action is triggered by a write to **86A0**, so either write that parameter last, or, better, write all 24 bytes as one structure, in one Write. If you are using VB remember that hex values like 86A0 will be rendered as FFFF86A0 unless you take steps to ensure no sign propagation. When used with Prepar3D version 2 or later you can also select the specific camera, by name. To do this, before writing the 6 float values to offsets 86A0-86B4 as described above, write a zero byte to offset 8680. This will make it refer to the default camera. Alternatively you can first write the Camera Name to offset 8680-869F, as a zero-terminated ASCIIZ character string of up to 32 bytes (including terminator). This will than make FSUIPC use the function **SimConnect_CameraSetRelative6DofByName** instead of the default camera version. **Note** that, for this extra facility to work, you must be using Prepar3D version 2 or later (earlier versions do not have this function), and FSUIPC must be able to load the newer versions of SimConnectP3D2.DLL (or SimConnectP3D3.dll for P3Dv3).	**N/A**	**OK-Intl**
86E0	2	Traffic Limiter: limit value	**Intl**	**Intl**
86E2	1	Traffic Limiter: target frame rate value	**Intl**	**Intl**
86E3	1	Traffic Limiter: ground preference value	**Intl**	**Intl**
86E4	1	Traffic Limiter: planned airports preference value	**Intl**	**Intl**
86E5	1	Traffic Limiter: airports preference value	**Intl**	**Intl**
86E6	1	Traffic Limiter: near preference value	**Intl**	**Intl**
86E7	1	Traffic Limiter: reserved	**Intl**	**Intl**
9540	64	Current aircraft Profile name (63 chars aSCII + zero term).	**Intl**	**No**
9690	24	Details of the nearest ground AI aircraft, as follows: Byte Size Content 0 4 32-bit integer identifying the aircraft (FSUIPC type ID) 4 4 Latitude as 32-bit float 8 4 Longitude as 32-bit float 12 4 Altitude as 32-bit float 16 2 Heading as 16-bit integer 18 2 Ground speed as 16-bit integer 20 2 Vertical speed as 16-bit integer 22 2 COM1 frequency (1dd.dd where 0xdddd is the value here)		
96A8	24	Details of the nearest airborne AI aircraft, same data as above		
9800	1024	*Used by Wideclient's Lua display control*		
9C00	1024	*Used for the ASE Weather reading facilities*		
A000	4096	*Reserved*		
B000	4096	FSX and beyond: METAR weather reading and writing (i.e. using the special FSX extended METAR strings of up to 2000 characters each):	**Ok-SimC**	**Ok-SimC**

		B000–B7FF = Weather writing area (WRITE) Just write string in FSX METAR format. B800–BFFF = Weather at requested location (READ) For ICAO ID or Lat/Lon written in CCxx area.		
C000	4096	FS2004 style NWI ("New Weather Interface") areas, allowing both local and global weather data to be read and written. C000–C3FF = Interpolated weather at aircraft (READ)* C400–C7FF = Global weather "GLOB" (READ)** C800–CBFF = Weather **writing** area (WRITE) For GLOB or ICAO ID as specified. CC00–CFFF = Weather at requested location (READ) For ICAO ID or Lat/Lon* as specified. The "read at requested location" facility is extended to read the weather at the user aircraft position, by giving an ICAO of '????'. This is the same as giving the aircraft's Lat/Lon, but a bit easier. (Global is read by 'GLOB', as before). Additionally, the ICAO field can be set to "<??>" to get the weather set at the nearest weather station to the user aircraft. The ICAO id of that station is returned in the ICAO field. ** A facility is also provided to force FSX into global-only weather, so that instructor stations, for example, can set weather reliably. This is also automatic for the AWI and FS98 interfaces. * Note that interpolated weather (at aircraft or Lat/Lon) does *not* include local layer information (for visibility, winds and temperature) other than for the layer at the aircraft altitude. The other layers are obtained from the nearest Weather Station.	**Ok-SimC**	**Ok-SimC**
D000 (1st use)	20	<u>Detecting runways in use</u> This facility gives applications a better chance of detecting the runways in use at any selected airport in range (i.e. within 85nm or so of the user aircraft). The Weatherset2 program provided with FSUIPC makes use of this to show any runways currently assigned when AI traffic is active at a weather station selected by ICAO code. This is the interface for this: D000 32-bit signature (see below) D004 4 character ICAO of airport D008 32-bit timestamp D00C 4 bytes giving up to 2 departure runways, format: Number (1 byte), Designator (1 byte) D010 4 bytes giving up to 2 arrival runways, format: Number (1byte), Designator (1 byte) Runway numbers: 1–36 plus 37=N, 38=Ne, 39=E, 40=Se, 41=S, 42=Sw, 43=W, 44=Nw Designators: 0=none, 1=L, 2=R, 3=C, 4=W **Procedure**: 1. Write your signature value (generated by your program, to prevent simultaneous access by others), and the ICAO at the same time. If you use separate writes, write the ICAO first, but use one FSUIPC_Process call. 2. Read the timestamp. This is best done in the same FSUIPC_Process call as the writes. 3. Read the ICAO, timestamp and 8 bytes of runway details until the timestamp changes (or until you time-out). Then check that the ICAO you read is the one you want. If so,	**Ok-Intl** *(via SimC)*	**Ok-Intl**

		then the runway bytes are either zero (if there aren't any known) or they are filled in for you. 4. Write zero to the signature to free the interface for others. If you don't do this, FSUIPC will clear it in any case within about 12-15 seconds of action 1 above. **Notes**: The runways are gleaned from the data in the tables at D040 and D840, described below, but FSUIPC is here looking through ALL the traffic, i.e. all traffic within FS's own 80–90nm radius. It is not restricted it by the user-set radius, nor the smaller ground limit.		
D000 (2nd use)	16	<u>Reading full AI Traffic identity strings</u> The offset area at D000 can also be used to read full AI aircraft data strings. To do this, proceed as follows: 1. Write the selected command, from list below, to D004 (32-bit DWORD) 2. Read the timestamp at D008 (32-bit DWORD) 3. Write the AI id (from the TCAS table, see earlier) to D00C (32-bit DWORD) 4. Write a signature to D000 (32-bit DWORD) It is probably best to do all that in one FSUIPC Process call—in recent versions of WideFS the read should be separated out for you in any case. The order isn't important except that you must write the signature last. If you want to do another within 14 seconds, use the same signature. Use a signature of zero to allow anyone to do the same thing at the same time, but then be aware that your data may not be what you asked for. 5. Wait till the timestamp in D008 changes. 6. Read string result (up to 48 bytes including terminating zero) from D010. The command values available are: 1 = Tail Number 2 = Airline name + Flight number 3 = ATC aircraft type, plus ATC aircraft model * 4 = Aircraft title 5 = ATC aircraft type + last 3 digits of tail number * The aircraft type is one zero-terminated string, and the model is another, following immediately. If either are missing you'll still get the null string (i.e. just the zero terminator). Except for the last case where 3 digits are extracted deliberately (in accordance with ATC practice), none of these strings are likely to be abbreviated, except perhaps any long Aircraft Titles. In other words don't expect the string read in command 2 to be the same as the 14 character version in the TCAS tables—though the beginning and end will be, of course.	**?-Intl/SimC**	**?-Intl**
D040	1920 (96 x 20)	AI ground aircraft additional traffic data. An array of 96 x 20 byte structures as follows: <u>TCAS DATA2</u> 0 BYTE bGateName This is a numeric representation of the gate name, when one is assigned. Otherwise it is zero. The values are as in the BGL, as follows: 0 No name	**Ok-SimC** *(excepting items marked **)*	**N/A**

		1 Ramp parking 2 N Ramp parking 3 NE Ramp parking 4 E Ramp parking 5 SE Ramp parking 6 S Ramp parking 7 SW Ramp parking 8 W Ramp parking 9 NW Ramp parking 10 Gate 11 Dock 12–37 Gate A to Gate Z 1 BYTE bGateType This is a numeric representation of the gate type, when one is assigned. Otherwise it is zero. The values are as in the BGL, as follows: 1 Ramp (GA) 2 Ramp small 3 Ramp medium 4 Ramp large 5 Ramp Cargo 6 Ramp Military Cargo 7 Ramp Military Combat 8 Gate small 9 Gate medium 10 Gate heavy 11 Dock (GA) 2 WORD wGateN This is the gate number, if it is actually numbered. 4 WORD wSpare Reserved for future use 6 short sPitch Aircraft pitch in degrees * 65536 / 360 8 char chICAO[4] Departure airport ICAO Identifier 12 char chICAO[4] Arrival airport ICAO identifier 16 BYTE runway 0 if not assigned for take-off or landing. Else 1-36, or one of 37=N, 38=NE, 39=E, 40=SE, 41=S, 42=SW, 43=W, 44=NW 17 BYTE runwaydes 0 or runway designator: 1=L, 2=R, 3=C, 4=W (water) 18 short sBank Aircraft bank in degrees * 65536 / 360 Note that only those slots marked as valid in the *equivalent* slot in the main TCAS ground tables at E080 are valid here. You should check there first, before using any of this data.		
D840	1920	AI airborne aircraft additional traffic data (same format as the entry for D040). The equivalent main TCAS tables start at F080.		
E000	64	AI ground aircraft tables, housekeeping information as follows: E000 WORD this gives the size of each slot (currently 40) E002 WORD maximum number of slots which will be used (N=96) E004 WORD number of slots used so far (keeps increasing, never decreases) E006 WORD changes count: incremented every time *any* slot is changed E008 BYTE slotChanges[]: an array of N bytes, each one being incremented when relevant slot is changed E068 BYTE[8] option settings for Ground tables. See * below. E07E WORD the FSUIPC offset for the slot with the nearest ground aircraft to the user aircraft. * The 8 bytes at offset E068 contain the current option settings	**Ok-Intl**	**?-Intl** *(For options at E068 only)*

		for Ground aircraft. They are used as follows: Byte 0 Range in nm (0 = unlimited). For ground, this is the range when the user aircraft is airborne. Default is 6nm. Byte 1 Range in nm (0 = unlimited) for Ground aircraft, when the User aircraft is also on the ground. Default is 3 nm. Byte 2 The TCASid option setting, thus: 0 = Tail number 1 = Airline + Flight number 2 = Type 3 = Title 4 = Type + last 3 digits or tail number 5 = Model Byte 3 = 0 normally, giving preference to nearer aircraft when the table is full [1] 0 to give preference to active aircraft. An aircraft is considered inactive if it is in states x80 or x81 (initialising or sleeping). Bytes 4–7 Reserved. Normally most of these options will be as set by the user via the FSUIPC options dialogue or INI file. Applications can change them by writing to these bytes, independently for ground and airborne traffic (the latter at F068). However, FSUIPC will automatically re-instate the user's settings in approximately 20 seconds after the last write to any one of these bytes (airborne or ground). If an application wants to continue with changed settings it must re-write that changed setting at regular intervals. I would suggest using an interval of no more than 5 seconds in order to allow for delays when Networking is being used or FS is under other loads.		
E080	3840 (96 x 40)	AI ground aircraft traffic data. An array of 96 x 40 byte structures as follows: <u>TCAS DATA</u> 0 DWORD id 0 = empty, otherwise this is an FS-generated ID. FSUIPC makes this negative to distinguish FS entries from user added ones. 4 float lat 32-bit float, degrees, –ve = South 8 float lon 32-bit float, degrees, –ve = West 12 float alt 32-bit float, in feet 16 WORD hdg Heading. 360 degrees == 65536 format. Note that this is degrees TRUE, not MAG 18 WORD gs Knots Ground Speed 20 short vs signed feet per minute V/S 22 char idATC[15] Zero terminated string identifying the aircraft. By default this is the Airline & Flt No., or Tail no. 37 BYTE bState a status indication—see list below. 38 WORD com1 the COM1 frequency set in the AI aircraft's radio. (0Xaabb as in 1aa.bb). NOTE that this is set to 0x9999 whilst the aircraft is in "SLEW" mode rather than normal flight mode. The "state" byte provides this information: 0x80 128 Initialising 0x81 129 Sleeping 0x82 130 Filing flight plan 0x83 131 Obtaining clearance 0x84 132 Pushback (back?) 0x85 133 Pushback (turn?)	**Ok-SimC**	**N/A**

<table>
<tr><td></td><td></td><td>
0x86 134 Starting up
0x87 135 Preparing to taxi
0x88 136 Taxiing out
0x89 137 Take off (prep/wait?)
0x8A 138 Taking off
0x8B 139 Departing
0x8C 140 Enroute
0x8D 141 In the pattern
0x8E 142 Landing
0x8F 143 Rolling out
0x90 144 Going around
0x91 145 Taxiing in
0x92 146 Shutting down
</td><td></td><td></td></tr>
<tr><td>F000</td><td>64</td><td>
AI airborne aircraft tables, housekeeping information as follows:

F000 WORD this gives the size of each slot (currently 40)
F002 WORD maximum number of slots which will be used (N=96)
F004 WORD number of slots used so far (keeps increasing, never decreases)
F006 WORD changes count: incremented every time any slot is changed
F008 BYTE slotChanges[]: an array of N bytes, each one being incremented when relevant slot is changed
F068 BYTE[8] option settings for Airborne tables. See * below.
F07E WORD the FSUIPC offset for the slot with the nearest airborne aircraft to the user aircraft.

* The 8 bytes at offset F068 contain the current option settings for Airborne aircraft. They are used as follows:

Byte 0 Range in nm (0 = unlimited). Default is 40nm.

Byte 1 Not used.

Byte 2 The TCASid option setting, thus:

0 = Tail number
1 = Airline + Flight number
2 = Type
3 = Title
4 = Type + last 3 digits or tail number
5 = Model

Byte 3 Not used

Bytes 4–7 Reserved.

Normally most of these options will be as set by the user via the FSUIPC options dialogue or INI file. Applications can change them by writing to these bytes, independently for ground and airborne traffic. However, FSUIPC will automatically re-instate the user's settings in approximately 20 seconds after the last write to any one of these bytes (airborne or ground). If an application wants to continue with changed settings it must re-write that changed setting at regular intervals. I would suggest using an interval of no more than 5 seconds in order to allow for delays when Networking is being used or FS is under other loads.
</td><td>Ok-Intl</td><td>?-Intl
(For options at F068 only)</td></tr>
<tr><td>F080</td><td>3840</td><td>AI airborne aircraft traffic data (same format as the entry for E080)</td><td>Ok-SimC</td><td>N/A</td></tr>
</table>