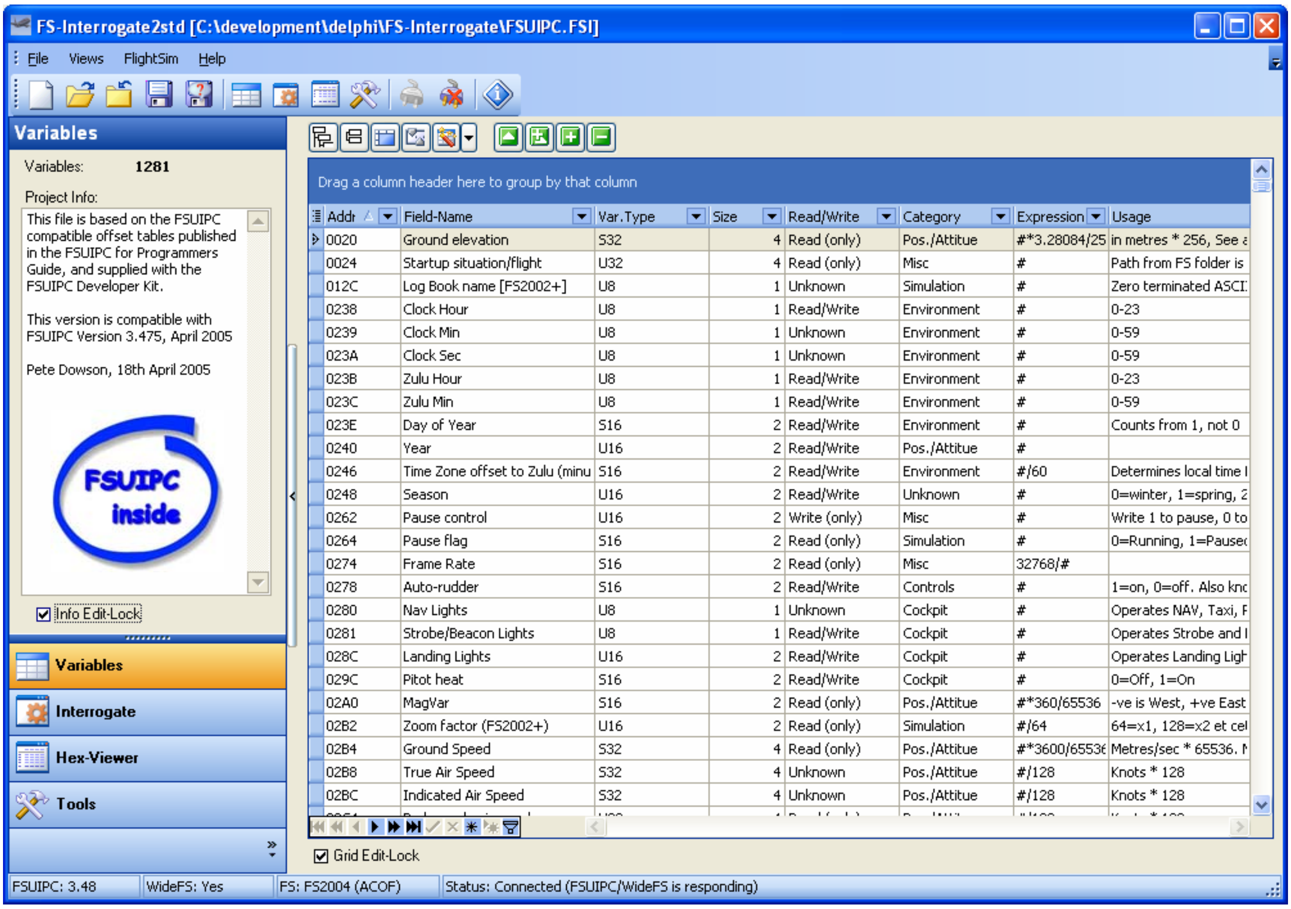
FS-Interrogate 2

Manual


FS-Interrogate2std [C:\development\delphi\FS-Interrogate\FSUIPC.FSI]
File
Views
FlightSim
Help
Variables
Variables
Interrogate
Hex-Viewer
Tools
Drag a column header here to group by that column

Variables: **1281**

Project Info:

This file is based on the FSUIPC compatible offset tables published in the FSUIPC for Programmers Guide, and supplied with the FSUIPC Developer Kit.

This version is compatible with FSUIPC Version 3.475, April 2005

Pete Dowson, 18th April 2005

☑ Info Edit-Lock

Addr	Field-Name	Var.Type	Size	Read/Write	Category	Expression	Usage
0020	Ground elevation	S32	4	Read (only)	Pos./Attitue	#*3.28084/25	in metres * 256, See a
0024	Startup situation/flight	U32	4	Read (only)	Misc	#	Path from FS folder is
012C	Log Book name [FS2002+]	U8	1	Unknown	Simulation	#	Zero terminated ASCI
0238	Clock Hour	U8	1	Read/Write	Environment	#	0-23
0239	Clock Min	U8	1	Unknown	Environment	#	0-59
023A	Clock Sec	U8	1	Unknown	Environment	#	0-59
023B	Zulu Hour	U8	1	Read/Write	Environment	#	0-23
023C	Zulu Min	U8	1	Read/Write	Environment	#	0-59
023E	Day of Year	S16	2	Read/Write	Environment	#	Counts from 1, not 0
0240	Year	U16	2	Read/Write	Pos./Attitue	#	
0246	Time Zone offset to Zulu (minu	S16	2	Read/Write	Environment	#/60	Determines local time I
0248	Season	U16	2	Read/Write	Unknown	#	0=winter, 1=spring, 2
0262	Pause control	U16	2	Write (only)	Misc	#	Write 1 to pause, 0 to
0264	Pause flag	S16	2	Read (only)	Simulation	#	0=Running, 1=Pausec
0274	Frame Rate	S16	2	Read (only)	Misc	32768/#	
0278	Auto-rudder	S16	2	Read/Write	Controls	#	1=on, 0=off. Also knc
0280	Nav Lights	U8	1	Unknown	Cockpit	#	Operates NAV, Taxi, F
0281	Strobe/Beacon Lights	U8	1	Read/Write	Cockpit	#	Operates Strobe and I
028C	Landing Lights	U16	2	Read/Write	Cockpit	#	Operates Landing Ligh
029C	Pitot heat	S16	2	Read/Write	Cockpit	#	0=Off, 1=On
02A0	MagVar	S16	2	Read (only)	Pos./Attitue	#*360/65536	-ve is West, +ve East
02B2	Zoom factor (FS2002+)	U16	2	Read (only)	Simulation	#/64	64=x1, 128=x2 et cel
02B4	Ground Speed	S32	4	Read (only)	Pos./Attitue	#*3600/65536	Metres/sec * 65536. I
02B8	True Air Speed	S32	4	Unknown	Pos./Attitue	#/128	Knots * 128
02BC	Indicated Air Speed	S32	4	Unknown	Pos./Attitue	#/128	Knots * 128

☑ Grid Edit-Lock

FSUIPC: 3.48 | WideFS: Yes | FS: FS2004 (ACOF) | Status: Connected (FSUIPC/WideFS is responding)

The official website for FS-Interrogate can be found at:
www.liljendal.dk/fsinterrogate.htm

pelle@liljendal.dk

Introduction

Having worked for several years as a Developer and for even longer time been interested in Flight Simulation it was only natural I tried to join these two worlds. I got hold of Peter Dowson's FSUIPC and since I am into (Borland) Delphi the first task at hand was to transform the SDK source from the C source Peter wrote into Delphi.

At the time I was under the impression that all data that Microsoft Flight Simulator (MSFS) had to offer to 3rd party developers was physically placed within a "magical" 64 Kb buffer (little did I know it was actually FSUIPC that placed various data within this "buffer"). So in order to better find out what this "buffer" contained I in 2000 wrote a small piece of software that I called FS-Interrogate. Both writing this program and the Delphi port of the SDK code, got me in contact with Peter Dowson and he made me aware of my mistake, however at the same time he saw some potential in FS-Interrogate and made a lot of suggestions as how to improve it. I can't but help being a bit proud that it is still bundled with the FSUIPC SDK (either I did something right or there have not been anybody else trying to do the same thing <G>). Due to work and other interest I've been away from the worlds of MSFS for some time. I haven't been completely away and I have had several sub-versions of FS-Interrogate in various states of Development, however I never got to finish any of those versions.

In 2005 I was once again bitten by "the Flight Simulator bug" and I got myself a copy of MSFS 2004 ACOF (my luck was it was not longer "new", so it was on sale <G>). From earlier contact with Peter I knew many of the shortcomings of the old version of FS-Interrogate (it had no support for 32 bit Floating Point, 64 bit Integers nor Strings). I contacted Peter in order to learn what else could benefit a new version of FS-Interrogate. So with regard to many of the new features added you would have to thank Peter for his suggestions, however don't blame him if I implemented them badly <G>. Anyway here it is, hope you like it.

FS-Interrogate is a complete rewrite and with few/minor exceptions nothing from the original source has been reused. The GUI has had a major (and much needed) facelift, and I hope it will be well received (it has so far <G>). From very early on I decided to spilt it up into a (Free) **Standard Version** and a (payware) **Professional Version**.

Standard Version

This version is completely free and you are allowed to use it on as many PCs as you like whether or not you are simply "toying around", writing freeware applications or writing big/expensive commercial applications. **However if you do write commercial software with the assist of FS-Interrogate the least you could do is to offer me a free version/license for your software ☺**

Beside the ("missing") Export feature the Standard Version will offer the same features as the old version of FS-Interrogate did (to say the least), however the new improved GUI is a lot easier to work with. The aim is that the Standard Version will still be bundled with the FSUIPC SDK, and that it will contain the features needed by the average FS Add-On developer and/or Home-Cockpit Builder

Professional Version

I don't expect to get rich making a Professional Version of FS-Interrogate (there are only so many Developers building Add-Ons for MSFS), however over the years I have had various expenses of which some are directly related to the creation of FS-Interrogate. Also I guess many of the tools/add-ons that I have bought over the years have been written by people who have had a better understanding of FSUIPC through the use of FS-Interrogate, so on some level its only fair I get paid as well !?

As of now I am not 100% sure exactly which features will be put into the Professional version however I have several on the "drawing board":

- Export (as known from FS-Interrogate 1.x).
- Import (Compare 2 FSI-files and import the differences you like).
- Log-File interpretation (makes it easier for Developers to Debug their software using FSUIPC's Log-files).
- Build in Scripting Engine (Beats loading up your development tool, and compile each time you just want to test a simple feature).

At the time of Writing this, not even a single line of code has been written for the Pro-Version, and the Std. Version is still in early Development – however feel free to mail me any requests/suggestions to improve the software.

What is FS-Interrogate?

With the release of MSFS 2000 Microsoft also began making SDKs for 3rd party programmers wanting to write add-ons for MSFS, however the community still wanted to use some of the "old" Add-ons that were originally made for FS95/FS98. These "old" Add-Ons communicated with MSFS via FS6IPC (by Adam Szofran), which didn't run with FS2000. This problem was fixed when Peter Dowson made his FSUIPC since it presented the same data that Adam's FS6IPC did (in the same way) so in one stroke it was possible to use several FS95/98 add-ons together with FS2000. To many FSUIPC is a lot more appealing than using the MSFS SDK directly since Peter is taking care of all the complex work and the rest of us can simply interface with MSFS basically only having to read data from/write date to various offset addresses. *Also to the people (like myself) who is not using C/C++ the SDK is not worth much, however FSUIPC is a lot easier to work with, and today the FSUIPC SDK has support for most development tools thanks to a large community.*

However in order to successfully write add-ons using FSUIPC you'll have to know where you read the various data and how you should interpret these. E.g. the IAS (Indicated Air Speed) is read as a LongInt (32 bit signed Integer) at offset $02BC (0x02BC to C-Developers or &H02BC to VB-Developers) however in order to get it in a useful form (we prefer IAS expressed as Knots) we will have to know that the LongInt read from MSFS should be divided with 128. So if we read the value 34560 we divide it by 128 and get the result 270, hence the IAS is 270 Knots.

Likewise if you want to switch on/off the various Lights you will both have to know that these are located as a SmallInt (16 bit signed Integer) at offset $0D0C, however you'll also have to know which bit to use (e.g. if you want to switch on/off the Beacon Lights you'll have to know these are controlled by bit-2).

The primary objective of FS-Interrogate is to maintain a "list" of all known "variables" (which data is available at which addresses and how should/can we interpret/use these data). ***Lets us ALL use the moment to Thank Peter Dowson and others for keep maintaining this "list", so the rest of us know where to find what !!!.***

The most basic form of these various addresses (in FS-Interrogate regarded as "variables") is an Offset address, a Variable Type (eg. "SmallInt" - being a 16 bit signed Integer) and a descriptive name (e.g. "Frame Rate"). However you will also have to know if you are only able to read, write or both read/write these variables. For instance the IAS at $02BC is read-only hence you can't "set the speed of the plane" by writing data to this address. However you can both read the current state of the lights and change this state by reading/writing a SmallInt from/to offset address $0D0C. As an addition to the features available in FS-Interrogate 1.x you can now indicate (for each single variable) in which version of MSFS this particular variable is supported, if the variable is a part of an array (e.g. there are 96 slots of identical TCAS data-variables), and if the variable is for a particular 3rd party add-on (e.g. Project-Magenta, RC, PMDG and others who are using particular offsets for variables that are only available when using their add-on products), and knowing these addresses is often essential for home cockpit builders.

Due to the (vastly) improved grouping, sorting and filtering of data it is both possible and feasible to place both the "standard variables" and these various 3rd party variables within the same FSI-file. Also if a variable is used differently in various versions of MSFS (for instance it might have been "read/write" in FS95,98,2000, however its only "read" in FS2002,2004) the "***correct way to use FS-Interrogate***" (at least as I see it <G>) is to create 2 separate variables where one is marked as "Read/Write" supported by FS95,98,2000 whereas the other variable is marked as "Read only" and being supported by FS2002,2004. When working with the variables the user would simply setup FS-Interrogate to filter out all, but the variables supported for the platform he/she is using.

The Interrogate Part of FS-Interrogate

Also FS-Interrogate lets you "play with" these various variables, as it is able to display the values read from MSFS and it lets you sent data back to MSFS so that you can see in MSFS what happens when you write certain values to certain addresses. Also it contains various means of detecting changes within the data so you can locate the address associated with certain aspects of the MSFS. For instance you can read data from MSFS, flip a switch in a panel, read a 2nd time, flip the switch back to its original position, read again a 3rd time and then let FS-Interrogate inform you which offset-addresses that were changed during the process (this is the true "Interrogate" part of FS-Interrogate, hence the name). Having an option to "play" with the data also gives a better understanding of how its working and what can be done. A "hands on" approach is often better than "theoretical naked facts" – however FS-Interrogate lets you have both (so now there is no longer any excuse for not writing that killer MSFS Add-On you have always thought about <G>).

Who is it for

Basically FS-Interrogate is for all who want to know what (MSFS) Data they can access through FSUIPC, whether they write Add-on applications for MSFS or for home cockpit builders know want to know which data is available and how to use then (its not easy to impress you best friend if all that happens when you pull that big gear-lever is that you turn on the Altitude Hold of your Autopilot <G>).

FSI-Files (SDI)

FS-Interrogate is build around a **SDI (Single Document Interface)**, which simply means that you are only able to work with one FSI file at the time, however for most users they will only need a single file (being the FSUIPC.FSI bundled with the FSUIPC SDK). Since the very first version of FS-Interrogate these files have been stamped with a “File Structure Version ID” which enables all versions of FS-Interrogate to load older files (however old versions of FS-Interrogate will not be able to read files saved with the new version of FS-Interrogate). Also as a new feature the FSI-files are now stamped with a CRC32 value so that FS-Interrogate will detect faulty files. There is no difference in the file-structure between the Std. and Pro. versions so, FSI-files can freely be interchanged between the two.

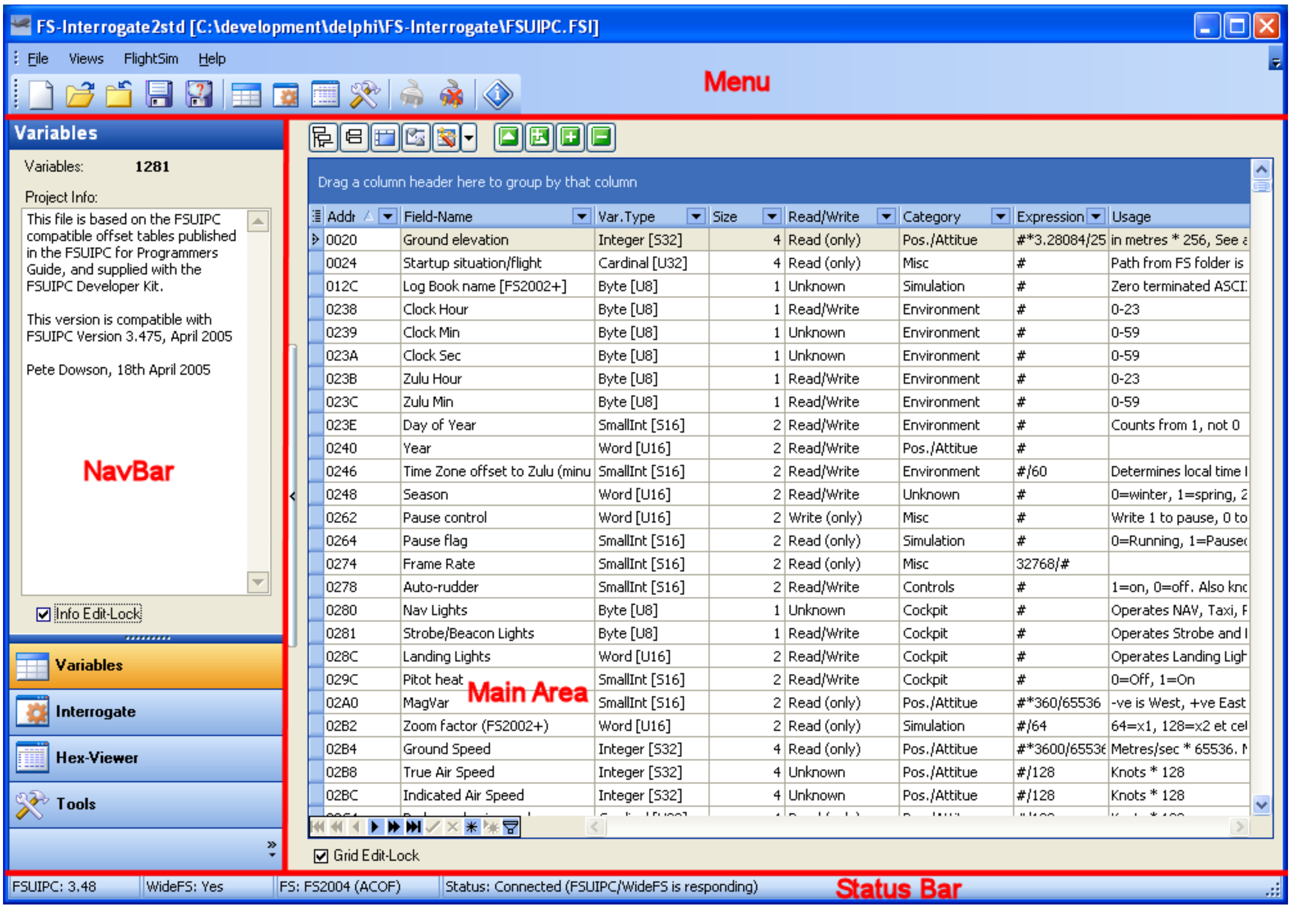
FS-Interrogate will monitor the changes you make to the FSI-file that is currently loaded, so whenever you have unsaved changes the program will warn you if you try to exit the program or load another file without first saving your changes (so unless you don’t want to loose the changes you’d better pay attention to those warnings <G>).

Automatic load the default file (FSUIPC.FSI)

As stated above most users will only need the FSUIPC.FSI file that is bundled with the FSUIPC SDK, hence for the very same reason whenever you start FS-Interrogate it will check if there is a FSUIPC.FSI file at the same location (in the same folder) as the FS-Interrogate EXE-file. If it successfully locate such a file it will be loaded. If you want to disable this function or you would like to automatically load another FSI-file you can do so via the Settings-Dialog (found in the File menu).

The Interface (GUI)

Compared to FS-Interrogate 1.x this new version has had a complete (and much needed) Facelift. The whole interface has been build around a Menu/Navbar, where the screen has been spilt up into 4 segments. At the top you’ll find the main menu, and bottom you’ll find the status bar that will inform you if you are connected to MSFS via FSUIPC/WideFS. In between you will find the NavBar in the left side of the screen and to the right you see the “main area” of the screen (its here you will do most of your work).



Variables: 1281

Project Info:

This file is based on the FSUIPC compatible offset tables published in the FSUIPC for Programmers Guide, and supplied with the FSUIPC Developer Kit.

This version is compatible with FSUIPC Version 3.475, April 2005

Pete Dowson, 18th April 2005

☑ Info Edit-Lock

Drag a column header here to group by that column

Addr	Field-Name	Var.Type	Size	Read/Write	Category	Expression	Usage
0020	Ground elevation	Integer [S32]	4	Read (only)	Pos./Attitue	#*3.28084/25	in metres * 256, See a
0024	Startup situation/flight	Cardinal [U32]	4	Read (only)	Misc	#	Path from FS folder is
012C	Log Book name [FS2002+]	Byte [U8]	1	Unknown	Simulation	#	Zero terminated ASCI
0238	Clock Hour	Byte [U8]	1	Read/Write	Environment	#	0-23
0239	Clock Min	Byte [U8]	1	Unknown	Environment	#	0-59
023A	Clock Sec	Byte [U8]	1	Unknown	Environment	#	0-59
023B	Zulu Hour	Byte [U8]	1	Read/Write	Environment	#	0-23
023C	Zulu Min	Byte [U8]	1	Read/Write	Environment	#	0-59
023E	Day of Year	SmallInt [S16]	2	Read/Write	Environment	#	Counts from 1, not 0
0240	Year	Word [U16]	2	Read/Write	Pos./Attitue	#	
0246	Time Zone offset to Zulu (minu	SmallInt [S16]	2	Read/Write	Environment	#/60	Determines local time I
0248	Season	Word [U16]	2	Read/Write	Unknown	#	0=winter, 1=spring, 2
0262	Pause control	Word [U16]	2	Write (only)	Misc	#	Write 1 to pause, 0 to
0264	Pause flag	SmallInt [S16]	2	Read (only)	Simulation	#	0=Running, 1=Paused
0274	Frame Rate	SmallInt [S16]	2	Read (only)	Misc	32768/#	
0278	Auto-rudder	SmallInt [S16]	2	Read/Write	Controls	#	1=on, 0=off. Also knc
0280	Nav Lights	Byte [U8]	1	Unknown	Cockpit	#	Operates NAV, Taxi, F
0281	Strobe/Beacon Lights	Byte [U8]	1	Read/Write	Cockpit	#	Operates Strobe and I
028C	Landing Lights	Word [U16]	2	Read/Write	Cockpit	#	Operates Landing Ligh
029C	Pitot heat	SmallInt [S16]	2	Read/Write	Cockpit	#	0=Off, 1=On
02A0	MagVar	SmallInt [S16]	2	Read (only)	Pos./Attitue	#*360/65536	-ve is West, +ve East
02B2	Zoom factor (FS2002+)	Word [U16]	2	Read (only)	Simulation	#/64	64=x1, 128=x2 et cel
02B4	Ground Speed	Integer [S32]	4	Read (only)	Pos./Attitue	#*3600/65536	Metres/sec * 65536. N
02B8	True Air Speed	Integer [S32]	4	Unknown	Pos./Attitue	#/128	Knots * 128
02BC	Indicated Air Speed	Integer [S32]	4	Unknown	Pos./Attitue	#/128	Knots * 128

☑ Grid Edit-Lock

FSUIPC: 3.48 | WideFS: Yes | FS: FS2004 (ACOF) | Status: Connected (FSUIPC/WideFS is responding)

Menu

If you have used any other windows program I am pretty sure you will be able to navigate this simple menu <G>. I have added both menu items and speed-buttons for most function so there is several ways to interact with the program (please remark that the keys SHIFT-F1…F4 offers easy access to the various "views" of FS-Interrogate.

Status Bar

Here you can see if FS-Interrogate is connected to MSFS (via FSUIPC and/or WideFS), and if it is you can see which version of FSUIPC you are using.

NavBar

This is here you can switch between the main "Views" for FS-Interrogate. As of now there are 4 different views to choose between (all being described later on).

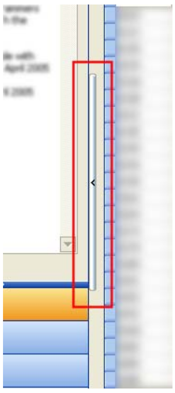
Main Area

Whenever you switch between the "Views" of FS-Interrogate (either via the *NavBar* of via the *Menu*) these various Views will be shown in the "Main Area".

General GUI components

Before I begin describing the various features of FS-Interrogate I feel I better start out describing some of the components that are used throughout FS-Interrogate (so I won't have to describe their usage each time they appear <G>).

Splitter

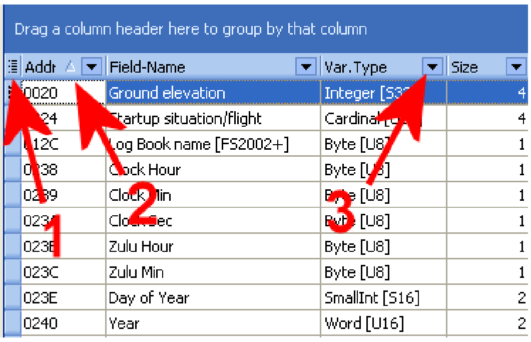


Between then *NavBar* and the *Main Area* there is a *Splitter* that is used to divide the amount of space that you want to use for the *NavBar* and for the *Main Area*. If you click and hold the LMB (Left Mouse Button) on the *Splitter* you can drag it left/right in order to make more room for one of the views. But you can also simply click the *Splitter* in order to completely "remove" the *NavBar*.

You will also find the *Splitter* in other forms and it can be found both in an Vertical and Horizontal version (however they work in the same way (you probably seen one before <G>).

Grids (Tables)

A lot for the data displayed by FS-Interrogate is displayed in Grids. Most of these grids are *Read-Only* so you won't be directly able to change the content of these grids (basically you can only change within the Variable-Grid). The Grids used in FS-Interrogate lets you customize the appearance of the data it shows by enabling you to change the sorting, grouping, filtering and which columns to display at any one time. I really don't have the time to go through each and every thing that is possible with these *Grids* so I suggest you use some time familiarize yourself with these and learn what they can do. However I do have a few pointers that might otherwise be overlooked.



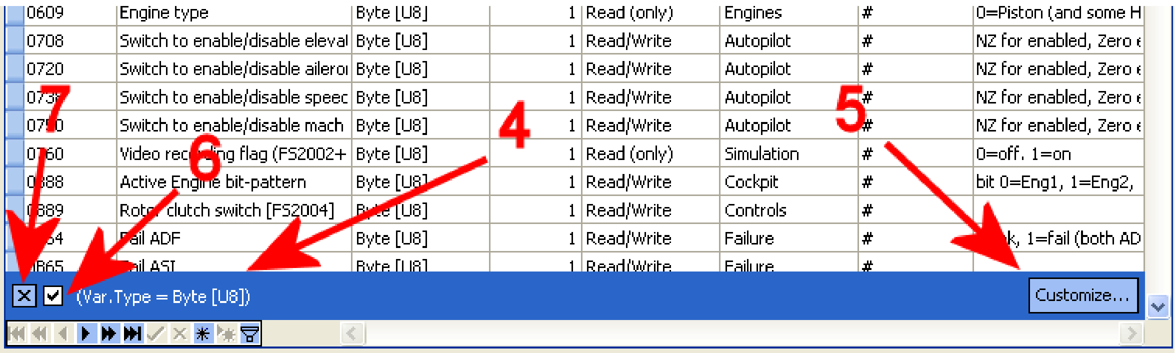
Drag a column header here to group by that column

Addr	Field-Name	Var.Type	Size
0020	Ground elevation	Integer [S32]	4
0024	Startup situation/flight	Cardinal [U32]	4
012C	Log Book name [FS2002+]	Byte [U8]	1
0238	Clock Hour	Byte [U8]	1
0239	Clock Min	Byte [U8]	1
023A	Clock Sec	Byte [U8]	1
023B	Zulu Hour	Byte [U8]	1
023C	Zulu Min	Byte [U8]	1
023E	Day of Year	SmallInt [S16]	2
0240	Year	Word [U16]	2

1: Here you find the *Quick Customization* that lets you pick which columns to show in the Grid.

2: Left Clicking a column header will change the sorting (and by clicking again you can change between Ascending and Descending). If you want to sort using multiple columns first click one column and then hold SHIFT while you click the next column(s)

3: The small "arrow in the box" lets you access the filter option where you can setup the filter so that even though your grid contains many rows of data, only a few will be shown. Whenever a filter is active the lower part of the Grid will contain additional information regarding the active Filter (see next Screenshot).

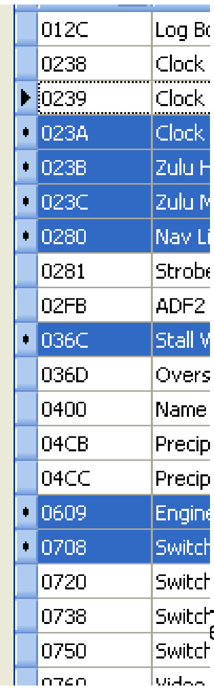


0609	Engine type	Byte [U8]	1	Read (only)	Engines	#	0=Piston (and some H
0708	Switch to enable/disable eleval	Byte [U8]	1	Read/Write	Autopilot	#	NZ for enabled, Zero e
0720	Switch to enable/disable ailero	Byte [U8]	1	Read/Write	Autopilot	#	NZ for enabled, Zero e
0738	Switch to enable/disable speec	Byte [U8]	1	Read/Write	Autopilot	#	NZ for enabled, Zero e
0750	Switch to enable/disable mach	Byte [U8]	1	Read/Write	Autopilot	#	NZ for enabled, Zero e
0760	Video recording flag (FS2002+	Byte [U8]	1	Read (only)	Simulation	#	0=off. 1=on
0888	Active Engine bit-pattern	Byte [U8]	1	Read/Write	Cockpit	#	bit 0=Eng1, 1=Eng2,
0889	Rotor clutch switch [FS2004]	Byte [U8]	1	Read/Write	Controls	#	
64	Fail ADF	Byte [U8]	1	Read/Write	Failure	#	k, 1=fail (both AD
B65	Fail ASI	Byte [U8]	1	Read/Write	Failure	#	

☒ ☑ (Var.Type = Byte [U8])

4: Here you can see the currently active filter. In this case the Grid has been setup only to show variables that are "**Byte [U8]**".
5: By clicking the Customize button you can get into the more advanced filter options (only for the Nerds among you – yes you know who you are <G>).
6: In stead of fully erase the filter you can temporary disable it. So by removing the checkmark the filter will be put out of effect and the Grid will again display all data, and when you again want it to filter (only show "Byte [U8]" variables) you simply check the checkbox again.
7: When you have no need for the filter you simply click the "X" and the filter will be completely removed and the grid will show all data (only the filter is removed, nothing bad will happen to your data <G>).

Also in the previous screen shot you can see the *Navigator* that is visible in most grids (however not all grids have the same buttons). The first 6 buttons is for navigating between the rows. The next 2 are for when you edit data (either accept or discard your changes). The next 2 are for setting at bookmark and the for later returning to it, and finally the last is for setting up a filter.



	012C	Log Bo
	0238	Clock
▶	0239	Clock
•	023A	Clock
•	023B	Zulu H
•	023C	Zulu M
•	0280	Nav Li
	0281	Strobe
	02FB	ADF2
•	036C	Stall W
	036D	Overs
	0400	Name
	04CB	Precip
	04CC	Precip
•	0609	Engine
•	0708	Switch
	0720	Switch
	0738	Switch
	0750	Switch
	0760	Video

Various functions of FS-Interrogate require that you first select some rows. I suggest you use "the Blue Box" that is to the left of each row for this purpose.

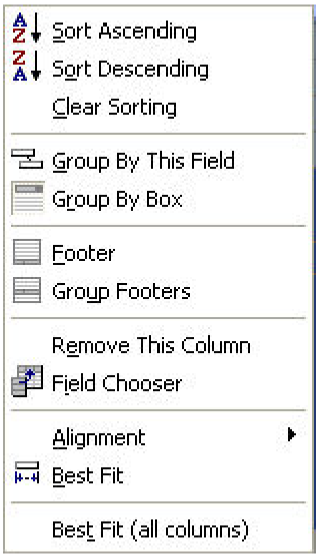
Nothing fancy here, its all "standard windows functionality". Click the first row you want to select, scroll down to the last and hold SHIFT while you click it. Also you can hold down CTRL and then click individual rows in order to either select or deselect them.

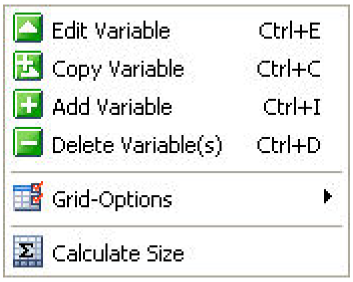
In some cases you might want to consider sorting, grouping or filtering prior to select the rows in order to make it easy to select the rows you need. E.g. if you want to pick all variables in the "Weather" category between "0F64" and "0F70" that are "Words":

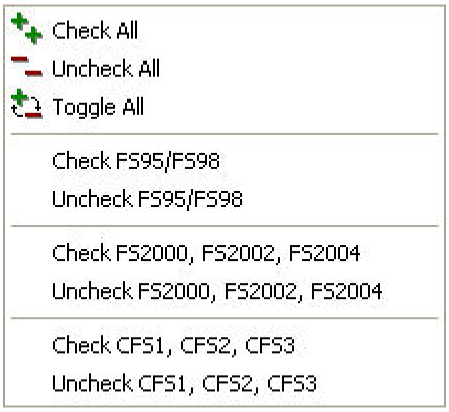
you would set up a filter to show only “Words”, group by category and sort by Address (and now you simply pick those you need).

Context sensitive Menus

When you get 10 minutes where you don’t know what else to do, start right-clicking. Most of the Grids has one Context sensitive menu when you right-click the header and another when you right click the within the Grid. Likewise many of the checkbox-lists has a Context sensitive menu that helps you easy pick many items in one operation – here is a few examples:







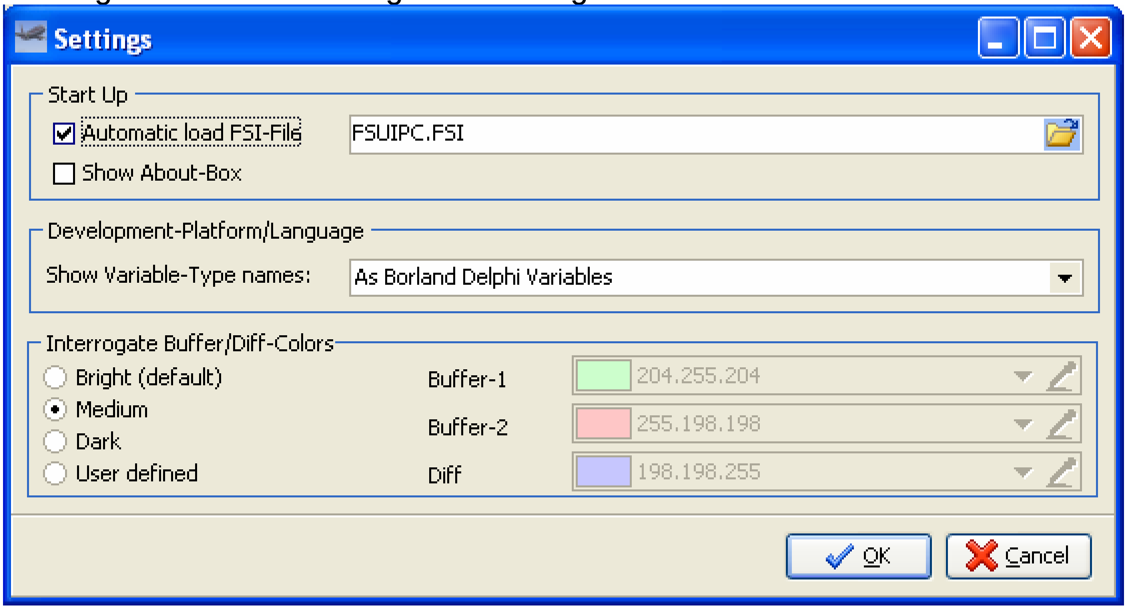
Connection to MSFS (via FSUIPC/WideFS)

You can use FS-Interrogate without being connected to MSFS since it is able to read the FSUIPC.FSI file (that is part of the FSUIPC SDK), and you can then use it to look through the various variables. However you won’t see the full potential of FS-Interrogate until you connect to MSFS and start “playing with the variables” and in the process learn what you can do (and sometimes what you can’t do <G>).

FS-Interrogate will automatically detect if it is able to connect to MSFS either directly using FSUIPS (if it is running on the same PC as MSFS) or via WideFS (if running on another PC). So as long as FSUIPC/WideFS is running and configured correctly, you wont have to do anything in FS-Interrogate (**you can see in the *Status Bar* of FS-Interrogate if its connected or not**). Also if the connection goes down and comes back up (eg. If you shutdown and restart MSFS and/or WideFS), it will automatically reconnect. You can manually disconnect and if you do it will not automatically try to reconnect until you click the Connect button (or pick Connect from the menu).

Settings

In the File-menu you will find the Settings-form that lets you set the various settings that FS-Interrogate is using:



Settings

Start Up

☑ Automatic load FSI-File FSUIPC.FSI

□ Show About-Box

Development-Platform/Language

Show Variable-Type names: As Borland Delphi Variables

Interrogate Buffer/Diff-Colors

○ Bright (default)
● Medium
○ Dark
○ User defined

Buffer-1 🟩 204.255.204

Buffer-2 🟥 255.198.198

Diff 🟦 198.198.255

Start Up

Most users will only have use for the FSUIPC.FSI file that is bundled with the FSUIPC SDK and for that very same reason you can setup FS-Interrogate to automatically load this file each time you run the program. If you do not specify a drive/path (as in the example above where it simply says: "FSUIPC.FSI"), it will look for this file in the same folder as where the FS-Interrogate program is located. If this file does not exist (or at least does not exist in that folder) you won't see any error messages however it will simply not load any file.

Development-Platform/Language

No doubt about it I am a die hard (Borland) Delphi fan and I doubt that I will ever want to want to try any other development tool, however that doesn't mean that the rest of you should be bothered with the Delphi-names for the various variable types (as it was the case with FS-Interrogate 1.x).

In this new version you are able to pick between several different Development-Platforms/Languages. As of now this selection is ONLY used when FS-Interrogate shows the names for the various Variable Types. So for a "32 bit unsigned integer" I as a Delphi Developer sees the name "Integer" and C/C++ developer will want to see "int" and a VB developer would want to see "Long".

These various (different) names can complicate things when a Delphi-, VB- and C/C++ -developer is communicating and for that purpose the default setting is "As they appear in FSUIPC Monitor" (a feature of FSUIPC). These names are short and precise (at least once you learn how to use them <G>). No matter which other setting you might choose these short name are shown in the end of the Variable Type Name.

Your choice of "Development-Platform/Languages" will also affect how HEX values will be shown in the 8-32 bit Integer (and Factored) columns in the Interrogate View. If you select "Borland Delphi" these will be prefixed with "$" if you select "VB" or "VB.Net" they will be prefixed by "&H" and in all other cases they will be prefixed by "0x".

IMPORTANT: Please note that not all Development-Platforms/Languages has support for all the variables used (e.g. VB doesn't have a signed 8 bit value) so that would simply be shown using the "FSUIPC Monitor name" which is "[S8]".

Interrogate Buffer/Diff Colors

Last but not least, the settings form lets you set up the columns colors used for the columns in the Interrogate view (the first 2 are also used by the "3 Scan Locater").

The Different Views (Overview)

FS-Interrogate is build around 4 different "Views" that you can either pick via the *NavBar* or via the *Menu*. Even though some of these views seem to present the same data, **its <u>important</u> that you understand the difference between them** since they have each been optimized for a single purpose: so use the right tool (View) for the right Job. Before describing each in detail I will give a short introduction to them here:

Variable-View (overview)

This is the place to maintain the "list" of variables. So if you need to change any details for the variables or you need to add new variables – this is the place.

Interrogate-View (overview)

The Interrogate-View is the playground where you read data from MSFS and display them. If you do locate a new (not previously described variable) you will be able to add it to the variable list. Also it is in this view you'll find the "3 scan locater" that can help you search where changes are taking place e.g. when you flip a switch on the panel.

Hex-Viewer (overview)

In many ways the Hex-Viewer is just like the Interrogate view since it allows you to read and display data. However in stead of presenting these per variable they are presented in one continuous (64 Kb) buffer presented as HEX values and chars (however moving the cursor around you can also see these data as various variable types).

Tools-View (overview)

This view is "the Black sheep of the Family". In the future this view might get some more sub-views however as of now it simply contains the Scratch-Pad (the place where you e.g. can take notes while learning where which variables are located, and how to use them).

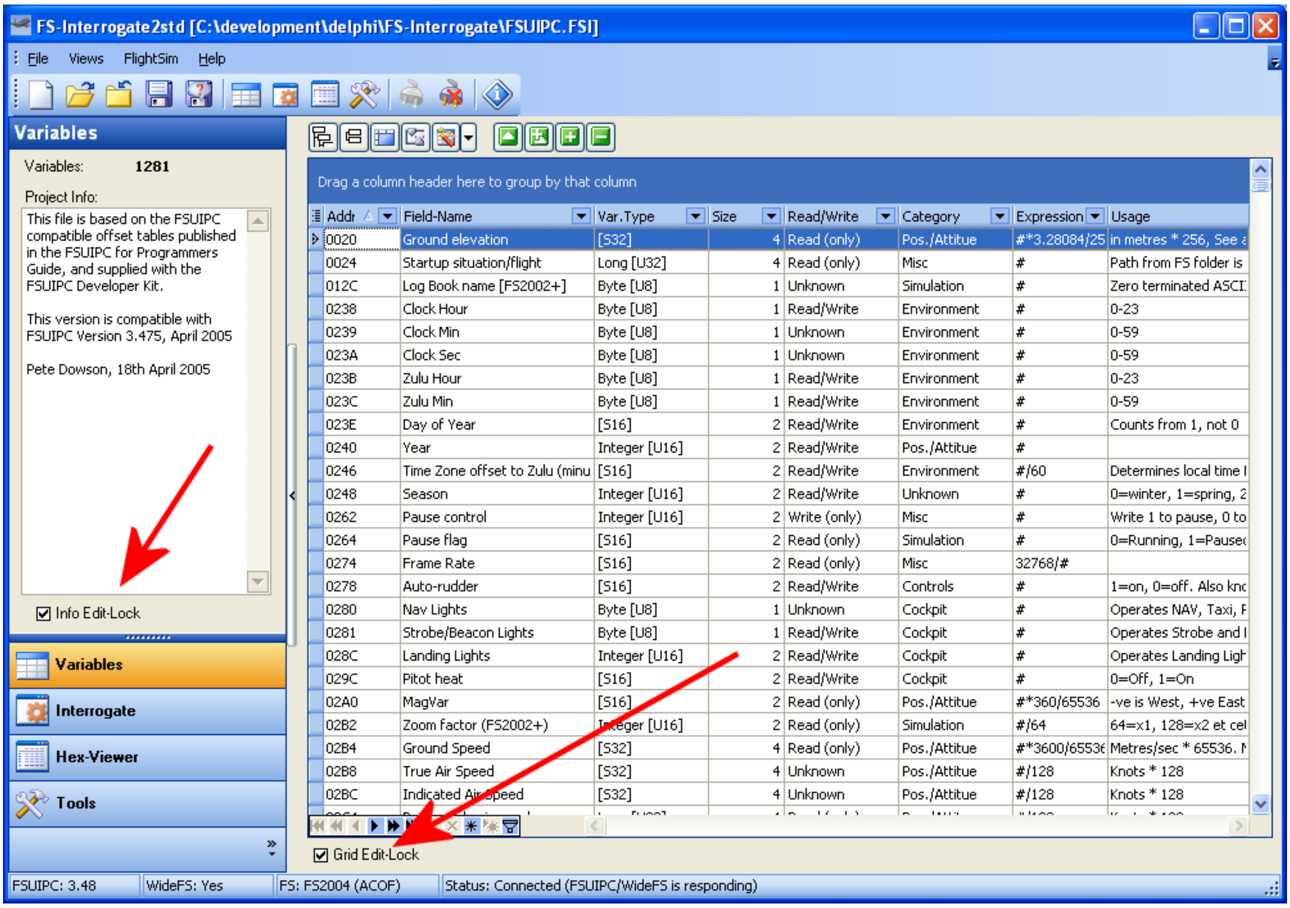
Variable View (Detailed)

This view is the one optimized for maintaining variables, and there are 2 different ways to do it: either directly within the grid or via the dedicated variable edit forms - the latter gives access to ALL the details of the variables, hence some info is not visible within the Grid (e.g. the "See Also variables" which are only accessible through the Variable edit form).

IMPORTANT: Please note that the changes you make within the Variable View (either directly within the Grid or via the Variable Edit Form) are not automatically visible in the Interrogate view until you next time click the "Setup Fields" button of the Interrogate View.

Edit variables Directly within the Grid

To prevent you from making changes by accident, you need to remove the *Lock* (most of us only need to use the FSUIPC.FSI file from the FSUIPC SDK and don't want to make any changes to it). In the NavBar there is an "Info Edit-Lock" (a Check-Box) that needs to be unchecked if you want to edit the *Project Info* (which is simply a text describing the particular FSI-file). Likewise if you want to be able to edit the variable info directly within the Grid you need to uncheck the "Grid Edit-lock" Check-Box, see the "Red-Arrows" below (no don't begin to look up in the skies, its not those Red-Arrows you are to look for now <G>)


FS-Interrogate2std [C:\development\delphi\FS-Interrogate\FSUIPC.FSI]
File
Views
FlightSim
Help
Variables
Variables
Interrogate
Hex-Viewer
Tools
Drag a column header here to group by that column

Variables: 1281

Project Info:

This file is based on the FSUIPC compatible offset tables published in the FSUIPC for Programmers Guide, and supplied with the FSUIPC Developer Kit.

This version is compatible with FSUIPC Version 3.475, April 2005

Pete Dowson, 18th April 2005

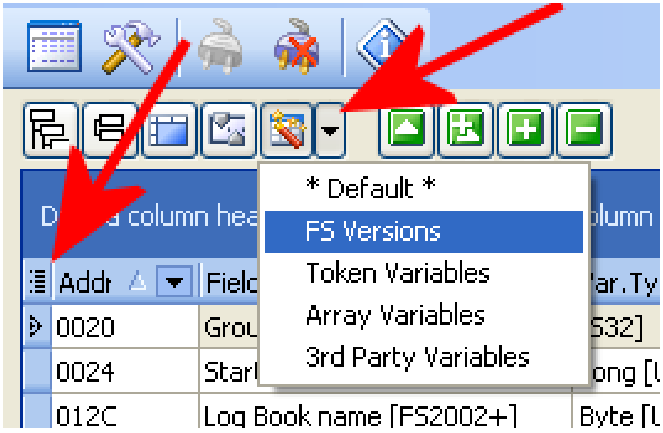
☑ Info Edit-Lock

Addr	Field-Name	Var.Type	Size	Read/Write	Category	Expression	Usage
0020	Ground elevation	[S32]	4	Read (only)	Pos./Attitue	#*3.28084/25	in metres * 256, See a
0024	Startup situation/flight	Long [U32]	4	Read (only)	Misc	#	Path from FS folder is
012C	Log Book name [FS2002+]	Byte [U8]	1	Unknown	Simulation	#	Zero terminated ASCI
0238	Clock Hour	Byte [U8]	1	Read/Write	Environment	#	0-23
0239	Clock Min	Byte [U8]	1	Unknown	Environment	#	0-59
023A	Clock Sec	Byte [U8]	1	Unknown	Environment	#	0-59
023B	Zulu Hour	Byte [U8]	1	Read/Write	Environment	#	0-23
023C	Zulu Min	Byte [U8]	1	Read/Write	Environment	#	0-59
023E	Day of Year	[S16]	2	Read/Write	Environment	#	Counts from 1, not 0
0240	Year	Integer [U16]	2	Read/Write	Pos./Attitue	#	
0246	Time Zone offset to Zulu (minu	[S16]	2	Read/Write	Environment	#/60	Determines local time I
0248	Season	Integer [U16]	2	Read/Write	Unknown	#	0=winter, 1=spring, 2
0262	Pause control	Integer [U16]	2	Write (only)	Misc	#	Write 1 to pause, 0 to
0264	Pause flag	[S16]	2	Read (only)	Simulation	#	0=Running, 1=Pausec
0274	Frame Rate	[S16]	2	Read (only)	Misc	32768/#	
0278	Auto-rudder	[S16]	2	Read/Write	Controls	#	1=on, 0=off. Also knc
0280	Nav Lights	Byte [U8]	1	Unknown	Cockpit	#	Operates NAV, Taxi, F
0281	Strobe/Beacon Lights	Byte [U8]	1	Read/Write	Cockpit	#	Operates Strobe and I
028C	Landing Lights	Integer [U16]	2	Read/Write	Cockpit	#	Operates Landing Ligh
029C	Pitot heat	[S16]	2	Read/Write	Cockpit	#	0=Off, 1=On
02A0	MagVar	[S16]	2	Read (only)	Pos./Attitue	#*360/65536	-ve is West, +ve East
02B2	Zoom factor (FS2002+)	Integer [U16]	2	Read (only)	Simulation	#/64	64=x1, 128=x2 et cel
02B4	Ground Speed	[S32]	4	Read (only)	Pos./Attitue	#*3600/65536	Metres/sec * 65536. N
02B8	True Air Speed	[S32]	4	Unknown	Pos./Attitue	#/128	Knots * 128
02BC	Indicated Air Speed	[S32]	4	Unknown	Pos./Attitue	#/128	Knots * 128

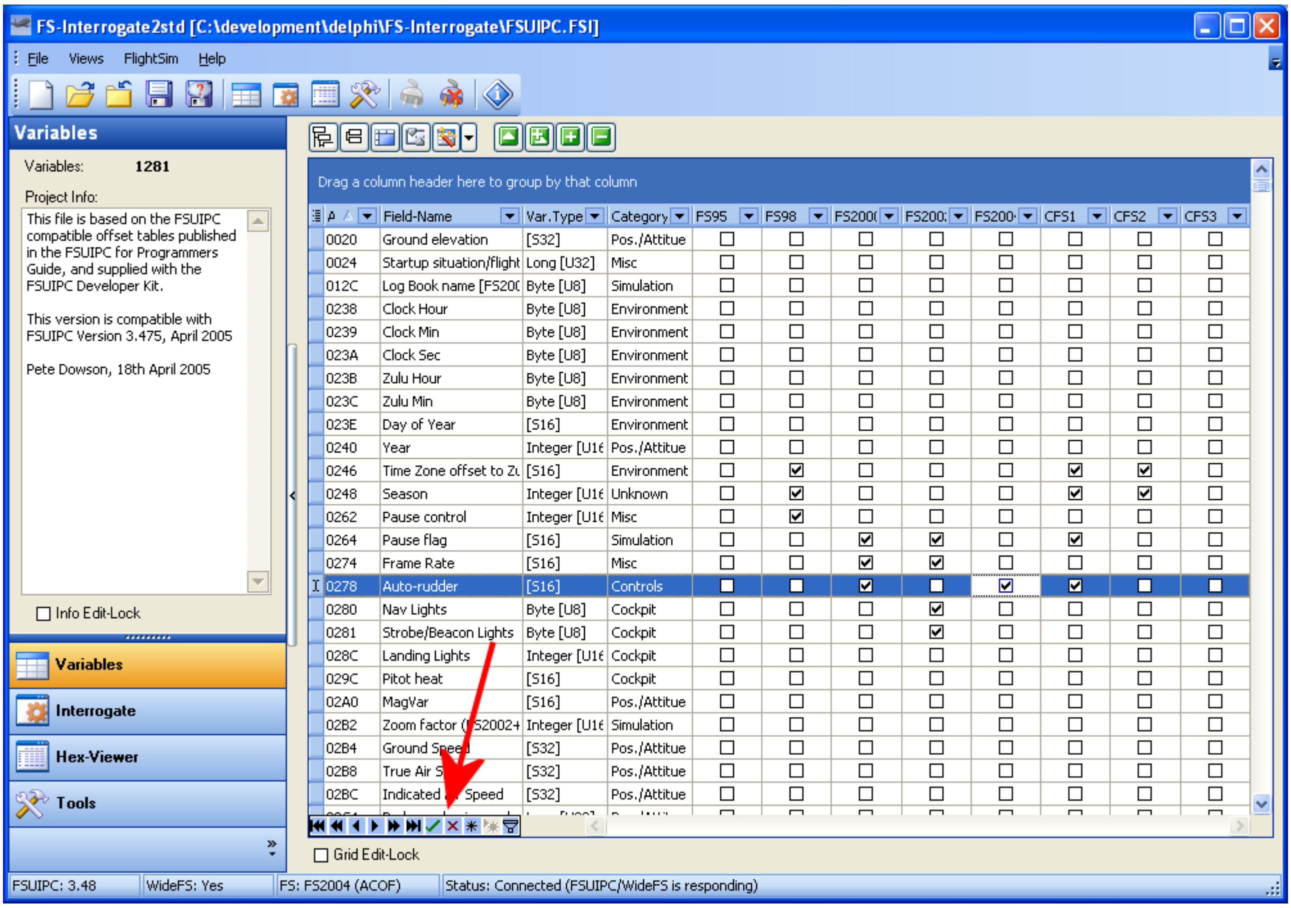
☑ Grid Edit-Lock

FSUIPC: 3.48 | WideFS: Yes | FS: FS2004 (ACOF) | Status: Connected (FSUIPC/WideFS is responding)

One of the new features of FS-Interrogate is that you can per. Variable set checkmarks for which FS-versions that particular variable is valid for. If you have to go through a lot of variables that need changes like these it's a lot faster to edit directly within the grid (instead of having to open a form for each variable that needs to be altered). So first we would need to setup the grid to show the columns with FS-Version info. We can either do this manually by clicking the Quick Customization button of the Grid. This will show a Check-Box list where you can check which columns you want to be present in the Grid.



Another way would be to pick one of the pre-defined Grid-setups that have been created for this very purpose (also visible on the previous Screen-shot).



FS-Interrogate2std [C:\development\delphi\FS-Interrogate\FSUIPC.FSI]
File
Views
FlightSim
Help
Variables
Variables
Interrogate
Hex-Viewer
Tools
Drag a column header here to group by that column

Variables: 1281

Project Info:

This file is based on the FSUIPC compatible offset tables published in the FSUIPC for Programmers Guide, and supplied with the FSUIPC Developer Kit.

This version is compatible with FSUIPC Version 3.475, April 2005

Pete Dowson, 18th April 2005

□ Info Edit-Lock

A	Field-Name	Var.Type	Category	FS95	FS98	FS200(	FS200:	FS200·	CFS1	CFS2	CFS3
0020	Ground elevation	[S32]	Pos./Attitue	□	□	□	□	□	□	□	□
0024	Startup situation/flight	Long [U32]	Misc	□	□	□	□	□	□	□	□
012C	Log Book name [FS20(	Byte [U8]	Simulation	□	□	□	□	□	□	□	□
0238	Clock Hour	Byte [U8]	Environment	□	□	□	□	□	□	□	□
0239	Clock Min	Byte [U8]	Environment	□	□	□	□	□	□	□	□
023A	Clock Sec	Byte [U8]	Environment	□	□	□	□	□	□	□	□
023B	Zulu Hour	Byte [U8]	Environment	□	□	□	□	□	□	□	□
023C	Zulu Min	Byte [U8]	Environment	□	□	□	□	□	□	□	□
023E	Day of Year	[S16]	Environment	□	□	□	□	□	□	□	□
0240	Year	Integer [U16	Pos./Attitue	□	□	□	□	□	□	□	□
0246	Time Zone offset to Zu	[S16]	Environment	□	☑	□	□	□	☑	☑	□
0248	Season	Integer [U16	Unknown	□	☑	□	□	□	☑	☑	□
0262	Pause control	Integer [U16	Misc	□	☑	□	□	□	□	□	□
0264	Pause flag	[S16]	Simulation	□	□	☑	☑	□	☑	□	□
0274	Frame Rate	[S16]	Misc	□	□	☑	☑	□	□	□	□
0278	Auto-rudder	[S16]	Controls	□	□	☑	□	☑	☑	□	□
0280	Nav Lights	Byte [U8]	Cockpit	□	□	□	☑	□	□	□	□
0281	Strobe/Beacon Lights	Byte [U8]	Cockpit	□	□	□	☑	□	□	□	□
028C	Landing Lights	Integer [U16	Cockpit	□	□	□	□	□	□	□	□
029C	Pitot heat	[S16]	Cockpit	□	□	□	□	□	□	□	□
02A0	MagVar	[S16]	Pos./Attitue	□	□	□	□	□	□	□	□
02B2	Zoom factor (FS2002+	Integer [U16	Simulation	□	□	□	□	□	□	□	□
02B4	Ground Speed	[S32]	Pos./Attitue	□	□	□	□	□	□	□	□
02B8	True Air Speed	[S32]	Pos./Attitue	□	□	□	□	□	□	□	□
02BC	Indicated Air Speed	[S32]	Pos./Attitue	□	□	□	□	□	□	□	□

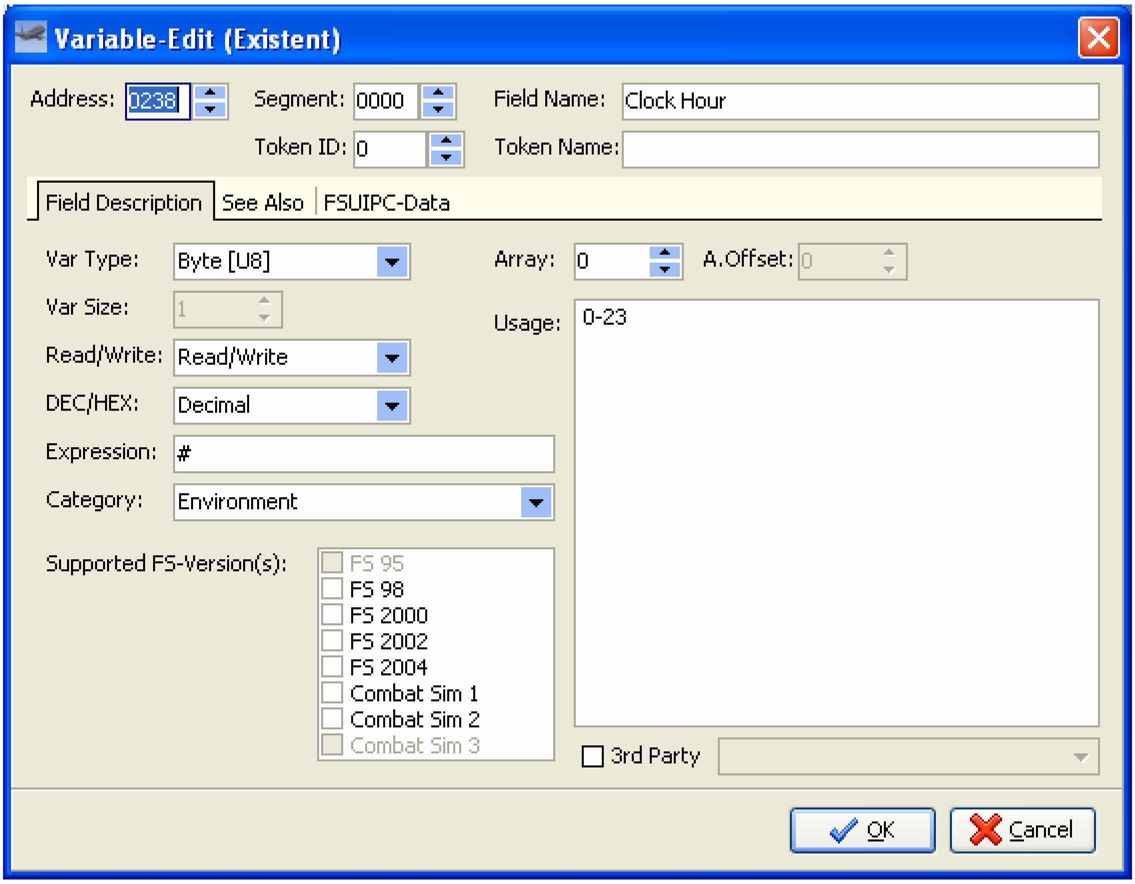
□ Grid Edit-Lock

FSUIPC: 3.48 | WideFS: Yes | FS: FS2004 (ACOF) | Status: Connected (FSUIPC/WideFS is responding)

You are now ready to "click away" and begin setting all those check-marks and change any other information you want to change. Whenever you start changing a particular row, that particular row is being put into *Edit mode* and its up to you if you want to keep those changes or not. So if you by accident change a value click the red "X" (see where the Arrow is pointing) to cancel that change or click the green "Check Mark" if you want to post your changes (whenever you scroll to another Row it will automatic accept your changes).

Edit variables via the Edit Variable Form

If you need access to all the details of a variable, you need to add a new variable or you want to copy a variable then you have to use the dedicated "Variable Form". To access this form either right-click within the Grid and pick the appropriate menu item, click one of the speed buttons above the grid, or simply use one of the keyboard shortcuts (what are shown in the popup menu that appears when you right click within the Grid).



Address: 0238
Segment: 0000
Field Name: Clock Hour
Token ID: 0
Token Name:

Var Type: Byte [U8]
Array: 0
A.Offset: 0
Var Size: 1
Usage: 0-23
Read/Write: Read/Write
DEC/HEX: Decimal
Expression: #
Category: Environment

Supported FS-Version(s):
□ FS 95
□ FS 98
□ FS 2000
□ FS 2002
□ FS 2004
□ Combat Sim 1
□ Combat Sim 2
□ Combat Sim 3

□ 3rd Party

In the Screen Shot above its obvious I am editing a variable (the Caption-Bar of the form tells you whether you are Editing, Copying or Adding a new variable). Compared to the version 1.x of FS-Interrogate (if you knew that one) there are additional info you can enter. In its basic form a “variable” is a description of a particular offset address: what you find at this offset address and how to use it.

Address, Segment and Fieldname

Under normal circumstances your would only need to concentrate on the info that you can access in the offset area between 0000 and FFFF so most of us don’t need to concentrate about the Segment (it should always be 0000). Setting the segment to 0002 simply means you in stead get access to the offset region between 00020000 and 0002FFFF (but these are read only). In this case the segment is 0000 and the Offset address is 0238, and the Field Name tells us that at this location you find the “Clock Hour” (the “Hour” part of the current “Clock”-time).

Token ID/Name

If a particular variable happens to be a Token Variable here you can enter its Token ID and its Token Name (don't ask me what they are fore, probably something Gauge/MS SDK related <G>).

Var Type and Var Size

For this Variable the "Var Type" is a Byte ([U8] means that it's a "Unsigned 8 bit Integer") and therefore the "Var Size" has the value "1". Only when the "Var Type" has been set to be a String/ASCIIZ will you be able to manually specify a size (up to a maximum of 1024, which is the largest size supported by FS-Interrogate for any string – to my knowledge there is non longer than 256 today).

Read/Write

Some variables are read only like for example the Frame Rate (however it would be nice if you simply could write a bigger value here in stead of having to spend thousands on a new faster computer <G>). The "Clock Hour" is as you can see both Readable and Writeable (you can read what the time is now, or you can change the time – within MSFS that is).

DEC/HEX

Most variables are displayed as Decimals however there are a few odd ones that give more meaning when displayed as a HEX value (e.g. those regarding radio frequencies). At address 034E you find the COM1 frequency and it's a [U16] (unsigned 16 bit Integer). Displaying it as a decimal value (e.g. 6471) gives no meaning, however if you convert this decimal value to HEX you in stead get "1947" which in itself doesn't give more meaning. However add a "1" in front of it and a decimal point before the last 2 digits you get "119.47" (which was the COM1 frequency I had dialed in MSFS). *It wouldn't make sense displaying Floating Point- or String- values as HEX (hence if the Var Type is set to one of these types it is forced to use Decimal).*

Expression

You will seldom be able to use the variable directly as its read from MSFS. Therefore FS-Interrogate lets you enter a mathematical expression describing how to "calculate" a (perhaps) more usable value (to be shown in the *Factored* column of the Interrogate View). In this Expression there must be a "#" sign since it will be replaced by the actual value read from MSFS. The variable in the previous Screen Shot (Clock Hour at 0238) is actually one of those variables you should use as it is read, hence the Expression should be set to "#". In effect this means that if the value is read from MSFS as "12" it will also be displayed in the Factored Column as “12” (meaning the “Hour-part” of the Clock is 12).

If you in stead look at variable as “Indicated Air Speed” (at Offset Address 02BC), you will notice that the Expression is “#/128”. This simply means if you want to display IAS displayed as Knots you have to divide the (signed 32 bit Integer) value you read from MSFS with 128. Hence if we read the value 34560 and you divide it by 128 and get the result 270 (meaning the IAS expressed in Knots is 270).

Category

The Category is simply a tool for you to classify the various variables into different Categories for the purpose of assisting you when searching. E.g. you could setup the Variable Grid to group on Category so all “Autopilot” related variables are grouped together, or you could setup a Filter so the Grid only shows variable in the “Fuel” Category.

Supported FS-Versions

As a new feature in FS-Interrogate 2.x you can now per variable indicate in which version of FS this particular variable is supported. E.g. at Offset Address 2DE8 you find “Wind speed at Aircraft” however only supported by FS2004, at 060C you find “Gear Type” however only supported in FS2000 and older.

IMPORTANT: *Since this is a brand new feature of FS-Interrogate I guess it will take some time before the FSUIPC.FSI file becomes fully updated, so at least for a while you shouldn’t take if for granted that the variable is not supported unless there is a checkmark*.

Array, Array Offset

Those of you with a background as software developer knows fully what an Array is however for the rest it can be described as “several alike variables” (I think an Example would be best <G>). If you look at the documentation for the FSUIPC SDK you will learn that the area above D000 is used for TCAS data (AI planes in the Air and on the Ground). Here in this area you (according to the FSUIPC SDK documentation) have 4 tables each holding 96 “slots” of *structs* (records if you have a Delphi background) and each of these holds several variables. Having to describe each of these variables in FS-Interrogate would be more or less a “lifetime full job” (anyone interested? The hours are terrible and there salary is non existent <G>). Since there are 96 slots you would set *Array* to 96 and according to the FSUIPC SDK the size of the *struct* is 40 bytes so you would set *Array Offset* to 40 (meaning there would be 40 bytes between each instant of this particular variable).

Usage

Basically the *Usage* is simply a description telling you more about the variable and often also more about how to use it (so pay attention <G>).

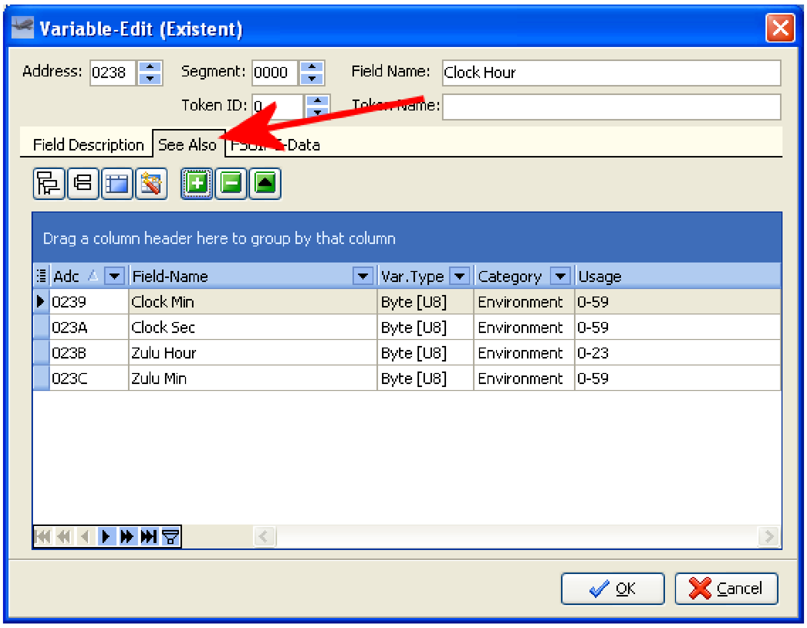
3rd Party

Several 3rd Party developers use various (otherwise unused) offset addresses for their products. In order to be able to distinguish between *The Official Variables* and the *3rd Party Variables* is possible to set a Checkmark which indicate that this particular variable is for a particular 3rd Party product. If you check this checkbox you will also be able to enter a name of this 3rd Party product. You can either manually enter the name or use the Drop-Down facility to simply pick one of those names already used in the FSI-file. *I suggest that you always use this Drop-Down facility since it ensure you always use the (exact) same name, hence it will be easier to filter/sort on this name*.

See Also (Variables)

As a new feature in FS-Interrogate 2.x you can now “link” to other related variables. The Screen Shot to the right shows the variable “Clock Hour” and how it has been “linked” to “Clock Min”, “Clock Sec”, “Zulu Hour” and “Zulu Min”.

The link is “one way” so when you link “Clock Hour” to “Clock Min”, it does not automatic cause “Clock Min” to be linking to “Clock Hour” also (this you will have to do manually if you want the opposite link to be effective as well.



Variable-Edit (Existent)

Address: 0238 Segment: 0000 Field Name: Clock Hour

Token ID: 0 Token Name:

Drag a column header here to group by that column

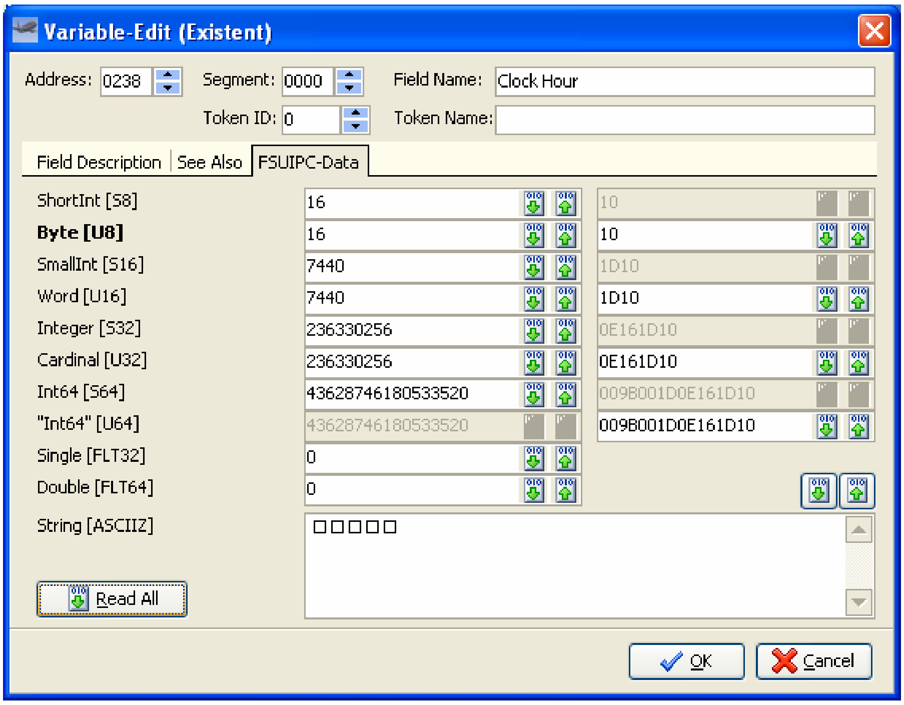
Adc	Field-Name	Var.Type	Category	Usage
0239	Clock Min	Byte [U8]	Environment	0-59
023A	Clock Sec	Byte [U8]	Environment	0-59
023B	Zulu Hour	Byte [U8]	Environment	0-23
023C	Zulu Min	Byte [U8]	Environment	0-59

If you click the speedbutton with the “+” sign you will be presented with a Form containing all variables. In this Form you can select multiple variables to link to, and when you click OK (in the *Browser Grid*) you will return to this view and the *See Also Variables* will be added to the Grid. Clicking the “-“ sign will remove the Variables from the *See Also Grid* (the variable will not be erased, they will simply be removed from this particular variables *See Also Grid*).

The key with “Triangle sign” is normally used when editing something (however then the Triangle is White). This button has a Black Triangle which simply means you can not edit those variables however you can see their details (since you can’t edit the fields they will be shown using a Blue’ish background).

FSUIPC-Data

The 3rd page of the Variable Edit form is used to interact with FSUIPC (read/write data). Even though you on the first page have selected the Var Type this form shows all various Var Types, however the “correct one” is shown in bold (in this example its “Byte [U8]).



Address: 0238 Segment: 0000 Field Name: Clock Hour

Token ID: 0 Token Name:

ShortInt [S8]	16	10
Byte [U8]	16	10
SmallInt [S16]	7440	1D10
Word [U16]	7440	1D10
Integer [S32]	236330256	0E161D10
Cardinal [U32]	236330256	0E161D10
Int64 [S64]	436287461805335520	009B001D0E161D10
"Int64" [U64]	436287461805335520	009B001D0E161D10
Single [FLT32]	0	
Double [FLT64]	0	
String [ASCIIZ]	□□□□□	

In the lower left corner of the form there is a “Read All” button that lets you read data into each of the various fields in one operation (when you select this page the first time it will automatically activate the button so for all fields should be filled with values).

Each of the Fields (for the different Var Types) contains two buttons. The one with the “arrow pointing down” simply means “download” (read the value from FSUIPC) and the one with the “arrow pointing up” means “upload” (write the value to FSUIPC).

The Fields has been arranged into 2 “columns” where the leftmost contains the Decimal-values and the rightmost contains HEX-values. Also you will note that some of the fields are (permanently) grayed out. In the HEX column it wouldn’t give any meaning to distinguish between *signed* and *unsigned* since its simply the most-significant-bit that is used for sign. Likewise it gives no meaning to show the Floating Point values as Hex values separately. Also you will probably note that the Field for a “Unsigned 64 bit Integer” has been disabled in the Decimal column this is simply because Delphi (the development tool I am using) have no support for “64 bit Unsigned Integers”. I doubt this will be a problem for anyone (it is does simply manually convert the displayed HEX-number to a Decimal value in your head or on paper if you are not “Hard-Core Assembler Guru” <G>).

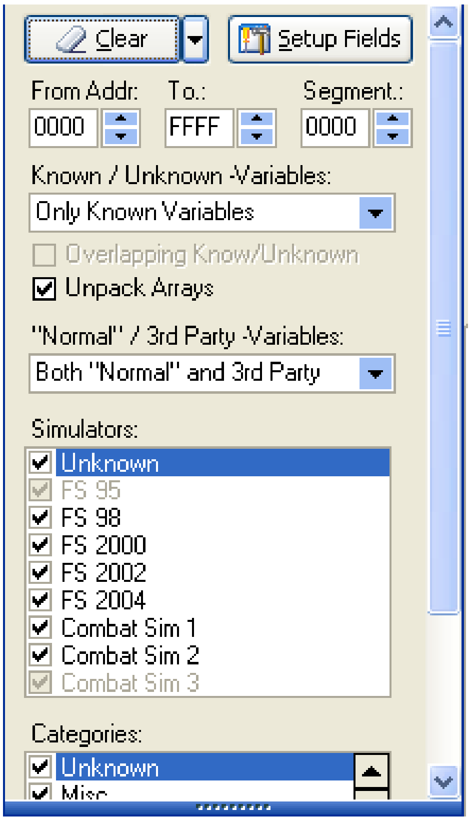
Interrogate View (Detailed)

Before you can use the Interrogate view you need to have it display some variables (*Known* and/or *Unknown* – as explained below). The Gird component used in FS-Interrogate is highly customizable however that comes with a price, it can appear somewhat *slow*, when it is setup to display thousands of rows (since it has to keep track of Grouping, Sorting and Filtering for each single row).

IMPORTANT: Therefore you should <u>ONLY</u> setup the Interrogate View to contain the variables you actually <u>need</u>. The fewer rows of data it contains, the faster the Grid will operate when refreshing (e.g. when you are reading data from MSFS). In this aspect it doesn’t help lowering the number of rows that are displayed, via the usage of the filtering function (it’s the total number of rows within the grid whether these are displayed or not).

Setting up the Interrogate View

As explained above you need to setup the grid to display some variables before you can use the Interrogate View and for this purpose you need to use the Nav-Bar which contains the controls for setting up the Interrogate View (fill the view with the variables you need to *Interrogate*).



From Addr: 0000 To.: FFFF Segment.: 0000

Known / Unknown -Variables: Only Known Variables

□ Overlapping Know/Unknown

☑ Unpack Arrays

"Normal" / 3rd Party -Variables: Both "Normal" and 3rd Party

Simulators:

☑ Unknown
☑ FS 95
☑ FS 98
☑ FS 2000
☑ FS 2002
☑ FS 2004
☑ Combat Sim 1
☑ Combat Sim 2
☑ Combat Sim 3

Categories:

☑ Unknown
☑ Misc

In its default appearance it will setup the Interrogate view to include all *Known* variables in Segment 0000. If you have changed any values you can return to this default setting by clicking the **Clear** button (which will also clear the *Work Area* if the Grid had already been setup to display any rows). The small “drop down button” next to the “Clear” button lets you pick some predefined settings.

When you in the NavBar have setup which fields to display (in effect which fields you will be able to *Interrogate*), you simply click the **Setup Fields** button, which will then begin filling the Grid with the selected variables.

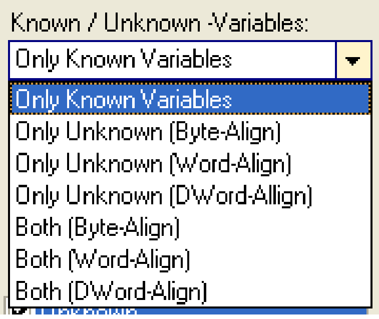
IMPORTANT: Its very important that you understand once you click **Setup Fields**, any changes made elsewhere will not be visible until next time you click "Setup Fields", hence the Variables in the Interrogate View is a *snapshot* of how they appeared when you clicked "Setup Fields". To clarify if you (after you have clicked "Setup Fields") add new variables, edit variables or even delete variables in the Variable View, these changes will not automatically be visible in the Interrogate View (at least not until you click "Setup Fields" again).

Segment, From/To Addresses

As explained previously most of us only need to concentrate about segment 0000 (in effect giving us access to variables in the range 0000 to FFFF). If you only need to interrogate variables in a certain area (e.g. between 0E8A and 0F8D – where you find the Weather related variables) you can use the From/To Addresses to only include variables within this range (both addresses included).

Known/Unknown Variables (and variable Alignment)

Variables defined in the *Variable view* are regarded as *Known* variables, however during the Interrogate process you might also want to *Interrogate* the unknown areas and since the *Interrogate view* only works on a Variable Basis it needs to setup some "dummy variables" occupying the unknown areas, and these are called *Unknown Variables* (since their name/type/usage is *Unknown*).

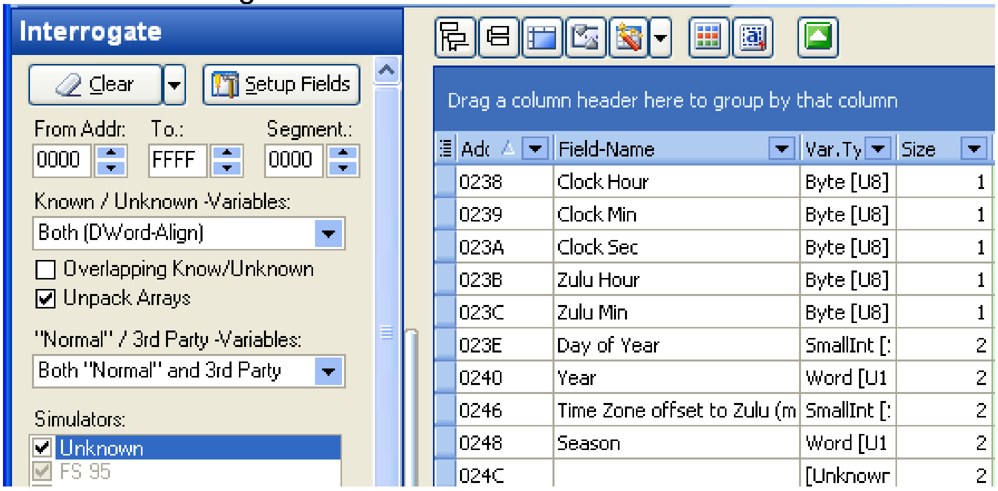

Known / Unknown -Variables:
Only Known Variables
Only Known Variables
Only Unknown (Byte-Align)
Only Unknown (Word-Align)
Only Unknown (DWord-Allign)
Both (Byte-Align)
Both (Word-Align)
Both (DWord-Align)

The Drop Down Box lets you decide if you want to only include *Known*, *Unknown* or *Both (know/unknown)* variables. But it also enables you to setup the *Alignment* (the space between each) of the *Unknown* variables. Setting the alignment to byte (e.g. via "Both (Byte-Align)") would setup *Unknown* Variables at 5000, 5001, 5002. Setting the alignment to Word (e.g. via "Both (Word-Align)") would setup *Unknown* variables at 5000, 5002, 5004 and so on.

As explained previously the Interrogate view is "becoming slower" the more rows it has to display, so picking the DWord-Align in stead of Byte-Align would lower the number of generated *Unknown* Variables by a factor 4. In some cases you would need to use Byte-Alignment however try to combine this with the usage of the To/From Addresses so you can lower the number of Rows the Grid has to display (if this is not feasible then you will simply have to come to terms with the slower Grid <G>).

Overlapping Known / Unknown

Clicking the *Setup Fields* button will only generate *Unknown* variables for the Offset Areas where no *Known* Variable exists, and under normal circumstances it requires that the entire area is free for *Known* Variables. If you have set “Known/Unknown –Variables” to the value “Both (DWord-Allign)” pressing the *Setup Fields* button will generate *Unknown* Variables with a 4 byte Allignment, you could see something like this:



Interrogate

From Addr: 0000 To.: FFFF Segment.: 0000

Known / Unknown -Variables: Both (DWord-Align)

☐ Overlapping Know/Unknown

☑ Unpack Arrays

"Normal" / 3rd Party -Variables: Both "Normal" and 3rd Party

Simulators:

☑ Unknown

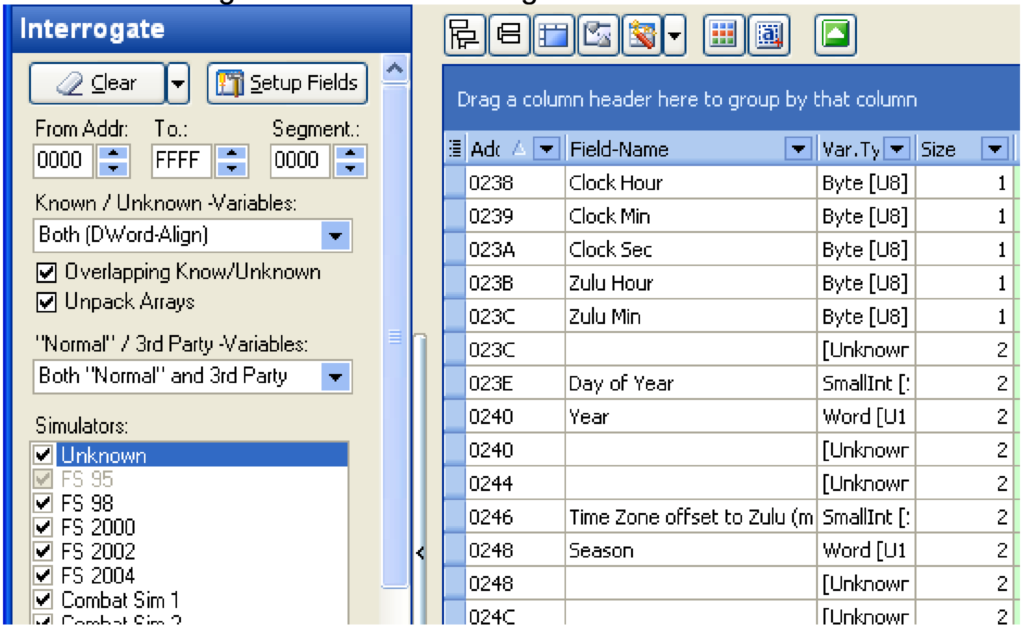
☑ FS 95

Drag a column header here to group by that column

Adc	Field-Name	Var.Ty	Size
0238	Clock Hour	Byte [U8]	1
0239	Clock Min	Byte [U8]	1
023A	Clock Sec	Byte [U8]	1
023B	Zulu Hour	Byte [U8]	1
023C	Zulu Min	Byte [U8]	1
023E	Day of Year	SmallInt [:	2
0240	Year	Word [U1	2
0246	Time Zone offset to Zulu (m	SmallInt [:	2
0248	Season	Word [U1	2
024C		[Unknowr	2

Address 0238 is a 4 byte Aligned-address however all four bytes (0238, 0239, 023A and 023B) are occupied so there will not be generated any variables here. Address 023C is also a 4 byte Aligned-address and within these 4 bytes 023C is occupied by “Zulu Min” and 023E/023F are occupied by “Day of Year” (023D is not occupied at all).

Checking the “Overlapping Known/Unknown” checkbox before clicking *Setup Fields* would in stead generate something like this:



Interrogate

From Addr: 0000 To.: FFFF Segment.: 0000

Known / Unknown -Variables: Both (DWord-Align)

☑ Overlapping Know/Unknown

☑ Unpack Arrays

"Normal" / 3rd Party -Variables: Both "Normal" and 3rd Party

Simulators:

☑ Unknown

☑ FS 95

☑ FS 98

☑ FS 2000

☑ FS 2002

☑ FS 2004

☑ Combat Sim 1

☑ Combat Sim 2

Drag a column header here to group by that column

Adc	Field-Name	Var.Ty	Size
0238	Clock Hour	Byte [U8]	1
0239	Clock Min	Byte [U8]	1
023A	Clock Sec	Byte [U8]	1
023B	Zulu Hour	Byte [U8]	1
023C	Zulu Min	Byte [U8]	1
023C		[Unknowr	2
023E	Day of Year	SmallInt [:	2
0240	Year	Word [U1	2
0240		[Unknowr	2
0244		[Unknowr	2
0246	Time Zone offset to Zulu (m	SmallInt [:	2
0248	Season	Word [U1	2
0248		[Unknowr	2
024C		[Unknowr	2

As you can see there has still not been generated an *Unknown* Variable at 0238 since all 4 bytes (0238, 0239, 023A and 023B) are occupied, however now an *Unknown* Variable has been generated at address 023C since only 3 of the 4 bytes are occupied by *Known* Variables (the same is the case at 0240 where "Year" only occupy 2 of the 4 bytes).

Using the "Overlapping Known/Unknown" checkbox is only useful when "Known/Unknown-Variables" is set to "Both (Word-Align)" or "Both (DWord-Align)".

Unpack Arrays

As of now there are 96 slots of identical TCAS related variables. In stead of defining each of these variables 96 times (in the *Variable View*) a new feature of FS-Interrogate 2.x lets you define these variables as an array (so each variable is defined only once). This reduces the workload when you create the variables, and when you have to maintain the variables. However during the Interrogate Process you would like to see all instances (e.g. you want to see all 96 slots of TCAS data). For this purpose the "Unpack Arrays" checkbox should be checked (it is by default). This will ensure that all 96 instances of the variables appear separately within the grid.

When displaying these Array-variables within the grid, an "Array-Index" number is appended to the Field-Name (e.g. "TCAS-Ground Heading[12]"). Also the Grid contains a special "Ary.Idx" (Array-Index) column that can be enabled via the Grids *Quick Customization* button (this column is not visible by default). Having this extra column lets you filter Array-Variables.

Normal/3rd Party-variables

The default setting is that the Grid shows all variables (whether or not these are defined as 3rd party). However through the "Normal / 3rd Party-Variables" drop-down you can define to only show 3rd party Variables, or to only show "Normal" Variables (Variables not defined as 3rd Party). You can not in the NavBar select Variables for a particular 3rd Party Add-on, however if you pick "Only 3rd Party (Not Normal)" and then use the Grids "3rd P.Name" column to set up a filter that only display variables for a particular 3rd party Add-on (e.g. setting up a filter with "3rd P.Name like %PMDG%" would only show 3rd party variables where "PMDG" is a part of the 3rd Party Name).

Simulators

As a new feature in FS-Interrogate 2.x you can now per variable define in which version of Flight Simulator that particular variable is supported, and setting up the *Interrogate View* you can set it up to only show variables for particular versions of Flight Simulator (right-clicking the checkbox-list opens a context sensitive menu helping you checking what you need). *Since this is a brand new feature of FS-Interrogate, you shouldn't (at least for now) take it for granted that a Variable is only supported for a particular version of FS based on these checkboxes.*

Categories

When a Variable is created its possible to place this variable within (one) of several pre-defined categories. So if you are only going to Interrogate variables that have anything to do with the Fuel System you would only check the “Fuel” Category (right-clicking the checkbox-list opens a context sensitive menu helping you checking what you need).

Interrogate

Having setup the Grid to display what variables you are going to Interrogate (as outlined above) is the first step you need to accomplish. Above the grid you will find a button that lets you select all rows in one operation (this can also be accomplished via the context menu that is displayed when right-clicking within the Grid), but you can naturally also manually select rows (e.g. holding SHIFT/CTRL while picking rows).

IMPORTANT: Most functions within the Interrogate View requires that you select <u>at least one row</u> (in effect one *Known* or *Unknown* Variable) since these functions will <u>only</u> be performed on the selected Variables.

Buffer-1 and Buffer-2

When reading data from MSFS into FS-Interrogate you are able to read these data into 2 separate (64 Kb) buffers. The reason for using 2 separate buffers is to enable you to compare the data read at 2 different points of times (e.g. before and after you do something particular within MSFS). When you enter the *Interrogate View* the Grid will be set up to only show the content of Buffer-1 (the columns with a Green’ish background). You can however either by picking a predefined view or setting up manually which columns to display containing Buffer-2 data as well (displayed with a Red’ish background). *If you don’t like the colors, these can be changed via the Setting form (opened via the File menu) or you can click a button above the Grid to temporary disable the coloring.*

Difference (between Factored Value in Buffer-1 and Buffer-2)

The difference between the Factored values (described below) read into Buffer-1 and Buffer-2 is automatically calculated by FS-Interrogate and if you setup the Grid (as outlined above) you can have this “Diff.Factored” column displayed (using a Blue’ish background). Below the grid you find a Drop-down that lets you decide if the difference (between Factored Buffer-1 and Buffer-2) should be shown “as Value”, “As %” or “As Factor“.

IMPORTANT: If the Factored column contains HEX values (Prefixed by either "$", "&H" or "0x") you can't rely on the value shown in the Difference column (e.g. if "Factored (Buffer-1)" contains the value "0x0900" and "Factored (Buffer-2)" contains the value "0x1745" then the difference will simply appear as "845"). This happens because the process that calculates the differences simply removes all non-numeric chars before turning it into a numeric value (e.g. "0x0900" simply becomes "900").

Reading data ("Read Buffer-1" and "Read Buffer-2)

Before reading data you need to select the variables (*Known* and/or *Unknown*) for which you want to read data (as explained previously) and then you simply click "Read Buffer-1" (or "Read Buffer-2") and the data will be read and displayed (if the columns for holding data are visible).

Progress Bar / Please wait (hiding the Progress Bar)

While the data are being read/the Grid is being updated a small blue message appear above the grid in the right side saying "Please wait…". Also a Dialog holding a progress-bar will appear to show you the progress of the operation. If the Grid is holding many rows (e.g. if setup to show all *Known* and *Unknown* variables in the range 0000-FFFF with a 1 byte alignment) the Progress Bar can appear to stay at 100% for a long time. This is simply because all data have been read and updated however the Grid components needs to inspect these data to see if the update has any influence on (sorting, grouping, filtering) and there is no way to predict neither the time or the progress of this operation (get used to it <G>).

When you are doing Continues Readings (describe next) the Progress Bar will most likely become annoying as it keeps "flashing" (it appears during the Read/Update and disappears before the next read). Also the Progress Bar will appear on top of those variables you try to monitor. For this very same purpose there is a checkbox below the Grid with the text "Hide Progressbar". If you uncheck this checkbox the Progress Bar will not appear during the read/update process however the Blue "Please Wait" will keep appearing (it can't be disabled).

Continues Reading (Monitor changes real-time)

In some cases (e.g. for Developers debugging) you want to keep track of one or several variables as changes take place and for this purpose you can use the "Continues" checkboxes below the Grid (the first/left-most is for Buffer-1 and the last/right-most is for Buffer-2). Only one of these checkboxes can be active at any one time and it will automatically be unchecked if you manually click either "Read Buffer-1", "Read Buffer-2" or "3-Scan Locater". At the time of writing this the delay time (after a scan has been finished/before the next will begin) is fixed to 200 ms.

Displaying data (8 bit, 16 bit … Factored)

The Data read from MSFS is displayed using several different columns: “8 bit”, “16 bit”, “32 bit”, “64 bit”, “Float32”, “Float64” and “Factored” (the first 4 being used for Integers, the next 2 for Floating Point, and the last for the “Calculated value” and Strings).

Normally when Interrogating *Known* Variables FS-Interrogate knows the *Variable Type* for that particular variable so the value read will only be shown in the column for that particular variable type and based on the *Expression* of that variable the calculated value will appear in the *Factored* column.

When Interrogating *Unknown* Variables FS-Interrogate has now idea which column to use (as the Variable Type is Unknown) therefore the read data will appear in all columns (except the *Factored* column). Since there is no *Expression* for *Unknown* variables a *Factored* value can not be calculated, hence it can’t be displayed. As long as there is no Factored Value FS-Interrogate can not calculate and display the difference between Buffer-1 and Buffer-2 since this is the difference between the Factored values.

Below the Grid there is a Drop-Down where you can setup if all Unknown should be regarded as Signed (the value can both be positive and negative) or they should be regarded as Unsigned (only positive values is possible). Using the same Drop-Down you can also force *Known* variables to be shown in all columns (either as signed or as unsigned). Likewise using another Drop-Down (also below the Grid) you can also force either only *Unknown* or both *Unknown* and *Known* variables to be displayed as DEC(imal) or HEX values. If displayed as HEX values these will be prefixed either by “$”, “&H” or “0x” (based on your selection in the Settings Form).

3-Scan Locater

Not all data between Address 0000 and FFFF is known and there can be several reasons for these data to be there - perhaps some 3rd party Developers are using these address for their particular Add-On, however have chosen not to publish this information (I won’t suggest that you “reverse engineer” when you are being prevented by copyrights/licenses). When you are searching for data the 3-Scan Locater can help you locate where changes are taking place when you for instance flip a switch on a panel.

If you have used version 1.x of FS-Interrogate you might recall that the 3-Scan locater function used the same buffers as the Interrogate View did, so when you returned (from the 3-Scan Locater to the Interrogate View) the grid would show the values read during the Scan Process. However in FS-Interrogate 2.x the 3-Scan locater has its own buffers so the values read during the 3, 4 or 5 scan's won't appear in the Interrogate View Grid when you return (this is one of the reasons why the 3-Scan Locater operates faster in FS-Interrogate 2.x than it did in FS-Interrogate 1.x).

IMPORTANT: FS-Interrogate can only locate changes taking place if the Developers has placed these changes into FSUIPC (write data back to FSUIPC) so there will be things that can't be located at all.

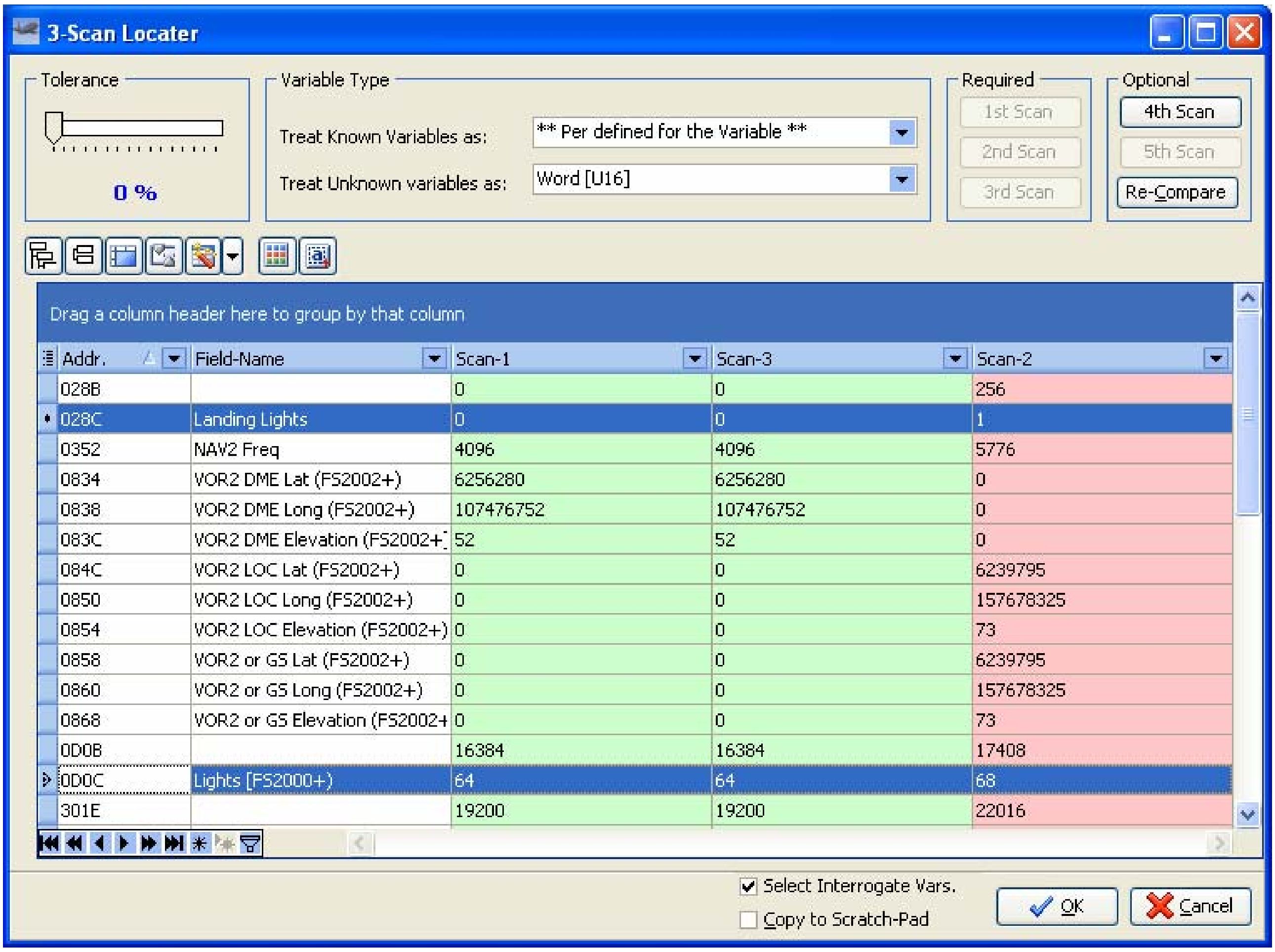
IMPORTANT: The "3–Scan Locater" will only function on those variables that have been selected so normally you would select all rows before clicking the "3-Scan Locater". Also you wouldn't normally know where to search, so it's a good idea to set "Known/Unknown –Variables" (in the NavBar) to "Both (Byte-Align) before clicking the "Setup Fields" button in the NavBar.

The required 3 Scans

When I first made this function in FS-Interrogate 1.x I wasn't very imaginative therefore it was simply called "3-Scan Locater" since you <u>have to</u> perform 3 scans in order to locate the changes you are looking for. The first scan is your "Basis-Ground" (read the data as they appear before you make any changes). Then you perhaps switch on the Landing Lights (please try first in a Default MSFS plane), and then you scan a second time. Before you scan a 3rd time you need to switch off the Landing Lights so that you have returned to your "Basis-Ground".

Comparing data

As soon as you have performed the 3rd scan FS-Interrogate has the data it need to locate any changes. It will begin going through the *Known* and/or *Unknown* variables (that where selected before you entered the "3-Scan Locater") locate the changes. It will look for variables that had the exact same value in the 1st and 3rd scan however had a different value in the 2nd scan. In our case – looking for the landing lights - you should at least see changes at 028C ("Landing Lights") and at 0D0C ("Lights [FS2000+]"):



Tolerance: 0 %

Treat Known Variables as: ** Per defined for the Variable **

Treat Unknown variables as: Word [U16]

Addr.	Field-Name	Scan-1	Scan-3	Scan-2
028B		0	0	256
028C	Landing Lights	0	0	1
0352	NAV2 Freq	4096	4096	5776
0834	VOR2 DME Lat (FS2002+)	6256280	6256280	0
0838	VOR2 DME Long (FS2002+)	107476752	107476752	0
083C	VOR2 DME Elevation (FS2002+	52	52	0
084C	VOR2 LOC Lat (FS2002+)	0	0	6239795
0850	VOR2 LOC Long (FS2002+)	0	0	157678325
0854	VOR2 LOC Elevation (FS2002+)	0	0	73
0858	VOR2 or GS Lat (FS2002+)	0	0	6239795
0860	VOR2 or GS Long (FS2002+)	0	0	157678325
0868	VOR2 or GS Elevation (FS2002+	0	0	73
0D0B		16384	16384	17408
0D0C	Lights [FS2000+)	64	64	68
301E		19200	19200	22016

☑ Select Interrogate Vars.

□ Copy to Scratch-Pad

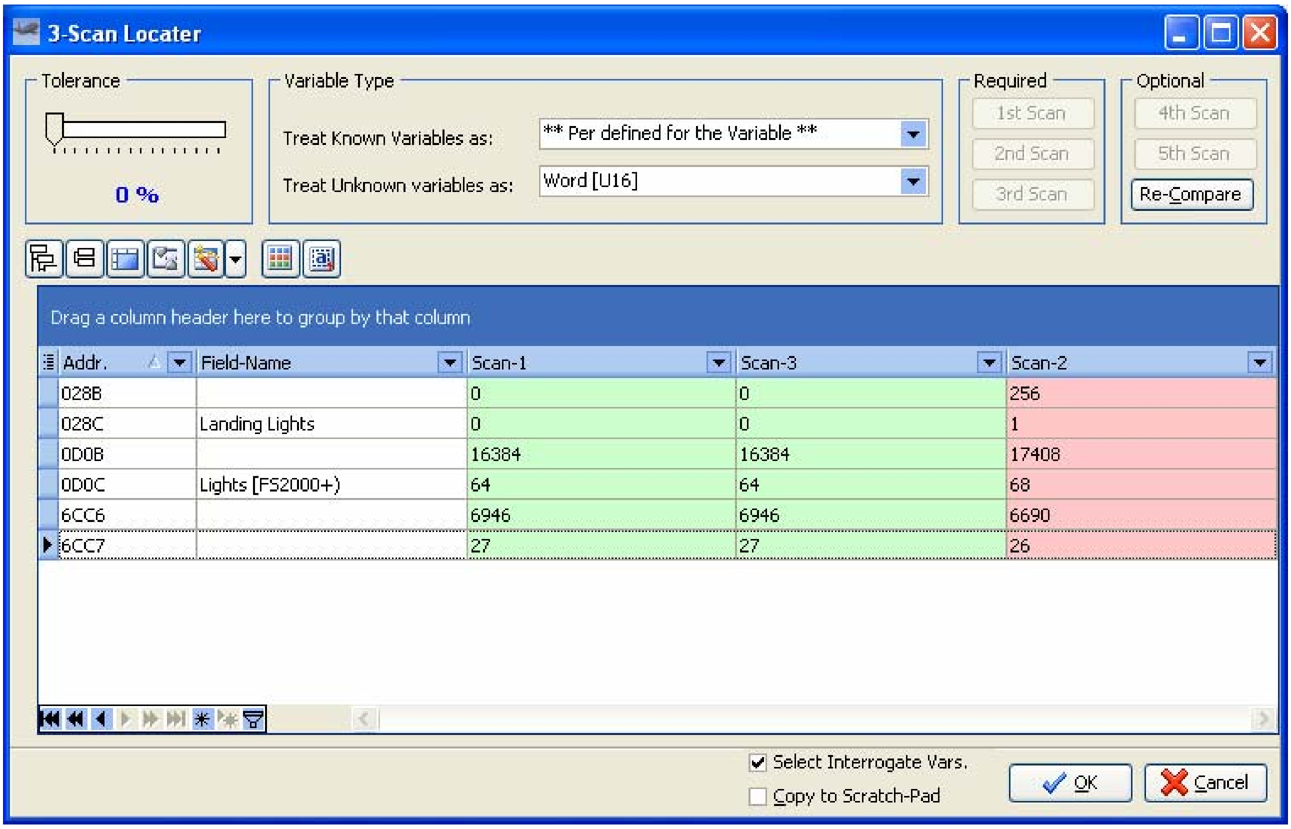
As you can see (in the screenshot above) there are 3 separate columns that shows exactly which data that was read during all 3 scans.

The optional 4th and/or 5th scan

The screenshot above also reveals that the "3 Scan Locater" has also found other variables than those that have anything to do with the Landing Lights. FS-Interrogate don't know anything beyond the data it reads, so even though "VOR2 DME Elevation" has nothing to do with Landing Lights it is shown here simply because it had the value "52" during the 1st and 3rd scan and the value "0" the 2nd scan.
These falsely located changes are called "False Positives" since it is "correct" that they appear (had the same value in 1st and 3rd scan, but a different value during the 2nd scan) however to us they are "False" (we are only interested in Landing Lights related variables). We can never be 100% percent sure that we can get rid of all "False Positives" however we can often eliminate most of them by performing a 4th and perhaps even a 5th scan.

Normally a variable has to have the same value during the 1st and 3rd scan but a different value during the 2nd scan in order to appear in the Grid. If you perform a 4th scan that value has to be identical to the one read in the 2nd scan and if you also perform a 5th scan that value has to be identical with the values read during the 1st and 3rd scan.

In our case you need to switch on the Landing Lights again, and then perform the 4th scan. We decide also to do a 5th scan so please turn of the Landing Lights and then perform the 5th scan.



Tolerance: 0 %

Treat Known Variables as: ** Per defined for the Variable **

Treat Unknown variables as: Word [U16]

Drag a column header here to group by that column

Addr.	Field-Name	Scan-1	Scan-3	Scan-2
028B		0	0	256
028C	Landing Lights	0	0	1
0D0B		16384	16384	17408
0D0C	Lights [FS2000+)	64	64	68
6CC6		6946	6946	6690
6CC7		27	27	26

☑ Select Interrogate Vars.
□ Copy to Scratch-Pad

As you can see all those variables regarding VOR2 has now disappeared either because they didn't have the same value during the 4th scan as they had during the 2nd scan and/or because they didn't have the same value during the 5th scan as they had during the 1st and 3rd scan (but its not 100% bullet proof, if they have had the same values they'd still be on the list – its your job to apply the final logic as what seems right and what seems wrong).

Variable Type

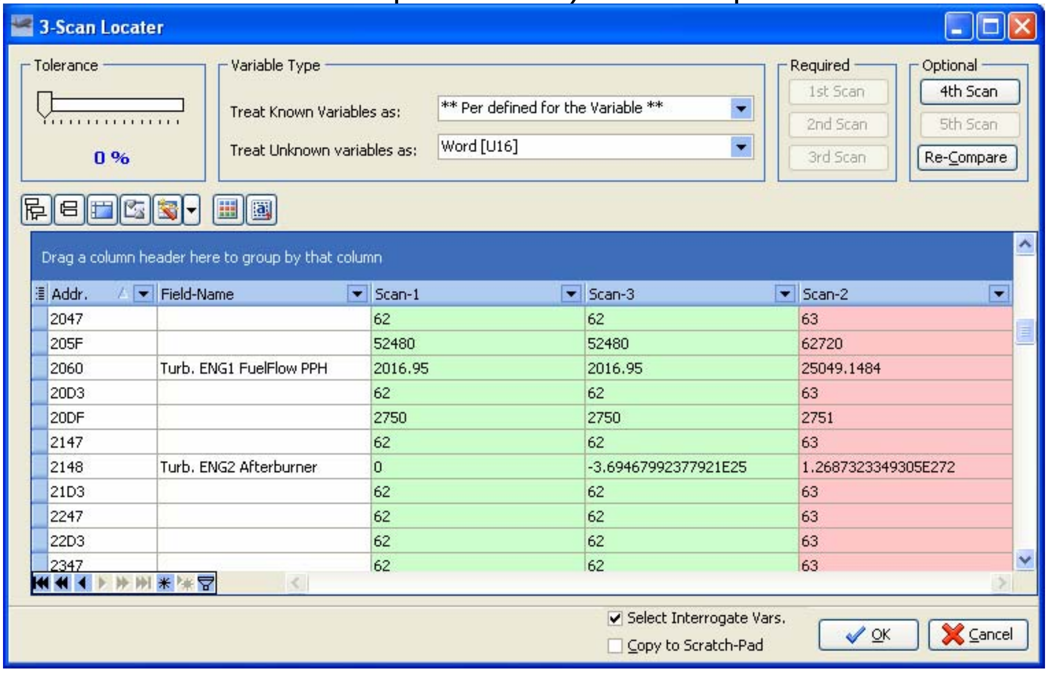
Above the grid you'll see to Drop-Downs that lets you decide which Variable Types the 3 Scan Locater should use during the Compare process. Under normal circumstances (the default behavior) it knows the Variable Type of each and every *Known* variable and will therefore use the one defined for each *Known* Variable, whereas it would simply use "[U16]" (Unsigned 16 bit Integer) for all *Unknown Variables*. Especially with the regard to the *Unknown* Variables its likely that you want to try to treat them as different types (perhaps you don't know if you are looking for a Byte, a 64 bit Floating point or something completely different).

Whenever you have changed the Variable Type you need to compare the data already read during the scans (whether you have performed 3, 4 or 5 scans) and therefore you need to manually click the "Re-Compare" button. However you can change the Variable Type and click this button as many times as it take for you to locate what your are searching for.

Tolerance

If you have used the "3 Scan Locater" from FS-Interrogate 1.x, then the Tolerance will be new to you. In the version 1.x it was more or less impossible to locate engine related data since these would never have the EXACT same value during the 1st and 3rd scan (the values keeps varying).

To demonstrate the Tolerance I have started the engines on my plane (PMDG's B747-400 is my new Favorite <G>) and I let the engine settle on Idle before performing the 1st scan. Then I moved the Throttle to max (only Engine 1 started) and let it climb for a bit of time before performing the 2nd scan. Then finally I moved the throttle back to idle and waited a couple of seconds (to let the engine have some time to spool down) before I performed the 3rd scan:



Tolerance: 0 %

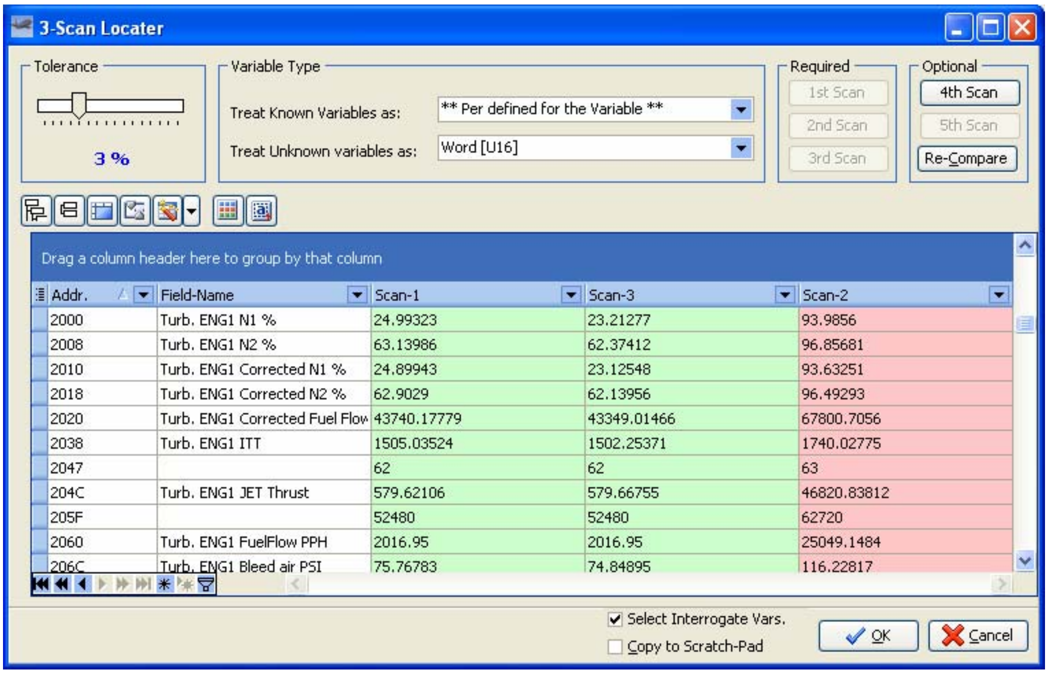
Variable Type
Treat Known Variables as: ** Per defined for the Variable **
Treat Unknown variables as: Word [U16]

Drag a column header here to group by that column

Addr.	Field-Name	Scan-1	Scan-3	Scan-2
2047		62	62	63
205F		52480	52480	62720
2060	Turb. ENG1 FuelFlow PPH	2016.95	2016.95	25049.1484
20D3		62	62	63
20DF		2750	2750	2751
2147		62	62	63
2148	Turb. ENG2 Afterburner	0	-3.69467992377921E25	1.2687323349305E272
21D3		62	62	63
2247		62	62	63
22D3		62	62	63
2347		62	62	63

☑ Select Interrogate Vars.
□ Copy to Scratch-Pad

It appears that only the "Turb. ENG1 FuelFlow PPH" changed during my "engine warm up", however try setting the Tolerance to other settings (beside the default 0%) and each time click the Re-Compare button:



Tolerance: 3 %

Treat Known Variables as: ** Per defined for the Variable **

Treat Unknown variables as: Word [U16]

Addr.	Field-Name	Scan-1	Scan-3	Scan-2
2000	Turb. ENG1 N1 %	24.99323	23.21277	93.9856
2008	Turb. ENG1 N2 %	63.13986	62.37412	96.85681
2010	Turb. ENG1 Corrected N1 %	24.89943	23.12548	93.63251
2018	Turb. ENG1 Corrected N2 %	62.9029	62.13956	96.49293
2020	Turb. ENG1 Corrected Fuel Flow	43740.17779	43349.01466	67800.7056
2038	Turb. ENG1 ITT	1505.03524	1502.25371	1740.02775
2047		62	62	63
204C	Turb. ENG1 JET Thrust	579.62106	579.66755	46820.83812
205F		52480	52480	62720
2060	Turb. ENG1 FuelFlow PPH	2016.95	2016.95	25049.1484
206C	Turb. ENG1 Bleed air PSI	75.76783	74.84895	116.22817

☑ Select Interrogate Vars.

□ Copy to Scratch-Pad

As you can see now there is a lot of values that now qualify (N1, N2, Corrected N1/N2, Corrected FF and so on). Also by examining the values in the columns Scan-1 and Scan-3 you can verify that these values are not exact the same.

I found what I searched for, Now what ?

Hex-Viewer (Detailed)

Tools-View (Detailed)

Scratch-Pad (Take notes as you go along)