

These gauges are provided as an aid on how to interact with XMLTools classes.

As they are totally programmed in XML, the complete code can be inspected and, of course, adapted to each user's custom needs.

They also contain advanced techniques used in XML scripts that are briefly detailed in the *Comment* section of each example.

Examples and utilities included in v2.01

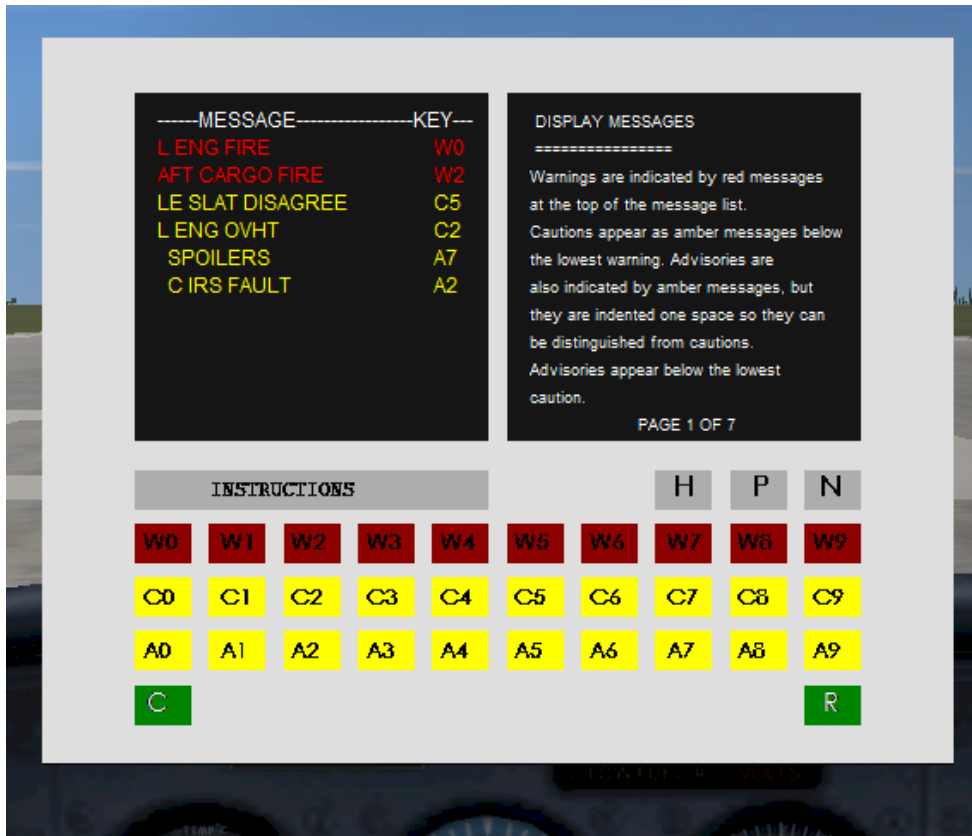
- XMLVarsExample.xml (example) page 2
- ReloadPanels.xml (example) page 4
- LocalVarsLogger.xml (utility) page 4
- EventHandler.xml (example) page 9
- XMLEventsLogger.xml (utility) page 9
- TableAndSimExample.xml (example) page 12
- KeysHandler.xml (example) page 15
- FlightDataRecorder.xml (example) page 17
- SaveAndLoadInitValues.xml (example) page 18
- SaveAndLoadFlightPlan.xml (snippet) page 18
- FDRforGoogleEarth.xml (example) page 18

Installation

To install the gauges in FSX, simply copy XMLTools folder inside FSX\Gauges subfolder.

XMLVARS class examples

- XMLVarsExample.xml



This gauge partially simulates the Crew Alert System of a Boeing 757 aircraft.

Messages are displayed initially in order of importance and then in order of occurrence. Warnings show first, followed by Cautions and Advisories. A total of 30 events (10 per category) can be triggered/canceled by clicking on the squares identified by W/C/A capitals.

Complete instructions on how to operate are shown on screen when the gauge is first displayed, or by clicking on the INSTRUCTIONS bar.

How to install

Edit the `panel.cfg` file of an aircraft of your choice, and add the following:

- In [Window Titles] section:

```
windownn=XMLVarsExample
```

- In [Window~~nn~~] section:

```
size_mm=245,270
window_size_ratio=1.000 (variable, depending on display resolution)
position=1 (variable, user choice)
visible=0 (could be 1 if starting in 2D mode)
ident=10080 (variable, user choice)
gauge00=XMLTools!XMLVarsExample, 0,0,245,270
```

Comments

This gauge makes extensive use of:

- Simple and structured macros, including nested operations with parameter passed values.
- Array processing loops of numeric and string variables.
- Detailed <String> properties, including color handling and internal loops.

LOCALVARS class examples

- **ReloadPanels.xml**

This is a simple gauge that makes (L:) variables keep their values after an aircraft's reload, either by using RELOAD_USER_AIRCRAFT event or by using Select aircraft... menu option.

How to install

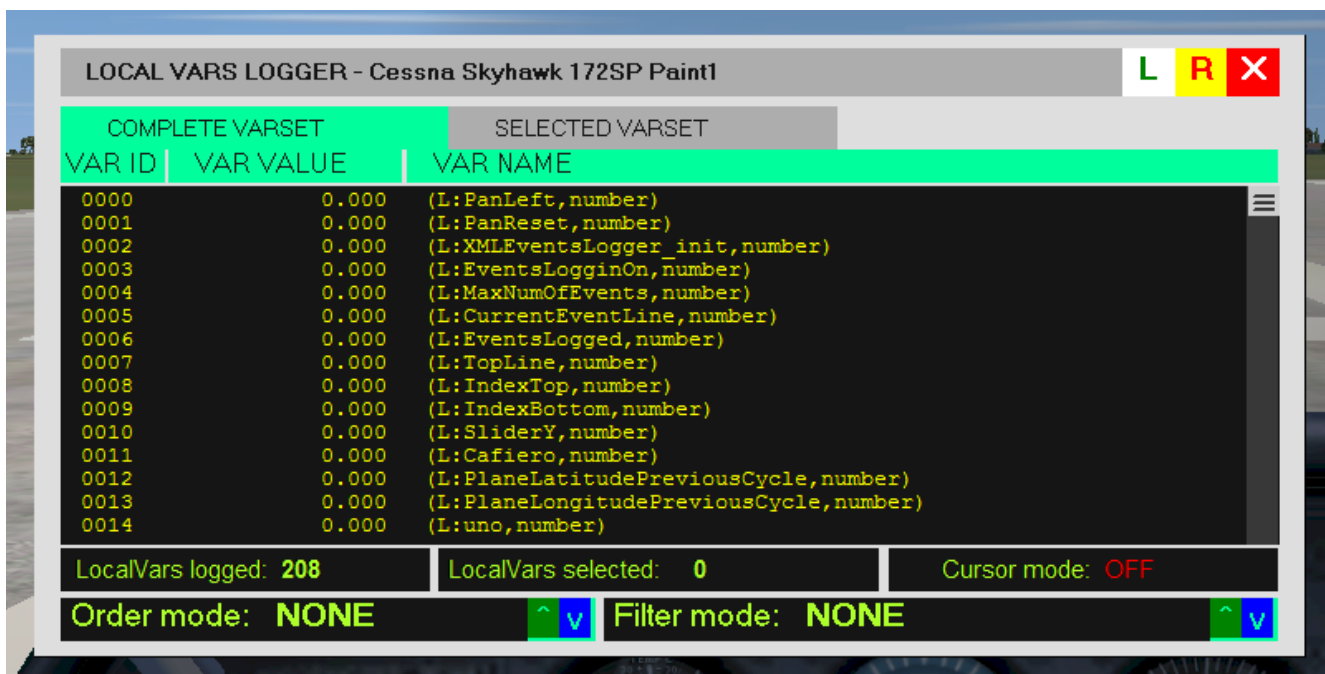
Edit the `panel.cfg` file of an aircraft of your choice, and add the following:

- In `[VCockpit01]` section:

```
gauge00=XMLTools!ReloadPanels, 0,0,1,1
```

It is preferable that this gauge be the first one to be loaded so that (L:) variables get their values restored before they are affected by the panel's operation.

- **LocalVarsLogger.xml**



This utility gauge displays the complete set of Local (L:) variables initialized by the current flight session, including model (.mdl) defined Lvars and those dynamically created at runtime.

Present vars can be selected and displayed separately in another tab, where it is possible to change their values and/or units.

Sort and filters can be applied to the complete varset as to facilitate search and diagnose operations.

How to install

Edit the `panel.cfg` file of an aircraft of your choice, and add the following:

- In [Window Titles] section:

```
windownn=LocalVars Logger
```

- In [Window~~nn~~] section:

```
size_mm=245,190
window_size=0.6, 0.5 (user choice, depending on display resolution)
position=3 (variable, user choice)
visible=0 (could be 1 if starting in 2D mode)
ident=11111 (fixed, a panel ID recognized by the gauge)
gauge00=XMLTools!LocalVarsLogger, 0,0,245,190
```

Operation

Local variables are loaded and displayed in the [COMPLETE VARSET] tab:

COMPLETE VARSET		SELECTED VARSET
VAR ID	VAR VALUE	VAR NAME
0000	0.000	(L:PanLeft,number)
0001	0.000	(L:PanReset,number)
0002	0.000	(L:XMLEventsLogger_init,number)
0003	0.000	(L:EventsLogginOn,number)
0004	0.000	(L:MaxNumOfEvents,number)
0005	0.000	(L:CurrentEventLine,number)
0006	0.000	(L:EventsLogged,number)
0007	0.000	(L:TopLine,number)
0008	0.000	(L:IndexTop,number)
0009	0.000	(L:IndexBottom,number)
0010	0.000	(L:SliderY,number)
0011	0.000	(L:Cafiero,number)
0012	0.000	(L:PlaneLatitudePreviousCycle,number)
0013	0.000	(L:PlaneLongitudePreviousCycle,number)
0014	0.000	(L:uno,number)

The list can be scrolled either by using the mouse wheel or dragging the vertical slider button.

Variable values are initially displayed in **number** units.

To select a variable, it is necessary first to enable cursor operation:

Cursor mode: OFF

Cursor mode: ON

Click on this field to enable/disable the cursor

When cursor is enabled, the display shows its current position:

0003	0.000	(L:EventsLogginOn,number)
0004	0.000	(L:MaxNumOfEvents,number)
0005	0.000	(L:CurrentEventLine,number)
0006	0.000	(L:EventsLogged,number)
0007	0.000	(L:TopLine,number)
0008	0.000	(L:IndexTop,number)
0009	0.000	(L:IndexBottom,number)

Mouse wheel and/or vertical slider button are used to position the cursor over a certain variable, which can be selected then by a double click of the mouse's left button.

Once the variable is selected, cursor changes to red:

0003	0.000	(L:EventsLogginOn,number)
0004	0.000	(L:MaxNumOfEvents,number)
0005	0.000	(L:CurrentEventLine,number)
0006	0.000	(L:EventsLogged,number)
0007	0.000	(L:TopLine,number)
0008	0.000	(L:IndexTop,number)
0009	0.000	(L:IndexBottom,number)

And the variable is displayed in the [SELECTED VARSET] tab:

COMPLETE VARSET		SELECTED VARSET
VAR ID	VAR VALUE	VAR NAME
0005	0.000	(L:CurrentEventLine,number)
0007	0.000	(L:TopLine,number)
0198	47.000	(L:AF_GlobalAirON,number)
0187	50.000	(L:AF_ClearAirTurbulence,number)

In the [COMPLETE VARSET] tab, a selected variable that is not highlighted by the cursor shows also in red:

0003	0.000	(L:EventsLogginOn,number)
0004	0.000	(L:MaxNumOfEvents,number)
0005	0.000	(L:CurrentEventLine,number)
0006	0.000	(L:EventsLogged,number)
0007	0.000	(L:TopLine,number)
0008	0.000	(L:IndexTop,number)
0009	0.000	(L:IndexBottom,number)

In the [SELECTED VARSET] tab, a variable can be deselected by positioning the cursor over it and double clicking with the mouse's left button. This operation is not available in the [COMPLETE VARSET] tab.

To change the **value** of a selected variable, in the [SELECTED VARSET] tab first position the cursor over the desired variable and then click on the [EDIT] button:

Value: 0.0000 EDIT

Type a proper value and then click [OK] /type ENTER to confirm or click [X] / type ESC to cancel.

Value: 456.2 OK X

The same operation applies to **units**:

Unit: feet OK X

Supported units are described in the FSX SDK, or the ESP 1.0 SDK.

Edition of **values** and **units** is not available when cursor mode is OFF:

Value: NONE EDIT Unit: NONE EDIT

To **sort** the variable set by name or value, in the [COMPLETE VARSET] tab click either [↑] or [↓] buttons to cycle through available order modes:

Order mode: A -> Z ^ v

A **filter** may be applied to the variable set also in the [COMPLETE VARSET] tab by clicking on either [↑] or [↓] buttons to cycle through available filter modes:

Filter mode: (X) = 1 ^ v

Some filter modes are pretty straightforward and some need a brief explanation.

For example, filter mode [TOGGLE (0) → (1)] detects when a variable changes its status from being 0 in a previously saved cycle to 1 in the current one. This can be very useful in case of inspecting a third party *virtual cockpit* while trying to figure out which variables are linked to 3D switches; the proper name should display when the switch is clicked. Filter modes [TOGGLE (1) → (0)] and [(X1) != (X)] work much the same.

To **close** this utility gauge, click on this button:



To trigger a RELOAD_USER_AIRCRAFT event, click on this button:



To toggle visibility of (L:) variables between Local and Global mode, click on this button:



Varset in Local mode

or



Varset in Global mode

Note: to keep this button's functionality, `ReloadPanels.xml` gauge must not be installed together with this utility in the same panel.

Comments

- This gauge makes extensive use of LOCALVARS and XMLVARS variables, including some array formats.
- The entire code is partially optimized, with much of it being pretty straightforward for a better understanding of the different techniques employed.
- It provides a good example on how to program a scrolling display with vertical sliders.
- Also contains keyboard captures and dynamic text's management.

XMLEVENTS class examples

- **EventHandler.xml**

This gauge provides an example on how to properly program event trappings to be used with this class. It contains specific macros to handle normal events and also a custom event that is triggered within a typical `<On Event>` function.

How to install

Edit the `panel.cfg` file of an aircraft of your choice, and add the following:

- In `[VCockpit01]` section:

```
gaugenn=XMLTools!EventHandler, 0,0,1,1
```

Operation

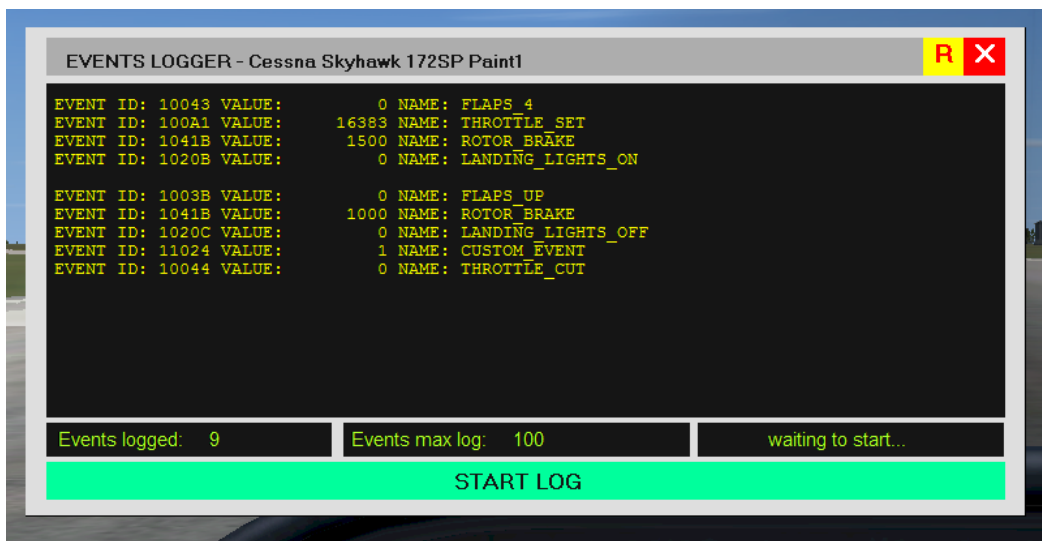
In any view mode, pressing **F8** applies flaps down together with full throttle and landing lights on. Pressing **F5** applies flaps up together with landing lights off. If current view is Virtual Cockpit, `LANDING_LIGHTS_OFF` event is trapped and triggers a custom event that cuts the throttle.

Comments

The gauge shows an example on how to take benefit of an unspecific aircraft event like `ROTOR_BRAKE`, which only affects rotorcraft variables, to command different actions depending on the value passed to this one.

A custom event example is also provided as a reference to understand the way it should be coded.

- **XMLEventsLogger.xml**



This is another utility gauge that captures and displays normal and custom events as they are triggered. It can log a maximum of 100 events per capture.

How to install

Edit the `panel.cfg` file of an aircraft of your choice, and add the following:

- In [Window Titles] section:

```
windownn=XMLEvents Logger
```

- In [Window~~nn~~] section:

```
size_mm=245,190
window_size=0.6, 0.5 (user choice, depending on display resolution)
position=4 (variable, user choice)
visible=0 (could be 1 if starting in 2D mode)
ident=11112 (fixed, a panel ID recognized by the gauge)
gauge00=XMLTools!XMLEventsLogger, 0,0,245,190
```

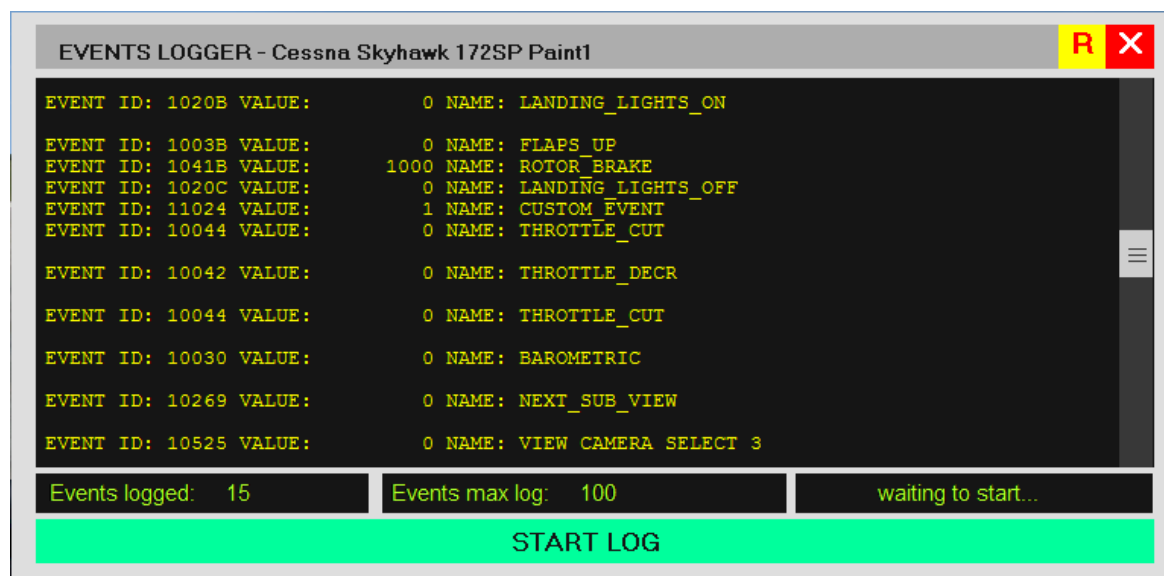
Operation

Click on [START LOG] bar to begin capturing events.

Each capture might contain a single event or a group of them if there are others triggered in the same action.

A blank line is displayed between captures to help visualizing the number of events included in a single capture.

To stop logging, click on [STOP LOG]. The entire log can be inspected using the mouse wheel or dragging the vertical slider button to scroll the display:



Once the process is restarted, events previously displayed are erased from the log.

While the log is in capture mode events are caught even when the 2D window is not visible.

To **close** this utility gauge, click on this button:



To trigger a `RELOAD_USER_AIRCRAFT` event, click on this button:

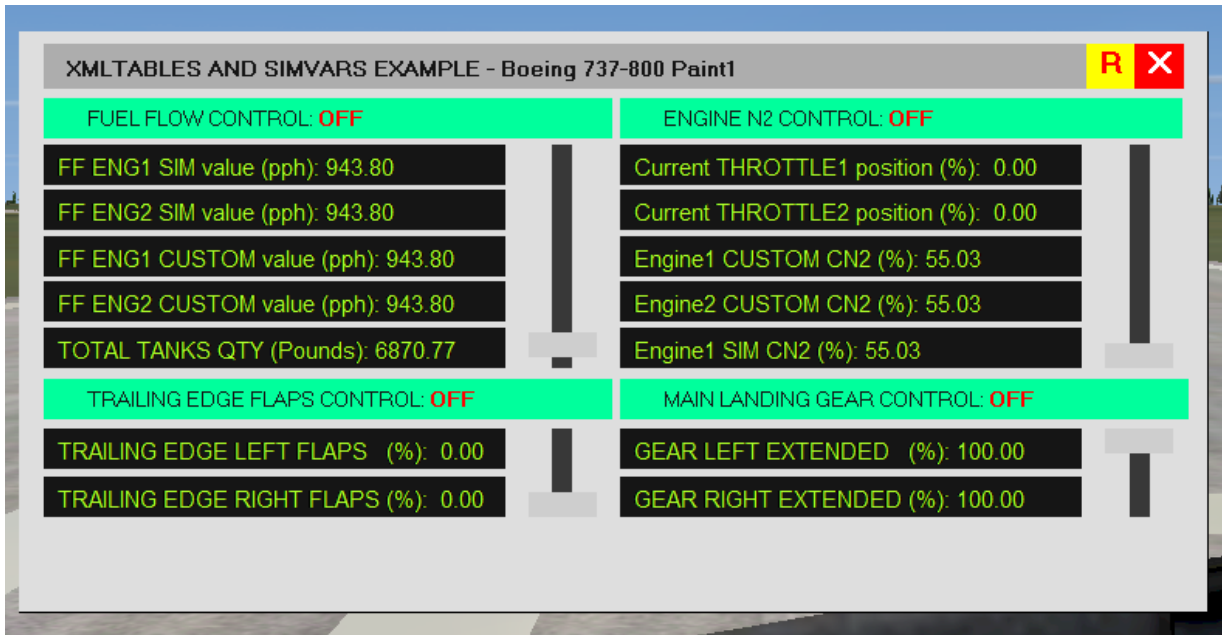


Comments

- This gauge makes use of processing loops of numeric and string variables.
- The entire code is partially optimized, with much of it being pretty straightforward for a better understanding of the different techniques employed.
- It provides a good example on how to program a scrolling display with vertical sliders.

SIMVARS and XMLTABLES class examples

- **TableAndSimExample.xml**



This gauge combines examples of SIMVARS and XMLTABLES classes. It uses sliders to control different aircraft variables and to obtain interpolated values from a table.

How to install

Edit panel.cfg file of the default B737-800, and add the following:

- In [Window Titles] section:

```
windownn=XMLTables and Sim
```

- In [Window~~nn~~] section:

```
size_mm=245,190
window_size=0.6, 0.5 (user choice, depending on display resolution)
position=1 (variable, user choice)
visible=0 (could be 1 if starting in 2D mode)
ident=11113 (fixed, a panel ID recognized by the gauge)
gauge00=XMLTools!TableAndSimExample, 0,0,245,190
```

Operation

Each one of four vertical sliders' position corresponds to a simulator controlled value when its associated green bar shows OFF, and to a custom controlled value when the bar shows ON.

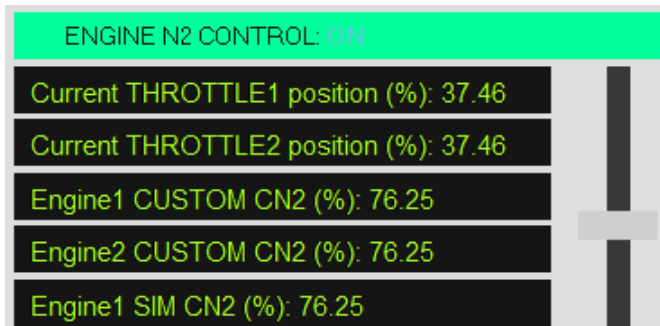
Click on a green bar to enable custom control of the vertical slider associated with it.

To control FUEL FLOW (SIMVARS class):



Move the slider to increase/decrease both ENG1 and ENG2 CUSTOM fuel flow, which is represented by TOTAL TANKS QTY increase/decrease in pounds. Simulator fuel flow data is not affected by the process.

To operate ENG CUSTOM CN2 (XMLTABLES class):



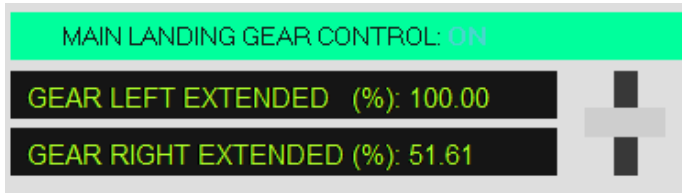
Move the slider to increase/decrease THROTTLE position. This is used as a source to obtain CUSTOM CN2 values by interpolating data from default B737-800 .air tables 1503 and 1504, which have been replicated within the gauge. CUSTOM CN2 values will show the same as SIM CN2 value once this one becomes stable.

To manage asymmetric FLAPS position (SIMVARS class):



In flight, move the slider to increase/decrease TRAILING EDGE LEFT FLAPS in percent. Switch to External view and spot the asymmetry; then try to maintain the aircraft stable!

To control MAIN LANDING GEAR extension (SIMVARS class):



In flight and with landing gear down, move the slider to extend/retract the RIGHT MAIN GEAR. Switch to External view and spot the asymmetry; then try to land safely with right gear retracted!

XMLKEYS class examples

- **KeysHandler.xml**

This gauge provides an example on how to properly program key and joystick trappings to be used with this class. It contains useful macros as a reference for making the code more understandable.

How to install

Edit panel.cfg file of the default B737-800, and add the following:

- In [Window Titles] section:

```
windownn= Keys Handler
```

- In [Window~~nn~~] section:

```
size_mm=120,30  
window_size_ratio=1.000  
position=8  
visible=1  
child_3d=1  
type=Special  
ident=10120 (variable, user choice)  
gauge00=XMLTools!KeysHandler, 0,0,120,30
```

Operation

Open Keys Handler's 2D window with the aircraft on the ground. Watch the small rectangle on the bottom right side of the screen:



Press the SPACEBAR to enable reverse thrust:



Now switch to External view and move the joystick's slider, which should be mapped to throttles by default. Notice the throttles are now operating in reverse thrust only.

Press the SPACEBAR again to put throttles back into normal operating mode.

Pressing SPACEBAR when airborne has no effect (to avoid going into reverse thrust in flight).

LOGGER class examples

- **FlightDataRecorder.xml**

A common use of LOGGER is as a Flight Data Recorder to record to HDD file flight variables for later playback, or to record interactions of various variables for detailed debugging analysis.

How to install

Edit panel.cfg file of the default C172, and add the following:

- In [Window Titles] section:

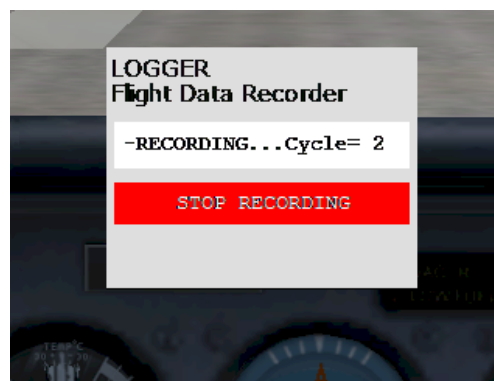
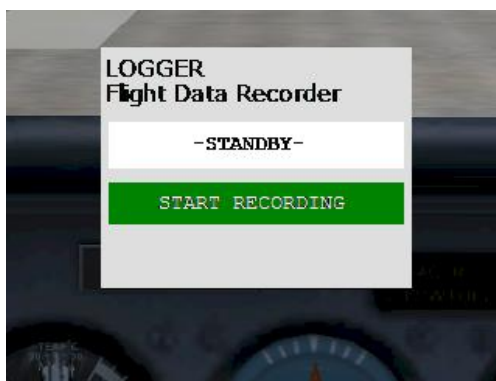
```
windownn= Flight Data Recorder
```

- In [Window~~nn~~] section:

```
size_mm=100,100  
window_size_ratio=1.000  
position=4  
visible=1  
ident=11119 (variable, user choice)  
gauge00=XMLTools!FlightDataRecorder, 0,0,100,100
```

Operation

Open Flight Data Recorder 2D window:



Press the green bar to start data recording. Press the red bar to stop the process.
Data is saved into FSX's Modules subfolder as FlightRecorderFile.txt

- **SaveAndLoadInitValues.xml**

LOGGER can be used to save and load initial values or settings of gauges you create – values that cannot otherwise be saved when you perform a Save Flight.

How to install

Edit the `panel.cfg` file of an aircraft of your choice, and add the following:

- In `[VCockpit01]` section:

```
gaugenn=XMLTools!SaveAndLoadInitValues, 0,0,1,1
```

Operation

On exiting FSX in VC view, some custom data values are saved with Csv format into FSX's main folder as `MyInitValues.txt`. Next time the aircraft is loaded those values are read from that file and assigned to Local Variables.

- **SaveAndLoadFlightPlan.xml (snippet)**

The Flight Simulator gps module provides the capability to easily create new *Flight Plans*, *SIDs* or *STARs* consisting of *Waypoint* coordinates using the `FlightPlanDirectToDestination`, `DeleteWaypoint` and `AddWaypoint` variables. Combining gps XML instructions with LOGGER enables the user to create, save and load *Flight Plans* to and from HDD file from within an *XML Flight Management System* gauge or GPS gauge without needing to use Flight Simulator's Flight Planner application.

Limitations to this method of creating *Flight Plans* include the inability of the Flight Simulator gps module to write (set) *Flight Plan Type* (VFR or IFR), *Route Type* (Direct, VOR, Low Alt or High Alt Airways), *Cruising Altitude* and automatic engagement of Flight Sim's ATC module. Flight Simulator's Flight Planner is still required to set those variables. On the other hand, one advantage of creating Flight Plans from within XML and using LOGGER to store the Flight Plan is the ability to add *VNAV Target Altitude* and *VS Profile* to the *Waypoint* information.

- **FDRforGoogleEarth.xml**

This is a complete xml gauge that can be used to record the flight information needed to generate *Google Earth* playback. It is recommended that the gauge be set up as a separate window in your `panel.cfg` file. The gauge size is 200 X 200 pix.

Before running the flight recorder, open the xml file and edit line 5 to show the filepath that you want used for the output file. Indicate the file path, but exclude the file name and .csv file extension – those are entered while Flight Simulator is running.

Operation

To operate the recorder while Flight Simulator is running, first click the white FILENAME.CSV text box to enable keyboard entry, then type the desired file name. Include the file extension .csv, but do not include the entire file path here. Click the text box again to disable keyboard entry.



The screenshot shows a window titled "FLIGHT DATA RECORDER" with a subtitle "FOR GOOGLE EARTH PLAY BACK". Inside the window, there is a text box labeled "FILENAME.CSV" containing "MY-FLIGHT.CSV" and a blue "CLEAR" button to its right. Below this, there is a section for "CYCLES TO SKIP" with a text box containing "18" and a blue "CLEAR" button to its right. To the right of the "CYCLES TO SKIP" section, there is a red "RECORD" button and a red "ON/OFF" button. Below the "RECORD" button, the word "CYCLES" is written in red. At the bottom left, there is a green button labeled "INSTRUCTIONS".

Click the white CYCLES TO SKIP entry box and enter the number of gauge update cycles to skip in between LOGGER recordings. A value of 0 or 1 causes LOGGER to record flight variables every update cycle. A value of 18 causes LOGGER to record flight variables every 18 cycles, or once every second at the default update frequency of 18 cycles per second, and so forth. Click the number entry box again to disable keyboard entry. Finally, click the ON/OFF button to start and stop the recorder.

- **KML generation spreadsheet: LOGGER – Google Earth v0.90 Beta.xls.**

LOGGER – Google Earth v0.90 Beta.xls is a Microsoft Excel 2003 application that contains macros to generate a Google Earth tour or track kml file. It functions also in Excel 2007.

Copy this file to any directory, and then open it in Excel. Excel may issue a Security Warning asking permission to Enable Macros. Adjust Excel macro security settings if necessary (Tools > Macro > Security > Medium), click "Enable Macros", and then open the Instructions Worksheet. Follow directions on the Instructions Worksheet.

Files and examples are included in *Google Earth Playback* folder.