

FS Recorder 2.0 Converter

Disclaimer & License

This software is provided "as is" and without any warranties. Installation and usage of this software is completely at your own risk! In no event shall the author be liable for any problems or damages caused directly or indirectly by installing or using this software.

This software is freeware. It may be freely distributed, as long as no money is charged for it, and none of the included files is removed or modified. Personal usage is free. Using this software for any commercial purpose is not allowed without written permission of the author!

Introduction

The FS Recorder converter is a simple tool to convert FS Recorder .frc files to text files and vice versa. It only supports the file format of FS Recorder 2.0, which is identical for the FS2004 and FSX versions.

Usage

When starting the converter a simple dialog box opens:



To convert a .frc file to a text file press the button *FS Recorder (.frc) → Text*. To convert a text file to a .frc file press the button *Text → FS Recorder (.frc)*. You will then have to select a source file and enter a target file name, after that the conversion starts. If an error occurs during the conversion, an error message is displayed, otherwise the conversion was successful.

By default altitude values in the text files are in feet, you can change this to meters by enabling the *Altitude in Meters* check-box.

About .frc Files

The .frc files produced by FS Recorder contain one or multiple tracks. The first track always contains the flight data of the user aircraft. If AI traffic was recorded, there is one additional track for each AI aircraft. During playback (using *Play*, not *Play as traffic*) the data in the first track is always used for the user aircraft, while all other tracks are played back as AI traffic (if enabled in the FS Recorder settings).

Format of converted Text Files

If you want to convert a text file to a .frc file, the format of the text data has to comply with the following format description. The actual flight data is formatted into rows and columns, where each row contains one set of parameters (in the following called a data sample), and each column represents a specific parameter. Before the actual flight data at least two special entries are required, which define the title of the aircraft and identify the data columns.

For better understanding of the format also have a look at the included sample files, which can be found in the samples directory.

Special Entries

Special entries start with a # character, followed by the entry name, a : character, and finally the parameters. A special entry is effective until the end of the file or until the next occurrence of the same entry.

Aircraft Entry

At least one aircraft entry needs to be present before the first [data entry](#), otherwise the conversion will fail with an error message. An aircraft entry has the following format:

```
#Aircraft: "<aircraft title>"
```

where you replace *<aircraft title>* with the title of the aircraft (as specified in the aircraft.cfg file). The aircraft title has to be enclosed in double quotes. The aircraft entry has to be placed before the [data entry](#) for the track to take effect!

Example:

```
#Aircraft: "Cessna Skyhawk 172SP"
```

TailNumber Entry

This entry is optional and defines the aircraft tail number:

```
#TailNumber: "<tail number>"
```

where you replace *<tail number>* with the tail number for the aircraft. The tail number has to be enclosed in double quotes. This entry has to be placed before the [data entry](#) of the track to take effect!

Example:

```
#TailNumber: "G-BAFM"
```

StaticCGtoGround Entry

This entry is optional and defines the height of the aircraft's reference point above the ground in feet, when the aircraft is sitting on the ground. FS Recorder uses this value to correct the aircraft altitude, if a recording is played back with a different aircraft as it was recorded with. The correct value for an aircraft can be found in the aircraft.cfg file (as *static_cg_height*). The format of this entry is:

```
#StaticCGtoGround: <CG height>
```

where you replace *<CG height>* with the height value in feet. This entry has to be placed before the [data entry](#) of the track to take effect! If this entry is not present, a default value of zero is used.

Example:

```
#StaticCGtoGround: 4.3
```

Data Entry

The data entry is required and defines, which parameters are present in the following [flight data section](#) and in which order. Each data entry starts a new track. A data entry has the following format:

```
#Data: <column descriptors>
```

where *<column descriptors>* is a sequence of strings separated by white space (tabs or spaces), and each of these strings defines one column according to [Table 1](#). Using any string not listed under column descriptors in the table will result in an error. Columns listed as required in the table have to be present. Before the first data entry an [aircraft entry](#) is required, or an error will occur!

Examples:

```
#Data: latitude longitude altitude pitch bank heading
```

```
#Data: timestamp altitude longitude latitude heading pitch bank onground
```

Flight Data Section

The flight data section is formatted as follows: Each line represents one data sample and consists of a sequence of values separated by whitespace (tabs or spaces). These values have to match the column definitions made with the [data entry](#) according to Table 1:

Table 1

<i>column descriptor</i>	<i>parameter</i>	<i>type</i>	<i>required</i>
timestamp	time code of data sample in seconds	floating point	no
latitude	latitude in degrees (positive=north, negative=south)	floating point	yes
longitude	longitude in degrees (positive=east, negative=west)	floating point	yes
altitude	altitude in feet (meters if <i>Altitude in Meters</i> enabled)	floating point	yes
pitch	pitch angle in degrees (-90 to 90)	floating point	yes

<i>column descriptor</i>	<i>parameter</i>	<i>type</i>	<i>required</i>
bank	bank angle in degrees (-180 to 180)	floating point	yes
heading	true heading in degrees (0 to 360)	floating point	yes
onground	on ground flag (0=in the air, 1=on ground)	boolean	no
velocityX	velocity in eastern direction in feet/sec	floating point	no
velocityY	vertical velocity in feet/sec	floating point	no
velocityZ	velocity in northern direction in feet/sec	floating point	no
elevators	elevators position (-16383 to 16383)	integer	no
ailerons	ailerons position (-16383 to 16383)	integer	no
rudder	rudder position (-16383 to 16383)	integer	no
throttle1	engine 1 throttle setting (-4096 to 16384)	integer	no
throttle2	engine 2 throttle setting (-4096 to 16384)	integer	no
throttle3	engine 3 throttle setting (-4096 to 16384)	integer	no
throttle4	engine 4 throttle setting (-4096 to 16384)	integer	no
prop1	engine 1 propeller setting (0 to 16384)	integer	no
prop2	engine 2 propeller setting (0 to 16384)	integer	no
prop3	engine 3 propeller setting (0 to 16384)	integer	no
prop4	engine 4 propeller setting (0 to 16384)	integer	no
mix1	engine 1 mixture setting (0 to 16384)	integer	no
mix2	engine 2 mixture setting (0 to 16384)	integer	no
mix3	engine 3 mixture setting (0 to 16384)	integer	no
mix4	engine 4 mixture setting (0 to 16384)	integer	no
spoilers	spoilers position (0 to 16383)	integer	no
flaps	flap handle position (0 to n)	integer	no
gear	gear handle position (0=up, 1=down)	boolean	no
lights	light switches (bitfield)	integer	no
parkbrake	parking brake (0=off, 1=set)	boolean	no
engine1	engine 1 running	boolean	no
engine2	engine 2 running	boolean	no
engine3	engine 3 running	boolean	no
engine4	engine 4 running	boolean	no
nav1frq	NAV radio 1 frequency in MHz	floating point	no
nav1obs	NAV radio 1 OBS in degrees	integer	no
nav2frq	NAV radio 2 frequency in MHz	floating point	no
nav2obs	NAV radio 2 OBS in degrees	integer	no
adffrq	ADF frequency in kHz	floating point	no
com1frq	COM radio 1 frequency in MHz	floating point	no

<i>column descriptor</i>	<i>parameter</i>	<i>type</i>	<i>required</i>
com2frq	COM radio 2 frequency in MHz	floating point	no
squawk	transponder code	integer	no
smoke	smoke system (0=off, 1=on)	boolean	no
IAS	indicated airspeed in knots	floating point	no
TAS	true airspeed in knots	floating point	no
mainexit	main aircraft exit	floating point	no
tailhook	tailhook position	floating point	no
wingfoldL	left wing fold	floating point	no
wingfoldR	right wing fold	floating point	no
windDir	wind direction in degrees (0 to 360)	integer	no
windSpd	wind speed in knots	integer	no
autopilot	autopilot master switch	boolean	no
concnose	Concorde nose visor	integer	no

Data values of type *integer* have to be whole numbers in the specified range. Values of type *boolean* can either be 0 or 1. Values of type *floating point* have to be real numbers in the specified range. *Integer* and *boolean* values **may not** use a decimal point! For floating point values the use of a decimal point is optional.

Lines, which do not comply with the format defined by the [data entry](#) will either be skipped or result in wrong converted data, which can lead to unpredictable results when played back with FS Recorder!

The *latitude*, *longitude*, *altitude*, *pitch*, *bank* and *heading* columns are required. If one of them is not present in the [data entry](#), an error will occur.

The *velocityX*, *velocityY* and *velocityZ* columns contain aircraft velocity along the three world axes (which point to the east, upward and to the north). If these columns are not present, the converter will approximate velocity data using the aircraft position data.

The *lights* column describes the state of the light switches as a bit field:

Bit 0	1	Nav lights
Bit 1	2	Beacon
Bit 2	4	Landing lights
Bit 3	8	Taxi light
Bit 4	16	Strobe lights
Bit 5	32	Panel lights
Bit 6	64	Recognition lights
Bit 7	128	Wing lights
Bit 8	256	Logo lights
Bit 9	1024	Cabin lights

The *timestamp* column contains the time code of the data samples. Timestamps have to be positive and monotonically increasing, otherwise a conversion error will occur! If no timestamps are

present, the converter uses fixed intervals, which default to one second, but can be changed with the [interval option](#).

At least two data samples have to be present in a data section, otherwise conversion will fail.

A simple Example

The simplest example for a valid text file could look like this:

```
#Aircraft: "Cessna Skyhawk 172SP"
#Data:  latitude longitude altitude pitch bank heading
        49.5      11.1    1050.0   -2.1   0.0    278.0
        49.5      11.1    1050.0   -2.1   0.0    278.0
```

Multiple Tracks

Multiple tracks can be specified in a text file by adding a new data entry and flight data section for every additional track. If no [aircraft entry](#) is present for the new track, the last specified aircraft will be used. The same is true for the [StaticCGtoGround entry](#).

Example for using three tracks:

```
#Aircraft: "Cessna Skyhawk 172SP"
#StaticCGtoGround: 3.765
#Data:  latitude longitude altitude pitch bank heading
        49.5      11.1    1050.0   -2.1   0.0    278.0
        49.5      11.1    1050.0   -2.1   0.0    278.0
#Data:  latitude longitude altitude pitch bank heading flaps
        49.8      11.1    1050.0   -2.1   0.0    278.0      1
        49.8      11.1    1050.0   -2.1   0.0    278.0      1
#Aircraft: "Cessna Skyhawk 172SP Paint 2"
#Data:  latitude longitude altitude pitch bank heading timestamp
        49.5      11.5    1050.0   -2.1   0.0    278.0      0.0
        49.5      11.5    1050.0   -2.1   0.0    278.0      1.0
```

Option Entries

Option entries start with a \$ character, followed by the option name, a : character and finally the arguments. They can occur anywhere in the file, also within a [flight data section](#). However, option entries are only effective until the next [data entry](#), so to take effect they need to be placed after the [data entry](#) of the respective track.

Offset Option

The offset option allows to shift the aircraft's position by a specified offset. The offset can be along world axes, aircraft axes or a combination of them. The format of the offset option looks like this:

```
$Offset: <x offset> <y offset> <z offset> [rotation]
```

where <x offset>, <y offset> and <z offset> are the three components of the offset vector in meters. The optional argument [rotation] specifies, around which axes the offset vector turns with the

aircraft, and can be a combination of the three letters *p*, *b* and *h* (for pitch, bank and heading).

If the *[rotation]* argument is not present, the offset will be along the world axes, where the *x* axis points east, the *y* axis points up and the *z* axis points north.

If *pbh* is used, the offset vector will be fixed to the aircraft and the offset will be along aircraft axes, where the *x* axis points to the right, the *y* axis points up and the *z* axis points forward.

This option can be used to correct an offset between flight data and scenery or to create multiple flights or tracks with different offsets from the same data for simultaneous playback as a perfect formation. Probably the best way to understand the rotation parameter is to have a look at the included offset samples, convert them to .frc files and play them with FS Recorder.

Examples:

```
$Offset: 5 1.5 -3  
$Offset: 10 0 15 pbh  
$Offset: 6.5 0 0 hb
```

Time Offset Option

This option allows to add an offset to the time code of a track, using the following format:

```
$TimeOffset <seconds>
```

where *<seconds>* is the number of seconds you want to add to the timestamps. This option can be used when synchronizing multiple flights for simultaneous playback, or to simulate multiple aircraft flying in a perfect sequence by creating multiple tracks from the same data with different time offsets.

Example:

```
$TimeOffset: 3.5
```

Interval Option

The interval option defines the interval used for timestamps, if no timestamp column is present in the [flight data section](#). The format looks as follows:

```
$Interval: <seconds>
```

where *<seconds>* specifies the time code interval in seconds. If timestamps are present in the [flight data section](#), this option has no effect.

Example:

```
$Interval: 0.2
```

Comments

Lines starting with // will be treated as comments by the converter and have no effect on the output.

Example:

```
// This is a comment
```

Skipped Lines

Any line in the file, which does not start with either #, \$ or // and cannot be interpreted as a sequence of data values according to the last [data entry](#), will be skipped by the converter.

Command Line Usage

Developers of tools for editing/creating/analyzing data files in the text format might want to automatically call the converter in their tool. The converter therefore can also be called with command line arguments:

```
FrcConverter.exe <input file> <output file> [-m]
```

```
<input file>    the file you want to convert
```

```
<output file>  the output file name
```

```
-m             altitude in meters
```

If the input file has the extension .frc the converter does an frc-to-text conversion, otherwise it tries a text-to-frc conversion.

If the converter is called with command line arguments, the user interface is not shown and conversion starts immediately. In case of conversion errors or if an invalid command line is used, an error message is displayed.

Support

For support information see the FS Recorder manual.

Copyright © 2007-2010 Matthias Neusinger