

File Formats

From LFS Manual

[Jump to navigation](#)[Jump to search](#)

Contents

- 1 DDS
- 2 PTH
- 3 TXT
- 4 SET
- 5 LYT
- 6 DRV
- 7 BANS
- 8 RAF
- 9 SPR
- 10 MPR
- 11 CAR_info.bin

DDS

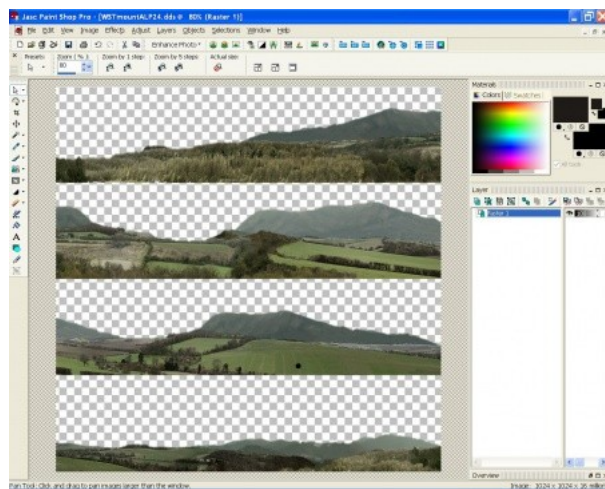
Texture files

So you want to edit LFS files that are in DDS format? Several textures in LFS are in DDS format including lights, interiors, track signage, seats, steering wheels and several other game and car components. To customise some textures in LFS You will need to edit the appropriate DDS file. DDS files can be handled by Paint Shop Pro and Photoshop after installing the DDS plugin which can be found here:
http://developer.nvidia.com/object/nv_texture_tools.html

Although these files are hosted by Nvidia, they are not GPU-specific, so you can use them with any brand card. You don't need all of the files shown on that site. You can download a DDS viewer which can enable thumbnails in windows explorer or my computer and view them easily. There are various viewers available and all seem to work well enough. Grab the plugin if you want to edit DDS with one of the programs mentioned above.

The DDS files can be found in your LFS/data/dds folder. There are a few things to be aware of before you begin:

- Once you have changed the DDS file all of the cars of that type will show that file, the texture is used universally. For example, if you customised the interior of the XF GTi, then all XF GTi's in your sim will have the customised interior. Unlike you car skin, these files are used on every car of that type.
- Be wary of texture size. Yes I know, you've got a fast PC and it can do anything except make your bed, but still be wary of texture size. The default textures in LFS work very well, replacing a 30KB



texture with a 300KB texture will use more resources. It is very easy to overdo it so be sensible and back up your files first.

- When you save your customised file you will probably be presented with some complicated save options. Just try the default settings, in other words: don't worry too much about all the bells and whistles. If the option to generate mip maps is not selected then select it first but it should be on by default in most cases.
- There is a plugin that you will need and it works for Adobe Photoshop and Jasc (recently purchased by Corel) Paint Shop Pro only. Both of these programs use the same plugin file. There is a 3DS MAX plugin too, but MAX users check your version first to see if the plugin is even required.

Please note that DDS files use an alpha layer in many cases. If you don't know what that means I suggest you consult your software's documentation and try experimenting with the DDS format files.

Remember to save your files back into DDS format!

PTH

Path nodes are a series of points with direction and width that describe the track that you drive along. LFS uses it to watch your progress along the track, decides if you are driving in reverse. They provide the data for the echoes and the lightmaps, hold information about which objects you can see from that point, define the left and right boundaries for the AI drivers and are also used in yellow and blue flag systems, the position list, timing and some other things. Their length is not constant but there is approximately 0.2 seconds of time between passing one node and the next, when you are driving at a reasonable speed.

```
PTH VERSION 0 - Path files for Live for Speed S2
=====
```

```
The nodes are given by a fixed point position (X, Y, Z) and a
floating point direction (X, Y, Z)
```

```
The node can be considered as a line perpendicular to its direction.
```

```
Outer and driving area left and right limits are given.
```

```
TYPES :
=====
```

```
1) X,Y,Z int : 32-bit fixed point world coordinates (1 metre = 65536)
```

```
X and Y are ground coordinates, Z is up.
```

```
2) float : 32 bit floating point number
```

```
FILE DESCRIPTION :
=====
```

```
num    unit    offset  description
---    ---    -

```

```
HEADER BLOCK :
```

```
6    char    0      LFSPTH          : do not read file if no match
1    byte    6      version          : 0 - do not read file if > 0
1    byte    7      revision         : 0 - do not read file if > 0
1    int     8      num nodes        : number
1    int     12     finish line      : number
.....NODE BLOCKS
```

```
NODE BLOCK :
```

```
1    int     0      centre X         : fp
1    int     4      centre Y         : fp
1    int     8      centre Z         : fp
```

```

1    float 12    dir X      : float
1    float 16    dir Y      : float
1    float 20    dir Z      : float
1    float 24    limit left : outer limit
1    float 28    limit right: outer limit
1    float 32    drive left  : road limit
1    float 36    drive right : road limit

```

TXT

Language Files, they are used as language packs to translate LFS to a selected list of languages.

Visit [How to translate](#) page for more information.

Thanks to Eold, we have a translation utility program which makes it easier to make language packs. See the enclosed README.txt for more information.

http://www.lfs.net/file_lfs.php?name=LFSTranslator.zip

SET

Thanks to colcob for originally working this format out for v0.3H ([1] (<http://forum.rscnet.org/showthread.php?t=213190>)), Bob Smith for updating it for v0.5P([2] (<http://www.lfsforum.net/showthread.php?t=14477>))), Woz for updating the bit field values for passengers, and back to Bob Smith for updating it for v0.5X/Y. The format was updated again following v0.6V.

TYPES :
=====

char : 1-byte ASCII character
byte : 1-byte integer
word : 2-byte unsigned integer
float: 4-byte float

Offset	Type	num	Description
-----	----	---	-----
0	char	6	Description (always SRSETT)
6	byte	1	0
7	byte	1	LFS Internal Version (was 250 for ages, then 251, currently 252) [ignore this number]
8	byte	1	File format version (currently 2) [check this number]
9	byte	3	0
12	byte	1	Bit 7 (Patch X Setup=1, older set=0) - used for pre-load handling Bit 2 (ABS On=1, Off=0) Bit 1 (Traction Control On=1, Off=0) Bit 0 (Asymmetrical On=1, Off=0)
13	byte	1	Unknown, seems to hold random values
14	byte	1	Handicap Mass Position
15	byte	1	Tyre Brand (0=Cromo, 2=Torro, 3=Michelin, 4=Evostar)
16	float	1	Brake Strength (Nm)
20	byte	1	Rear Wing Angle
21	byte	1	Front Wing Angle
22	byte	1	Voluntary Handicap Mass
23	byte	1	Voluntary Intake Restriction
24	byte	1	Max Steering Lock
25	byte	1	Parallel Steering
26	byte	1	Brake Balance
27	byte	1	Engine Brake Reduction
28	byte	1	Centre Diff Type (0=Open, 1=Viscous)
29	byte	1	Centre Diff Viscous Torque
30	byte	1	0
31	byte	1	Centre Diff Torque Split
32	word	1	Gear Ratio 7 (0 to 65534 = 0.5 to 7.5)
34	word	1	Gear Ratio Final (0 to 65534 = 0.5 to 7.5)
36	word	1	Gear Ratio 1 (0 to 65534 = 0.5 to 7.5)
38	word	1	Gear Ratio 2 (0 to 65534 = 0.5 to 7.5)
40	word	1	Gear Ratio 3 (0 to 65534 = 0.5 to 7.5)
42	word	1	Gear Ratio 4 (0 to 65534 = 0.5 to 7.5)
44	word	1	Gear Ratio 5 (0 to 65534 = 0.5 to 7.5)

46	word	1	Gear Ratio 6 (0 to 65534 = 0.5 to 7.5)
48	byte	1	Passenger (4 2bit fields). Passengers are located in the byte at the following locations
<pre> 76 54 32 10 ---+---+--- RR RC RL FR </pre>			
The individual passenger types are identified as follows.			
<pre> 00 = None 01 = Male 10 = Female </pre>			
49	byte	1	Car Config (roof on LX4/6 and UF1)
50	byte	1	Traction Control Slip (divide by ten)
51	byte	1	Traction Control Engage Speed
52	float	1	Rear Ride Height (NOT spring motion range)
56	float	1	Rear Spring Stiffness (N/mm)
60	float	1	Rear Compression/Bump Damping (N/mm)
64	float	1	Rear Rebound Damping (N/mm)
68	float	1	Rear Anti Roll Bar Stiffness (N/mm)
72	float	1	Handbrake Strength (N.m)
76	byte	1	Rear Toe (0=-0.9deg, 9=0deg, 18=0.9deg)
77	byte	1	Rear Caster (i.e. always zero)
78	byte	1	Rear Tyre Type (0 through 7 is R1 through Knobbly, in order of grip)
79	byte	1	0
80	byte	1	Rear Left Camber Adjust (45=0.0deg, 0=-4.5deg, 90=4.5deg)
81	byte	1	Rear Right Camber Adjust (45=0.0deg, 0=-4.5deg, 90=4.5deg)
82	byte	1	Rear Tyre Size (when using ALTERNATE configuration on GTR cars, 0 through 5 to 9 depending on the car)
83	byte	1	Rear Diff Clutch Pack Pre-load (multiply by ten)
84	byte	1	Rear Diff Type (0=Open, 1=Locked, 2=Viscous, 3=Clutch Pack)
85	byte	1	Rear Viscous Torque
86	byte	1	Rear Power Locking
87	byte	1	Rear Coast Locking
88	word	1	Rear Left Tyre Pressure (kPa)
90	word	1	Rear Right Tyre Pressure (kPa)
92	float	1	Front Ride Height (NOT spring motion range)
96	float	1	Front Spring Stiffness (N/mm)
100	float	1	Front Bump/Compression Damping (N/mm)
104	float	1	Front Rebound Damping (N/mm)
108	float	1	Front Anti Roll Bar Stiffness (N/mm)
112	byte	4	0
116	byte	1	Front Toe In (0=-0.9deg, 9=0deg, 18=0.9deg)
117	byte	1	Front Caster (need to divide by ten)
118	byte	1	Front Tyre Type (0 through 7 is R1 through Knobbly, in order of grip)
119	byte	1	0
120	byte	1	Front Left Camber Adjust (45=0.0deg, 0=-4.5deg, 90=4.5deg)
121	byte	1	Front Right Camber Adjust (45=0.0deg, 0=-4.5deg, 90=4.5deg)
122	byte	1	Front Tyre Size (when using ALTERNATE configuration on GTR cars, 0 through 5 to 9 depending on the car)
123	byte	1	Front Diff Clutch Pack Pre-load (multiply by ten)
124	byte	1	Front Diff Type (0=Open, 1=Locked, 2=Viscous, 3=Clutch Pack)
125	byte	1	Front Viscous Torque
126	byte	1	Front Power Locking
127	byte	1	Front Coast Locking
128	word	1	Front Left TyrePressure (kPa)
130	word	1	Front Right TyrePressure (kPa)

Note that some values can be "wrong" for mods: for instance, if the mod does not allow ABS, the setup file can still have ABS=1 if the default setup has this setting (it can be fixed by allowing ABS, disabling it, and then disallowing it).

LYT

These are Layout files, and govern how things are set out on an AutoX track. You will notice that on some tracks you find a limit to the number of objects you can add; this is a limitation in the LFS engine. It is believed (although untested) that this has not significantly changed since S1 ([3] (<http://forum.rscnet.org/showthread.php?t=185896>)).

TYPES :

=====

- 1) short : 16 bit signed integer
- 2) word : 16 bit unsigned
- 3) char : 8 bit signed integer
- 4) byte : 8 bit unsigned

FILE DESCRIPTION :

=====

num	unit	offset	description
---	----	-----	-----

HEADER BLOCK :

6	char	0	LFSLYT	: do not read file if no match
1	byte	6	version	: do not read file if > 0
1	byte	7	revision	: do not read file if > 252
1	word	8	num added objects	: number of OBJECT BLOCKS
1	byte	10	laps	: number
1	byte	11	flags	: see NOTE4

.....OBJECT BLOCKS

OBJECT BLOCK :

1	short	0	X	: position (1 metre = 16)
1	short	2	Y	: position (1 metre = 16)
1	char	4	Zchar	: height (1m = 4) - see NOTE3
1	byte	5	Flags	: 0 for objects - see NOTE1
1	byte	6	Index	: object index - see NOTE5
1	byte	7	Heading	: heading - see NOTE2

NOTE1 :

How to distinguish between physical objects and control objects, like start positions, checkpoints, finish line and marshal circles.

```

if (Index >= 192) // either a circle or an unknown object
{
    if (Index==255) // it's a marshall circle
    {
        if (Flags & 0x80) // highest bit set : restricted area
        {
            // Heading has its usual meaning

            // Flags byte contains the following data :
            // bits 0 to 1 :
            // 00 = no marshall
            // 01 = standing marshall
            // 10 = marshall pointing left
            // 11 = marshall pointing right
            // bits 2 to 6 :
            // radius in metres (1 to 31 - shifted left by 2 bits)
        }
        else // highest bit of flags is not set : route checker
        {
            // Heading is used not for heading, but the route index

            // bits 2 to 6 :
            // radius in metres (1 to 31 - shifted left by 2 bits)
        }
    }
    else
    {
        // unknown object - ignore
    }
}
else // could be an actual object or a control object
{
    if (Flags & 0x80) // highest bit set : control object
    {
        // Heading has its usual meaning

        // Flags byte contains the following data :
        // bits 0 to 1 :
        // 00 = Start position (if width = 0) or finish line (if width > 0)
    }
}

```

```

    // 01 = Checkpoint 1
    // 10 = Checkpoint 2
    // 11 = Checkpoint 3
    // bits 2 to 6 :
    // half width in metres (1 to 31 - shifted left by 2 bits)
}
else // highest bit of flags is not set : autocross object
{
    // Heading has its usual meaning

    // Flags byte contains the following data :
    // bits 0 to 2 :
    // Colour - only used for chalk (0-3) and tyres (0-5)
}
}

```

NOTE2 :

Heading represents 360 degrees in 256 values.

Heading = (heading_in_degrees + 180) * 256 / 360

128 : heading of zero
 192 : heading of 90 degrees
 0 : heading of 180 degrees
 64 : heading of -90 degrees

NOTE3 :

About Zchar, the approximate altitude :

LFS does contact checks to place objects accurately on the ground.

For output purposes : Zchar indicates the approximate altitude with a value from -80 (-20 metres) to 127 (31.75 metres).

For input purposes : The ground check is performed with a test ray starting from 2 metres above Zchar. Using a value lower than 127 allows objects to be placed on the road below a bridge, for example. If you are creating objects from scratch and you are not sure of the approximate altitude, you can set Zchar to its maximum value (127). This will usually work unless there is a physical surface above the road where you are trying to place the object.

NOTE4 :

If the file is saved in the most recent format, the two lowest bits of the flags byte must be set (total value 3).

bit 0 : the file does not need correction for the Blackwood scaling
 bit 1 : the file does not need conversion for the new object indices
 bit 2 : unknown
 bit 3 : unknown

Some very old layouts may not have bit 0 set.
 Layouts saved up to 0.5Z34 will not have bit 1 set.

(Newly created lyt files (ver 0.7f) have a flag value of 8 - Flag information is outdated)

NOTE5 :

Object indices are now the same at all tracks.
 The first valid object index is 4 (AXO_CHALK_LINE).
 There are many gaps to allow for future objects.
 Valid object indices are all less than 192.

AXO_NULL
 AXO_1
 AXO_2
 AXO_3
 AXO_CHALK_LINE
 AXO_CHALK_LINE2
 AXO_CHALK_AHEAD
 AXO_CHALK_AHEAD2
 AXO_CHALK_LEFT
 AXO_CHALK_LEFT2
 AXO_CHALK_LEFT3

```
AXO_CHALK_RIGHT
AXO_CHALK_RIGHT2
AXO_CHALK_RIGHT3
AXO_14
AXO_15
AXO_16
AXO_17
AXO_18
AXO_19
AXO_CONE_RED
AXO_CONE_RED2
AXO_CONE_RED3
AXO_CONE_BLUE
AXO_CONE_BLUE2
AXO_CONE_GREEN
AXO_CONE_GREEN2
AXO_CONE_ORANGE
AXO_CONE_WHITE
AXO_CONE_YELLOW
AXO_CONE_YELLOW2
AXO_31
AXO_32
AXO_33
AXO_34
AXO_35
AXO_36
AXO_37
AXO_38
AXO_39
AXO_CONE_PTR_RED
AXO_CONE_PTR_BLUE
AXO_CONE_PTR_GREEN
AXO_CONE_PTR_YELLOW
AXO_44
AXO_45
AXO_46
AXO_47
AXO_TYRE_SINGLE
AXO_TYRE_STACK2
AXO_TYRE_STACK3
AXO_TYRE_STACK4
AXO_TYRE_SINGLE_BIG
AXO_TYRE_STACK2_BIG
AXO_TYRE_STACK3_BIG
AXO_TYRE_STACK4_BIG
AXO_56
AXO_57
AXO_58
AXO_59
AXO_60
AXO_61
AXO_62
AXO_63
AXO_MARKER_CURVE_L
AXO_MARKER_CURVE_R
AXO_MARKER_L
AXO_MARKER_R
AXO_MARKER_HARD_L
AXO_MARKER_HARD_R
AXO_MARKER_L_R
AXO_MARKER_R_L
AXO_MARKER_S_L
AXO_MARKER_S_R
AXO_MARKER_S2_L
AXO_MARKER_S2_R
AXO_MARKER_U_L
AXO_MARKER_U_R
AXO_78
AXO_79
AXO_80
AXO_81
AXO_82
AXO_83
AXO_DIST25
AXO_DIST50
AXO_DIST75
AXO_DIST100
AXO_DIST125
AXO_DIST150
AXO_DIST200
AXO_DIST250
```

AXO_92
AXO_93
AXO_94
AXO_95
AXO_ARMC01
AXO_ARMC03
AXO_ARMC05
AXO_99
AXO_100
AXO_101
AXO_102
AXO_103
AXO_BARRIER_LONG
AXO_BARRIER_RED
AXO_BARRIER_WHITE
AXO_107
AXO_108
AXO_109
AXO_110
AXO_111
AXO_BANNER1
AXO_BANNER2
AXO_114
AXO_115
AXO_116
AXO_117
AXO_118
AXO_119
AXO_RAMP1
AXO_RAMP2
AXO_122
AXO_123
AXO_124
AXO_125
AXO_126
AXO_127
AXO_SPEED_HUMP_10M
AXO_SPEED_HUMP_6M
AXO_130
AXO_131
AXO_132
AXO_133
AXO_134
AXO_135
AXO_POST_GREEN
AXO_POST_ORANGE
AXO_POST_RED
AXO_POST_WHITE
AXO_140
AXO_141
AXO_142
AXO_143
AXO_BALE
AXO_145
AXO_146
AXO_147
AXO_RAILING
AXO_149
AXO_150
AXO_151
AXO_152
AXO_153
AXO_154
AXO_155
AXO_156
AXO_157
AXO_158
AXO_159
AXO_SIGN_KEEP_LEFT
AXO_SIGN_KEEP_RIGHT
AXO_162
AXO_163
AXO_164
AXO_165
AXO_166
AXO_167
AXO_SIGN_SPEED_80
AXO_SIGN_SPEED_50
AXO_170
AXO_171
AXO_172


```

AX0_173
AX0_174
AX0_175
AX0_176
AX0_177
AX0_178
AX0_179
AX0_180
AX0_181
AX0_182
AX0_183
AX0_184
AX0_185
AX0_186
AX0_187
AX0_188
AX0_189
AX0_190
AX0_191

```

DRV

These files contain the data on the AI drivers. This format was "discovered" around 0.3G, and it is unknown if they have been changed recently ([4] (<http://forum.rscnet.org/showthread.php?t=215013>)).

```

TYPES :
=====

```

```

char : 1-byte ascii character
byte : 1-byte integer
word : 2-byte integer
int  : 4-byte integer, lowest byte first

```

```

FILE DESCRIPTION :
=====

```

num	unit	offset	description
---	---	----	-----
6	char	0	SRAINM : do not read file if no match
1	byte	6	unknown : 0x00 ?
1	byte	7	unknown : 0xF6 Version?
1	byte	8	num AIs : Number of AI Names in that file?
3	byte	9	unknown : 3 bytes unknown

AI Data: Repeat (num AIs) times.

24	char	0	Name	: AI's playername (Fill with 0x00)
8	char	24	Plate	: Numberplate label (Fill with 0x00)
1	byte	32	Gender	: 0x00 == Male, 0x01 == Female
3	byte	33	unknown	: 3 bytes unknown

BANS

The file format of the bans file ([5] (<http://forum.rscnet.org/showpost.php?p=2426455&postcount=9>)).

```

Notes :

```

```

The 64 bit "Time" values are obtained from GetSystemTimeAsFileTime.
Meaning : number of 100-nanosecond intervals since January 1, 1601.
One hour (HOURL_TIME) = 3600000000

```

```

Demo ban expiry : time - ban->Time > 12 * HOURL_TIME
Name ban expiry : time - ban->Time > ban->BanHours * HOURL_TIME

```

```

The bans are loaded into memory when :
- the program starts up.

```

```

The bans are saved to disk when :
- bans are cleared
- a new ban is added

```

```

- the program exits

file format
-----
6 chars      LFSBAN
1 byte       0
1 byte       version (246 - do not read file if increased)
1 integer    num_demo_bans
[demo ban * num_demo_bans]
1 integer    num_name_bans
[name ban * num_name_bans]

demo_ban
-----
in_addr      IP address
__int64      Time

name_ban
-----
24 chars     user name
__int64      Time
integer      BanHours
integer      Space

```

RAF

LFS Replay Analysers work by reading a RAF file created by Live for Speed and displaying the data in their own way on graphs or map displays.

To create a RAF file, record a single player replay and then watch the replay. At any time during the lap BEFORE the lap you wish to analyse, press **Esc** and click on **Output lap data** and type in a name for the RAF file. Output will start when the car crosses the finish line. The RAF will be stored in the `lfs/data/raf` folder.

```

Replay Analyser File Version 2
=====

RAF format for LFS S2

NOTE 1 : (about reading the file)
=====

Please take note of and use the following variable fields :

(A) header size      (first data block's offset within file)
(B) block size       (size of one data block)
(C) wheel block size  (size of a dynamic wheel block)
(D) wheel block offset (offset of wheel block within data block)

The above numbers may be increased with no change to the RAF version.

NOTE 2 : (about centre of gravity and reference positions)
=====

In S1 time, position of car was the position of the CoG, so
wheel positions were given relative to the CoG.

In earlier versions of S2, CoG changed with fuel usage,
so the wheel positions were given relative to the CoG with no fuel.

Since S2 Alpha 0.5V the car's position is given as the position of
a fixed, central reference point at approximately the lowest point of
the car's body. Wheel positions are given relative to this point.

More static info can be extracted from the CAR_info.bin files,
which can be exported from the Garage by pressing the letter O.

NOTE 3 :
=====
Added for version 0.5X : short track name (at offset 24)

```

NOTE 4 :

=====

Added for version 0.5Z : approximate G values in data blocks

Added for version 0.5Z : update interval and slip fraction

FILE DESCRIPTION :

=====

num	unit	offset	description
---	----	-----	-----

HEADER BLOCK : 1024 bytes (A)

6	char	0	LFSRAF	: do not read file if no match
1	byte	6	game version	: ignore
1	byte	7	game revision	: ignore
1	byte	8	RAF version (2)	: do not read if increased
1	byte	9	update interval	: ms (normally 10 / hlvc 100)
2	byte	10	0	:
1	word	12	header size	: data blocks start (A)
1	word	14	block size	: size of a data block (B)
1	word	16	wheel block size	: inside data block (C)
1	word	18	wheel block offset	: within data block (D)
1	int	20	number of blocks	: total number of data blocks
4	char	24	short track name	: e.g. BL2R
1	float	28	track ruler length	: total index distance
32	char	32	player	: text
32	char	64	car	: text
32	char	96	track	: text
16	char	128	config	: text
16	char	144	weather	: text
8	char	160	LFS version	: text
1	byte	168	player flags	: driver aids etc (see NOTES)
1	byte	169	number of wheels	: usually 4
1	byte	170	HLVC legal	: 0=UNKNOWN 1=LEGAL 2=ILLEGAL
1	byte	171	number of splits	: including lap time
1	int	172	split 1	: ms
1	int	176	split 2	: ms
1	int	180	split 3	: ms
1	int	184	split 4	: ms
1	float	188	mass	: kg including driver
1	float	192	sprung mass	: kg including driver
1	float	196	R antiroll	: N/m
1	float	200	F antiroll	: N/m
1	float	204	final drive	: final drive ratio
1	byte	208	number of gears	: forward gears
3	byte	209	0	:
7	float	212	gear ratios	: forward gear ratios
272	byte	240	0	:
4	wheel	512	STATIC WHEEL INFO	: (see below)

STATIC WHEEL INFO : size 128 bytes per wheel

1	float	0	X	: relative to reference point
1	float	4	Y	: relative to reference point
1	float	8	Z	: relative to reference point
1	float	12	radius	: unloaded
1	float	16	width	: at widest point
1	float	20	maximum deflect	: suspension travel
4	byte	24	0	:
1	byte	28	0	:
1	byte	29	tyre type	: (see NOTES)
2	byte	30	0	:
1	float	32	spring constant	: N/m
1	float	36	damping (C)	: Ns/m
1	float	40	damping (R)	: Ns/m
1	float	44	max brake torque	: Nm
80	byte	48	0	:

DATA BLOCKS : 192 bytes (B) every 100th of a second

1	float	0	throttle	: 0 to 1
1	float	4	brake	: 0 to 1
1	float	8	input steer	: radians
1	float	12	clutch	: 0 to 1
1	float	16	handbrake	: 0 to 1
1	byte	20	gear	: 0=R, 1=N, 2=first gear
1	char	21	lateral G * 20	: -120 to 120 = -6 to 6 G

```

1 char 22 forward G * 20 : -120 to 120 = -6 to 6 G
1 char 23 upwards G * 20 : -120 to 120 = -6 to 6 G
1 float 24 speed : m/s
1 float 28 car distance : m - travelled by car
1 int 32 position X : map X (1m = 65536)
1 int 36 position Y : map Y (1m = 65536)
1 int 40 position Z : altitude (1m = 65536)
1 float 44 engine speed : radians/s
1 float 48 index distance : m - track ruler measurement
1 short 52 RX : x of right-vector
1 short 54 RY : y of right-vector
1 short 56 RZ : z of right-vector
1 short 58 FX : x of forward-vector
1 short 60 FY : y of forward-vector
1 short 62 FZ : z of forward-vector
4 wheel 64 (D) DYNAMIC WHEEL INFO : (see below)

```

DYNAMIC WHEEL INFO : size 32 bytes (C) per wheel

```

1 float 0 suspension deflect : compression from unloaded
1 float 4 steer : including Ackermann and toe
1 float 12 X force : force right
1 float 16 Y force : force forward
1 float 8 vertical load : perpendicular to surface
1 float 20 angular velocity : radians/s
1 float 24 lean rel. to road : radians a-c viewed from rear
1 byte 28 air temperature : degrees C
1 byte 29 slip fraction : (0 to 255 - see below)
1 byte 30 0 :
1 byte 31 0 :

```

NOTES :
=====

Axes

X - right

Y - forward

Z - up

To work out heading from the forward-vector

```

float b = FX / 32767.0f; // convert FX to a float from -1 to 1
float e = FY / 32767.0f; // convert FY to a float from -1 to 1
float heading = atan2(-b, e); // heading (anti-clockwise from above)

```

Player flags

```

LEFT HAND DRIVE 1
GEAR CHANGE CUT 2
GEAR CHANGE BLIP 4
AUTO SHIFT 8
SHIFTER 16
RESERVED 32
BRAKING HELP 64
AXIS CLUTCH 128

```

Tyre types

```

RACE R1 0
RACE R2 1
RACE R3 2
RACE R4 3
ROAD SUPER 4
ROAD NORMAL 5
HYBRID 6
KNOBBLY 7

```

Slip fraction

This is the dynamic value of the current combined slip ratio relative to the combined slip ratio that would provide the greatest force.

0 to 254 - slip ratio increasing up to maximum force available
255 - slip ratio exceeds the maximum force slip ratio

Single player replay file header format.

SPR file header format : Live for Speed 0.6B

=====

TYPES :

=====

char : 1-byte ascii character

byte : 1-byte integer

word : 2-byte integer

int : 4-byte integer, lowest byte first

time : 4 bytes [min | sec | 100ths | 1000ths]

FILE DESCRIPTION :

=====

num	unit	offset	description
---	----	-----	-----
6	char	0	LFSSPR : do not read file if no match
1	byte	6	file version high : ignore
1	byte	7	file version low : ignore
1	byte	8	SPR version : ignore
1	byte	9	reserved : -
1	byte	10	reserved : -
1	byte	11	qual mins : qualifying time
1	byte	12	laps byte : laps / hours (see NOTES)
1	byte	13	skill : skill level (0,1,2,3,4)
1	byte	14	wind : 0=off 1=weak 2=strong
1	byte	15	hotlap mode : 0=no 1=yes 2=custom 3=invalid
8	char	16	LFS version : text, ends 0
4	char	24	short track name : e.g. BL2R
1	byte	28	added mass : hotlap only
1	byte	29	intake restrict : hotlap only
1	byte	30	abs enabled : hotlap only
1	byte	31	0 : -
32	char	32	track name : text, ends 0
32	char	64	user name : text, ends 0
32	char	96	car name : text, ends 0
1	byte	128	config : 1,2,3... (first config is 1)
1	byte	129	reversed : 0=no 1=yes
1	byte	130	weather : 0,1,2... (first weather is 0)
1	byte	131	number of drivers : total cars in race
1	word	132	player flags : driver settings (see NOTES)
1	byte	134	hlvc best lap : the lap of the split times
1	byte	135	number of splits : including lap time
1	time	136	split 1 : msht time (first check point)
1	time	140	split 2 : msht time
1	time	144	split 3 : msht time
1	time	148	split 4 : msht time
1	int	152	flags : -
1	int	156	replay length : centiseconds (0 if unknown)
32	char	160	local driver name : including colours

The rest of the file is the actual SPR data

NOTES :

=====

Laps Byte (lb) has various meanings :

0	:	practice
1-99	:	number of laps... laps = lb
100-190	:	100 to 1000 laps... laps = (lb - 100) * 10 + 100
191-238	:	1 to 48 hours... hours = lb - 190

Player Flags

PIF_SWAPSIDE	1
PIF_RESERVED_2	2
PIF_RESERVED_4	4
PIF_AUTOGEARS	8
PIF_SHIFTER	16
PIF_RESERVED_32	32
PIF_HELP_B	64
PIF_AXIS_CLUTCH	128

```

PIF_INPITS      256
PIF_AUTOCLUTCH  512
PIF_MOUSE       1024
PIF_KB_NO_HELP  2048
PIF_KB_STABILISED 4096
PIF_CUSTOM_VIEW 8192

```

MPR

Multiplayer file header format.

```

MPR file header format for LFS S2 : 0.5X9
=====

```

```

TYPES :
=====

```

```

char : 1-byte ascii character
byte : 1-byte integer
word : 2-byte integer
int  : 4-byte integer, lowest byte first

```

```

FILE DESCRIPTION :
=====

```

num	unit	offset	description
---	----	-----	-----
6	char	0	LFSMPR : do not read file if no match
1	byte	6	game version : ignore
1	byte	7	game revision : ignore
1	byte	8	MPR version : ignore
1	byte	9	immediate start : joined already running game
1	byte	10	reserved : -
1	byte	11	reserved : -
1	int	12	rules : -
1	int	16	flags : -
1	byte	20	laps byte : laps / hours (see notes)
1	byte	21	skill : skill level (0,1,2,3,4)
1	byte	22	wind : 0=off 1=weak 2=strong
1	byte	23	num players : players at start of race
8	char	24	LFS version : text, ends 0
4	char	32	short track name : e.g. BL2R
1	int	36	start time (UTC) : seconds from 00:00 1/1/1970
32	char	40	track name : text, ends 0
1	byte	72	config : 1,2,3.. (first config is 1)
1	byte	73	reversed : 0=no 1=yes
1	byte	74	weather : 0,1,2.. (first weather is 0)
1	byte	75	num finished (NF) : players in results table
1	int	76	0 : -
NF	result	80	RESULT INFO : (see below)

RESULT INFO : size 80 bytes per finished player

24	char	0	player name	: text, ends 0, no colours
8	char	24	number plate	: text, NOTE : NO ZERO AT END
4	char	32	short car name	: text, ends 0
24	byte	36	lfs user name	: text, ends 0
1	word	60	laps done	: total laps completed
1	word	62	player flags	: driver settings (see NOTES)
1	byte	64	confirm flags	: penalties (see NOTES)
1	byte	65	number of stops	: pit stops count
1	word	66	penalty seconds	: penalty time added
1	int	68	overall time	: milliseconds
1	int	72	best lap time	: milliseconds
1	byte	76	0	: -
1	byte	77	start position	: 0 = unknown, 1 = pole, etc.
1	byte	78	handicap mass	: kg
1	byte	79	intake restriction	: %

The rest of the file is the actual MPR data

NOTES :

```
=====
```

```
Laps Byte (lb) has various meanings :
```

```
-----
0           : practice
1-99       : number of laps... laps = lb
100-190    : 100 to 1000 laps... laps = (lb - 100) * 10 + 100
191-238    : 1 to 48 hours... hours = lb - 190
```

```
Player Flags
```

```
-----
LEFT HAND DRIVE      1
GEAR CHANGE CUT      2
GEAR CHANGE BLIP     4
AUTO SHIFT           8
SHIFTER              16
RESERVED             32
BRAKING HELP         64
AXIS CLUTCH          128
RESERVED             256
AUTO CLUTCH          512
MOUSE                1024
KB NO HELP           2048
KB STABILISED        4096
```

```
Confirmation Flags
```

```
-----
MENTIONED            1
CONFIRMED            2
PENALTY_DT           4  <-- disqualified
PENALTY_SG           8  <-- disqualified
PENALTY_30           16
CONF_PENALTY_45       32
CONF_DID_NOT_PIT     64  <-- disqualified
```

CAR_info.bin

Press letter in the garage to export CAR_info.bin

```
// File format for CAR_info.bin files (LFS 0.5U15)
// -----

offset  description

0       identifier : LFS_CI
6       0 byte
7       version : currently 1 - do not read file if increased
8       short car name (3 chars + zero)
12      byte : passengers - bit 0:driver, 1:front, 2:r_left,
                        3:r_midde, 4:r_right

13      0 byte
14      0 byte
15      0 byte

// Body matrix - 9 floats - in LFS, X right, Y forward, Z up

16      Body matrix : (a, d, g) is right vector,
                        (b, e, h) forward,
                        (c, f, i) up

// From this point, most values are 4 byte floating point or
// 4 byte integer values

52      Reference point position in WORLD coordinates
                        x, y, z (integer : 65535=ONE)
64      CoG position in WORLD coordinates
                        x, y, z (integer : 65535=ONE)
76      CoG relative to reference point, in LOCAL coordinates
                        x, y, z (float)
88      Fuel Tank Pos rel to ref point, in LOCAL coordinates
                        x, y, z (float)

100     0
        0
        0
```

```
112      0
        0
        0
        0

// Now 4 aero blocks - rear wing, front wing,
                    undertray, body : 20 bytes each)

128      Position : x, y, z
        Lift : multiply by speed squared to get lift value
        Drag : multiply by speed squared to get drag value

// Moment of inertia matrix, required for dynamic modelling

208      moment of inertia matrix (9 floats : 36 bytes)
        0
        0
        0

256      max torque (Nm)
        @rpm
        max power (kW)
        @rpm

272      fuel litres
        mass total
        approx wheelbase in metres (at zero suspension compress)
        approx weight distribution (at zero suspension compress)

288      byte : number of forward gears
        byte : Drive - 1=rwd, 2=fwd, 3=all
        0 byte
        0 byte
292      torque split
        drivetrain efficiency
        0

304      reverse gear
        1st gear
        ...
        7th gear

336      final drive ratio
        0
        0
        0

352      parallel steer (1 = parallel)
        brake strength
        brake balance
        0

368      0
        0
        0
        0

// offset 384 - wheels : rear then front
                    left then right
                    128 bytes each

0        tyre type byte
        0 byte
        0 byte
        0 byte
4        pressure
        air temperature
        toe in

16       centre of contact patch (local coords x, y, x)
        unsprung mass

32       width of tyre
        sidewall height proportion of width
        rim radius
        rim width

48       spring constant (vertical, at wheel)
        damping - compression
        damping - rebound
        anti roll
```


64	camber inclination caster scrub radius
80	moment of inertia current suspension deflection (vertical, at wheel) maximum suspension deflection 0
96	tyre spring constant at current deflection / temperature current vertical tyre deflection 0 0
112	0 0 0
0	

Live for Speed guides and tutorials

v

Basic Setup Guide | Advanced Setup Guide | Technical Reference | League Racing Guide | Skin Tutorial | Autocross Editor | LFS Editor Guides | Hosting | File Formats | LFS Programming | Movie Tutorial | Scripting | Translating

Retrieved from "https://en.lfsmanual.net/index.php?title=File_Formats&oldid=11913"

- This page was last edited on 6 June 2025, at 15:53.