



navigation

■ Main Page

■ Recent changes

■ Random page

■ Help

links

■ LFS Main Site

■ LFS Forum

■ LFS World

search

Search LFS Manual

GoSearch

toolbox

■ What links here

■ Related changes

■ Upload file

■ Special pages

■ Printable version

■ Permanent link

■ Page information

Script Guide

by AndroidXP

Contents [hide]

1 Introduction

2 How to use

2.1 Basics

2.2 Reserved filenames

3 Commands

3.1 Options & Miscellaneous

3.2 Binding & Executing Commands

3.3 In-game

3.4 Parameter reference

4 Examples

4.1 Automatic shifter configuration (for Logitech G25/G27 users)

4.2 Indicator Control

4.3 (Obsolete) "F9 > F10 > F11 > F12 > Off" Toggle

4.4 (Obsolete) Brake balance adjustment (increase/decrease)

Introduction

With the release of patch V in December 2006, the LFS developers introduced a new scripting system, quite similar to the implementation seen in most first person shooters. Generally, it allows the automation of common tasks, customisation of options on a per-car basis as well as simple toggles to change the binding of keys (and more). This guide is aimed to give you a general understanding of the LFS script system as well as a few examples for simple scripts.

Version 0.7E (December 2023) introduced more commands to use in-game (all bindable), including controlling status screens (F9-F12), adjusting live settings and pit instructions (with math operators), switching mode of car lights.

How to use

Basics

Script files have the extension ".lfs" and can be found under `<LFS install dir>\data\script`. To edit these files, simply open them with a text editor of your choice (Notepad, etc). You can also create new script files, just make sure they have the right extension. If you want, you can even organize your scripts into subdirectories to keep the place clean.

In general, each line in the script file represents a command that is executed by LFS, going from top to bottom. The only exception are empty lines, and lines that start with *"//"*, which are treated as comments and thus ignored. Basically you can imagine the system as LFS reading each line, writing it into the chat window and hitting return. Of course it is not exactly like that, but it should give you the general idea.

Version 0.7E introduced the possibility of using multiple commands on a single line, which sometimes can avoid the need for a script file, e.g. to set a button to change both front and rear tyres in pit stop, you could use a double command like: `/pitins ftyre super /pitins rtyre super`. This could also be used for more complicated setup changes, though with the need of pre-programming the whole setup beforehand.

Note: some commands can NOT be immediately followed by another command, on the same line:
`/say, /echo, /join, /rcm, /pass, /msg, /altf and /ctrlf`

Reserved filenames

While you can generally name new script files however you want, a few names have a special meaning to LFS. All affected files already exist in the script folder when you first install the patch, so as long as you don't move or overwrite any files everything will be fine.

autoexec.lfs

Like the name suggests, this script is executed automatically every time LFS starts and is thus the ideal place to initialize your scripts.

CAR.lfs, SKINID.lfs

"CAR" being replaced with one of the three-letter car shortcuts (see [here](#) or below), or "SKINID" being replaced by the 6-character [vehicle mod](#) SkinID, these scripts are executed everytime the corresponding car is selected in the garage - for example if you select the XR GT Turbo, LFS will execute "XRT.lfs". They are usually used to give car specific settings, like a custom FOV or force feedback strength.

road.lfs, sequential.lfs, paddle.lfs

These filenames are reserved and automatically executed. Their main purpose are custom settings that should apply to cars with a certain shifter type, *road.lfs* is the equivalent to cars with a H-gate shifter, *sequential.lfs* to a sequential stick shifter and *paddle.lfs* to having paddles. If you have a wheel that supports such shifter types (like the [Logitech G25](#) for example), then these script files are handy to automatically change the shifter setup so it fits the currently selected car.

To match the cars to their gearbox, they are assigned as following:

Road:

UF1, XFG, XRG, XRT, RB4, FXO, LX4, LX6, FZ5, FZR, UFR, RAC

Sequential:

FBM, FOX, F08, MRT, XFR, XRR, FXR

Paddle:

BF1

Commands

Main article: Commands

The following commands are just an excerpt of the more commonly used ones.
A list of all commands can be found under `<LFS install dir>\docs\Commands.txt`.

Options & Miscellaneous

```
/run [scriptname] - run a script (data\script\scriptname.lfs), e.g. "/run myScript" or "/run myFolder\myScript"
/say [message] - same as typing a chat message
/echo [text] - show text (only on local screen)
/fov [degrees] - field of view
/ff [0-200] - force feedback strength
/wheel_turn [degrees] - specify turn angle of controller
/autoclutch [0-1] - turn autoclutch off / on
/gccut [0-1] - throttle cut on upshift **
/gcblip [0-1] - throttle blip on downshift **
/shifter [auto/sequential/shifter] - shift type
/view [foi/heli/cam/driver/custom] - select view
/view [save/reload] - save or reload current custom view**
**NB: gccut and gcblip were removed with a recent patch (patch Y).
***NB: new command since patch Z19
```

Binding & Executing Commands

Binding

/axis [axis]

[function]

- e.g. "/axis 2 throttle" (to unbind, use -1)

/invert [0/1]

[function]

- e.g. "/invert 1 brake"

/button [button]

[function]

- e.g. "/button 5 shift_up" (to unbind, use -1)

/key [key]

[function]

- e.g. "/key Q handbrake"

/ctrlf [num]

[text]

- change shortcut text, e.g. "/ctrlf 1 hello"

/altf [num]

[text]

- change shortcut text, e.g. "/altf 1 bye"

Simulating key presses

/press [key]

- simulate key press

/shift [key]

- SHIFT + key

/ctrl [key]

- CTRL + key

/alt [key]

- ALT + key

In-game

Status screens

/status [none/F9/F10/F11/F12/next/prev]

- controls status screen
e.g. "/status next" will cycle through the F9 to F12 status screens, including the "none" status, e.g. "/status F9" will open F9 status screen, regardless of whats already open or not

Car Lights

/light ind [off/loff/right/all]

- switch indicators/hazard

/light head [off/side/low/high/low_off/low_high/next/prev]

- switch headlights

/light [rfog/ffog/extra] [off/on/toggle]

- switch fog/extra

/light all [off/on]

- all switchable lights off/on at once

Controlling Live Settings (F11) and Pit Instructions (F12)

You can use operators:
= (set value)
++ (add to value)
-- (subtract from value)

Examples (see all available options below, in the [Parameter reference](#) section):
/pitins ftyre = r3 : change front tyres to R3 in pit stop
/pitins rtyre = super : change rear tyres to road super
/pitins fpressure = 1.1 : set front tyre pressure to 1.1 bar
/pitins fpressure ++ 0.1 : increase requested pressure by 0.1 bar
/pitins fpressure 30 psi : increase requested pressure to 30 psi
/pitins cancel : cancel all pit instructions
/pitins tyres always : change all tyres
/pitins tyres 20 : change tyres if wear > 20%
/pitins tyres never : do not change tyres
/liveset bbal 60 : set brake balance to 60%
/liveset bbal -= 1 : decrease brake balance by 1%
/liveset farb += 2 : increase front anti-roll bar by 2
/liveset rarb -= 0.1 : decrease rear anti-roll bar by 0.1

Parameter reference

Function names for the /button and /key commands

steer_left, steer_right, steer_fast, steer_slow
throttle, brake, shift_up, shift_down
clutch, handbrake, left_view, right_view, rear_view
horn, flash, reset, pit_speed, tc_disable, zoom_in, zoom_out
reverse, gear_1 - gear_7, ctrl_f1 - ctrl_f12, ignition

To find out the number for a controller button, go to [Options > Controls](#) and press the button in question. You will see the button # and currently assigned function in the middle right part of the screen.

Function names for the /axis and /invert commands

steer, combined, throttle, brake
lookh, lookp, lookr
clutch, handbrake, shiftx, shifty

Parameters for the key press commands (press / shift / ctrl / alt)

Letters A to Z
Numbers 0 to 9
F1 to F12
up, down, left, right
space, enter, esc, tab
less, more (< >) or (, .) depending on your locale

Available options for /pitins

fuel, tyres, repair, symmetric
fryre, fcamber_l, fpressure_l, fcamber_r, fpressure_r, fwing
rtyre, rcamber_l, rpressure_l, rcamber_r, rpressure_r, rwing
cancel, fcamber, fpressure, rcamber, rpressure

Available options for /liveset

bbal, farb, rarb

Examples

On a general note: as you might have noticed from the command listing, it is currently impossible to bind other script commands to a button; for example, "button 1 /say hello" wouldn't work. Luckily we can use the text shortcut keys (Ctrl+F1-F12, Alt+F1-F12) for this, so we can use "ctrlf 1 /say hello" (which binds "say hello" to Ctrl+F1) and then "button 1 ctrl_f1" (binds the press of Ctrl+F1 to button #1). However, this also means that you have to give away a few of your shortcut keys for the scripts, which shouldn't really be a problem, considering there are 32 available shortcuts total (24 bindable via scripts).

Automatic shifter configuration (for Logitech G25/G27 users)

Shortcut keys used: None

To automatically adjust the controller bindings for the shift mechanism used by the currently selected car, all you have to do is modify the existing *road.lfs*, *sequential.lfs* and *paddle.lfs* as shown below.

Optionally you can also add re-bindings of whatever you want to use the free shifter/paddle buttons for. For example, when using the shifter unit, my right paddle acts as handbrake and the left one as horn, whereas when using the paddles, pulling the sequential stick is the handbrake and pushing towards the horn. Doing this maximises the capabilities of the few easy-to-reach buttons on the G25, these re-bindings are not included in the example scripts.

road.lfs

```
/button 8 gear_1
/button 9 gear_2
/button 10 gear_3
/button 11 gear_4
/button 12 gear_5
/button 13 gear_6
/button 14 reverse

/shifter shifter
/autoclutch 0

/echo Shiftmode: H-Gate
```

sequential.lfs

```
/button 9 shift_up
/button 8 shift_down

/shifter sequential
/autoclutch 0

/echo Shiftmode: Sequential
```

paddle.lfs

```
/button 4 shift_up
/button 5 shift_down

/shifter sequential
/autoclutch 1

/echo Shiftmode: Paddles
```

Indicator Control

Shortcut keys used: Ctrl+F10, Ctrl+F11

This example shows a simple way to comfortably control the indicators with two buttons. You need three scripts and a few entries to *autoexec.lfs*, shortcut keys used are Ctrl+F10 and F11. The indicators will work as follows; first press of left button activates left indicator, second press deactivates it. Vice versa for the right button. Switching between left and right is possible. For ease of use, the script will output green arrows indicating the currently selected mode into the chat area.

By using the same logic and adding another script, it is also easy to implement the control of the hazards by a third button.

indicateLeft.lfs

```
/light ind left
/ctrlf 11 /run indicateStop
/ctrlf 10 /run indicateRight
/echo "2< ◀ ◀"
```

indicateRight.lfs

```
/light ind right
/ctrlf 10 /run indicateStop
/ctrlf 11 /run indicateLeft
/echo "2> ▶ ▶"
```

indicateStop.lfs

```
/light ind off
/ctrlf 10 /run indicateRight
/ctrlf 11 /run indicateLeft
/echo "80000ff"
```

autoexec.lfs

```
/button 16 ctrl_f11
/button 18 ctrl_f10
/ctrlf 10 /run indicateRight
/ctrlf 11 /run indicateLeft
```

(Obsolete) "F9 > F10 > F11 > F12 > Off" Toggle

Since 0.7E, you can use the bindable `/status [none/F9/F10/F11/F12/next/prev]` command to cycle through the F9-F12 menus easily. Therefore, this example is kept only for historical reasons.

Shortcut keys used: Ctrl+F12

The basic idea of this script is, that we want to have one button (for example on the wheel) cycling through the F9-F12 menus. To accomplish this, we need five scripts plus an entry in *autoexec.lfs*.

For this script we will use the Ctrl+F12 text shortcut for our cycle, and bind button #0 (whichever button that is on your controller) to execute it. The multiple usage of the /press command ensures that the menu is properly displayed, in case a menu was already opened manually beforehand. For example, if the F11 menu is already open and we would run *viewF11.lfs* without pressing the F9 key in it, it would just close the F11 menu instead, which is something we definitely don't want to happen in the heat of a race.

It will work like this:

- Button 0 -> Ctrl+F12 -> viewF9 -> press F9 + rebind Ctrl+F12 to run viewF10
- Button 0 -> Ctrl+F12 -> viewF10 -> press F10 + rebind Ctrl+F12 to run viewF11
- Button 0 -> Ctrl+F12 -> viewF11 -> press F11 + rebind Ctrl+F12 to run viewF12
- Button 0 -> Ctrl+F12 -> viewF12 -> press F12 + rebind Ctrl+F12 to run viewOff
- Button 0 -> Ctrl+F12 -> viewOff -> turn off menu + rebind Ctrl+F12 to run viewF9

viewF9.lfs

```
/press F10
/press F9
/ctrlf 12 /run viewF10
```

viewF10.lfs

```
/press F9
/press F10
/ctrlf 12 /run viewF11
```

viewF11.lfs

```
/press F9
/press F11
/ctrlf 12 /run viewF12
```

viewF12.lfs

```
/press F9
/press F12
/ctrlf 12 /run viewOff
```

viewOff.lfs

```
/press F10
/press F9
/press F9
/ctrlf 12 /run viewF9
```

autoexec.lfs

```
/button 0 ctrl_f12
/ctrlf 12 /run viewF9
```

(Obsolete) Brake balance adjustment (increase/decrease)

Since 0.7E, you can use the bindable `/liveset bbal += 1` command to adjust brake balance (and front-, rear arb values) on the fly easily. Therefore, this example is kept only for historical reasons.

Shortcut keys used: Ctrl+F11, Ctrl+F12

A simple, one-key adjustment of brake balance either front or backwards, without having to navigate through the F11 menu and using the arrow keys to adjust. After each brake bias change, it also goes back to the F9 menu to show tyre temperatures (as most racers use this while racing). Change buttons 6 and 5 to your preference to assign the functions. Echo command messages are just informative, can be deleted. Be aware, the key binding runs every time LFS is launched, therefore when changing controller devices (when button number mapping might change), make sure to double check the key number.

bb_increase.lfs

```
/press F11
/press right
/press F9
/echo "1SCRIPT: ^8Brake Balance +1, back to F9"
```

bb_decrease.lfs

```
/press F11
/press left
/press F9
/echo "1SCRIPT: ^8Brake Balance -1, back to F9"
```

autoexec.lfs

```
/button 6 ctrl_f12
/ctrlf 12 /run bb_increase
/button 5 ctrl_f11
/ctrlf 11 /run bb_decrease
```

Categories: [Guides](#) | [Programming](#)