

# **Pilots Flight Log Book**

Including full support for Engineers, Navigators, Radar and Radio technicians and GIB flight crew, recording on all aircraft types

**For release v2.01**

This document forms part of the Pilot Flight Log Book and is :  
Copyright © 1986 – 2019, Jointly Applewood Computers and Vincent B Coen, FBCS.  
along with the Flightlog software.

17 Stag Green Avenue  
Hatfield  
Hertfordshire  
AL9 5EB  
United Kingdom

Phone: 44-1707-263712 – local time hours 12:00 to 21:00 Zulu.

General support email: [vbcoen@gmail.com](mailto:vbcoen@gmail.com)

For sending CSV files/data only use [vbcoen@btconnect.com](mailto:vbcoen@btconnect.com) as it has higher limits for size of file attachments but still only send compressed archive file/s with your surname, initials and license number i.e., coenvb-atpl123456.zip with subject as Flightlog CSV. This helps prevent duplicated file names arriving.

Otherwise use Flightlog in subject line then software or manual as needed along with BUG, FEATURE request or registrations.

Registrations must include: User name, Town and Country as well as your email address that can receive updates as email archived attachments and these will be around 10Mb in size.

Archives used are produced using RAR, ZIP and \*nix's TAR – your choice.

We do **NOT** pass on such details to anyone.

See the section of Appendix R - Registrations in this manual for more details.

Written            Started 6th November 2018  
LibreOffice    v6.02 or later.

1 3 5 7 9 ~~0~~ 2 4 6 8

Release. 2.01.40

Latest update:    18. December 2018 23:39:48  
Doc. Version     74

## Table of Contents

1 Manual Update Record.....	5
2 Introduction.....	6
2.1 General Overview of Flightlog.....	6
3 Pre-Operations.....	10
4 Normal Flight log Processes.....	11
5 Work Flow Overview.....	13
6 Menu 2 – Flight Log.....	16
6.1 Option A & B – Flight log Data Entry and Amend Functions.....	16
6.2 Option C and D – Flight Log Reporting.....	22
6.3 Option E – Analysis & Totals Report.....	23
6.4 Option F – Analysis & Totals Display.....	23
6.5 Option G – Cert of Ext. Analysis Report.....	23
6.6 Option H – Change Log Book Airfield Code.....	24
6.7 Option J – Change Log Book Aircraft Type.....	24
6.8 Option K - Edit Airfield Name.....	25
6.9 Option S - Create Sequential files from ISAM.....	27
6.10 Option T – Import CSV Data and Parameters.....	28
6.11 Option U – Enter User details for Reports.....	28
6.12 Option X - Quit Log Book System.....	28
7 CSV Parameter Structure.....	29
7.1 CSV Overview.....	29
7.2 Testing your CSV parameter configuration with CSV data.....	29
7.3 CSV Prerequisites.....	32
7.4 CSV Configuration Overview.....	33
7.5 CSV Definition Record Types.....	35
7.5.1 Record 1.....	35
7.5.2 Record 2.....	36
7.5.3 Record 3.....	37
7.5.4 Record 4.....	38
7.5.5 Record 5.....	39
7.5.6 Record 6.....	40
7.6 Configuration File Example.....	42
7.6.1 Example CSV data file.....	43
Figure 12 - Post CSV data import.....	46
7.6.2 Delimiters and data formats.....	46
8 Appendix A - Keyboard operations or actions available on data entry.....	47
9 Appendix B – Warning and Error messages.....	48
9.1 Overview.....	48
9.2 SSDs and Garbage Collection.....	48
9.3 Warning and Error messages produced by Flightlog.....	49
9.3.1 Extra guidance on specific messages which detail fixes for the source file.....	56
9.3.1.1 Changing settings for CSV field limits.....	56
9.3.1.2 Changing settings for Aircraft or Airfield Tables.....	56
9.4 File handling status codes and their meaning.....	58
9.4.1 File Access Error numbers.....	58
9.5 RDBMS error codes.....	59
9.5.1 MySQL SQL Status messages.....	59
10 Appendix C – Installing the Cobol Compiler.....	60
10.1 Install the Cobol compiler.....	60

10.2 Set up and build the Cobol compiler.....	63
10.3 Test the Cobol compiler.....	63
10.4 Validating the Cobol compiler.....	66
11 Appendix D - Flightlog Build and Installation.....	67
11.1 Installing the Flightlog executable or binaries.....	67
11.2 Installing other optional Packages.....	68
11.3 Compiling the Source code.....	68
11.4 Installing Flightlog.....	69
11.5 Preparation for running Flightlog.....	71
11.6 Running Flightlog Manually.....	71
11.6.1 Parameter 1 (P1):.....	72
11.6.2 Parameter 2 (P2):.....	72
11.6.3 Parameter 3 (P3):.....	73
11.6.4 Parameter 4 (P4):.....	73
11.6.5 Setting of LC_TIME for your preferred date format.....	73
11.6.6 Running Flightlog.....	74
12 Appendix R - Registering Flightlog.....	75
13 Appendix T - GnuCOBOL v3.0 Install Readme file Contents.....	77
13.1 Cobol Compiler requirements, Instructions for Build and Testing.....	77
13.2 Requirements:.....	78
13.3 Compiler Installation.....	79
13.4 Compiler Development.....	83
13.5 GnuCOBOL v3. after build Cobol 85 Validation Tests.....	84

# 1 Manual Update Record.

Because no manual much like any software will be free of typo's, contain only valid or perfect details or just plan miss information, we have the Manual update facility and log file. Here is a detailed list of all fixes to the manual as found or reported by users or the development team.

So, if you find any, please report them. To help aid this, the first few versions of this manual is only supplied as A4 format so every one is reporting to the same standard least of all when users specify a page within a chapter. A US letter formatted version will be also be issued after three or so months. Hey, it is less pages to print :)

06/11/2018 vbc 2.0	Initial written document with details taken from program specifications, source notes and source code. This version is a major upgrade from releaser v1.0 containing many new features and bug fixes.
20/11/2018 vbc	Added extra crew capacities to P1, 2, 3 by including E1, 2, N1, 2, R1, 2, T1, 2. More text for Flightlog functions.
23/11/2018 vbc	Added help and optional parameter fields. Made it use only one menu. Changed manual to match. This now is for Flightlog v2.01.25.
27/11/2018 vbc	2.01.27 now uses line sequential back up files for ease of use. More layout clean ups.
30/11/2018 vbc	Added comments for install of enscrip and postscript-common for printing as needs ps2pdfwr, ps2pdf and ps2pdf14 to work with supplied script prtpdf. Updated comp-Flightlog.sh.
10/12/2018 vbc	Proof read results. Extra notes regarding CSV data file formats as used with CSV file. CSV is the correct term for CSV. - Must change it here.
12/12/2018 vbc	All references to CDF changed to CSV. Menu screen shot replaced for change of CDF to CSV. BUGS #001 & 002 fixed. Changed CSV data to reflect the config file.
17/12/2018 vbc	Bug #003 finally found, but left in CSV-TEST code. Software rev. now 2.01.40. More content clean ups.
18/12/2018 vbc	Update for Parameter CSV= (.44).

NEEDS TO BE PROOF READ, then as a New USER!!!! -  
Bound to be duplicated text, errors or omissions.

## 2 Introduction

The FlightLog software allows **all** flight crews, Pilots, Flight Engineers, Reo's and other GIB's etc, to record their flight time in a unified location namely their computer, regardless of platform, i.e., Linux, OSX, and other \*nix systems as well as Windows.

This is the only purpose of Flightlog, it does not do other functions such as flight planning, weight & balance etc. as there are plenty of tools around if needed, to help do these functions.

The OS (Open Source) versions allows users to build and install Flightlog having also installed (or built) the free Cobol compiler GnuCOBOL. This is free as in beer, too I.

For detailed instructions on doing this, see chapter 10 Appendix C – Installing the Cobol Compiler page 60 onwards along with instructions for installing Flightlog executables and other useful tools for the reports. Note that Appendix T only has the original README type files supplied with the compiler. So there is no need to actual print out these pages when printing this manual – just specify to acrobat or other pdf printing tool to omit these tail end pages.

### 2.1 General Overview of Flightlog

Flightlog is a user specific tool and uses three primary files per user in one directory namely :

1. Aircraft : To record aircraft types flown, last registration flown and the last flight date.
2. Airfield : To record the ICAO coded airfield/airport along with the name & date last used.
3. Flitelog : To record all flight details including date, start and end times, airfields used, flight capacity, captains name and other details including flight by IFR and multi-engine etc.

These files, like a paper flight log book are for one user / crew member only, and therefore cannot be shared.

Clearly these files, record the aircraft and airfields used as entered into Flightlog over the many more that exist in the world. Note that the airfield file can be pre-loaded with every airfield in a country, continent or even the world but most if not all pilots only use a very small fraction even within one country so this is not included as an option. That said, a problem would exist if this is done, in that many countries use an alphabet that is outside the normal A – Z, a – z range i.e., accents, and would require special programming to cope let alone the special set up for both the monitor and printer.

For the airfield database I use, all names are actually converted to just using the normal A-Z, a-z letters as the usage of name with these accents as explained would create issues with screen displays and more importantly printing unless your system can cope with both as a normal font set (without the use of two bytes used for each character) and this software does not support that.

These three files that have the extension of “.dat”, hold their data in an indexed file structure to keep the speed of access as fast as possible by only accessing the required record.

Disk storage wise, the requirements is very modest :

airfield file record size is 32 bytes and holds one airfield.

aircraft file record size is 24 bytes and holds the details of one aircraft type.

flitelog file record size is 112 bytes and hold the details of one flight.

A byte is the exact equivalent of one character.

So for a 30,000 hour pilot say with the same number of flights (it would be less, and very much so, for long haul) is under under 5 MB (megabytes) with the other two only adding at most, say 2 Mb. There is also the size of the indexes but lets assume an extra 50% so all in all, total size would be under 11MB. A very small amount that could be backed up on to a memory stick, SD card or even your mobile phone or tablet.

In addition to the use of these three files, there is a menu option that allows you to copy each of them to a line sequential file which allows all three files to be transferred to another computer system, platform or even Cobol compiler with minimum issues. These sequential files can be examined if really needed, using any text editor or system command like 'less' or 'cat' in \*Nix (Linux, Mac OSX, BSD or Unix) or 'type' in Windows or via a file manager but you must not change them. Likewise if running Flightlog in a mainframe environment there are also plenty of tools, one can use to view the records.

Normally these sequential files do **not** need to ever be read by a user – they are there for two reasons, back up or as migration, should you ever wish to move to another system or more likely the sub system used with the GnuCOBOL compiler that handles indexed files of which there is at least four different one's, all with their own ways of storing data and they are **not** compatible.

All of these files, should be backed up after a session of data entry and it is recommended to use an archiving program such as ZIP, RAR or for \*nix systems, tar. The archive name should be say some thing similar to backup-yyyy-mm-dd.ext where ext is zip, rar, or tar and the date is date of last flight recorded. Here, see the sample script for running the Flightlog program as used in the UK for ideas, but this is only for Linux and for use in the UK (United Kingdom) as the setting for LC\_TIME shows, so that the date format used for date entry and for display and reports will be dd/mm/yyyy. The date format for date entry and display is not necessarily the same as used for the back up file name, see later.

```
—
#!/bin/bash
# 18/12/2018 vbc - v1.01 Added backup directory if not exists.
#                               and do backup at end.
#
if [ ! -d backups ]; then
  mkdir backups
fi
#

export LC_TIME=en_GB
../Flightlog
zip -9 backups/backup-$1.zip *.dat *.seq
exit 0
—
```

**Note:** For the USA, use LC\_TIME=en\_USA for mm/dd/yyyy in place of en\_UK or blank for the yyyy/mm/dd date format.

Here \$1 is an added parameter to the script run-Flightlog.sh with the date of last flight in the form of yyyy-mm-dd or if you wish dd-mm-yyyy or any other variation. Here, it is run as:  
run-Flightlog.sh 1998-09-21 (but could be say run-Flightlog.sh 19980921).

This produces a backup file with the name : **backup-1998-09-21.zip** in the backups directory. The reason for this format is that a directory listing will show them in date order.

If the parameter is not included when running Flightlog using the supplied script, the back up file name will be backup-.zip. If there is not a back up directory called backups in the directory where Flightlog runs from it is created.

Remember the rule that there are two types of users – One that has lost data and one that that will (at some point).

It cannot be stressed enough to **always** make these back ups in case of power, or hardware problems and these do happen let alone just typing issues or in the rare case of program bugs (defects). Program defects will always exist in one form or another despite heavy testing as all combinations tested does not always match what a user does. However that said, testing has been conducted in functional (including use-cases), system and UAT (user acceptance testing) to help ensure bugs are minimised but there is no such thing as bug free software even if used by many and over very many years as many mainframe users can testify.

The Flightlog software supports Private, Commercial and Military pilots as well other crew members such as flight engineers (yes they are still used), radar targetting officers also known (mostly by pilots) as gib's, navigators, radio officers etc., all with their own capacity designators. Therefore for crew moving over say to becoming pilots etc, the same Flightlog system can be used to record all of their flights.

For reports, a listing of all flights or between two dates with an option of totals by month, lists of all airfields visited, all aircraft types flown both with dates last visited or used. There is an option to restrict the report of airfields to only those that have been entered in a flight log record, i.e., you have used it and sorted by the last flight date to that airfield.

In addition, an optional short report containing CoE (certificates of experience) for the following periods, 1, 3, 6 and 13 months which can be passed to the examiner as a separate sheet along with an aircraft type report when needed. This is created after accepting a start date for which Flightlog looks at flight records and counts backwards for the various month periods.

So a CoE for one month counts backward from the entered date for one whole month. Likewise for the other periods producing a report line for each one in turn. Saves trying to count the totals in your log book/s. Just provide the CoE report page along with the aircraft type report so that the examiner can easily check and then sign and stamp your license or paper log book.

Of course this does not avoid the requirement for type ratings every six months (I wish) but as it records flights by aircraft types it does help to validate such.

Flightlog can also be used to record flight simulator training conducted in an approved trainer or software package and here use the aircraft type to show the simulator used along with the remarks field to specify the process or procedures conducted.

Now 'Approved' does depend on your licensing authority and you need to contact them for what is approved but 3 and 2 axis simulators should be there. For many authorities, single axis D-Link (D-4) is also ok. Software based one's may be a bit more difficult as does depend on what is being practised and using what facilities etc and here I would suggest that only using a keyboard is not going to cut it, so a wheel/stick and pedals are kind of important as well as a printed log / report of what processes and procedures was conducted and these should be retained. Very useful for refresh of specific airport IFR arrival, departure procedures.

Likewise emergency procedures for specific a/c types although a 3 axis simulator would be some what better :)



Flightlog is supplied as Open Source, that is the source for the program is included and is supplied totally free of any charges.

If you cannot find an archive containing the executable copy of Flightlog then you can install the free GnuCOBOL compiler as sources, build and install it on your computer then compile / build Flightlog and install on your computer ready to use. See Appendix D and onwards for detailed instructions to do this. Note that any updates to Flightlog will have to be compiled again before use, if not getting the executables from the website (when available) but it will take all of 5 seconds to do.

If you run under Linux or similar, one option instead of downloading and building the Cobol compiler is to install it via your software manager, which if nothing else will install the extra packages required for it. If the version of the installed Cobol compiler is very old such as version 1 releases, you can install it (which will also install the required packages) then uninstall it which should leave the extra packages present and therefore saving some extra steps.

Note that here, the current full release is v2.2 and that v3.0 RC2 is available from their website see 10 Appendix C – Installing the Cobol Compiler page 60 for details.

While Flightlog is supplied as free (as in beer) and open source, it is appreciated that some users would prefer to have a full, update and support service just like any other piece of software they purchase. This is where you can register your use of Flightlog with us and for a small fee get updates and support when required. See the inside front cover of this manual for the contact email addresses for doing so.

All users are encouraged to register their use of the software regardless of requiring the support and update service and doing so will show us the number of users and the computer platforms (Linux, OSX, Windows, etc.) used.

See section 12 Appendix R - Registering Flightlog on page 75 that covers this, in more detail.

### 3 Pre-Operations

In order to help validate data entered using Flightlog, your primary log book/s need to be totalled on each page with counts shown for brought forward and carried forward figures, e.g., totals at the top and bottom of each page for all flight capacities such as P1, P2 etc, day and night, instrument/IFR & multi engine etc, if so rated but you should be doing this for your own usage to avoid mistakes and for CoE records anyway. Doing so, you can verify the entered data and last entry report totals against your log books to confirm data has been correctly entered. If not, it just a case of selecting the menu option of 'Amend Log book' and change the element/s in error having spotted what the discrepancies are, and most likely they will be an equal lower total on one column and the same amount higher in another, again more on this later in this manual.

These above mentioned totals, are produced by Flightlog on the report listing by month (menu option D) and at the start and end of each report page, so you can verify all flight data entered to match your paper log book/s. Where you have more than one log book, only enter them one at a time and in date order when possible by log book if you can, i.e., starting with the oldest book first. It does not matter if dates are not in order other than to help confirm valid data by comparing the totals by log book page and option C, D (or F to just display) reports, see figure 2 for an example. Personally, I check after each completed log book page to validate the totals as I have found it saves a lot of time if I do make a mistake, so I stop data entry, run option F and if needed D, then use another virtual screen and examine the logbook.rpt file by going close to the end of the report, for the last set of totals and use this to match against the paper log book. Often using option F to display the closing totals is enough to indicate if there is a problem and then, if needed use D for a full report. However it does sometimes show up errors in my mental arithmetic during log book totalling :). Maybe, I should stop entering flight info on a long, getting boring flight?

Some minor but important points, do not do major data entry when tired or when getting that way so just do 1 – 5 log book pages (max) at a time. This I found, when entering my own log books into the system and yes I have over five books but for two of them, one is always in one of my two flight bags. I have one for (private) VFR flying and the other for IFR which for me is the norm. I have also been known to use flying diaries which has log book pages at the back (from IALPA and others, just for rough notes on flights) which at some time I then update my main log books when I remember, or more likely when I have a check coming up. That said, it is now a lot easier just maintaining the Flightlog data but I do believe in double record keeping as I have lost log books in the past and no doubt will again. Mostly moving homes !

## 4 Normal Flight log Processes

All Flightlog functions are selected via the menu.

Menu options only require the letter being entered, i.e., no return key required. The keys such as the Escape and function keys F1, 2, 3, 4, 5 and 10 are used within data entry or amend. There are other options within data entry or amend which are described in that section. These function keys help speed up data entry. Note that the screen shots can be of different builds. Note that the program name and version number is shown top left of the screen along with the current date and time as set in your computer. This version number must always be used to report a defect / bug in the program along with your operating system (Windows, Linux, OSX, other \*nix, OS/2 etc.).

Now to cover the menu options and what they do.

Looking at Figure 1 – Menu, you can see that it provides access to the following functions where the first one is the primary one and the others, on an as needed basis

```
LOG BOOK (2.01.31)          Flight Log Book Program  at 17:47:48 on 12/12/2018

Select one of the following by letter :- [ ]

  (A) Enter Log Book Data
  (B) Amend Log Book Data
  (C) Log Book Reports
  (D) Log Book Report & Monthly Analysis
  (E) Analysis & Totals Report
  (F) Analysis & Totals Display
  (G) Cert of Ext. Analysis Report

  (H) Change Log Book Airfield Code
  (J) Change Log Book Aircraft Type
  (K) Edit Airfield Name

  (S) Create Sequential files from ISAM
  (T) Import CSV Data and Parameters
  (U) Enter User details for Reports

  (X) Quit Log Book System

Copyright (c) 1986-2018 Vincent B Coen FMBS, Applewood Computers, Hatfield, UK
```

Figure 1 Menu.

This is the list of all menu functions:

- |     |                                    |  |
|-----|------------------------------------|--|
| A - | Enter Log Book Data                | This is used to enter your flight record data.                               |
| B - | Amend Log Book Data                | Here to amend a flight log record created from (A) or via CSV imported data. |
| C - | Log Book Reports                   | Simple report of all flight entries or date selective.                       |
| D - | Log Book Report & Monthly Analysis | Like (C) but includes monthly sub totals as well aircraft &                  |

- airfield reporting.
- E - Analysis & Totals Report Like (D) but without flight data, i.e., Analysis only.
  - F - Analysis & Totals Display As for (E) but displayed only. Useful to validate totals against your paper log book.
  - G - Cert of Ext. Analysis Report  
Will produce CoE (Certificate of Experience) details for 1, 3, 6 and 13 months if flights go back that far. If any do not fit within these periods the specific report line will be omitted.
  - H - Change Log Book Airfield code  
Changes **all** flight entry records replacing one airfield (ICAO code) for another one & update the airfield record.
  - J - Change Log Book Aircraft Type  
Change **all** flight entry records for a specific aircraft type as well as the aircraft file. Useful when you entered it wrong during data entry and only spotted it later, i.e., PA28-235 should have been PA28-236.  
Yes, got that tea shirt and likewise for the odd heavy :)  
Wonder why this was included ?
  - K - Edit Airfield Name  
For one or more airfields to Amend (the name), Insert, Delete, View or List one or more to printed output.
  - S - Create Sequential files from ISAM  
Create transferable and readable copies of all three data files (indexed) to line sequential data format.
  - T - Import CSV Data and Parameters)  
Import a company supplied file containing your flights flown over a period of time provided in CSV format.
  - U - Enter User details for Reports  
Although this information will be requested after you request options D through G this acts as one place to do the same. Limited use but could be handy.
  - X - Quit Log Book System  
Closes all files included print report and return to the O/S (operating system) prompt.

The Escape key does the same as the 'X' key.

All reports are cumulative e.g., one after the other up to the point of quitting Flightlog.  
More information of these options in detail, later.

Well, that is the top level overview of the menu, now to get in to details for each menu option but lets cover the work flow required to enter data in to the system before doing so just in case you are eager to get going.

## 5 Work Flow Overview

Now a description of the work flow starting at the beginning, start a terminal program such as konsole (although there are many other you can use).

Adjust the terminal programs settings to be 106 characters wide and 24 or more deep and the longer the better as many of the data displays will make use of the extra depth and this is done by using the mouse with the cursor at the left and then bottom of the box and dragging it over until you get the right size which will be displayed while you drag the edge.

Create the directory that will hold your flight log data such as "Flightlog".

Move to that directory.

Starting the program by typing Flightlog (with or without any parameters) followed by enter.

For windows do similar, using a terminal box via Start->run command, and you will be (like Linux) at your home directory so, for the first time so follow the above steps

For Linux or variations (including Mac OSC, BSD or any other \*nix system)

```
mkdir Flightlog  
cd Flightlog
```

```
Flightlog
```

Note above commands might well be different for your version of Windows.

For the actual steps to run Flightlog (under Linux) see Preparation for running Flightlog on page 71 for detailed instructions.

Right, for now and assuming you have read the above mentioned section, just use the name Flightlog and return.

Warning: If the terminal screen sizes is not => 106 wide or length => 24 then you will get an error message SY010 or SY012 depending on the error, so before you do hit return adjust the width and/or length then press enter and the program will re-check the settings.

The menu will appear see figure 1.

Select A to enter your flights with your paper log book by your side turned to the first flight log page.

For details of data entry see Option A & B – Flight log Data Entry and Amend Functions on page 16.

Continue adding your paper log book entries but after a page or two and to validate the information entered, quit Data Entry when you are asked for a new flight date having entered the last entry on a page, by hitting the ESCape key. Now select option F, you will see a display of totals followed by aircraft statistics see Figure 2 for an example.

Grand	1406				1056	119	218	11	355	222	Hrs			
Total	05	Totals	Carried	Forward	45	30	50	00	50	30	Mins			
AA5A		6.00		28.25	1.40			1.20	37.25	27.20				14/03/1987
BE200					1.22				1.22		1.22			19/03/1989
BE23		5.50							5.50	2.35				09/08/1993
BE60		13.40			1.30				15.10	2.35	15.10	0.35		27/10/1993
BE90		0.42							0.42		0.42			12/05/1988
C150		20.05		16.12	2.00		0.30		38.47			4.55		13/01/1991
C152		44.29		54.35	2.20		3.00		104.24	24.57		5.20		09/10/1986
C172		57.24		1.50	3.50				63.04	6.40		26.55		14/04/1991
C340		134.03			50.58				185.01	35.05	185.01	138.05		13/05/1990
DR400		2.00							2.00					21/04/1986
G159				1.00					1.00		1.00			10/10/1981
J5P		0.25							0.25					02/07/1986
M20C		29.00							29.00	20.50		18.25		01/09/1986
MS894		1.45							1.45					05/06/1986
PA18		0.45							0.45					15/04/1973
PA23		6.35			1.00				7.35	0.50	7.35			03/08/1989
PA28-140		417.41		6.00	47.45		6.10		477.36	41.07		219.35		12/08/1991
PA28-160		13.25							13.25	1.20		12.05		25/10/1989
PA28-161		2.45							2.45			2.45		08/04/1990
PA28-180		10.40		2.55	1.25				15.00	2.45		3.45		03/03/1991
PA28-236		237.10			96.10				333.20	176.55		188.10		25/06/1989
PA28R180		7.30			7.10				14.40	5.40		5.25		05/03/1989
PA28R200		3.36							3.36					01/12/1981
PA30				7.35					7.35		7.35			18/08/1985
PA31-310		2.50							2.50	1.51	2.50			18/12/1985
PA32		4.00			1.10				5.10	1.15		3.00		24/07/1989
PA32-260		2.50			0.30				3.20	3.00		3.20		04/08/1987
PA34		1.15							1.15		1.15			17/06/1987
PA38		16.15		0.40					16.55			14.25		21/09/1998
R22				0.18					0.18					03/07/1986
SF205F		0.20							0.20					03/07/1986
TB9		10.15							10.15	1.05				30/04/1988
U206		3.30							3.30					09/06/1983

Hit return when ready to continue

Figure 2 – Menu option F – Analysis & Totals Displayed

Now examine the display looking at the first two lines and compare the totals shown with that of your log book page and it should match. As you can see, you have the following values shown from left to right :Grand Total (all your flight hours), then the break down as Day – P1, P2 and P3 followed by Night – P1, P2 and P3 totals, IFR, Multi-Engine time and Instructing and for these last three, they are already within the preceding six totals as it is just total times. For commercial flight crew, you may well not need to keep track of IFR flights as all flying is as such. In which case do not make any entries for it. But for flying on a private license it is useful information when submitting any CoE's elements of the report for an examiner.

Note that If you entered non P(ilot) times, such as E1, 2 or N1, 2 or R1 or, T1, 2 etc (instead of capacity P1 – 3) then these values are so related instead. This means that, i.e., E1 is the 1st and 4th figures, E2 is 2<sup>nd</sup> and 5<sup>th</sup>. etc.

Now back to the displayed total figures, by comparing your paper log book totals with these, you can spot immediately if they are the same.

If not, it is more often than not likely to be the fact that you entered a flight as P1 instead of P2 or P3 or more likely, entered it for day when it should be night or the other way around or a mixture of the four and that you forgot that one or more flights have a split between day and

night and you did not enter it that way. So from the menu select option C and print out your flights. Now using the 'less' command or a text editor to look at the report output file, logbook.rpt So find the flight/s in error and then in Flightlog select B for Amend and change the incorrect record entries. During Amend mode you will be asked for the date and time of the flight and you will see the screen refreshed with the flight details with the cursor back on Date. This allows you to change all entries including the date, start and end times along with all other flight data.

When you have amended all the flights in error, come out of Amend mode (Escape key) select C report once more, then quit the program and check the report log file (logbook.rpt) to confirm the totals now match up. You can of course use option F again to just check the totals and only then if there are still total errors select a C or D report. This does of course assume that you have, for each paper log book page added up each column, added the (top) brought forward totals and create at the bottom of the page the Carried Forward totals. These **must** match up before entering any more data.

After a few pages of this, you will soon know when typing, if you have made a mistake during entry and you can always go back to the field in error by pressing the Field back key which is F10, just repeat the F10 key until the cursor get to the field in error.

Now you can re-enter the correct data. If you are at one of the Pn entries where the field is non zero but the correct value is zero you can type the value 00.00 or just Z or z in the first position followed by return and the program will change it to – yep 00.00.

Correct, been there, and got the plantation :)

Many of these short cuts have been added after I encountered time consuming typing and wanted ways to reduce it.

As you are entering new flights whenever you enter a new airfields ICAO code you will be asked to confirm if the entered code is correct (if not just Escape) otherwise it will ask for the airfield name.

Likewise when entering a new aircraft type it will again ask you to confirm it is a new one and not incorrectly typed (the Escape key will return to the aircraft type field) otherwise you will get asked some question regarding the type like M/S multi-engine or Single, Complex type (all Multi engines are complex so this question will be omitted) but for singles fixed gear & props its N (or space) instead of Y for any other facilities such as retractable gear and/or variable pitched propellers. This is, as applies to the Civil Aviation Authority, but might be different in your country or even the UK in the future. It is there for more meaningful information when printing out a set of CoE's along with the aircraft types.

If you find that you made a mistake in entering the Aircraft details you can go into the option that will allow a global change for **all** flight entries should it be needed otherwise just amend the record in error.

Well, that covers “quick lets get it going” and enter flight data, so now for a detailed look at each of the menu options.

## 6 Menu 2 – Flight Log

### 6.1 Option A & B – Flight log Data Entry and Amend Functions

From the menu, select A, after which the data entry screen will appear.  
You now get the screen as shown in Figure 3.

```
File Edit View Bookmarks Settings Help
LOG BOOK (2.01.31) Log Book Data Entry at 18:02:35 on 12/12/2018

Flite Date A/C Type A/C Reg Captain From ICAO To ICAO Start End
[21/09/1998] [abcdefgh] [Gabcd] [AAABBBCCDDDEEE] [YYYY] [XXXX] [99.99] [99.99]

-----Day-----
M/S Cap P1 P2/3 P1 P2/3 IFR ----- Inst Remarks -----
[x] [xxx] [99.99] [99.99] [99.99] [99.99] [99.99] [

-----Night-----
M/S Cap P1 P2/3 P1 P2/3 IFR ----- Inst Remarks -----
[x] [xxx] [99.99] [99.99] [99.99] [99.99] [99.99] [

F1=View Afld, F3=View Acft F10=Prev field, Esc=Quit
Shell
```

Figure 3 – Data Entry.

Note that the Amend screen is the same, other than the word Amend instead of Entry.

Hopefully the screen layout is clear. It consists of a title line which provides the program name and version (you will need this if reporting any issues). Next is two blocks for data entry consisting of a heading line then the first data entry line, a two line gap (for caution or information messages) then the next line headings, followed by a data line.

Now the cursor is on the first of two data entry lines waiting for the date in your chosen format as defined by the system variable LC\_TIME. If this is wrong, press escape twice to get back to the prompt. There might a second or two delay depending on the speed of your computer.



If this is the case, you need to look at your system setting for LC\_TIME but if you are based outside the UK and USA and do not use Unix format then you need to run the supplied run-Flightlog.sh script / batch file having changed this variable using a text editor first.

However, lets check if you have a setting for this environment variable, so in a terminal type the following :

```
set | grep LC_TIME
```

You will see something along the line of :

```
LC_TIME=en_GB.UTF-8
```

Here is the setting for the UK and will use a date format of : dd/mm/yyyy

If this is not set at all, the results from the set command will be blank and you will need to set up your locale but most if not all Linux distributions do set this up having asked for your country location.

The problem arises if you are not in the UK or USA and do not use these two locale settings but still use one or the other date formats, in which case use the run-Flightlog.sh script having changed it to one of the two options as now detailed. Note that if you use the international or standard Unix format of yyyy/mm/dd, again follow below:

Using a text editor (only) change the line starting with export LC\_TIME=  
Change the word that come next on the line  
to 'en\_UK' for dd/mm/yyyy format  
or 'en\_USA' for mm/dd/yyyy format  
or any other for the default Unix one using yyyy/mm/dd. Otherwise set it as:  
LC\_TIME= E.g., empty value.

Save the file, and restart the program by running the script : run-Flightlog.sh  
If you get an error message about it cannot be started, then type (in Linux) still using the terminal:

```
chmod +x run-Flightlog.sh
```

and this will make it executable, so now retype the script name again. It should now work.

So back to the data entry screen by having started Flightlog from the script or directly, then having selected option A.

So fields from left to right should be self explained, but it is in order:

Flite Date, A/C Type, A/C Reg, Captains (name), ICAO from (airfield), ICAO to (airfield), followed by the Start and End flight times (as a 24 hour clock) and here most pilots use the local time converted as GMT/UTC. As are flight planning documents.

Abbreviations used here are: Flite = Flight, A/C = Aircraft, Reg = Registration number, Captain is he's| her name as surname space initials, ICAO = official icao code designation for the airfields name as four alphabetic characters, start and end times in format of "hh.mm".

Date format will be in your required format (reports and displays will be the same) as 10 chars (characters) and instead of the '/' symbol you can use a comma, period, '\ ' or any thing else as it does not check them (as of v2.). If you have other records present on file when starting up Flightlog, the date shown will be of the last (latest) flight you have entered, just in case you did more than one flight for the same date and as a reminder of when you last entered information into the Flightlog system.

The date is checked to be a valid date, i.e., February has 28 or 29 days, depending on the year etc same for all the months.

Next is A/C type and this allows for 8 chars although most only need 4 however in my days of flying PA28's I found it is a wide range of types from very simplex like the PA28-140, PA28-180 to the more complex like PA28-236 Dakota (var. prop), PA28R180, PA28R200 (var prop and retractable gear) so I did use all 8 chars for it. Most of the other types was kept for 4 chars though and for commercial A/C, it is almost all 4 char ICAO coding or manufacturers coding although models within types are not very well supported using these coding standards. It will take the type from the last flight and display that allowing you to change or just press enter to accept it.

If it is a new type you will be asked to confirm it (or you can re-enter the type code having pressed the escape key). If new, you will be asked some questions such as M/S (Multi or Single engine) and if it is a Complex type (all Multi engine's are complex) on completion of these questions it will save the answers for storing on the aircraft file after completing the rest of the flight data for this flight. See figure 4. This also is the same for a new airfield, only here it asks for confirmation and then the Airfield name. See figure 5.

```
LOG BOOK (2.01.31)          Log Book Data Entry          at 18:02:35 on 12/12/2018

  Flite Date   A/C Type   A/C Reg   Captain From   ICAO   ICAO   End
[12/12/2018] [B874]     ] [Gabcd] [AAABBBCCDDDEEE] [YYYY] [XXXX] [99.99] [99.99]

M/S   Cap   P1   P2/3 P1   P2/3 IFR   -----   Inst   Remarks   -----
[X] [XXX] [99.99] [99.99] [99.99] [99.99] [99.99] [99.99] [99.99] [99.99]

B874      is a new entry. Enter Y if ok, else N to re-enter [y] M or S - [S]

F1=View Afld,          F3=View Acft F10=Prev field, Esc=Quit
```

Figure 4 – New Aircraft type

```

LOG BOOK (2.01.31)      Log Book Data Entry      at 18:02:35 on 12/12/2018

  Flite Date   A/C Type   A/C Reg   Captain From   ICAO   ICAO   Start   End
[12/12/2018] [B874]     ] [GBJYN] [SELF CCCDDDEEE] [EGAA] [XXXX] [99.99] [99.99]

  -----Day-----      ----Night-----      Inst
M/S   Cap   P1   P2/3 P1   P2/3 IFR   ----- Remarks -----
[X] [XXX] [99.99] [99.99] [99.99] [99.99] [99.99] [

EGAA is a new entry. Enter spaces to re-enter or give name
[ ] Or to re-enter, press ESCape or F10

F1=View Afld,      F3=View Acft F10=Prev field, Esc=Quit

```

Figure 5 – New Airfield

Now Flightlog will take the A/C Reg., from the last flight of that a/c type if flown before and will remember it by a/c type, all to cut down typing, so amend it or press enter to accept it.

An important point regarding the registration, which is nice and simple for all countries other than the USA and here, do **not** enter the first letter 'N', just the other five digits. Don't worry, it will appear on the reports in full.

Now the Captain's name, again it will use the name from the previous flight so if it is the same, just enter or type in the full name or if you are in P1 Capacity, just type S and a space then enter, it will replace that with the text "SELF". Correct, I have been known for being lazy, when typing a lot of entries in over a day or week when I was also a flying instructor, which tends to be a lot of flights per day. It is anything to save more data entry.

Next four fields, so the ICAO code for the From and To airfields, again it will display the one from the last flights "To field". So if it is wrong change it and it is then copied it to the To field letting you press enter for correct or just change it, if needed.

Next is the Start and the End flight times as 24 hour clocks normally as GMT or Zulu. If the flight lands after midnight it will handle that correctly but not so sure if you are flying for over 24 hours. Sorry, not had that as an issue as managed to avoid it, even flying direct across the pond in twins, as it is only 10 – 15 hours max (yes extra tanks in cabin).

So now the next data line with fields M/S, Cap, Day P1, P2/3, Night P1, P2/3 followed by Inst. IFR and Remarks.

M/S is preset for you, based on the a/c type you entered although you can over ride this.

Cap = Now the first of the quick keys :

I = P1I for flying instructor, this is the letter I or i

T = P1T for flight trainer (commercial) Training Captain and this is the letter T or t.

This field will repeat the capacity of the last flight to help speed up typing so if the last flight was as P1 and this is P1I just enter I and it will replace it correctly. Same for P1T.

You can enter any of the following combinations: P1, P2, P3 (If PUT entered it is changed to P3), E1, E2, N1, N2, R1, R2, T1 and T2 are all valid as well as P1S (P1 under supervision like a checkout for A/C or airport , route etc).

Note that from the above, the first letter classification means:

P=Pilot, E=Engineer, N=Navigator, R or T=Radio operator or for military R=Reo or GIB.

If you enter P2 for a single engine aircraft it will reject it as incompatible with A/c Type coding M/S = S as can't be second officer on a single engine aircraft.

There are a few other checks but they should not appear and relate to military crew. These are being upgraded after version testing by such flight crew.

More for Amend mode but also for Entry, so for each of the Pn and the IFR times it is a five char field in the format hh:mm so enter the leading zeros or if they are already present, use the right arrow key to move over to the correct position. The period must be present for validation checks. Now another fast key option, if the Pn field you are on, has a non zero value but you want it as zero enter 'Z' or 'z' then enter will refresh as 00:00. If you make a mistake after pressing enter, the F10 key will go back one field so doing it twice it will go back two fields and so on. Wait there are more, but these are shown in Appendix A - Keyboard operations or actions available on data entry.page 47.

Namely function keys F1, F2, F3, F4, F4 as well as the already mentioned F10.

These keys become available depending on what you are doing i.e., some only appear on the bottom line of screen for Amend mode like when you can delete the current record (F4)

The others are most always present note the word mostly.

F1 = Displays a list of airfields and escape at the end returns back to the entry screen exactly where you were.

F3 = Display a list of Aircraft and escape is exactly the same function as for F1.

F2 = During Amend will appear and be available to get the next record in date/time order. This will no longer be available after the Capacity data entry sequence.

F4 = Delete the current displayed record in Amend mode only.

F5 = In amend mode and at the point for entering P1 day/night values will save the current record then return to enter date for a new flight. Return here will cancel Amend mode and return to the menu.

F10 = As explained will jump back one field. Can be used many time until back at Date entry.

ESCAPE key will go back to date entry.

The last field is remarks and it is 32 chars long and is for any characters in the range of A to Z, a to z and 0 to 9 along with the standard special keyboard keys like ( ) [ ] { } + = ? < > , . etc. For example for P11, I use coding for the student exercises conducted, then routing information if IFR ex's so use symbol coding like />Fnnn climb to level Fnnn or Fnn\> descend to level etc. There was a stack of others including opinions of flying skills as these had to be inserted in their training records. I am sure you have your own.

Well that ends, Data Entry.

Amend is very similar except that it shows 'Amend' instead of 'Entry' and it starts off asking for Flight Date then start time then having found the record (or you get an error message with a return to Date Entry) to try again. Check what you wanted, as no doubt you typed it wrong.

You then go to Date Entry again, where you can change date and then the rest of the data fields.

If the date or time has changed it will create a new record and delete the old one otherwise it will update the record entry for that flight. It will also update or create a new record for Aircraft or Airfield if a new one has been entered during the amend process.

For both of these, airfield and aircraft, the flight date entry is updated to the date of the flight if the date is later than that currently recorded. This also applies to Data Entry.

Various function keys become valid by text appearing and disappearing that says so, during the course of entry data, this is similar but not exactly the same as Data Entry.

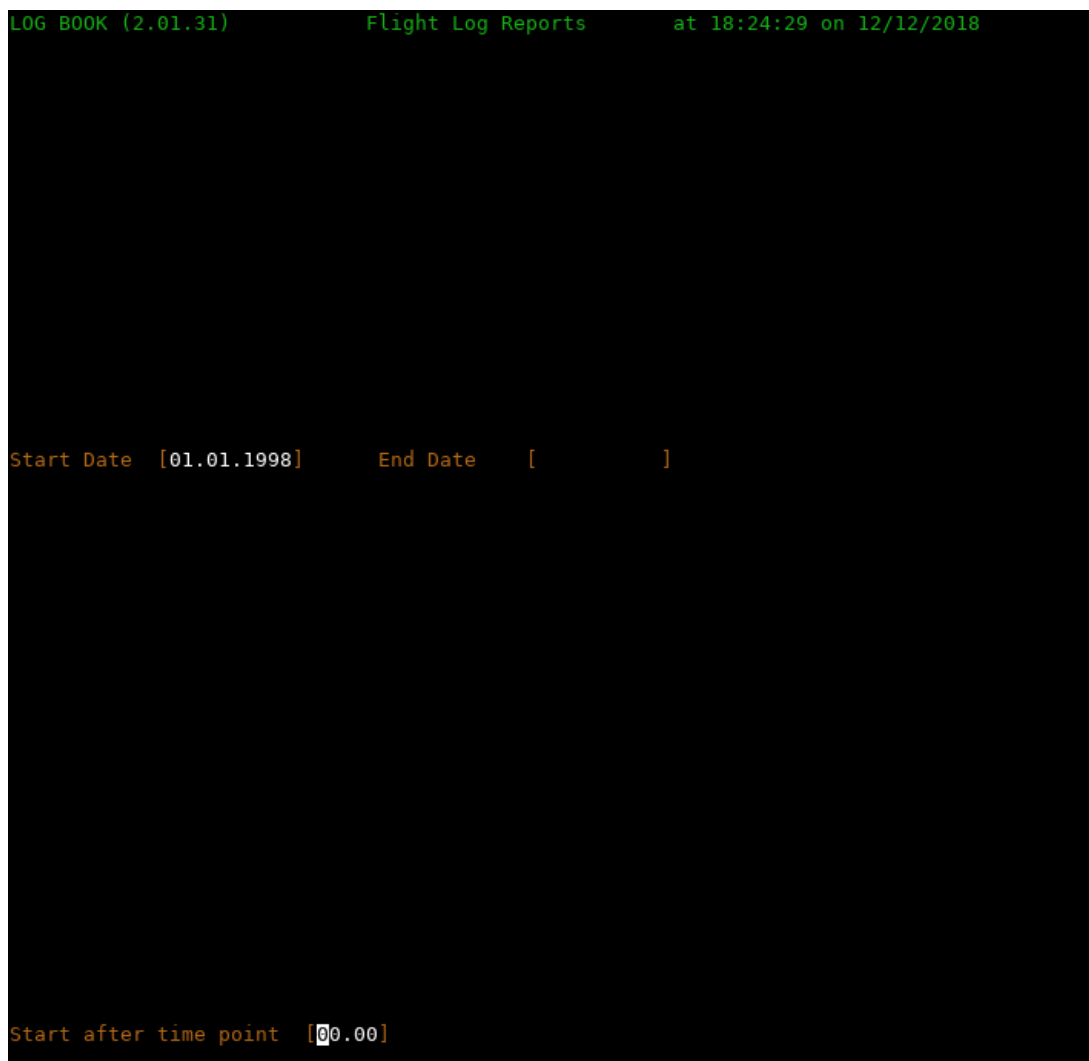
Short cuts for keys are the same, also see Data Entry for information as well as Appendix A - Keyboard operations or actions available on data entry. on page 47.

## 6.2 Option C and D – Flight Log Reporting

Menu options C and D are very similar, in fact the difference is that for (D) monthly sub totals are also printed out along with the Brought forward and Carried forward totals on each printed page. In addition a report for Airfields flown to or from and Aircraft flown along with last dates for both as well as number of flights for each airfield.

On option C and D, the Aircraft list is not produced only the Airfields and that is in last visit date order with one's not visited ignored just in case you have loaded an entire countries worth. This does not apply to menu options E, F or G.

On entry you are offered the option to enter a Start and End Date for the report and for both you can leave blank and it will, for Start Date start at the beginning and for end will continue for all flights flown. If you specify a Start but not an End date you are also offered a Start Time, again you can leave this blank for all from the specified date.



The screenshot shows a terminal window with a black background and green text. At the top, it displays 'LOG BOOK (2.01.31)' on the left, 'Flight Log Reports' in the center, and 'at 18:24:29 on 12/12/2018' on the right. Below this, there are two lines of input fields. The first line shows 'Start Date' followed by '[01.01.1998]' and 'End Date' followed by '[ ]'. The second line shows 'Start after time point' followed by '[00.00]'. The text is in a monospaced font, typical of a terminal.

Figure 6 – Reports Start date & Time

It is there if you need to be selective, to compare the totals against your paper log book.

You are now asked for your name and license number but is free form so you can include any text and this will appear on the top of all report pages as well as to the CoE page if selected, as well as Airfield and Aircraft reporting.

These questions for name and license, will not appear if you have already entered the user data through menu options “(U) Enter User details for Reports” during this run of Flightlog for selection of options C, D or E. It does not save the details after ending the program. In my case, I add the log book and page numbers in for interim reporting.

### 6.3 Option E – Analysis & Totals Report

This provide a report showing the flight totals for Grand, and each of the capacity categories, followed by an aircraft report. This is the same as produced for option F  
Handy for a quick totals check against your log book.

It does request the name and license number first for the Aircraft page/s if not previously entered. This will produce a lists of all Airfields, visited or not in ICAO code order.

### 6.4 Option F – Analysis & Totals Display

This provides as a display exactly the same as option E. See Figure 2.

If the displayed data exceeds a screen then you are requested to press enter for the next block of data otherwise return for the menu. It will not ask for pilot details as in option E. Useful to see on screen, the totals to compare against your paper log book. This will produce a lists of all Airfields, visited or not in ICAO code order.

### 6.5 Option G – Cert of Ext. Analysis Report

This option like the other reports is added to any existing reports will provide, assuming there is flight records going back far enough give statistics for 1, 3, 6 and 13 months for your log book or licence entry stamps starting backwards from the supplied date that you will be asked for as in the example below here using – [11/10/1987]. Note that if your flights do not go back beyond a CoE point where the totals are all zero this report will finish.

The Date requested will be in the format as all dates are, from the environment variable of LC\_TIME as discussed earlier in the manual.

It always starts with today’s date so amend it as needed. This is the date used to count backwards for the required flight data.

Here is an example of the output :

```
LOG BOOK (2.01.17)          Cert. of Experiences for          21/11/2018  00:43:54  PAGE  1

The following to be read in conjunction with the Aircraft report page/s
Active date used for all Certificate of Experience data is 11/10/1987
```

CoE Type	P1	DAY FLYING P2	P3	Inst. Flts	P1	NIGHT FLYING P2	P3	Inst. Flts	TOTAL	(IFR	Single	Multi	Inst.)
1 Month	33.50	0.00	10.10	21	20.50	0.00	1.00	8	54.40	37.20	54.40		11.10
3 Months	71.45	0.00	42.55	72	22.00	0.00	2.10	12	93.45	43.10	93.45		45.05
6 Months	143.55	0.00	100.35	153	29.10	0.00	3.25	18	173.05	54.50	171.50	1.15	104.00
13 Mths	209.40	0.00	134.15	228	54.10	0.00	10.00	41	263.50	71.25	262.35	1.15	144.15

Figure 7 - CoE report

Sorry about that, had to make the font 6.5 pt so it fits on a A4 page.

## 6.6 Option H – Change Log Book Airfield Code

As discussed earlier, this option will allow you to change an Airfield ICAO code for another, replacing the original in the Airfield data file, as well as all instances of where it is used in all flight log records. This mean that if you only have an error in a few flights but not all of them use Option B – Amend Log Book Data instead.

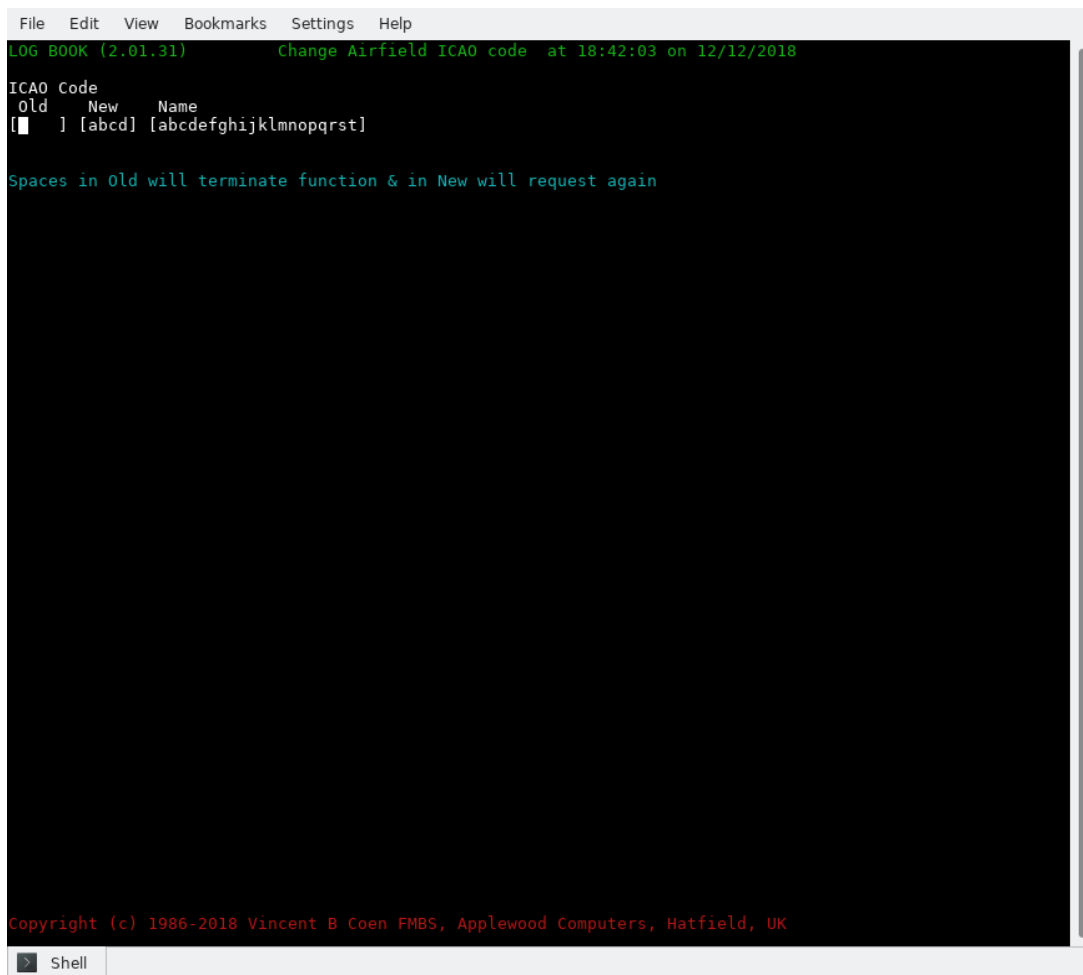


Figure 8 – Global change of Airfield

If in doubt about using this option, make a back up of your data files first and that could mean running menu 1 option S - Create Sequential files from ISAM just do not do it again until you have verified the flight log data via the options C or D reporting, and examined the records in error to verify it did what you were expecting, and no more.

## 6.7 Option J – Change Log Book Aircraft Type

As discussed earlier, this option will allow you to change an Aircraft type for another, replacing the original in the Aircraft data file, as well as all instances of where it is used in all flight log records. This mean that if you only have an error in a few flights but not all of them, consider using Option B - Amend Log Book Data instead.



```
LOG BOOK (2.01.31)          Change Aircraft type          at 18:42:49 on 12/12/2018
- - - - Type - - - -
Old      New
[bcdefgh] [abcdefgh]

Spaces in Old will terminate function & in New will request again

Copyright (c) 1986-2018 Vincent B Coen FMBS, Applewood Computers, Hatfield, UK
```

Figure 9 – Global change of Aircraft type

If in doubt about using this option, make a back up of your data files first and that could mean running option “S - Create Sequential files from ISAM” just do not do it again until you have verified the flight log data via the options C or D reporting, and examined the records in error to verify it did what you were expecting, and no more.

## 6.8 Option K - Edit Airfield Name

A facility to allow you to just work on the Airfield records, but with the CSV record types 4 & 5 for adding airfield and aircraft details, many of these facilities are redundant, other than the D(elete) function as the D(isplay) and L(ist) functions are available elsewhere. This option was one of the first created when the software was first written in the 1980's The CSV facility was a later addition but it has been left in just in case of a requirement to delete an incorrectly entered airfield during log book data entry or a need for a full list of every airfield stored in the system.

A to amend an Airfield name.

Enter A (for amend) and enter then the ICAO code and enter.

The current name is shown, and you can now change it, followed by enter when done. Use the cursor left and right keys to move the cursor without changing characters.

Other options under Action are :

**D** to delete a record but make sure you are **not** using it any where in your flight records as you will not get any warning if you are. Useful if you entered an incorrect code for an airfield but went into Flight log Amend mode and then changed the code but you know, you will not be

reusing what was originally entered. The Amend mode does not delete airfields previously entered in case there are other flights that do use it.

**I** to Insert (create) a new record with ICAO code and name. You can, when loading in a few or many use the CSV record type 4 and this can be run along with type 5 for aircraft types without any CSV definitions etc.

**L** to produce a listing of **all** Airfields in ICAO code order however, flight counts will be zero. This is because analysis is not run and that is the time when the counting occurs. The ICAO code entry is ignored so leave it blank.

**V** to produce a display of Airfields

For both V and L – entering a code will just report on the one ICAO code specified.

**X** to quit or use Escape key.

```
LOG BOOK (2.01.31)      Airfield Edit Program      at 18:56:30 on 12/12/2018

      Action      ICAO      Airfield Name
      [ ]      [ ]      [ ]

      ICAO code spaces for options V and L

                                +-+
                                | ACTIONS |
                                | A = Amend |
                                | D = Delete |
                                | I = Insert |
                                | L = List |
                                | V = View |
                                | X = Quit |
                                | Esc = Quit |
                                +-+

Copyright (c) 1986-2018 Vincent B Coen FMBS, Applewood Computers, Hatfield, UK
```

Figure 10 – Edit Airfield Name

## 6.9 Option S - Create Sequential files from ISAM

Selecting this option will create a sequential file for each of the indexed (ISAM) file namely flitelog, aircraft and airfield.

Indexed files have a search key that makes accessing a specific record fast and the key used, is different for each file and these are:

- |          |   |
|----------|---|
| Flitelog | - Flight Date and start time (in form yyyyymmddmmmm) where mmmm is actual time from (hh:mm) converted to minutes past midnight. |
| Airfield | - The ICAO code for the airport or airfield of 4 digits A-Z, 0-9..  |
| Aircraft | - The ICAO code for the specific aircraft of 4 to 8 digits A-Z, 0-9.  |

It will come back to the menu prompt without any extra screen output – it is very quick.

These files, other than another layer of back up are in case you wish to migrate the Flightlog software and the data to another platform, i.e., Windows to Linux, or to use another indexed file handler by recompiling Flightlog having installed and configured the new ISAM handler. This assumes you are using the source program and not the executable.

If migrating from using files to a rdbms system such as MySQL you will need to do this.

Note the option (using Mysql) is not currently available – It is subject to demand.

## 6.10 Option T – Import CSV Data and Parameters

This function allow you to import a company supplied CSV (Comma Delimited data file) containing records of all your flights over a period of time.

Well that's how the theory goes, as what they can supply is a file, containing the records for all flight crew. If that is the case, then Flightlog can handle it, providing you have set up the CSV Layout records correctly see record type 6.

Read chapter 7 CSV Parameter Structure page 29 where it discusses in detail how to create these important layout records that describe the data file you have or will receive.

This is needed, as companies supply different data with both Dates and Times formats that can vary widely between them. Sounds more daunting than it really is, but will need you to look at the first few CSV data records when creating the layout record type 1, 2 and possibly 3 and 6 if needed. By examining the received file you can also confirm that it is a data format that you can see, if not or it is blank or junk characters, you may well have have been sent a mainframe created file in EBCDIC format instead of the one used on PC type computers as ASCII (see CSV Overview page 29), in which case you will need to convert it first or ask the company for an ASCII data file. This should not happen but just in case one company does it.

See the supplied example in the manual of the configuration data, along with a sample CSV data file that matches up with each other.

For the first time of using a CSV data file, it is **strongly** recommended that you work from a new directory for testing i.e., creating one in Flightlog say, called csv-testing. Copy the CSV data and parameter files to it, start Flightlog and having run the Import CSV process, run option D and look at the logbook.rpt file to verify that all is good. If it is, then go back to the normal directory and run it again. If not find and fix the parameter error/s, using a text editor, then save the updated file, delete the flitelog.dat file and rerun Flightlog again using options T and D again to check that your changes are spot on. See 7.2 for more information.

If you are still having issues and have registered Flightlog, you can supply us with both files (as email file attachment) and we will try and fix it for you but using the same process as above.

If you wish to send such an archive change the file name to your surname, initials and licence number., i.e., "coenvb-atpl123456.zip", just in case some one else does the same, and use this name in any email to us.

## 6.11 Option U – Enter User details for Reports

If you enter the name and license number here, it will save you the need to enter the same information when you request many of the reports. There again, you can just enter this when you request a report.

If should only request it once as Flightlog checks if this information is present before requesting it. This only applies during the Flightlog processing so if you quit the program it will have to be entered the next time you run it.

This option is there, if you need it.

## 6.12 Option X - Quit Log Book System

Typing X or using the Escape key will quit the program closing all files.

## 7 CSV Parameter Structure

### 7.1 CSV Overview

Here we discuss the CSV formatted data file which stands for Comma Separated Value formatted (another name is Comma Delimited) file but herein, we will just refer to it as a CSV file. When requesting such a file from your company you must tell them you want a CSV data file.

Next is the format for a CSV file and here is a problem as they can vary somewhat, so at the basic level, each field is always separated by a comma and can be in quotes but they do not have to be, so in record type 1 definitions it is 'assumed' that all have quotes between text regardless of alphabetic text or numeric but they can also not have quotes at all. Flightlog will check for both possibilities including using double and single quotes ( " and ' ).

For commercial flight crews, this will allow you to import a \*Airline supplied file containing your flight records supplied in a CSV formatted file into Flightlog. Saves typing. Make sure you get them often enough as for one company, I used get them every 5 – 6 months and then they went bust in the middle of a gap, between getting them. Keep it to a month or even less. If you can, get them so they include the last day of the previous month or week.

\*Some Flight training establishments and Executive Hire businesses also supply these on request. Make sure you get the file as ASCII (used with PC's) and not in IBM mainframe format otherwise you will have to find software to do it for you, prior to use with Flightlog.

If you obtain this data using an online connection to the mainframe using X3270 terminal software or similar, you can specify that on the data transfer, it converts it to ASCII before saving it on your computer or memory stick etc.

Technically the data conversion between EBCDIC and ASCII can be done in Flightlog but you will not be able to look at the data before hand if you need to on a personal computer. If this facility is really needed, ask us.

Most \*airlines, create this data from aircraft technical or flight operation logs, held on servers, so they should always supply it as ASCII. It would be rare for them to use mainframes for this purpose but, you never know. This information is mostly created by you as flight crew as you have to fill in date and times (and may be flight crew names) etc., on to your flight document/s but it may well not include day and night flight times.

This facility can be used by **all** flight crew who wish to add into Flightlog, a large number of airfields in one go such as the modified lists from the Airfield text files supplied with Flightlog although, you will need to use a text editor to create the correct format for type 4 records.

You can do the same with Aircraft types if needed. These two record types can be imported by themselves with no CSV data just create a configuration file containing what is needed.

### 7.2 Testing your CSV parameter configuration with CSV data

On using this facility for the first time it is **seriously** recommended that you protect your flight log data by using another data directory to test your parameter configuration file and CSV data.

To do this (under Linux or \*nix) when in your working Flightlog directory do the following:

```
mkdir csv-testing  
cd csv-testing
```

[If Flightlog program is in working directory -

```
cp ../Flightlog .
```

Note the space and full stop at the end ]

<code>cp ../csv-conf.txt .</code>	And again, but change file name if different.
<code>cp ../csv-flitelog .</code>	And again, but change file name to <u>your</u> data file.
<code>cp ../aircraft.dat .</code>	And again but now for the aircraft and airfield data files.
<code>cp ../airfield.dat .</code>	As above.

Now you can run Flightlog which will create a new flight data file just using your CSV data. To do this, select menu option T and on return select D to produce a detailed report then exit Flightlog and examine logbook.rpt using a text editor or the less command etc. and verify that the flight data with totals is correct. Look closely at date and time formats to verify that your configuration parameters type 1 and more importantly, type 2 records are correct.

If it **is correct**, then you should copy the CSV parameter file back to the normal directory if you had to change it, then go back to the normal working directory and can then delete the test directory and its contents, run the CSV process using your normal flight log data and to do so here is the actual Linux commands :

<code>cp csv-conf.txt ..</code>	Change the name of the file if not the same. Note the space and 2 full stops at the end. Only do this if you changed the file while testing.
<code>cd ..</code>	Change to your normal data directory. Note the space and 2 full stops
<code>rm -fr csv-testing</code>	Remove the test directory and it's contents.

Now continue and run Flightlog rerunning option T to input your CSV data file.

But if it is **not correct**, check and change your parameter file type 1 & 2 settings then delete the created data files (.dat) by running:

<code>rm -f flitelog.dat</code>	This will remove the data file just created.
---------------------------------	--

Now re-run the above test again and continue doing so until you get the correct results.

If the problem is odd start and end times or times for capacity (P1, 2, 3) check record type 2 settings fields 3 or 4 respectively, against the CSV data file as likely to be an incorrect time format used.

If the flight dates are wrong check record type 2 settings field 2 and make sure you are using the correct date format. See CSV Definition Record Types record type 2.

Some important points as found when testing for myself.

With the CSV data shown in a separate window via a text editor etc. and looking at the first record, start creating the csv configuration file type 1 records starting with the first csv field by typing in:

```
101,
102,
103,
```

etc., until you have all of the csv fields present so for example if the csv record has 20 fields separated by a comma (plus the last one which does not have a comma at the end) you should have typed in "101," through "120," each on new lines. Now go through the csv data again this time looking for the fields that match up with the Flightlog data, see 7.5.1 Record 1 for more detail as to what these are. Also refer to 7.6 (Configuration File Example) and 7.6.1 (Example CSV data file). Having completed that now it is a case of going through the type 1 records again for the one's not used by Flightlog and adding 00 after the comma to specify that it will be

ignored, again using the examples indicated above. You must create a type 1 record for every csv field – **You must not omit any** as Flightlog need them to work out where it is in the csv data file when reading it. Failure to do so will result in errors possibly reported by Flightlog but more likely an unusable flitelog file that only shows up when requesting a full flight data report.

I repeat, every CSV data field must be recorded as a record type 1 entry.

On setting my own up I got some very odd behaviour and it took an hour or more before I spotted that I had missed one unused field).

Luckily you normally only have to do this once per \*airline.

\*Airline could also be a Flight Training Establishment, Aircraft Executive Hire etc.

### 7.3 CSV Prerequisites

This section is only of use to professional crew members who work for an organisation such as an airline, large flight training establishment, executive flight hire etc., or the military who can provide a record of all flights over a period of time usually a week or month undertaken by a given crew member. You would need to ask for this from operations or another department and might need to remind them you require it for running on a PC (personal computer).

The CSV file to be imported has a Flightlog maximum size limit of 512 characters per record.

If your supplied records are longer, contact us for an updated version of Flightlog with the size required.

It is not expected that any airline uses anywhere near this limit but we could be wrong.

There is also a predefined maximum number of type 1 records that can be used and this is 96.

This mean there cannot be more than 96 fields (separated by quotes) in a CSV data file. It is not expected that any airline uses anywhere near this limit but we could be wrong, again.

CSV data record fields are separated by a comma and optionally at each end of the data by one of two special characters namely a quote (") or (') and which one needs to be specified in record type 3 field 2, If neither are used leave as space, see below for **CSV Definition record types**.

It should be noted that the number of fields in the Flightlog flight record itself is 16 and that there is a requirement that a minimum of 9, are defined as targets within the type 1 records and these are for Flight Date, Start time, End time, Captain, Capacity, Airfield From, Airfield To (both using the ICAO codes).

If either of these sizes or other limits are a problem, contact us, see the details on inside front cover so we can update Flightlog and re-issue it.

If only these nine fields are available in the CSV data file, then capacity time recording for P1, P2, P3 etc., will be based on day flight times only, as automatically calculated day/night time can not be accurate. In these circumstances the crew member will have to manually amend the imported data and adjust such fields should they wish to maintain precision on night flight times, after the CSV data has been processed by Flightlog. The reference to P1 etc., could also be for any crew capacity such as E, N, R, T etc.

Note that a/c registration is just that, not a flight number which can be almost any aircraft reg, so if your company is supplying that, ask them to change the data created.

Flightlog must be told to ignore **all** CSV fields that are not used in the flight log data record so that every field in the CSV record is created in the CSV configuration file.



## 7.4 CSV Configuration Overview

There are four record types that can be used to define the CSV data file and its characteristics along with another two to record new Aircraft types flown and Airports or Airfields used. This file must be created using a **text editor** only such as Notepad, Notepad++ for Windows, vi, vim, emacs for \*nix type operating systems including Linux, Apple Mac OSX to name but a few.

Do not **use a word processor** as the format will not be in a text format and cannot be read by Flightlog.

Record type 1 - The purpose for type 1 records is to map the data in the CSV file with that in the Flightlog flight data file as the order of data provided in the CSV file will not match up nor does Flightlog know which to ignore. There **must** be a type 1 record for **every** CSV field present in the file regardless of whether it is used, otherwise Flightlog cannot locate the correct data.

Records 1 (CSV file and flight log layout definitions), 2 (Date and time data formats used) and 3 (CSV data file name), these three records are **always required**.

Record types 4 & 5 only needed if new airfields and aircraft exist in the CSV data file and should be created when the CSV record contain new airport/fields visited or/and new aircraft types flown. These must be entered at or before the time of importing a new CSV data file that contains this data.

Record types 4 & 5 can be entered at any time prior to the CSV data and these record types can be the only record types present.

It does not harm Flightlog if they are presented more than once, just takes more time.

Record 6 is used if the CSV data file does not have the exact name of the crew member or s/he wishes to use another name such as SELF in the Flightlog log book file, or if the crew member is not flying as P1 and the CSV data file needs to be search for valid data in other than the field holding the captains (or commanders) name. See discussion above in overview and below for record type 6.

Record type 1 must always be present when importing log book data and is specific for one Airline (unless another uses exactly the same data lay out). Here Airline can also include any other organisation that records the same data such as Executive Hire, Flight Training Establishments. This type of data is always maintained by commercial organisations or where the aircraft are registered and flown as a commercial operation and that does often include training schools etc. This is because they have to record aircraft technical logs which in turn records each and every flight with date, times and flight crew members names. However it does not mean that they can provide a CSV data file.

For pilots acting as P2 (second officer), you need to look through the data provided to see what field is used to hold your name, so that this field will be used to act as the record search criteria (see record type 6) but note that the Captain will always be the recorded pilot in charge. The format for recording such is down to the standards set up by your countries aviation authority such as CAA (for the United Kingdom) or FAA (for the USA).

The normal operation here for pilots flying as other than as P1 in that the pilot in command is in the captain's field and the CSV record only provided for you as P2 or other as flown. A problem would exist if you only get **all** pilots records as CSV data and this is where the Flightlog software has to search for the field that holds your name as the non P1 crew member. This requires the usage of the type 6 record where field 4 is used to find the CSV field holding your name so that the search only takes records with your name to process. Note that the following flight

capacities are recognised P1, P2, P3, P1S, (U/S treated as P1S), P1T, E1, E2, N1, N2, R1, R2, T1, T2 – more can be added on request. Same applies to using menu options A and B to input the data manually from your paper records such as a log book or diary.

Records for new airports or new aircraft types are only needed if a new one will be in the current CSV data file being processed and these must be defined before or during importing the CSV data file.

Entries for aircraft types and airfields will be checked. If they are not currently held, then CSV data will not be added to data bases but produce error messages and Flightlog will abort on the first instance of any missing Airfields or Aircraft and you will have to correct the problem before re-running. If this happens, then when re-running the CSV file you will receive error or warning messages FL041 and FL042 for any record that already exists on file which you can ignore.

Note that after the first instance of duplicated flights, any subsequent records will be shown as message FL041 with the next display line showing a full stop (".") for each CSV record. So if you get ten full stops it means that there was 10 similar duplicate records.

See Appendix B – Warning and Error messages page 48 for a full list and descriptions of these messages.

Record types can be in any order, i.e., records 1, 2, 3, Optional: 6, and if needed 4 and/or 5 as required, e.g., record 4, 5, 3, 1, 2, 6 etc.

When processing a CSV file, compulsory records are 1, 2, 3 with 6, 4 & 5 as and when needed.

When you need to process new Airfields and/or Aircraft types, then records 4 & 5 can exist by themselves with no other record types present. This way you can enter one or the other with no CSV data to be processed and even use it to preload both before using menu option A to enter flight log data manually.

## 7.5 CSV Definition Record Types

### 7.5.1 Record 1

This record *MUST ALWAYS* be present for data imports Defines each CSV field present in file regardless if it will be used to import data. Char as used below, is abbreviation for characters.

Record 1 – CSV field layouts

As four fields

Field 1 = Record Type	One char	'1' for field definitions for flight log record.
Field 2 = Source Position	Two chars	Field number starts with 01, with leading zero as needed – Must <i>always</i> be 2 numeric digits and <b>all</b> CSV fields must be accounted for ( <b>No Omissions</b> ).
Field 3 = Separator	One char	Use '=' or comma but don't care. Acts as a separator.
Field 4 = Target Field No	Two chars	As per field position in the Flightlog file, see below: Or as 00 to be ignored as data not used in flight log records.

<i>Flight log field name</i>	<i>Position in record</i>	<i>Comments</i>
No field	= 00	Omitted, field is not used in Flight log file.
FLT-Date	= 01 -	Must exist.
FLT-Start	= 02 -	Must exist.
FLT-End	= 03 -	Must exist.
FLT-AC-TYPE	= 04 -	Must exist.
FLT-AC-REG.	= 05 -	Must exist. (Aircraft registration number)
FLT-CAPTAIN	= 06 -	Must exist. (Name of captain, converted to capitals)
FLT-CAPACITY	= 07 -	Must exist. (Flight capacity)
FLT-FROM	= 08 -	Must exist. (Take off Airfield ICAO code – four chars)
FLT-TO	= 09 -	Must exist. (Landing Airfield ICAO code – four chars)
FLT-MS	= 10 -	(Multi / Single engine aircraft)
FLT-P1 Day	= 11 -	(Flight times for x1 Day)
FLT-P23 Day	= 12 -	( --- ditto – x2/3 Day)
FLT-P1 Night	= 13 -	( --- ditto – x1 Night)
FLT-P23 Night	= 14 -	( --- ditto – x2/3 Night)
FLT-INSTRUMENT	= 15 -	( --- ditto – as IFR time)
FLT-REMARKS	= 16 -	(Any comments)

[x can be P = Pilot, E = Engineer, N = Navigator, R = Radio / Radar operator, T = Radio, etc ]

No Field means a CSV data field that is not used in the Flight log file. It still **must** be specified but will be otherwise ignored. See CSV examples in Example CSV data file page 43.

An airline may well include fields that are not used in recording flights such as technical log information. It depends on where the data comes from within an Airline etc. as to what is supplied and providing the 'Must exist' data is included they can and will, be ignored if field 4 is set to 00.

For FLT-Captain there is an extra record type available – Record type 6.

### 7.5.2 Record 2

This record *MUST ALWAYS* be present for data imports. Defines the format used for flight dates, Flight start & end times and flight capacity (Pn) times.  
Check your CSV data file for format to be used for all three.

Record 2 – Date and time format layouts

*As four fields:*

Field 1 = Record type	One char	'2' for field date and time format definitions.
Field 2 = Date	Ten chars	Input Date format, space filled to the right.
Can be :		

DD/MM/YYYY	United Kingdom edited format.
YYYY/MM/DD	International edited format.
MM/DD/YYYY	USA edited format.
DDMMYYYY	United Kingdom format.
YYYYMMDD	International format
MMDDYYYY	USA format.
YYYYDDD	Julian date format. DDD = day number - 001 to 366.

YYYY = 4 digit year. (1900 to 2299 – but changeable depending on +today's date.

MM = 2 digit month (01 to 21).

DD = 2 digit day (01 to 31).

In these following cases, century is taken as if YY = > 70 = last one i.e., 19 if +today's Year is 20 or 20 if Year = 21 and if less than 70 it is today's century. This way you can enter historic data from 30 years before the current century. This also applies to manual data entry processing – menu option A and B.

*After the year 2000 these should not be used, but can also be defined (as above but with no century specified):*

DD/MM/YY  
YY/MM/DD  
MM/DD/YY  
DDMMYY  
YYMMDD  
MMDDYY  
YYDDD

Field 3 = Time 1	Five chars	Time format for Flight Start and End times
------------------	------------	--

Can be :

MMMM	Recorded as minutes after midnight.
HHMM	Hours, Minutes with no separator between HH and MM.
HH.MM	Hours, Minutes with separators , full stop or colon in character three. On time data import, character three is ignored.

Field 4 = Time 2	Five chars	Time format for Xn times
------------------	------------	--------------------------

Can be :

MMMM	Recorded as minutes only.
HHMM	Hours, Minutes with no separator between HH and MM.
HH.MM	Hours, Minutes with separators , full stop or colon in character three. On time data import, character three is ignored.

### 7.5.3 Record 3

Record must be present to read the CSV flight data file.

Record 3 – CSV data file name layouts

As four fields

Field 1 = Record type	One char	'3' for importing the CSV flight data file.
Field 2 = Not used	One char	Leave as A or space. In case we support other data formats. See the notes below the CSV example data at Delimiters and data formats on page 46.
Field 3 = Data delimiter	One char	Delimiter used in data file, single or double quotes, e.g., [""] or ['] [the single middle character. Here - Choice of one of the two options ]. If omitted, will use char 1 from first data record but will test it anyway. If fields do not use either leave as space.
Field 4 = CSV File Name	64 chars	File name to be imported if not 'csv-flitelog'. Can include full path if held in another directory but less than 64 characters in length. i.e., \users\fred\Downloads\csv-data.csv for Windows or <i>home/fred/Downloads/csv-data-20190531.csv</i> for *nix, such as Linux, OSX, BSD etc.

#### 7.5.4 Record 4

Only needed *once*, if a **new** airfield is included in log book data - You can add them in advance of Airfields being used.

If airfield already recorded, then only updated if name is different otherwise ignored.

Record 4 – Airfield layouts

As four fields

Field 1 = Record type	One char	'4' for NEW airport data.
Field 2 = ICAO Code	Four chars	ICAO code for the NEW airport
Field 3 = Separator	One char	Use '=' but don't care. Acts as a (data) separator.
Field 4 = Airfield name	20 chars	Airport name up to 20 chars with trailing spaces.

### 7.5.5 Record 5

Only needed *once*, if a **new** aircraft type used in log book data - You can add them in advance of being used. If aircraft type already present the type 5 record/s will be ignored.

#### Record 5 – Aircraft type layouts

As five fields

Field 1 = Record type	One char	'5' for NEW aircraft type data.
Field 2 = Aircraft type	Eight chars	Aircraft code for NEW aircraft types as used in logbook data, use trailing spaces. First four are significant and only four are normally used for Heavies using ICAO codes but all 8 characters can be specified i.e., light a/c.
Field 3 = Separator	One char	Use '=' but don't care. Acts as a separator.
Field 4 = Aircraft MS	One char	M for multi-engine or S for single engine.
Field 5 = Aircraft Complex	One char	'Y' for (yes) for all multi-engine (Default) otherwise 'N' (or space) for simple single engine A/C with fixed gear and propeller.

Note that records 4 and/or 5 can be the only records present, if you wish to add such to Flightlog even if not using CSV data import but use a different file name for such. If you do use this option then use the Flightlog parameter P1 or P2 to define the name with path if required. See 11.6 Running Flightlog Manually page 71 for details of these and the others and how to use them.

### 7.5.6 Record 6

*This record type is only needed if the following CSV data scenarios and cases are matched -*

**Scenario 1** : The data supplied is only for one named pilot -

Case 1 – The pilot wishes to change the name used as captain in the record say to SELF which might be classed as normal, for manually entering flight information in one's flight log book.

Case 2 – The pilots name is given as other than their name such as employee number, pilots license number or just not correctly typed etc. and you want to change the recorded name, again such as SELF.

**Scenario 2** : The data supplied is for many pilots and crew members and the data needs to be searched for only the specific crew members data records ignoring all others.

This could well apply for flight crew flying other than as P1 such as P2, P3, E1, R1 or any other classification (other than P1S, U/S or P1T as these are recorded as such but counted as just P1 in totals and other analysis).

Case 1 - The specific captains name needs to be searched for in the correct crew members name and may if the pilot wishes to change the name used, as in scenario 1 case 1.

Case 2 - If the crew member acts as anything other than as P1 such as P2, P3, E1 or R1 etc., and therefore the captain's name in the CSV record is not their own.

Where a CSV field other than the one for captain shows the crew members name and this field is needed to be used to search records for inclusion.

For Scenario 1, Case 1, field 7 can be spaces in which case, all instances of captain is changed to the content of field 2. This assumes that all CSV records are for only the one pilot.

Field 7 **must** be used for Scenario 2 Case 1.

For Scenario 2, Case 2

Field 3 set to '(' and Field 5 set to ')', with Field 4 set to the CSV field that needs to be searched for, to find the crew member acting as P2, P3, E1, etc. or any other non P1 pilot category. Where the field is not that of Captain as this would be Case 1.

This is an override facility, instead of searching for the Captains name in the CSV captains field. Here the specified field (other than captains field position) is used to search for records containing crew flying as other than P1 but is held in another specified CSV field as defined by CSV field 4 and this is what has to be searched for.

Here in a paper based flight log book and for crew working as P2, E1 etc, that is used for flying capacity, the captain's name is who is acting as P1 and then any comments held in the remarks column for base details if needed, may be.

The captain's name needs to be recorded in case of query or verification by the licensing authority or airline and for some aviation authorities, is a legal requirement.

Note that if 'PUT' is specified it is automatically changed to P3. This is in case, the data file comes from a flight training establishment.

Please note that the Captain name, Airfield ICAO code and Aircraft types are changed to upper case (CAPITALS) characters and this is used in the search and the replacement name in the CSV data and how the data is saved in the Flight log file. The same applies also in manual data entry, for the same fields.



Only the Remarks field is not so treated – it is stored as is, ‘warts and all’.

In the event that the CSV data can contain other instances of records containing other options of Captains name within the same file for example employee number and/or license number and/or pilots mistyped name etc. then you need other sets of the CSV definition file for each possible instance, i.e., one for wrong typing, one for a number, one for pilots name and then run the same CSV data file for each. One at a time, hopefully this is rare if not very unlikely. If this does happen you should let operations know so that can get it fixed but also tell us to consider allowing for more than one type 6 record to be present as at present if more than one exists only the last is used.

**Record 6 : *Only*** needed if you receive a CSV file with other flight crew records so you can be selective on the name to be searched for and/or you wish to change the name of Captain (as you) to say, SELF.

#### Record 6 – Captain name layouts

As seven fields

Field 1 = Record type	One char	'6' for captain name Search or Replacement.
Field 2 = Replaced name	15 chars	Field 7 name to be replaced with, such as SELF.
Field 3 = Separator	One char	Use '(' but don't care. Acts as a separator.
Field 4 = Source position	Two chars	Numeric between 01 and 96 which is field holding the name of the P2, P3 or any other crew member.
Field 5 = Separator	One char	Use ')' but don't care. Acts as a separator.
Field 6 = Capacity	Two chars	Capacity where CSV file does not specify it. If present this will be the capacity that is stored for all matched records. Normally used for crew not flying as P1, where you are searching a specific CSV field for a given name (yours). If spaces it will use the one in the CSV data. <b>Do not use</b> if you are flying as P1.
Field 7 = Search name	30 chars	Captains name to search for that is in CSV data where other names can exist and/or it is being replaced with name in field 2. (with trailing spaces). It will be converted to Upper Case as will the CSV data field 2 when comparing.

Leave fields 3, 4, 5 and 6 blank if only flying as P1.

When flying as any other crew category/capacity such as P2, E1 etc., field 3,4 & 5 is used for the CSV source field to locate the non P1 crew members name in CSV fields, other than the default one of target 06 (Captain). Field 6 specifies the actual operating capacity, See above references to Scenario 2, Case 2.

## 7.6 Configuration File Example

Here shown is a ruler that displays the character positions in the text file starting at column one for each record type that needs it (records 1, 2, 5 & 6) top figure is in 10's and bottom units. Highlight text shows the columns as the 10, 20 and 30 character positions to help you. So in the first example for record 1 you can see that in column 18 is the start of comments and these are ignored by Flightlog. The comments are only for your reference or notes. Note target field 10 is omitted as does not exist in the CSV data.

--- Ruler shown here ---

```
0000000000111111111122222222223
123456789012345678901234567890
```

```
101,01      date          TEST 1
102,04      a/c type
103,00      Not used - Tech log record #
104,05      a/c reg
105,06      Captain
106,07      Capacity
107,02      start time
108,03      end time
109,08      from airfld
110,09      to airfld
111,11      P1 day
112,12      P23 Day
113,13      P1 nite
114,14      p23 nite
115,15      IFR. time
116,16      Remarks
117,00      Date of next annual
118,00      Date of next 3 year maintenance
119,00      Snag list report number
120,00      Aircraft site base
121,00      Date of last inspection
122,00      Second officer (P2) Name
123,00      Absolutely no IDEA
```

--- Ruler shown here ---

```
0000000000111111111122222222223
123456789012345678901234567890
```

```
2yyyyymmdd hh.mmhh.mm      Empire Flt Training formats
```

```
3A"csv-test1.csv
```

```
4LEPA,Palma De Mallorca, SP
4EGBG,Leicester
```

--- Ruler shown here ---

```
0000000000111111111122222222223
123456789012345678901234567890
```

```
5BE23      ,SN      Beech Musketeer
5PA23-160,SN
5PA28-161,SN
5BE60      ,MY      Beech Duke
5CONC      ,MY      Concorde
5A337      MY      Airbus A330-700 Beluga XL
5A388      MY      Airbus A380-800
5B748      MY      Boeing 747-8
5B789      MY      Boeing 787-900
```

The first example shows changing any CSV records with Captain as COENVB which needs to change to SELF as usable in example data, see 7.6.1 Example CSV data file.

--- Ruler shown here ---

```
0000000000111111111122222222223
123456789012345678901234567890
```

```
6SELF          COENVB
```

[Second example for is for P2 pilot where a specific CSV data source field is needed for searching, instead of the CSV source field for Captain.]

Note that there can only be one record 6 present, but if more than one is present, only the last one will be used. Here it is checking source field 22 and if found against 'GOODX' inserts the record with capacity marked as 'P2' but captain name is maintained.

--- Ruler shown here ---

```
0000000000111111111122222222223
123456789012345678901234567890
```

```
6              (22)P2G00DX
```

Here you can see blank lines are used to separate record types which are ignored including any counts. Also comment text has been added after the data fields that are required for each record. Note that any comments placed after the last required field definitions are ignored. See type 1 record with comments starting on column 18.

Now the last example for the type 6 record :

```
6SELF          (22)P2G00DX
```

Here the pilot acting as P2 wishes to change the name of the captain to 'SELF' as s/he was acting as PIC (Pilot in Command) as it was his/her leg during a more than one segments and the csv file only had records of that type of flight as otherwise it all csv records that matched the search criteria would be added in and recorded that way. Here looking for field 22 containing users name, force change to capacity P2 with his name or SELF as captain.

Of course could use 'GOOD' instead of 'SELF' just by changing field 2 to 'GOOD'.

### 7.6.1 Example CSV data file

To match up with above configuration file data using quotes between data fields as well as commas.

```
"19900331" "pa28-161" "x1941" "gbppk" "Self" "P1" "14.35" "14.55" "egmc" "egmc" "00.20" "00.00" "00.00" "00.00" "C/O" "19910130" "19920130" "S1232" "egmc" "19891001" "" "JonesE"
"19900331" "C340" "x1942" "grita" "self" "p1" "15.50" "16.40" "egmc" "egtc" "00.50" "00.00" "00.00" "00.00" "00.00" "00.00" "19910130" "19920130" "S1235" "egmc" "19891001" "goodx" "JonesE"
"19900401" "PA28161" "x1941" "GBPPK" "COENVB" "P1" "14.35" "14.55" "EGMC" "EGMC" "00.20" "00.00" "00.00" "00.00" "19910130" "19920130" "S1232" "EGMC" "19891001" "" "JONESE"
"19900401" "C340" "x1942" "GRITA" "COENVB" "P1" "15.50" "16.40" "EGMC" "EGTC" "00.50" "00.00" "00.00" "00.00" "19910130" "19920130" "S1235" "EGMC" "19891001" "GOODX" "JonesE"
```

You will noticed that the first two CSV data records uses lower case text and the other two with one space line between, uses upper case (capitals) text.

Flightlog will convert all alphabetic text to capitals (upper case) on import.

This does not apply to the Remarks field which is left 'as is'.

Note that there are 2 or more lines shown here for every CSV data record and that there is nine type 1 record definitions showing CSV data fields not used in the flight log data file.

Here is an example of the same data but with captains name being COENVB and here the first above record 6 example could be used. Note that for the aircraft type pa28-161, the – was

removed but that is because the work processor used to create this manual treats such by starting a next line. Also the data shows a change from lower case to capitals and even mixed for pilot name, airfield and remarks.

```
"19900331" "pa28161" "x1941" "gbppk" "coenvb" "P1" "14.35" "14.55" "egmc" "egmc"
"S" "00.20" "00.00" "00.00" "00.00" "00.00"
"C/O" "19910130" "19920130" "S1232" "egmc" "19891001" "" "JONES E"
"19900331" "c340" "x1942" "grita" "coenvb" "P1" "15.50" "16.40" "egmc" "egtc" "M"
"00.50" "00.00" "00.00" "00.00" "00.00" "" "19910130" "19920130" "S1235" "egmc"
"19891001" "goodx" "JONES E"

"19900401" "PA28161" "x1941" "GBPPK" "COENVB" "P1" "14.35" "14.55" "EGMC" "EGMC"
"S" "00.20" "00.00" "00.00" "00.00" "00.00"
"C/O" "19910130" "19920130" "S1232" "EGMC" "19891001" "" "JONESE"
"19900401" "C340" "x1942" "GRITA" "COENVB" "P1" "15.50" "16.40" "EGMC" "EGTC" "M"
"00.50" "00.00" "00.00" "00.00" "00.00" "" "19910130" "19920130" "S1235" "EGMC"
"19891001" "GOODX" "JONESE"
```

Here is the same data from the last block but without the quotes around each data field but note that the Remarks field does on one record as it also contains special characters with another that has no data just the way it was supplied. Also times in the second block are in quotes may be because of the colon symbol. You will need to verify that as the type of data in Remarks is free form data and therefore can contain almost anything, it should be and must be in quotes if it contains a comma as it is the field separator. For times could be one or the other but often without.

```
19900331,pa28161,x1941,gbppk,coenvb,P1,14.35,14.55,egmc,egmc,00.20,00.00,00.00,00
.00,00.00,"C/O",19910130,19920130,egmc,19891001,"",JONES E
19900331,c340,x1942,grita,coenvb,p1,15.50,16.40,egmc,egtc,00.50,00.00,00.00,00.00
,00.00,"",19910130,19920130,S1235,egmc,19891001,goodx,JONES E

19900401,PA28161,x1941,GBPPK,COENVB,P1,"14.35","14.55",EGMC,EGMC,"00.20","00.00",
"00.00","00.00","00.00"
"C/O",19910130,19920130,S1232,EGMC,19891001,"",JONESE
19900401,C340,x1942,GRITA,COENVB,P1,"15.50","16.40",EGMC,EGTC,"00.50","00.00","00
.00","00.00","00.00","",19910130,19920130,S1235,egmc,19891001,GOODX,JONESE
```

Like previously said there is not hard and fast rules about this, at least that is adhered to. Flightlog covers all variations - hopefully.

So lets show both a configuration and another example of csv data used during one of Flightlog system tests and this shows both single and double quotes as well as none just the normal comma separating the csv fields.

## Configuration File

```
--
101,01      date          TEST 35 uses both quotes and non.
102,04      a/c type      with two type 6 records
103,00      Tech log info?
104,05      a/c reg
105,06      captain
106,07      capacity
107,02      start time
108,03      end time
109,08      from airfld
110,09      to airfld
111,11      P1 day
112,12      P23 Day
113,13      P1 nite
114,14      p23 nite
115,15      Inst. time
116,16      Remarks
117,00      Date of next annual
118,00      Date of next 3 year maintenance
119,00      Snag list report number
120,00      Aircraft site base
121,00      Date of last inspection
```

```

122,00      Second officer (P2) Name
123,00      Absolutely no IDEA

2yyyyymmdd  hh:mmhh:mm

3A"csv-test35.csv

4LEPA,Palma De Mallorca, SP
4EGBG,Leicester

5BE23      ,SN      Beech Musketeer
5PA23-160 ,SN
5PA28-161 ,SN
5BE60      ,MY      Beech Duke
5CONC      MY      Concorde
5A337      MY      Airbus A330-700 Beluga XL
5A388      MY      Airbus A380-800
5B748      MY      Boeing 747-8
5B789      MY      Boeing 787-900

6SELF      COENVB
6SELF      (22)P2GOODX

```

--

Note two type 6 records but only the last is used.

Now for the csv data records:

```

--
19900331,pa28161,x1941,gbppk,coenvb,P1,14.35,14.55,egmc,egmc,00.20,00.00,00.00,00
.00,00.00,C/O,19910130,19920130,egmc,19891001,JONESE
19900331,c340,x1942,grita,coenvb,p1,15.50,16.40,egmc,egtc,00.50,00.00,00.00,00.00
.00,00.00,19910130,19920130,S1235,egmc,19891001,goodx,JONESE
19900401,pa28161,x1941,gbppk,coenvb,P1,"14.35","14.55",EGMC,EGMC,"00.20","00.00",
"00.00","00.00","00.00","C/O",19910130,19920130,S1232,EGMC,19891001,""JONESE
19900401,c340,x1942,grita,coenvb,P1,"15.50","16.40",EGMC,EGTC,"00.50","00.00","00
.00","00.00","00.00","",19910130,19920130,S1235,egmc,19891001,GOODX,JONESE
--

```

This creates only two flight log records as there are only two out of all, that has 'GOODX' in csv data field 22, See figure 11.

```

--
LOG BOOK (2.01.43)  Personal Flying Log Book of Figure 11  2018/12/18 21:27:59  Page 1

-- Aircraft--      OP JOURNEY DEPART ARR - DAY - NIGHT
DATE  TYPE  REG  CAPTAIN  CAP FROM TO  TIME TIME  P1 P2/3 P1 P2/3 IFR MULT ----- REMARKS -----

Grand 0
Total 00 Totals Carried Forward 00 00 00 00 00 00 Mins
1990/03/31 C340  GRITA SELF      P2 EGMC EGTC 15.50 16.40 0.50      0.50
-----
Sub 0 Months Sub Totals 0 0 0 0 0 0 0 Hrs
Total 50      50 00 00 00 00 50 Mins
=====
1990/04/01 C340  GRITA SELF      P2 EGMC EGTC 15.50 16.40 0.50      0.50
-----
Grand 1
Total 40 Totals Carried Forward 40 00 00 00 00 40 Mins
--

```

Figure 11 - CSV report (D) for example with type 6 record selecting P2 user

Figure 12 shows the screen on completion of a csv data import. Note the bottom four displays showing the totals read in (4), written out (2) and lastly count of any that previously existed or were rejected (0).

```
File Edit View Bookmarks Settings Help
LOG BOOK (2.01.43) Flight Log Book Program at 21:13:37 on 2018/12/18

Select one of the following by letter :- [t]

(A) Enter Log Book Data
(B) Amend Log Book Data
(C) Log Book Reports
(D) Log Book Report & Monthly Analysis
(E) Analysis & Totals Report
(F) Analysis & Totals Display
(G) Cert of Ext. Analysis Report

(H) Change Log Book Airfield Code
(J) Change Log Book Aircraft Type
(K) Edit Airfield Name

(S) Create Sequential files from ISAM
(T) Import CSV Data and Parameters
(U) Enter User details for Reports

(X) Quit Log Book System

CSV Records in - 0004
CSV Records out - 0002
CSV Records Exist -0000 or rejected
FL017 Hit return to continue
```

Figure 12 - Post CSV data import

## 7.6.2 Delimiters and data formats

These examples show that the data delimiter is a double quotation symbol - “ and this is the normal one used for CSV files, **but** you could work for an airline who’s computer systems uses the single quote - ‘ or even no quotes - It could happen, so we allow for all three variations.

Flightlog supports both types of quotes and you register the one used in field 3 of record type 3. Regardless of this, Flightlog will still check the first character of the CSV data file to that specified in the type 3 record and if there is a difference will use the one used in the data record but if not present will then default to use both, including without quotes. It will even cope with a change of quote type between fields, although that should not happen – famous last words :).

When using your CSV configuration for the first few times compare CSV data against that stored in the flight log, looking closely at date, start, end times, capacity times, pilots name and Remarks to ensure nothing is wrong or in the case of remarks, missing. It could mean an error in your configuration file data or a software bug and if so, report the problem, if needed supplying the CSV data along with your configuration file and send as an archived email attachment – see email address on inside front cover but don’t forget to register your usage of Flightlog, see 12 Appendix R - Registering Flightlog page 75.

## 8 Appendix A - Keyboard operations or actions available on data entry.

During Data entry the following keys are available when shown on screen

- F1 = Displays list of Airfields by ICAO code. At end returns to data entry screen.
- F2 = Displays next flight log record after current one, when in amend mode.
- F3 = Displays list of Aircraft types. At end returns to data entry screen.
- F4 = Delete the current record being Amended. This is not available for data entry.
- F5 = In Amend mode save the flight data now, go back to the Date start capture for a new record to change.
- F10 = Go back one data entry field. Doing so more than once will get back to beginning of data entry as way was with Escape.
- Esc = Go back to beginning of data entry (Flight Date).

During numeric data entry for flight times for specific capacity within day and night.  
Z or z in field position 1 will set the field to 00.00 for both Data entry and Amend.

When entering Capacity :

I provides P1I

T provides P1T

When entering Captain name:

S (letter S and one space) then enter key will give SELF.

During entering flight duration times for P1, P2 & 3 Day or night:

F5 = Saves data record retaining all data previously entered if any, only if in **Amend** mode and if all changes required have been completed. Saves time.

During Amend mode having entered Flight date and time and ready to accept a changed date or later in the data amend process:

F4 = Delete current record from system ( there is no recovery from this ) it is gone !

Useful when you will re-enter the data or have done so already using a different date/time.

Right arrow = Move to the right one letter or number.

Left arrow = Move to the left one letter or number.

Back space = Move to the left deleting the character on the left.

Home and End = Move to left or right most position of current text based field.

[ May not work for numeric fields depending on O/S (operating system) and its set up. ]

Escape = In menu will quit program, same as letter X.

Del = Delete current character position as shown under the cursor.

Shift = Change case of letter on selected key (both used together).

INSert

or Ins = Enter text in insert mode within a field such as remarks or captain name but remember to press again to stop insert mode.

Warning if right most character is at the data fields right most end point you will not be able to enter anything more so must press INSert again to go back to over write mode. This is the normal mode.

There are others but may well depend on your O/S and set up, such as:

Print Screen = Pass screen content to your printer.

## 9 Appendix B – Warning and Error messages

### 9.1 Overview

Before listing the individual messages which for some, mention the use of SSD data drives a brief discussion on them might be helpful but is a little bit technical:

### 9.2 SSDs and Garbage Collection

If you are running with SSD's you should be aware of the problem of garbage collection on \*nix based systems such as Linux. This relates to the fact that all SSD's are not created equal, e.g., some have good controllers built-in and other – well not really.

An example, when we first installed a SSD into a Linux based system we found out very quickly that you had to run a Linux utility called fstrim on a regular basis like in nightly so we set up a cron job to do just that at midnight, running as sudo (or root) 'fstrim -av'.

Great you think, well not so much when running Crucial SSDs as their controller require 8+ hours of idle time to process the garbage and even then, they may be not, as it forced us to shutdown the system and boot into bios and leave it – No not a solution as the SSD filled up very quickly and I do mean quickly and I cannot spare the system time to make it offline. This is not the only brand that will cause this problem.

So SSD mk 2 – bought Samsung 850 as a test and found out that their controller did not require such idle time. Then bought two m.2 960's SSD's and installed in server and laptop (windows), a Samsung 850 into a Mac Pro dual quad Xeon CPU system and all work as advertised well apart from a media system that also has a m.2 that in some circumstances such as transferring lots of video media and yes I mean a lot – like 40 GB. As the SSD is the system partition all data coming in, goes to that first before being moved to the required location and that is with copying to the specific hard drive as the SSD is used as some temporary data areas although not sure where exactly.

So in this instance it cannot do a clear up quickly enough to cope, again do not know why. Just have to remember not to do a lot at a time :(

So what is this about I hear you ask? Well (now the technical bit) unlike a normal hard drive that, when you delete a file it is job done. It is not that straight forwards for a SSD. When a file is deleted or moved or updated it has sectors / clusters on the media that are no longer in use and the SSD controller has to go through a process called Garbage Collection and this mean the onboard SSD controller goes though each sector checking if it is not in use then clear the whole sector down by setting it to X'00000000' (X = Hex) for every byte in the sector, so takes a wee bit of time.

Now as we run the fstrim process every midnight it keep it under control at least totally on the servers and Mac pro. The Windows system seems to cope without doing a thing – just as well really as I have no idea what it does as I cannot find any docs on it.

So long story very short, if you run SSD's on a Linux based system you MUST make sure fstrim is run often enough to clean it up before it gets out of space, that has not been cleaned up because, regardless of its size, if all clusters have been used and have not been cleaned up it is the equivalent of a filled drive and we are using 256GB and 500GB despite them only used for booting and minor other data requirements, they can and do get clogged up.



You have been warned. So keep that in mind when looking at some of these error messages regarding a full drive. I recommend you do NOT use a SSD to hold application data on and their performance is not that heavy a usage problem as most of the time Flightlog is waiting for the user to key some thing in and the grunt work is done quickly enough. Note that the fstrim program is normally found in the util-linux package so this needs to be installed and most distro's do this if it finds a SSD in the system during a Linux installation. As said before, we have had no such problem using Samsung 850, 950, 960 SSD's.

### 9.3 Warning and Error messages produced by Flightlog

Program Error messages used: All supplied in English.

Programming Errors in file sizing	SY001 through SY003 (with filename & size).
Terminal size Errors:	SY010 through SY012.
CSV Warning / Error messages:	SY021 through SY024.
Operational, Warning/Error messages:	FL001 through FL050.

Messages in the range SY001 to SY003 are all critical errors and the program cannot be used without an update as they indicate a programming defect: Report immediately as a 'critical bug' on SF website for Flightlog or direct using contact information shown on inside front cover of this manual.

These messages mention three 'back up' files are in fact the sequential versions of each of the indexed primary data files.

They are used to recreate the indexed files if you change platforms or Cobol compilers or still using the GnuCOBOL compiler but change the ISAM file handler as they do not share the same format of data and these sequential files help to safeguard your data and recreate the indexed files having previously saved then removed them.

If indexed files are not present in the working directory when Flightlog starts but the sequential one's are, then the indexed files will be automatically created from them.

SY001 Fltlog length not same as back up

This file is not defined as having the same size as of the backup file (Sequential).

This is a programming error as both should be same size. You cannot continue to use the Flightlog program. Report immediately to name listed on inside front cover for a fast update. As this is tested for after each update these three messages should not appear, - Famous last words :)

SY002 Airfield length not same as back up.

See comments for SY001 but applies to the two Airfield files.

SY003 Aircraft file length not same as back up.

See comments for SY001 but applies to the two Aircraft files.

They will appear also with file name and size shown as '= nnnnn' etc, that helps to indicate the exact problem, so pass on all such output so that it can be fixed :

For SY010 & SY012, these appear if the terminal program is set up too small, must be width => 106), length =>24: [=> means equal or larger]. For width there is no need to be greater than 106, as any extra will not be used. Every line greater than 24 will be used for some data display lists such as requested via function keys F1 & F3.

SY010 Terminal program not set to length => 24.

Terminal program set to use less than 24 lines. Fix to 24 or longer.

Program rechecks after entering return.

SY012 Cannot display via menu option F as screen too narrow i.e., less than 106.

If screen less than 106 you will get a bad screen if requesting a display of menu option F - Analysis & Totals Display.

Adjust the width or length of screen as program rechecks after entering return.

Messages SY021 through SY024. These are for CSV layout definitions or data errors:

SY021 Bad data in CSV layout record – Aborting

Invalid data in CSV layout record/s record type 1 or 2. Fix and rerun - Aborting.

SY022 Too many type 1 recs (> 96) – Aborting

You have defined more than 96 CSV fields and the defined limit is 96. Check your created data and see if you have made a mistake. If not report problem in the SF Tracking tool for Flightlog requesting a larger CSV field limit. Note that if you have the OS (Open source) version with source code you can change this yourself using a **text** editor. See Changing settings for CSV field limits page 56 at the end of this section for more information.

SY023 Bad CSV type 2 record – Aborting

Format defined in type 2 record is incorrect, fix and re-run. Enter return to finish.

SY024 Not enough CSV fields defined - Aborting

The minimum number of CSV fields that must be defined is nine which excludes flight times for P1, 2 & 3 both day and night as well as IFR time.

You have defined below 9 and Flightlog cannot process CSV data until you increase the number of field defined. You should do so for **all** fields present in the CSV record. Enter return to finish. Now correct the layout definition file.

Primary Flightlog error and warning messages: Most if not all can be displayed during data entry or other processing steps including creating reports.

Note some message start with the text [*FLnnn not displayed*] in *Italics* and this just means that the message number is not displayed for this message.

FL001 Not in use.

FL002 record does NOT exist. Hit return to continue

Requested data does not exist, could be for any of the Airfield or Flightlog files so check any other message that also appears.

You need to enter return to continue if possible or stop this process.

FL003 record error on rewrite. Hit return to continue

When trying to update the flight log record an error occurred

Hit return to acknowledge, the process will be stopped.

You need to find out why – possible problem with no disk space or an SSD issues.

FL004 Record Exists

Warning that a record with the same flight date and start time already exists.  
Goes back to re-enter date.

[*FL005 not displayed*] is a new entry. Enter Y if ok, else N to re-enter

Exactly the same as FL007 but for Aircraft, i.e., a new aircraft type has been entered, so confirming it is ok to continue or to re-enter a/c type. There is no special keys for this warning available.

[*FL006 not displayed*] Hit return to abort Correctly

See previous message/s after pressing return Flightlog will terminate.  
You will need to fix the reported problem.

[*FL007 not displayed*] is a new entry. Enter spaces to re-enter or give name

You have entered an Airfield ICAO Code that is not on file, if ok, enter the name or to re-enter code use the ESCape or F10 keys or by entering spaces with enter key.

FL008 MS not compatible with Capacity

You have entered a conflicting value for capacity such as P2 for a single engine aircraft or even En for a one engine aircraft. Re-enter value.

FL009 Log time > Elapsed time Try again

Displayed normally with FL022, calculated times do not add up. Re-Enter.  
Flightlog keeps track of the time difference between end and start times as total elapsed time and the individual times added up are higher.

FL010 Error on (re)WRITE Flightlog file. Hit return to continue - ABORT

When creating or updated the currently entered flight log data an error was found.  
More information is displayed to help but check that you have enough free disk space or for any issues with an SSD.

[*FL011 not displayed*] Or to re-enter, press ESCape or F10

Showing what extra keys can be used at this point.

FL012 Give Pilots name & Lic no.

Enter the crew members name and license number for printing in reports.  
Can be any other useful information that will be printed.

FL013 Reading logbook, have found Aircraft type not on file, created.

This should not happen unless you are using an empty aircraft.dat file. If not, it should be reported as a bug along with the version of Flightlog you are using and the platform details. Use the Flightlog hosted SF website and the menu option Tickets.

FL014 Date Error.

You incorrectly typed the date – Check the format you are using. There are three date formats used in Flightlog and it depends on the system setting of LC\_TIME.

en\_GB for dd/mm/yyyy,

en-USA for mm/dd/yyyy

and any other (or omitted) will be for yyyy/mm/dd.

LC\_TIME must be defined before running Flightlog, see the supplied script run-Flightlog.sh for an example. It will NOT change any system wide settings and

only be valid for the duration of running Flightlog.

FL015 Re-Adjust screen then hit return.

On first starting Flightlog and also getting SY010 or SY012 beforehand.

Gives you a chance to fix the problem by dragging the right or left edge so that the display shows the correct width and dragging the bottom down to change the length to the one required but must be over 24 deep.

After changing the terminal settings by dragging the edges hit return for Flightlog to recheck the settings. If all is well it will continue otherwise you will get one or more SY010 or SY012, messages again, until fixed.

FL016 Not in use.

FL017 Hit return to continue

As specified, note any displayed information and hit return to continue.

FL018 Not in use.

FL019 CSV delimiter used from data record

Information only with message FL006 after, just note it press return to continue.

It means that the delimiter you specified in rec 3 field 3 does not match the data.

FL020 Enter Y or N only

Only responses can be Y or N (For Yes or No).

FL021 Enter M or S only

Only letter acceptable are M or S (for Multi-engine or Single).

FL022 or Inst. time > Elapsed time try again

Displayed normally with FL009, calculated times do not add up. Re-Enter.

FL023 Airfield not found

Specified Airfield not on file. You will need to create it if running a CSV file import otherwise it has now been created.

FL024 Record does not exist

Requested flight cannot be found on file. Enter another. Check both date and start time. Produced by an amend request.

FL025 Error on rewriting aircraft record. Hit return to continue.

This should not happen unless short of disk space or a problem with an SSD and its garbage collection or lack of it.

FL026 Aircraft table limit reached, Increase size and recompile

This and the next msg FL027 should not really happen.

The sizes for both of these as supplied are:

Aircraft is set to 1,000 different types.

Airfield is set to 2,000 different Airfields.

These values should be way above the number used by a commercial pilot even with 40,000+ hours flight time. The record for aircraft types is held by a UK navy test pilot with a count in the

hundreds (130). Which makes mine look feeble at 35 – 40 aircraft types and 130+ airports / airfields and 19+ countries.

See Changing settings for Aircraft or Airfield Tables page 56, at the end of this section for information to allow you to change these.

FL027 Airfield table limit reached, Increase size and recompile

Also see FL026 for more information as answer covers this message.

FL028 ISAM Data files do not exist, but Seq does – Recreate?

During program start the indexed files are not present but the Sequential files are.

Do you want to use these to create the working indexed files ? Y(es) or N(o).

If N(o) it will create new empty files. Here if you want to do airfield and aircraft but not flitelog then say Y to this and after files are created exit Flightlog, then just delete the file or files not wanted that end in .dat. Then run Flightlog again. The missing files will be recreated as empty files.

This is a way to migrate one or two files,. Airfield and Aircraft to a new Flightlog directory, maybe for a stand alone flight log record set. Yes, I did this for my Mil. Log Book.

[ The next six messages are caused by no space on disk OR if using an SSD, garbage collection process has not been run or completed.

FL029 Error on writing Flightlog Seq file

Received an error when creating sequential record, out of disk space? Aborting.

File will not have correct data so you need to rerun after fixing the problem.

FL030 Error on writing Aircraft Seq file

Received an error when creating sequential record, out of disk space? Aborting.

File will not have correct data so you need to rerun after fixing the problem.

FL031 Error on writing Airfield Seq file

Received an error when creating sequential record, out of disk space? Aborting.

File will not have correct data so you need to rerun after fixing the problem.

FL032 Error on writing Flightlog Dat file

Received an error when creating indexed data record, out of disk space? Aborting.

File will not have correct data so you need to rerun after fixing the problem.

FL033 Error on writing Aircraft Dat file

Received an error when creating record, out of disk space? Aborting.

File will not have correct data so you need to rerun after fixing the problem.

FL034 Error on writing Airfield Dat file

Received an error when creating record, out of disk space? Aborting.

File will not have correct data so you need to rerun after fixing the problem.

FL035 Aircraft not found

Information only, will create a new record for this aircraft.

To Delete press F4

As shown, press the F4 function key to Delete the current record if required.

- FL037 No CSV Param file present. Hit return for menu  
CSV configuration file not found in current directory. Aborted, fix and re-run.
- FL038 Invalid CSV record type - not = 1 – 6  
When reading CSV configuration file, record type read is not = 1 through 6.  
Aborted, Fix and re-run.
- FL039 CSV rec type 1 not found or bad, CSV process cancelled  
As shown there was no type 1 records present in CSV configuration file.  
Aborted, Fix and re-run. This does not appear if *only* processing type 4 & 5 records.
- FL040 CSV import file not found - Aborting import  
The file specified in the CSV configuration file rec. type 3 cannot be found in current directory used by Flightlog. Or are you in the wrong one? Aborted, Fix and re-run.  
If you cannot use the default CSV file name consider using rec type 3 field 4 to give the correct file name.
- FL041 Flight log entry already exists  
Usually comes with message FL042 but confirms the same CSV data file might have been used more than once. More than one data record having same problem will result in just this message with the next line containing a full stop for every record with the same same problem so 10 full stops means an extra 10 duplicate records.
- FL042 Writing flitelog from CSV Error nn  
An error was detected when trying to write a new flight record. Check disc space or access rights to do so. Flightlog will continue but you are likely to get this for every data record it is creating.  
It can and is more likely that you are rerunning the same CSV data again, and this record has already been stored in the flitelog file in which case you can ignore it but you might well see this message for every CSV record being processed. In this case the value for nn will be the specific error, so see  
File handling status codes and their meaning page 58 for these.
- FL043 Bad CSV Start/End time -  
CSV data is invalid or you specified the wrong format in type 2 record. You need to check this. Run aborted.
- FL044 No A/c type present or no match on Aircraft file  
As shown, a/c type in CSV data file not been set up using record type 5. You must do so. Run aborted. Note that record types 4 & 5 are processed before processing the CSV data file.
- FL045 No x1,2 or 3 data present/set up  
Warning, same as FL046 and often seen with it.
- FL046 Computed value may not be accurate.  
Warning that separate data for x1, x2 and x3 day and night is not provided therefore accurate totals can not be produced. Requires amending these records after CSV import process if required.
- FL047 No IFR time data present/set up  
Warning No IFR/Instrument time set up or present in CSV record. Will get this for

every record this applies to. You should consider amending each record this applies to if such time was actually logged for flight.

FL048 Flight Capacity data NOT x1, x2 or x3

First character of capacity is not valid type i.e., (P, E, R, N, T),

Error: Shown for CSV data in Capacity field, where x is P (pilots), E (engineer), R (REO) [Military], N (navigator) or T (Radio operator). Could be the CSV data uses different designators for this field.

Report as feature update and do not continue with processing CSV file data until Flightlog has been updated, however check first that it is a valid value anyway.

*[FL049 not displayed]* Note that dates are checked for valid format

Information when entering dates for reports (FL050 also displayed).

*[FL050 not displayed]* Spaces for dates gives ALL & space for last, means ALL from start

Information when entering dates for reports (FL049 also displayed).

### 9.3.1 Extra guidance on specific messages which detail fixes for the source file

If you only have the binary or executable Flightlog program contact us (see inside front cover for email, phone number etc) to ask for a larger table with an idea of just how large you need.

To change the Flightlog source file (flightlog.cbl) you need to use a **text editor** such as notepad for Windows or vim, kate for Linux, there are many others.

Under no circumstances use a *word processor* as it will not create a plain text file. Before making any changes make a back up of the original source file, i.e., by running

```
cp flightlog.cbl flightlog-cbl-original.
```

#### 9.3.1.1 Changing settings for CSV field limits

Using a text editor, open file Flightlog.cbl and search for the following text (without the quotes) : "WS-CSV-Table-Max-Size", is currently set to 96, change to the required value.

Now you have to do the same with the table, searchable as : "WS-Group" and you want the second line which starts as :

```
04 filler occurs 96.
```

Change the 96 to the required value (do not overwrite any other text) which **must** be the same as changed in WS-CSV-Table-Max-Size.

You now save the source file and rerun the Cobol compiler exactly the same way as you did the first time, for a reminder see Compiling the Source code on page 68.

#### 9.3.1.2 Changing settings for Aircraft or Airfield Tables

First make a back up copy of the source file, see top notes at 9.3.1 above.

Use the your text editor's search facility to find lines starting with "WST-AC-MAX" (but excluding the quotes), this should be around line number 971.

Now there are two tables with the first one for Aircraft and the next for Airfields (there are others) and for Aircraft, if you *need* to, change the lines starting with

```
03 WST-AC-MAX PIC 9999 COMP VALUE 1000.
03 WST-AIRCRAFT-TABLE VALUE spaces.
05 WST-ACFT-Groups occurs 1000
    ascending_key WST-Aircraft INDEXED BY QQ.
```

Here change the two instances of value 1000 as shown here in **bold**, (on the above lines 1 and 3) to the number you need to, such as 1500 or 2000 etc, but do not make it too large.

They **must** have the same **value** and do not change any other text.

Now if the Airfield table is not large enough, change the next table present, "AIRFIELD-TABLE" which shows these lines:

```
03 WST-AFLD-MAX PIC 9999 COMP VALUE 2000.
03 WST-AIRFIELD-TABLE VALUE SPACES.
05 WST-ICAO OCCURS 2000
    Ascending key WST-AIRFIELD INDEXED BY QQQ.
```

Again change the value shown here in **bold**, currently set as 2000. Do not make it larger than needed but round the value up to the next 500's, e.g., make Airfields to say 2500 or 3000.



Do **NOT** enter or change **any other text** unless you know what you are doing as the compiler will report errors when rebuilding the Flightlog program. Word of warning: If you make these tables very large it will slow down the running of Flightlog and/or prevent other programs from running as these tables can take up a lot of your computer's memory (ram).

Having made these changes save the source of Flightlog but make sure you have a copy of the original file somewhere before hand. Now just compile the program as you did when installing the system see 11 Appendix D - Flightlog Build and Installation.

**Note** that these values for both tables are only for your own flight records. If you are going to allow Flightlog to be used by others on your computer, ensure they use their own username when logging on and not yours, this is regardless of the operating system used.

Apart from the security issues in this, you **cannot** share the same flight records / data files, just the same as you cannot share a paper flight log book, although you could pass to them, in their own directory a copy of your aircraft and airfield data files but **NOT** your flightlog file as that is totally unique to you as it records all of your flights. We should point out that pilots etc, may not have flown the same aircraft types, let alone gone to the same airfields but there is no harm in doing this.

Using this process (copying the aircraft and airfield files) you can have separate flight data files to match each paper log book although I can't see the point but some pilots might want to keep them separate say if some flying was with the military before civilian flying. For myself paper log book wise I have got a lot of them having filled many of them up but for Flightlog I just add them into the one set of Flightlog files in the one directory.

***I will repeat this as it is important, if you wish to maintain more than one set of log book data files (flight, airfield and aircraft) you must store them in their own directory and that also includes other users using the same computer.***

## 9.4 File handling status codes and their meaning

### 9.4.1 File Access Error numbers

02	Creating a Duplicate key on alternative key which allows duplicate keys.
05	Success (Optional File Not Found)
07	Success (No Unit)
10	End of file reached if reading forward or beginning-of-file reached if reading backward
14	Out of key range
21	Key invalid
22	Duplicate key condition when duplicate keys are not permitted,
23	Start/Read has been attempted on an optional input file that is not present.
30	Permanent I/O error
31	Inconsistent filename
34	Boundary violation
35	File not found
37	Permission denied
38	Closed with lock
39	Conflicting attribute
41	Open has been attempted on an already open file.
42	Close has been attempted on an already closed file.
43	Read not done
44	Record overflow
46	Read error
47	"OPEN INPUT" denied (insufficient permissions to read file)
48	"OPEN OUTPUT" denied (insufficient permissions to write to file)
49	"OPEN I-O" denied (insufficient permissions to read and/or write file)
51	Record locked
52	End of page
57	"LINAGE" specifications invalid
61	File sharing failure
91	File not available

Note: There may be others but most if not all are here, see GnuCOBOL Documentation. Most if not all of these indicate a program error and you should report such, giving as much information as possible as to what you were doing at the time in an email to the support email address as shown on inside front page but note that one's discussing duplicate keys could be that you are trying to enter the same data or rerunning the same CSV data file and if the later such messages can be ignored.

## 9.5 RDBMS error codes

Here is the more common errors but there are a lot more see the RDBMS Error messages in the Mysql systems manual.

### 9.5.1 MySQL SQL Status messages

This is a small selection of what could occur, for others see the Mysql SQL error documentation. The software tries to rely on these more than on errno (Error numbers) below.

Sql-State      00000 = Operation completed successfully  
                 01 = Completed successfully  
                 0200n = No data found one way or another  
                         [fs-reply/we-error] = Get random = 23 else = 10].  
                 23000 = Dup primary key on insert same as fs-reply = 22.

Internal Errors :

99NKS = Invalid key # used.  
99NKU = No valid key used.  
99NKD = No valid key used for delete  
99RNP = Read next with no position (no start 1st)  
99GNS = Could not generate a start.

MySQL errno (Error numbers) that can possibly appear via Flightlog if RDBMS support is added.

There are many others see Mysql error documentation.

This section is presented in case MySQL support is ever added to Flightlog which will only occur if users request it.

## 10Appendix C – Installing the Cobol Compiler

If you require the various manuals for the Cobol compiler in both A4 and US letter formats and want the latest versions of each, as a PDF file go to:

<https://sourceforge.net/p/open-cobol/code/HEAD/tree/external-doc/guide/PDFs/>

The reading of this section is only needed if you are installing Flightlog as source code from the O/S (Open Source) area on SF (sourceforge) or direct from our servers.

If you are using the executable / binary program from our servers or from SF (when available) then just follow the instructions shown at under Installing the Flightlog executable or binaries page 67. Otherwise continue however also read the section Installing other optional packages for converting the report file to a PDF that is ready to output to your printer.

### 10.1 Install the Cobol compiler

Read this through first, then again if needed, and follow it:

You should check on the GC (GnuCOBOL) website

<http://sourceforge.net/projects/open-cobol> or ( <http://www.opencobol.org>

for the latest version and download and use it.

Having downloaded both the compiler and Flightlog along with all required packages / dependencies, you can start to build the compiler, test it and then build Flightlog if all tests complete correctly.

If you are a Windows user go to the specific SF website to get the instructions for building under Windows along with the compiler sources at this address:

[https://sourceforge.net/p/open-cobol/code/HEAD/tree/trunk/build\\_windows/](https://sourceforge.net/p/open-cobol/code/HEAD/tree/trunk/build_windows/)

Another option, if you are using Linux is to go to the software repository and see if they have an up to date version of the Cobol compiler. For this, use your Linux distribution's software manager. If the version they have is at least v2.0 you can use this instead of getting the latest version but when you compile Flightlog, if it produces warnings or error messages uninstall the compiler and get the latest. Do **not** use compiler versions, v1 or v1.1, as these are far too old and I do mean very as they are over ten years old.

Continuing, assuming you have downloaded the latest Cobol compiler. You also need to download and install all of the dependencies that GC (GnuCOBOL) requires, some if not most of which you can make use of your Linux distributions repos (repositories) however check that they are of a version equal, or better still, later than that as shown in the README and if needed, the HACKING file. You will need to read these files (including HACKING as some of the packages discussed are needed if you download from the SF code area). This gives details of what is required, but listed in Chapter 13 is one from the current GnuCOBOL which is for v3. Also included in the GC archive is the README from the tests/cobol85 directory shown here in How to run the NIST CCVS85 (aka. ANSI85) Test Suite 13.5 on page 84. These tests must also be done to validate the GC compiler. If these tests succeed you can run 'sudo make install' etc and any other instructions listed in the README's. Before building Flightlog. Note that these files show that the following packages for users of Linux, OSX, or other \*nix platforms, that will also need installing if not already. For Windows, all elements needed are already supplied but check the README file for confirmation :

These are the extra packages listed in the README that you also need to install prior to building the Cobol compiler with the ones highlighted in ***bold italics*** used here:

GnuCOBOL **requires** one of the following external libraries to be installed for implementation of decimal arithmetic:

BOTH runtime **and** development components required.

- ***GNU MP (libgmp) 4.1.2 or later***  
<https://gmplib.org>

OR

- MPIR (libgmp - MPIR gmp-compat) 1.3.1 or later  
(preferred when compiling on Windows with compilers other than GCC)  
<http://mpir.org>

GNU MP and MPIR are distributed under GNU Lesser General Public License.

**NOTE** - libltdl is **NOT** needed when installing on Linux, SUN Solaris, MAC OS, CentOS or Windows (including Cygwin, MingW and native windows).  
It is also NOT needed with later versions of AIX and HP-UX.

- GNU Libtool (libltdl)

<https://www.gnu.org/software/libtool/libtool.html>  
libltdl is used to implement dynamic CALL statements.

For Indexed file processing you have choices of what one to use but for your own needs use the first one described as it is the one used during development and testing :

BOTH runtime **and** development components required.

- ***Berkeley DB (libdb) 4.1 or later***  
<https://www.oracle.com/technology/products/berkeley-db/db/index.html>

Download page is :

<https://www.oracle.com/technetwork/database/database-technologies/berkeleydb/downloads/index.html>

Berkeley DB is distributed under Oracles own open-source license.

For use only on your computer this product can be used with no problems and is the one used for development and the programmers own usage. Note that I use a now old version of this (currently Berkeley DB 12cR1 (12.1.6.2.32)), downloaded from the oracle website. As most of its features are not used it does not have to be 'the' latest or even close to it. Remember Flightlog will, if requested create (line) sequential versions of all three files that can then be used as sources to rebuild the indexed files, so changing the version or even switching to a different indexed file package is not a problem.

Note that if you linked your software with Berkeley DB, you must distribute the source code of your software along with your software, or you have to pay royalty to Oracle.

Flightlog complies with this as the source code is available from the website or you install the package on your computer before running the Flightlog executable. The only other option is to use v1.85 which has no restrictions.

OR

- VBISAM - ISAM file handler (libvbisam) 2.0 or later  
<https://sourceforge.net/projects/vbisam/>

VBISAM is distributed under GNU Lesser General Public License.

OR [ *Don't use this one* ]

- DISAM File handler (libdisam)  
<http://www.isamcentral.com>

There are a few others with C interfaces that can also be used but these are still being tested.

SCREEN SECTION and/or extended ACCEPT/DISPLAY used (it is in Flightlog)

BOTH runtime **and** development components required and these are used with Flightlog.  
One of the following but only the first one (ncurses) used here :

- **Ncurses (ncurses or ncursesw) 5.2 or later**  
<https://www.gnu.org/software/ncurses/ncurses.html>  
Ncurses is distributed under a BSD style license.

OR

- PDCurses (pdcurses) for MinGW/native windows ports  
<http://pdcurses.org/> or <https://github.com/Bill-Gray/PDCurses/>  
PDCurses is distributed as Public Domain.

OR

- Unix curses

There is another but it is not used in Flightlog, but in case you wish to program in Cobol you might want :

XML runtime support

BOTH runtime AND development components required.

- libxml2 - <http://xmlsoft.org>  
libxml2 is distributed under MIT License.

## 10.2 Set up and build the Cobol compiler

With all that done you can start with building Flightlog see chapter 11:

Detailed overview for building the GC (GnuCOBOL) compiler

Having transferred the Flightlog source archive to your home directory in a directory say "cobolsrc/Flightlog-Source" and here I will show you all of the command line instructions needed. To do so, you need to select and load a terminal program such as 'konsole' under KDE but any will do. After loading, you will be placed in your home directory, This is the terminal program (but can be almost any other) you will also use for running Flightlog.

In the following instructions note that names and version numbers might change. These instructions start with the basics for the build and installation of the Cobol compiler:

First off having started in your home directory .

<code>mkdir -p cobolsrc/Flightlog</code>	Create new directory called cobolsrc and Flightlog to hold the Flightlog source.
<code>mkdir cobolcompilers</code>	Create new directory for the compilers source directory
<code>cd cobolcompilers</code>	Change to the top level directory for the compiler/s.
<code>unzip ../open-cobol-code.zip</code>	Change the file name containing the compiler if needed. Assumes the file is in the higher level directory. This is the source file that you obtained from the sourceforge website.

Build the GnuCOBOL compiler according to the README instructions including the test procedures but once in the directory holding the sources you can run (having noted the preceding . (full stop) :

<code>cd open-cobol-code</code>	Change if name is different.
<code>./configure</code>	Runs to end without error note leading './'.
<code>make &gt; build.log 2&gt;build.err</code>	Will build the compiler. Check build.err by doing ('less build.err') for errors but warnings regarding translations is OK. Don't enter the quote symbols. This will take around 3 – 6 minutes. If you have a 4 core or more CPU you can add after 'make ' -j3 and this will speed the process up a bit.

## 10.3 Test the Cobol compiler

Make checkall > checks.log 2>checks.err

This runs two sets of tests and both must complete without errors although you might get some warnings (see the README for more information).  
This will take 3 – 10 minutes depending on system speed – be patient. Don't use the -j3 option.

Note for the above processing the screen will not show any output until each is finished.  
To do this :

1. Look at the checks.err file ( 'less checks.err' ) and this should be empty other than a message similar to :  
Total executed programs : 426 - Total performed tests : 9748  
but with no error messages.

2. Now look at checks.log ('less checks.log') and check that it does not have error messages (but could have a few warnings of skipped tests at the beginning for tests shown by number 1 – 900 or so). At the end of this first block of tests it should show a message like this:  
989 tests behaved as expected.  
2 tests were skipped.

Once past that block of messages you will then see the NIST tests that runs many test programs against GC and very near the end will show the differences of the summary file which must be zero. e.g.,

*Comparing total test results*  
*diff ./summary.txt summary.log*

i.e., nothing shown – blank line. Followed by a few lines saying leaving directory etc.

If all is good, lets continue with installing the Cobol compiler and building Flightlog.

sudo make install                      You need to know the admin password and if sudo is not set up for you can run su -c "make install"  
Again you need to know the admin password which you created when installing Linux.

With the compiler now fully installed, one more little job before moving on to Flightlog.  
You need to create file /etc/ld.so.conf.d/gnu-cobol.conf

**(This file is important).**

Note to do this, you will have to be root or run as sudo, see above notes. So create it using a text editor with the following content (ignoring the lines of – which indicates start and end of the data to be keyed in):

```
--  
/usr/local/lib/gnu-cobol  
/usr/local/lib  
/usr/lib  
--
```

If you are going to use the RDB MySQL system (not needed for Flightlog) also add this line at the end:

```
/usr/local/mysql/lib
```

So now run :

sudo ldconfig                      (Or su -c ldconfig) admin password needed.  
With the above, we have told the system where to find the new GC libraries. If you re-install the compiler you should do this each time as a safeguard .  
cd ../cobolsrc/Flightlog              Change to the Flightlog sources.

tar xfvz ../Flightlog-2.01.00.tar.gz  
Change name to the archive name supplied & unpack).  
This will unpack the source and other files including manual.



At this point, you ***must*** have installed all the GC dependencies (see above) unless you have installed a .RPM or .DEB GC v2 or later package from your distro's repo's (repositories), in which case, all should have been done for you, but if you get error messages during the GC package install, this may not be the case.

Using versions of the GC compiler before v2.2 is not recommended as they are considered too old but might still work. During the build of Flightlog error messages will give a good indication that the version is too old!

Now lets have a short reminder and recap

## 10.4 Validating the Cobol compiler

Yes, I know I am repeating this, because it is important !

At this point, you have installed GC (GnuCOBOL) by following the instructions in section 10.1 Install the Cobol compiler 10.1 and onwards.

You must do both sets of tests using `make checkall` and verify that all is well before moving on to the next step. Running these tests can take some time depending on the speed of your computer. It is ***vital*** that these tests are successful. This must be done even if you installed GC from the distro's repos as it validates the compiler. Note that during these tests there will be no output to the screen – that is to be expected as it is going to files.

After doing the tests, you can then complete the installation of GnuCOBOL by running `'sudo make install'` after which, you are now ready to compile the Flightlog program.

So now lets show you the steps involved and this is the easy and quick bit:

# 11 Appendix D - Flightlog Build and Installation

## 11.1 Installing the Flightlog executable or binaries

If you have received the executables or binaries (if available), you just have to extract the archive using ZIP, RAR or for \*nix platforms TAR into your newly created directory for running Flightlog in your home directory using a terminal program and this will be the directory you will run Flightlog, see section Installing Flightlog page 69. Here, Flightlog will create and use all of the data files it requires. Likewise, if you will be using a company supplied CSV flight log data, this is the directory where you will be copying the data file to, having renamed the file to the name you defined in the CSV configuration file. See the section CSV Parameter Structure page 29 on how to create this. The archive contains the program Flightlog, this manual and possibly a README for any need to read notes regarding installation or usage, etc. You can then start using Flightlog but see section 12 Appendix R - Registering Flightlog first on registering your usage of the program.

If you have downloaded the open source version of Flightlog, these are the files included within the source archive:

—	
Changelog	Shows the changes made to Flightlog including bug fixes and new features.
comp-Flightlog.sh	Script to compile Flightlog
Copying.pdf	The software licence.
Flightlog.cbl	Flightlog source file.
Flightlog.pdf	Flightlog Manual – this document.
prtpdf	Linux script file for printing without the ext.
prtpdf.sh	Linux script file for printing on A4 paper.
prtpdf-sh-A4	As above master for A4 paper.
prtpdf.sh-Letter	As above master for Letter paper.
	Copy to prtpdf.sh if needed but A4 setup should also work for Letter as margins have been set up to cover both paper formats.
run-Flightlog.sh	Script to run Flightlog using UK dates.
run-Flightlog-unix.sh	Script to run Flightlog using Unix dates.
run-Flightlog-usa.sh	Script to run Flightlog using USA dates.
README	Last minute notes.
Airfields/	Files in this directory :
List_of_airports_by_ICAO_code__C.pdf	World Airfields starting with C as pdf file.
List_of_airports_by_ICAO_code__C.txt	World Airfields starting with C as txt file.
List_of_airports_by_ICAO_code__E.pdf	World Airfields starting with E as pdf file.
List_of_airports_by_ICAO_code__E.txt	World Airfields starting with E as txt file.
List_of_airports_by_ICAO_code__G.pdf	World Airfields starting with G as pdf file.
List_of_airports_by_ICAO_code__G.txt	World Airfields starting with G as txt file.
List_of_airports_by_ICAO_code__K.pdf	World Airfields starting with K as pdf file.
List_of_airports_by_ICAO_code__K.txt	World Airfields starting with K as txt file.
List_of_airports_by_ICAO_code__L.pdf	World Airfields starting with L as pdf file.
List_of_airports_by_ICAO_code__L.txt	World Airfields starting with L as txt file.
And these two in a different layout.	
List_of_airports_in_the_United_Kingdom_and_the_British_Crown_Dependencies.pdf	
List_of_airports_in_the_United_Kingdom_and_the_British_Crown_Dependencies.txt	
and may be:	

TODO

Lists of possible new features but subject to user requests.

--

## 11.2 Installing other optional Packages

These optional packages are needed if you want to automatically convert the print file logbook.rpt to a printable PDF file for A4 or Letter paper.

For Linux or other \*nix, you need to install the following from your distributions software repo's (repositories) using your software manager :

enscript	This is the primary tool that converts logbook.prt to a postscript file but this would assume you have a printer that can process it correctly, so the next package is also needed.
postscript-common	This should also install other elements of postscript but this is the only one that contains the elements required.

That's it, and these will work in conjunction with the supplied script prtpdf.sh.

If however you have another way of going the same thing then use them and modify the script prtpdf.sh and this script needs to be run after Flightlog if you want printed output.

It is not automatic as you may well not require a fresh report just for a few flights so you control when doing so.

## 11.3 Compiling the Source code

Only needed if you have the OS (Open Source) version with the source code file flightlog.cbl.

First ensure that :

Compiler is fully built, tested and installed and that you have run ldconfig, see section from Appendix C – Installing the Cobol Compiler on page 60.

Compile Flightlog using the GnuCOBOL compiler by running the script comp-Flightlog.sh.

cd Flightlog	(change to the Flightlog source code directory)
bash comp-Flightlog.sh	(compiles the Flightlog program and any others)

The output of this step must not show any errors.

Assuming no errors occurred (but if so, fix and rerun) now continue with Installing Flightlog

Instead of running the compile script you can just enter :

```
cobc -x Flightlog.cbl
```

## 11.4 Installing Flightlog

Assuming no errors occurred during building the program if you did not get executables, (but if so, fix and rebuild) you can now run

`cp Flightlog ~/Flightlog` Assuming that is the directory you will be running the program from, and is in the level above that, for the source directory.

You can if you have a directory called bin in your home directory, copy the program there by instead typing :

`cp -vp flightlog ~/bin.` You can do both.

If others will use your computer to run Flightlog then instead of the above, copy flightlog to `/usr/local/bin` and to do so use : `sudo cp flightlog /usr/local/bin`

You may need to know the password for doing so depending on Linux distro used.

Now you can run Flightlog from any directory under your user name.

The first option of using `~/bin` assumes that your local bin directory is set up to be in your search path for programs.

Here is an example :

```
PATH=/home/vince/bin:/usr/local/bin:/usr/local/sbin
```

To see your path set up, run this:

```
set | grep PATH
```

You could have a lot of lines from this but the one you want to look at starts a line with:

```
PATH=/home
```

This is set up in my user name `.bashrc` file and this is set up at some point by the Linux installation scripts with a few changes by myself. Note that this could also be set up in file `.bash_profile` that also sits in your home directory.

Note that as these two files start with a period (full stop) you can only see them if you use `'ls -la'` as against the alias `'ll'` when browsing directories. Yes, this is for Linux so Windows will be a little different.

My `.bashrc` files amongst other information shows:

```
export PATH=~/.bin:./usr/local/bin:/usr/local/sbin:/usr/bin
```

I have removed excess extras, so just showing the basic path.

When running Flightlog you **must** ensure that the konsole or terminal program you are using has been correctly set up, namely set width to 106 columns, length of at least 24 lines and as long as you like, as some of the display processes can make use of it when displaying lists of Aircraft or Airfield within the Data Entry or Amend options via the function keys F1 or F3.

You **must** have set the width correctly.

The program will check that these settings have been done and will produce an error message (with you needing to press the enter key having adjusted the screen width and depth settings) and then checking again.

While changing these settings look for the konsole program “Edit profile” and also set “scrolling” to Unlimited Scrolling and although not needed for Flightlog, can be very handy.

For users running Mageia v6, I have to also run (under the terminal) a program called screen as the terminal program messes up some display areas and doing so before running Flightlog. Just run it with no parameters and you can install this from your distro’s repo’s.

## 11.5 Preparation for running Flightlog

### Actual Steps for Linux.

This will create a permanent set up for konsole, the terminal program, and providing it is done correctly only needs to be done once for a user as it assumes you are using the KDE graphic interface as against Gnome which may be a slightly different procedure.

Using the mouse move the cursor to an empty area on screen and right click, a box has appeared, select to the entry "Create new" by moving cursor to the area with > and here you have some choices but select "Link to Application" by left click.

Another box appears, so let set this up, first change to the Application tab (at the top) and change the description currently "Link to Application" to "Flightlog" (without the quotes).

In Advanced Options, Select top option "Run in terminal" using left click then again with left click for OK at bottom. Now to fill in the required areas, click Name and change to Flightlog, then Description do the same, Comment what ever you like or ignore it, and in Command select browse and you will get a new box that specifies at the top "Open File" so find the directory you created for running Flightlog and left click, so that the screen shows contents of directory Flightlog which has one file Flightlog, now select Flightlog (left click) the name appears next to lower tab "Name" now select Open on the bottom of the box. If done correctly within the Command tap you now have the path and filename of Flightlog. Now select OK (left click).

Start the Terminal program konsole as we have to set up the Profile for it with the width and length needed for Flightlog. Along to top of konsole select Settings and manage profile.

New box appears set to Profiles so now click on New Profile and for Profile name make it Flightlog then lower down change Columns to 106 and Rows to any larger than 24 and here does depend a bit on how big your monitor is but assuming a 21 inch or bigger, set to 50. Now select Apply then OK. Now to make sure it remembers the settings exit then restart a konsole session, now using the cursor drag the right hand edge to the right until the number 106 appears – Stop. Now do the same but on the bottom edge until 50 (or what ever length you have decided on) appears. That is it, so exit konsole and restart it and when it does it should now be set to the width and length you previously did.

Now we have an icon that when selected (double click) will run Flightlog and the Flightlog menu will appear.

So now you are ready to enter your flight log book data by selecting menu option A.

Now for some tweets to the settings we created for running Flightlog via the screen Icon that runs the application.

You should not do this now but it is a feature that you might need to add. By the way, there is no reason why you cannot set up other icons that run application for Flightlog but with different parameters to it just remember to name them to reflect what it will do, maybe the parameter list within the application description or name or both.

## 11.6 Running Flightlog Manually

Here are the Flightlog optional parameter options and are shown as if running from within a konsole window and the format is :

Flightlog P1 P2 P3 P4 or (HELP | help | -h | -H)

**Note** That a space must be included between the words/parameters and likewise after the program name.

First option is Flightlog HELP :

For help with the parameters, you can run:

Flightlog help | HELP | -h | -H - (the "|" means or, so multiple options for the same function) and this will produce a screen showing:

```
--
Parameter Help for LOG BOOK (2.xx.yy)

P1 = NONIGHT or NONITE for no night time calcs against table
P2 = CSV= CSV path and file name for Config file if not default
P3 = ACFT-DATE for report excludes unused Aircraft
P4 = EBCDIC conversion of CSV data [NOT CURRENTLY IN USE]
```

Hit return to abort Correctly

--

**Note** That the above characters "--" show the start and end of displayed data (that is in another type font) and xx is minor version and yy is build numbers of Flightlog.

The following optional parameters can be added in any order :

#### 11.6.1 Parameter 1 (P1):

Value NONIGHT | NONITE

Do not attempt to calculate start of night time when accepting x1, x2, x3 day and night entries, where x can be P, E, N, R, T. This will force Flightlog to accept the xn field based on the entered capacity value both for day then night (if any time is still unaccounted for).

Entering the misspelling NONITE also works just in case you are feeling lazy.

Why use this? This usually causes a problem if time is close to or after twilight. Note that the predefined values for night time is set in a table at "Night-In-Month" showing the month from January to December in order, so the current one set for the UK is "161718192021212019181716" and here the value for August is 20 for 20:00 hours. The program deducts one from these values to arrive at a 'safe' time when night falls. It does not recalibrate the table after any data is entered. It is just a basic calculation that seems to work best when time is clear of twilight, i.e., fully dark. If you have the OS version of Flightlog you can change these values in the source code file and then recompile the program, but do not forget an hour is deducted in the final calculation for the start of night so can always be up to an hour out, especially if it is the start of the hour. The deduction of 1 tries to help a calculation at the start of the hour but is not always right.

Correct, the program does not check for the flight start minute time as it would only change value by a maximum of 30 minutes and even that is not accurate.

#### 11.6.2 Parameter 2 (P2):

Value 'CSV=' CSV path and configuration file name, if not the default which is csv-conf.txt.

You can include the path if needed with the file name but must be less than 128 characters.

Example = "CSV=/home/username/Flightlog/csv-data/csv-config.txt" but without the quotes.

[Without the path it will only look in the current directory.]



Where username is normally your platforms user name that you login with and you have read access.

That said, you would normally keep this file in the same directory as the Flightlog data files. However this parameter allows you to change this.

### **11.6.3 Parameter 3 (P3)**

Value ACFT-DATE.

Using this option will change the way Aircraft lists are produced in that instead of a full list of all recorded, you will get the list ordered by last flight date and only those that have been flown. Useful if you have added a lot say from a different users directory than you have flown.

### **11.6.4 Parameter 4 (P4)**

This parameter is not available so ignore. Depends on users need and requesting it.  
Value EBCDIC - NOT yet AVAILABLE but will allow Flightlog to translate a CSV data file supplied from an IBM mainframe in EBCDIC to the PC data format of ASCII.  
As this function can be done when creating the CSV file it will be subject to user demand.

Example of calling Flightlog with all parameters :

```
flightlog NONIGHT CSV=/home/fred/flightlog/csv-config.txt ACFT-DATE
```

Note that the case of these parameters are very specific - they must all be in upper case (Capitals letters) other than the path and filename of the CSV configuration file.

### **11.6.5 Setting of LC\_TIME for your preferred date format.**

Before you run Flightlog, the environment variable LC\_TIME must be set up so that your locale date format is known:

Flightlog assumes you have the environment variable LC\_TIME set, in your system setup script (bashrc) to one of,

en\_GB (date format is dd/mm/yyyy), or  
en\_USA (date format is mm/dd/yyyy).  
Or any other including blank

If these two are not found, the date format is set to UNIX which is yyyy/mm/dd.  
If your system default is NOT one of these but you wish to make use of the UK or USA date formats just set the value of LC\_TIME before calling Flightlog (or in the run script file) as :

```
export LC_TIME=en_UK          For date format dd/mm/yyyy
```

*or*

`export LC_TIME=en_USA`

For date format mm/dd/yyyy

or

`export LC_TIME=`

For Unix format yyyy/mm/dd

This setting will only last while running Flightlog and will be forgotten after you exit the terminal program.

In the three supplied scripts that will run Flightlog, there is one version for each of the above settings so just copy the right one for you to `run-flightlog.sh`.

These scripts are :

`run-flightlog-uk.sh`

`run-flightlog-usa.sh`

`run-flightlog-unix.sh`

You might well have one called `run-flightlog.sh` and that is the UK one.

### **11.6.6 Running Flightlog**

The script `run-flightlog.sh` should be placed in the Flightlog directory if you need it and of course you can rename it. You must set it executable by running command:

`chmod +x run-flightlog.sh`

There is one optional parameter to this script which is the date to use of the backup file created at the end of running Flightlog and to make it easy to spot the latest we recommend that you use this format `yyyy-mm-dd` which will create a backup of `backup-yyyy-mm-dd.zip`. You can change the archiver used and the extension used such as `rar` so file is `backup-2019-05-30.rar` or `tar` so could be `backup-20190530.tar`

This script is also supplied in the Flightlog SVN code tree along with all other files.

Assuming the above has been done, lets get going by entering :

`run-Flightlog.sh`

Now continue with section Work Flow Overview on page 13 for a test work through of the various options in Flightlog. Don't worry if you just use test data as you can delete the three data files that have the extension of `.dat` and start again.

## 12Appendix R - Registering Flightlog

There are four types of registration:

1. To register your usage *only*, of Flightlog please send an email to the address shown on the inside front cover of this manual. This helps us know how many people are using Flightlog and who to contact if an important update becomes available including sending an updated archive.

Use "Flightlog Registration" in subject line (without the quotes).

Registrations details should include: Users name First and Last, Home Town and Country, platform used (windows, Linux (with distro and version and if x64 etc) as well as your email address that can receive updates as email archived attachments and these will be around 10Mb in size. This information will **not** be shared with anyone for any reason.

Support is only offered on a "as time is available" basis where all bug reports, forums and any updates are served via the Sourceforge website at:

[www.sourceforge.net/projects/Flightlog](http://www.sourceforge.net/projects/Flightlog)

ALL users are encouraged to use the tracker / bug reporting tool on the SF Flightlog website so that all users are aware of any issues and also the Forum / discussion boards for anyone to help resolve an issue. When reporting a problem always specify the version of Flightlog that you are using as well as your platform details (Linux, Windows etc). It is also a good place to see if any one else has reported a similar problem. We use them to report any updates or issues to Flightlog outside of the tracker service such as when there is a new version released.

2. For those users who require more conventional support (via email) you can again register providing the same information as in (1.), **and** via paypal sent to [vbcoen@gmail.com](mailto:vbcoen@gmail.com), pay a small fee depending on your location :

UK based £30 (pounds Sterling) includes VAT currently at 20%.

Or

Countries outside the UK £25 (pounds Sterling) no VAT.

These are one off payments for all updates within a major release, currently v1.

Invoices supplied on request. Note that fee is in Pounds Sterling,. Paypal may, make a small charge if your currency is not Sterling.

Below assumes that Flightlog is in use and any support is for Flightlog *itself* and its operations, or the manual or other supplied documentation.

For this you will get:

Support in a timely basis within level of issue priority of:

**Critical** – Software in use but cannot be used due to a bug.

**Normal** – Software in use but some aspects or function have a bug present but can still use it.

**Cosmetic** – Software or manual errors but cosmetic (non urgent) in nature. Does not prevent use of Flightlog in any way.

\*Estimated response times, from time email arrives:

Critical 4 to 24 hours\*.

Normal 1 – 5 days\* subject to any reported critical issues.

Cosmetic 5 days – 4 weeks\*.

\* Holidays or illnesses omitted, and remember that UK time is Zulu or GMT and work period is between the hours 12:00 and 20:00 per day, Monday through Friday usually, and even sometimes at the weekend.

The response times takes into account yours and our time zones and these times reflect worse case scenarios.

For example during a week day with a fault reported by mid day a fix should be dealt with same day at Critical level and hopefully Normal level as well.

This mean an updated copy of Flightlog will be sent directly to you, to test and use within the above response times.

Registration types 1 and 2 are for you, using the software on your own controlled computer for your use or that where there is no charge for any one else to use it. You may install Flightlog on other computers under your control or ownership. Remember to ensure that there are back up procedures in place both, on the system being used and away from it, such as using a cloud service as many do not change or it will be nominal for such a small amount of data.

While Flightlog could be run on a mobile device such as a tablet or even a mobile phone (yes, the Cobol compiler has been installed on both), it is not a recommended practice as such devices can be lost or damaged and as such, create a security risk for your data as users tend not to do back ups.

**3.** Commercial operations where Flightlog will be used by more than one person at no cost to them for doing so. Treated same as in **2**.

**4.** Commercial operations where Flightlog will be used by more than one person for a **cost** to them for doing so.

Contact us via email to discuss your requirements.

Flightlog “as is” is not really suitable for this type of operation without modifications unless each user has their own login and user name for the computer/s used.

## 13Appendix T - GnuCOBOL v3.0 Install Readme file Contents

*[These are here for your reference but you should read the files README and HACKING files included with the GnuCOBOL package in case of any updates].* Likewise also read the file NEWS for any later features added to the Cobol compiler if interested.

My version of these with extra notes, start at Appendix C – Installing the Cobol Compiler page 60.

Mine is shorter, as I have removed unneeded text but they are here for completeness.

If you require the various manuals for the compiler, available in both A4 and US letter formats and want the latest versions of each as a PDF file go to:

<https://sourceforge.net/p/open-cobol/code/HEAD/tree/external-doc/guide/PDFs/>

Where you will find, Programmers Guide, Quick Reference and the Sample Programs manuals.

### 13.1 Cobol Compiler requirements, Instructions for Build and Testing

GnuCOBOL

<https://www.gnu.org/software/GnuCOBOL/>

<https://sourceforge.net/projects/open-cobol>

<https://savannah.gnu.org/projects/GnuCOBOL>

GnuCOBOL is a free (like both in "free speech" and in "free beer") COBOL compiler, formerly known as OpenCOBOL.

It implements a substantial part of the COBOL 85, COBOL 2002 and COBOL 2014 standards, as well as many extensions included in other COBOL compilers.

GnuCOBOL translates COBOL into C and compiles the translated code using the native C compiler on various platforms, including Unix/Linux, Mac OS X, and Microsoft Windows.

This package contains the following subdirectories:

cobc	COBOL compiler
libcob	COBOL run-time library
bin	COBOL driver program
build_aux	Helper scripts
lib	Helper routines for missing OS functionality
config	Configuration files
po	International messages
doc	'info' and 'pdf' files
tests	Test suite (GnuCOBOL and framework for COBOL85)
extras	Useful COBOL programs

All programs except those in lib & libcob are distributed under the GNU General Public License. See COPYING for details.

Programs in lib and libcob are distributed under the GNU Lesser General Public License. See COPYING.LESSER for details.

See AUTHORS for the author of each file.

## 13.2 Requirements:

For all the following packages (required or optional),

BOTH runtime AND development components are necessary.

***All the following packages are normally part of a Linux distribution.*** Cygwin distribution also has these as installable packages, other operating systems also may have repositories for these - eg. MAC OS (OSX), CentOS and others all have package repositories. ALWAYS install the distribution packages when available at the current versions

GnuCOBOL REQUIRES one of the following external libraries to be installed for implementation of decimal arithmetic:

BOTH runtime AND development components are required.

- GNU MP (libgmp) 4.1.2 or later See <http://gmplib.org>

OR

- MPIR (libgmp - MPIR gmp-compat) 1.3.1 or later (preferred when compiling on Windows with other compilers than GCC) <http://mpir.org>

GNU MP and MPIR are distributed under GNU Lesser General Public License.

Please ALWAYS use the distro package whenever possible !! See NOTE above.

GnuCOBOL MAY require the following external libraries to be installed:

libltdl is NOT needed when installing on Linux, SUN Solaris, MAC OS, CentOS or Windows (including Cygwin, MingW and native windows).

It is also NOT needed with later versions of AIX and HP-UX. (AIX >= 5.1 and HP-UX >= 11.1 are known to NOT require this). (Check if you have the "dlopen" function).

- GNU Libtool (libltdl) <http://www.gnu.org/software/libtool/libtool.html>

libltdl is used to implement dynamic CALL statements.

GNU Libtool is distributed under GNU Lesser General Public License.

The following libraries ARE required WHEN :

1) Indexed-Sequential file I/O (ISAM) is used

BOTH runtime AND development components required.

One of the following:

- Berkeley DB (libdb) 4.1 or later [http://www.oracle.com/](http://www.oracle.com/http://www.oracle.com/technology/products/berkeley-db/db/index.html)  
<http://www.oracle.com/technology/products/berkeley-db/db/index.html>

Berkeley DB is distributed under Oracles own open-source license.

Note that if you linked your software with Berkeley DB, you must distribute the source code of your software along with your software, or you have to pay royalty to Oracle.

{For using Flightlog on your own or company systems this is not an issue so go ahead – This is the one used for all testing and production.}

OR

- VBISAM - ISAM file handler (libvbisam) 2.0 or later  
<http://sourceforge.net/projects/vbisam/>

OR

VBISAM is distributed under GNU Lesser General Public License.

- DISAM File handler (libdisam) <http://www.isamcentral.com>

DISAM is distributed under the proprietary License "Byte Designs Ltd. DISAM Software License".

2) SCREEN SECTION and/or extended ACCEPT/DISPLAY is used

BOTH runtime AND development components required.

One of the following:

- Ncurses (ncurses or ncursesw) 5.2 or later  
<http://www.gnu.org/software/ncurses/ncurses.html>

Ncurses is distributed under a BSD style license.

- Unix curses
- PDCurses (pdcurses) for MinGW/native windows ports <http://pdcurses.sourceforge.net>

## 13.3 Compiler Installation

***{ You have already done this if you followed the instructions in this manual.}***

The default installation path for GnuCOBOL is /usr/local.

The installation path may be changed by specifying `–prefix=<dir>` as a parameter to the configure program.

Further parameters may be specified to affect include/library search paths.

Execute `./configure --help` for further details.

To generate/install GnuCOBOL :

Configure and build

`./configure`

`make`

Here after successful completion, you must run

`make check` to run a series of GnuCOBOL test programs (must do!)

This MUST succeed - If not, please report. To one of the GC forums at sourceforge.

You should now perform a series of COBOL85 tests by.  
make test

It is highly recommended that you perform both these tests.

The language interpreter "perl" is required to run COBOL85 tests. If you build in Cygwin/MSYS you must use a Cygwin/MSYS version of perl.

Running "make test" will try to download the COBOL85 testsuite via the web, if it is missing. For details see tests/cobol85/README.

If you want to run both tests one after the other, you can run  
make checkall and if the compiler was previously not built this will also build the compiler.

Install  
make install

You generally need super-user privileges to execute "make install" unless you changed the installation directory with "./configure --prefix=<dir>" and have full access to <dir>.

To do so just run sudo make install

You need to know the spassword.

On Linux systems, if you are installing for the -first- time, you need to run "ldconfig" (as root). In fact, it does not hurt if you always do this every time you install an updated copy of the compiler. The special file for GnuCOBOL has been transferred to the /etc directory as */etc/ld.so.conf.d/gnu-cobol.conf* and this contains two or more lines as follows :

```
—  
/usr/local/lib/gnu-cobol  
/usr/local/lib  
/usr/local/mysql/lib  
/usr/lib  
—
```

You might want to add another line with  
/usr/lib64

but on my x64 system I do not.

- - - - Now to the official readme file.

On some Red Hat (Fedora) installations and possibly other Linux distros, /usr/local/lib is NOT automatically searched at runtime. Edit /etc/ld.so.conf (or the equivalent file) and add /usr/local/lib to the file.

Rerun "ldconfig".

If you think you have a problem or just want to record the make output, just redirect the output thus :

make 1>mymake.log 2>&1

make install 1>myinstall.log 2>&1



You can get back to a clean installation status by running : make distclean

The following is only interesting for advanced use. A normal user should not have recourse to use these options.

There are many configure options (see configure --help for a full list), these are the most important ones:

--with-db	Use Berkeley DB >= 4.1 (libdb) (ISAM handler) This is the default
--without-db	Do not use Berkeley DB / any other ISAM handler You will not be able to use indexed I/O
--with-vbislam	Use VBISAM (libvbislam) (ISAM handler)
--with-dl	Use the system dynamic linker This is the default
--without-dl	Use ltdl for dynamic program loading
--with-patch-level=<n>	Set internal patch level to n (default 0)
--with-varseq=<n>	Define the default format for variable length sequential files.

The default may be overridden at run time by  
setting the environment variable  
COB\_VARSEQ\_FORMAT to 0, 1, 2, or 3.

For values of 0, 1 and 2, four bytes are  
written preceding each record. The format of  
these four bytes for values of 0, 1, 2 is  
as follows :

n = 0 (default)

The first 2 bytes are the record length  
in big-endian order. This is compatible  
with mainframe. Bytes 3 and 4 are set  
to binary 0.

n = 1

The 4 bytes are the record length in  
big-endian order.

n = 2

The 4 bytes are the record length in  
native machine order (int).  
(This was previously the default)

For the value of 3, two bytes are written  
preceding each record :

n = 3

The first 2 bytes are the record length  
in big-endian order. The record follows

immediately after beginning at byte 3.

--enable-debug

Add '-g' debug option to make

## 13.4 Compiler Development

If you wish to hack the GnuCOBOL source code (or install it from the code repositories on Sourceforge), proceed as follows.

You need to install the following extra packages with specified minimum version before changing or building GnuCOBOL:

For compiling :

- Bison 2.3
- Flex 2.5.35

If you reconfigure and/or prepare a distribution (or install the compiler using the source code repositories)

- autoconf 2.63
- automake 1.10.1
- libtool 2.2.6
- m4 1.4.12
- gettext 0.17

Manual only:

- texinfo 4.12
- texlive (latest)

If you modify top-level configure.ac, Makefile.am in any directory, or any of the standard GC tests then you will need to run "autoreconf -I m4" to regenerate the necessary files.

If you have downloaded GnuCOBOL from the code repository assume the above but run :

`./build_aux/bootstrap` before running `configure`.

## 13.5 GnuCOBOL v3. after build Cobol 85 Validation Tests

### How to run the NIST CCVS85 (aka. ANSI85) Test Suite

{ You have already done this if you followed the instructions in this manual.}

You should run these tests and then compare results against the expected results.

#### \*NOTE:

It is expected that WARNING messages appear when running the test.  
The language interpreter "perl" is required to run these tests.

The final command of the test is a diff between expected results and actual results i.e.,  
diff summary.txt summary.log

If there is any output from this command, please tar and compress the complete cobol85 directory and report this to the GC list. You will receive further instructions where to send this.

This test can take a **long** time depending on your hardware.

#### 1. Run the whole test suite:

```
make checkall
```

#### 2. Test report summary will be put in summary.log.

#### 3. When rerunning tests as a result of a change, always do a "make clean" before "make checkall"

#### 4. The default GC configuration tests are NC SM IC SQ RL IX ST SG OB IF RW

### Make Options

- |                 |  |
|-----------------|--|
| - make test     | Run the test suite                               |
| - make save     | Save test reports in *.txt                       |
| - make diff     | diff from *.txt to the last reports              |
| - make clean    | Remove built files                               |
| - make checkall | Runs all tests and will compile first if needed. |

### Test Modules

#### Core tests:

NC - COBOL nucleus tests  
SM - COPY sentence tests  
IC - CALL sentence tests

#### File I-O tests:

SQ - Sequential file I-O tests  
RL - Relative file I-O tests  
IX - Indexed file I-O tests  
ST - SORT sentence tests

Advanced facilities:

RW - REPORT SECTION tests

CM - COMMUNICATION SECTION tests

IF - Intrinsic Function tests

SG - Segment tests

DB - Debugging facilities tests

OB - Obsolete facilities tests

Tests are being added all the time so this list might well be incomplete.