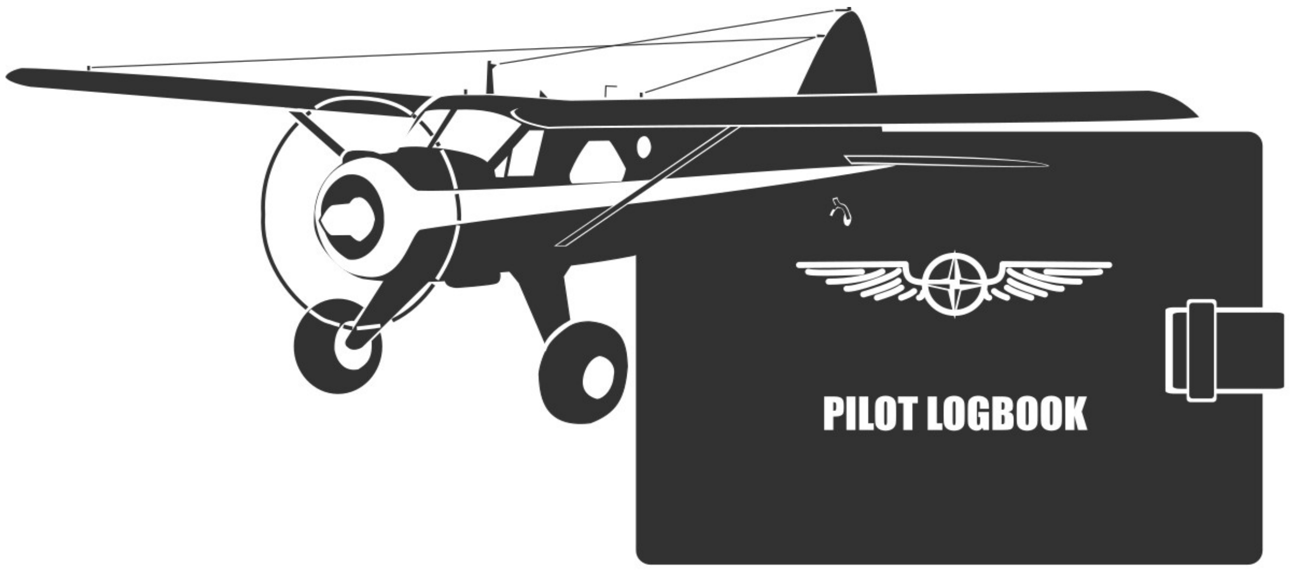


FS Logbook Editor

Old-School Freeware Edition

https://mouseviator.com

some thoughts someone may find helpful

Table of Contents

1. Legal Notices & Trademarks....8
    1.1. Trademarks....8
2. Introduction....9
    2.1. What is FS Logbook Editor?....9
    2.2. System Requirements....9
    2.3. Welcome to my PC (Installation)....9
    2.4. Again in my PC (Updating)....15
    2.5. Get OFF my PC (Uninstallation)....16
    2.6. Support & Updates....18
    2.7. First start....18
    2.8. News in version 1.86....19
    2.9. News in version 1.85....20
    2.10. News in version 1.80....20
    2.11. News in version 1.65....21
    2.12. News in 1.50 version....22
    2.13. News in 1.42 version....22
    2.14. News in 1.4 version....23
3. The Program....24
    3.1. Editor....24
    3.2. Edit logbook record dialog....25
        3.2.1. Airport List....26
    3.3. Importing flights from FSHub....27
    3.4. Importing flights from Microsoft Flight Simulator 2020....29
        3.4.1. Import using clipboard....29
        3.4.2. Import using network....31
    3.5. Load / save logbook from / as CSV file....33
        3.5.1. Loading logbook from CSV file....33
        3.5.2. Saving logbook into CSV file....34
    3.6. Load / save logbook from / as LittleNavMap CSV file....36
        3.6.1. Loading logbook from LittleNavMap CSV file....36
        3.6.2. Saving logbook into LittleNavMap CSV file....39
    3.7. Load / save logbook from / as Excel spreadsheet....41
        3.7.1. Loading logbook from Excel spreadsheet....41
        3.7.2. Saving logbook as Excel spreadsheet....42
    3.8. Load / save logbook from / as JSON file....44
    3.9. Viewing flight on map....44
    3.10. Filtering logbook records....47
    3.11. Printing logbook table....50
    3.12. Personal Minimums....51
        3.12.1. Add personal minimum....52
        3.12.2. Change personal minimum....52
        3.12.3. Delete personal minimum....52
        3.12.4. Checking personal minimums....52
    3.13. Pilot Statistics....53
        3.13.1. User aircraft categories....54
        3.13.2. Pilot certifications....56
    3.14. Log....57
        3.14.1. Log files....58

4. Menus.......59
4.1. File.......59
4.2. Record.......65
4.3. JasperReports.......65
4.3.1. How to make the report?.......66
4.4. Tools.......69
4.4.1. Little More Than Dumb Text Editor.......69
4.5. Help.......70
5. Settings.......71
5.1. General.......71
5.2. Load & Save.......75
5.2.1. Note on Encoding.......80
5.3. Logging.......82
5.4. Simulators.......83
5.4.1. Fields description.......83
5.4.2. Adding flight simulator.......84
5.4.3. Updating flight simulator.......84
5.4.4. Deleting flight simulator.......85
5.4.5. Automatically discovering flight simulators.......85
5.4.6. Example – adding “dummy” flight simulator for importing Navigraph airport data.......85
5.5. Database.......87
5.6. Airport Data.......89
5.6.1. Importing airport data for FS2004 / FSX / Prepar3D.......90
5.6.2. Importing airport data for Microsoft Flight Simulator.......92
5.6.3. Importing airport data from X-Plane.......94
5.6.4. Importing airport data from Navigraph data cycle.......95
5.7. Map Viewer.......98
5.8. JasperReports.......100
5.8.1. File virtualizer.......101
5.8.2. Swap file virtualizer.......102
5.8.3. GZip virtualizer.......102
5.9. FsHub.......103
5.10. MSFS.......105
5.10.1. Embed server options.......108
5.10.2. Mouseviator Enhanced Logbook for MSFS.......108
6. Customizing – a bit advanced topic.......110
6.1. Customizing date formats.......110
6.2. Customizing ods template.......111
6.3. Translating FS Logbook Editor.......113
6.4. Writing custom pilot statistics template.......115
6.4.1. Where are the templates stored?.......115
6.4.2. Template file structure.......115
6.4.3. Supported statistics types.......116
6.4.4. Template tags.......117
6.4.5. Paired tags.......117
6.4.6. Simple tags.......119
6.4.7. Value formatting.......122
6.4.8. Final words for statistics templates.......124
6.5. Creating JasperReports template for Logbook.......125
6.5.1. Examples.......125

6.5.2. Fields........126
6.5.3. Parameters........127
6.5.4. A scriptlet (with some useful functions)........127
6.5.5. Creating simple template using Jaspersoft Studio........129
6.5.5.1. Creating the title........131
6.5.5.2. Defining fields........133
6.5.5.3. Defining variables........135
6.5.5.4. Defining Scriptlet........138
6.5.5.5. Creating the body of the template, the Details band........140
6.5.5.6. Creating the airport list........143
6.5.5.7. Inserting page number, the Page Footer band........146
6.5.5.8. Creating the summary, the Summary band........146
6.5.5.9. The last steps, compiling the template........147
7. Information supported by different logbook files........150
7.1. Flight Simulator X binary logbook (.bin)........150
7.2. Flight Simulator 2004 logbook (.log)........150
7.3. X-Plane 10/11 logbook (.txt)........151
7.4. FS Logbook Editor XML file (.fslex)........152
8. Command line parameters........153
9. 3rd party resources used........154
9.1. Using another version of 3rd party library........155
10. Thanks........156

List of changes

FS Logbook Editor version 1.86

- Most changes are described in the ***News in version 1.86*** chapter.
- Updated the ***Load & Save*** chapter with description of new settings: **Only allow logbook.bin files in automatic logbook find dialog for FSX/Prepar3D**, **Use FPC when loading files** and **Use FPC when saving files**.
- Added chapter: ***Command line parameters***. I think its name is self explanatory. It is not a big chapter, FS Logbook Editor supports just one parameter for now :)

FS Logbook Editor version 1.85

- Updated the chapter: ***Importing airport data for FS2004 / FSX / Prepar3D*** and: ***Importing airport data for Microsoft Flight Simulator*** with description of new **Try to remove invalid XML characters** option.
- Updated the: ***Loading logbook from CSV file*** and the: ***Saving logbook into CSV file*** chapter describing new options added due to the usage of new CSV library to work with CSV files.
- Updated the chapter ***Viewing flight on map*** with description of new multi-flight Map Viewer window and also the ***Map Viewer*** section of ***Settings*** that describes new options related to this new functionality.
- Updated the ***Importing flights from FSHub*** chapter ***Importing flights from Microsoft Flight Simulator 2020*** with description on new functionalities added in FS Logbook Editor version 1.81 and 1.85. The ***FsHub*** section and ***MSFS*** section of ***Settings*** has also been updated with description of new options added (mostly related to aircraft type matching against user aircraft categories)
- Updated the ***User aircraft categories*** chapter with description of new Aircraft type combo box added in FS Logbook Editor version 1.85.
- Updated the ***3rd party resources used*** chapter.

FS Logbook Editor version 1.80

- New chapter - ***Importing flights from Microsoft Flight Simulator 2020*** and ***Settings*** → MSFS documenting new functionality of importing flights from Microsoft Flight Simulator 2020.
- New chapter – ***Importing flights from FSHub*** and ***Settings*** → ***FsHub*** documenting new functionality of importing flights from FsHub.
- New chapter – ***Little More Than Dumb Text Editor*** describing in short new embed text editor.
- Updated the ***Pilot Statistics*** chapter, describing new option to mark user aircraft category as containing multi-engine aircraft, new **Open in editor…** button and new possibilities to save pilot statistics as PDF, ODT and DOCX.
- Updated the ***File*** chapter with description of new file formats that logbook can loaded from / saved to. These file formats include JSON, CSV and Excel spreadsheets.
- New chapter – ***Load / save logbook from / as CSV file*** describing in more detail loading and saving logbook from / as CSV file.
- New chapter - ***Load / save logbook from / as Excel spreadsheet*** describing in more detail

new functionality to load /save logbook from /as Excel spreadsheet.

- Updated the ***Load & Save*** section of ***Settings*** with description of new **Rows to read for preview** option in the **CSV Files** options, and the same **Rows to read for preview option** in new **Excel Files** options.
- Updated the ***3rd party resources used*** chapter with new/changed libraries.
- Added **XLSX** and **DOCX** option to the list of export formats supported by JasperReports. See chapter: ***JasperReports***

FS Logbook Editor version 1.70

- New chapter: ***Importing airport data for Microsoft Flight Simulator*** about importing airport data from Microsoft Flight Simulator.
- Updated the ***3rd party resources used*** chapter with new/changed libraries.
- Added **PDF** option to the list of export formats supported by JasperReports. See chapter: ***JasperReports***

FS Logbook Editor version 1.65

- Updated chapter about ***Importing airport data for FS2004 / FSX / Prepar3D*** for new option and behavior.
- Updated the ***General*** chapter of ***Settings*** chapter for new options. And their explanation too. The new options allows to change many program fonts and also the Look and Feel of the application completely.

FS Logbook Editor version 1.60

- Added new paragraph describing the functionality of the up (↑) and down (↓) buttons for the Airport list at Edit logbook record dialog in the ***Airport List*** chapter.
- Added ***Picture 55: Print options*** and description of new printing options in the ***General*** section of ***Settings***. That would be kind of nonsense without also adding a chapter on printing - ***Error: Reference source not found***.
- Added chapter: ***JasperReports*** on generating JasperReport reports from logbook. Also added the related chapter on ***JasperReports*** setting options.
- Updated a ***3rd party resources used*** chapter with new libraries (and updated version of old libraries).
- Updated image ***Picture 53: General section of Settings*** settings. Added description of the new **Check for new version** option in the ***Settings*** → ***General*** chapter.

FS Logbook Editor version 1.50

- Updated the paragraph of **KML Export Settings** in the ***Load & Save*** chapter with the new **Coordinates format:** and **Magnetic variation format:** options.
- Updated the ***Database*** chapter to describe new controls and new support for MySQL, MariaDB and Microsoft SQL Server database systems.
- Added ***Viewing flight on map*** chapter about viewing flights in Map Viewer.
- Added ***Map Viewer*** chapter under ***Settings*** chapter about customizing the display of flight route and flight information at Map Viewer window.
- Added ***Filtering logbook records*** chapter describing the new logbook

records filtering feature.
- Updated the ***Load & Save*** chapter under ***Settings***, describing two new options to automatically load and save last filtering conditions.
- Updated images and description of import dialog at ***Importing airport data from X-Plane*** and ***Importing airport data from Navigraph data cycle*** chapter. These dialog have new option to select files encoding, for the case they are not UTF-8 encoded.
- Updated list of files being recreated by FS Logbook Editor in case they do not exists and that are left behind in case of uninstalling FS Logbook Editor in the ***Again in my PC (Updating)*** and ***Get OFF my PC (Uninstallation)*** chapters.
- Updated list of 3rd party resources used in the ***3rd party resources used*** chapter.
- Updated various images throughout this document to match current program state.

FS Logbook Editor version 1.42

- Removed description of Logbook encoding from ***Load & Save*** from ***Settings***. The description has been modified and placed to separate chapter ***Note on Encoding***.
- Added description for **Encoding** option to **FS2004 Files** options at ***Load & Save*** in ***Settings***.
- Added description **X-Plane Files** options at ***Load & Save*** in ***Settings***.
- Changed picture ***Picture 45: Dialog to select target simulator*** to display the new dialog. Changed **Note on Prepar3D v2 files** and added **Note on Prepar3D v4 files** to the description of file formats in ***File*** chapter under ***Menus*** chapter.

FS Logbook Editor version 1.40

- Added new chapter ***News in 1.4 version***, which describes the news of version 1.4.
- Added description of **Save Selected As…** and **Keyhole Markup Language (KML) – to view in Google Earth** file format to ***Menus*** → ***File*** chapter.
- Added description of **Airport Data Provider** at ***Editor*** tab chapter.
- Added description of **Date, time and other formatting** option group at the ***General*** section of ***Settings***.
- New chapters – ***Simulators***, ***Database*** and ***Airport Data*** under ***Settings*** chapter.
- Added description of new statistics types to ***Supported statistics types*** chapter ,new template tags to ***Template tags*** chapter, new paired tags to ***Paired tags*** chapter and description of **Tags for distance flown statistics** to ***Simple tags*** chapter.
- Added names of new styles for origin and destination airport name, city, country and state to ***Customizing ods template*** chapter.
- Updated list of translation files at ***Translating FS Logbook Editor*** with new *Database<your language code>.properties* file.
- Added new libraries to ***3rd party resources used*** chapter.

FS Logbook Editor version 1.31

- Updated ***Settings*** → ***Load & Save*** dialog image and description. There are two new options: **Save records in view order** and **FS Logbook Editor XML Files: Autosave when saving to other format**.
- Updated ***Menus*** → ***File*** chapter. The description of new menu items ***Open Logbook (Auto find...)*** and ***Recent Files...*** was added.

- Description of ***About*** menu item (About application dialog) in ***Help*** menu updated.

FS Logbook Editor version 1.30

- Updated ***Open Logbook*** and ***Save*** in the ***Menus*** → ***File*** chapter for the new file formats – **X-Plane 10 logbook (.log)** and **FS Logbook Editor XML file (.flex)**
- New chapter – ***Pilot Statistics*** that describes the new Pilot Statistics tab.
- New chapter – ***Writing custom pilot statistics template*** that describes how to write custom template for pilot statistics output.
- New chapter – ***Information supported by different logbook files*** that describes which logbook record attributes are supported by which file format (flight simulator).
- Updated list of translation files at ***Translating FS Logbook Editor*** with new *PilotStatistics<your language code>.properties* file.
- Updated chapter ***Again in my PC (Updating)*** for a list of default files related to pilot certifications, user aircraft categories and pilot statistics templates.
- Updated images.

FS Logbook Editor version 1.15

- ***Airport List*** chapter slightly updated.
- *Utils.properties* added to the list of translation files in the ***Translating FS Logbook Editor*** chapter.
- ***A Note on Prepar3D v2.2*** added to ***File*** → ***Save As*** paragraph. It is related to different saving of comments by Prepar3D v2.2 and the corrections introduced into FS Logbook Editor due to that.

FS Logbook Editor version 1.10

- Settings → ***General*** chapter updated to reflect changes in FS Logbook Editor version 1.10.
- Settings → ***Load & Save*** chapter updated to reflect changes in FS Logbook Editor version 1.10.
- ***Again in my PC (Updating)*** chapter added.

1. Legal Notices & Trademarks

The use of this software is bounded by End User License Agreement which you accepted during installation. For more details, please see the **License.txt** or **License.pdf** in the **docs** folder in program folder or click the License option in the Help menu. If you do not agree with the EULA, please do not use this software.

1.1. Trademarks

The following trademarks are referenced in this document:

- Flight Simulator 2004
- Flight Simulator X
- Microsoft Flight Simulator (2020)
- Prepar3D® - © 2013 Lockheed Martin Corporation. All Rights Reserved.
- X-Plane 10® - Laminar Research. All Rights Reserved.
- X-Plane 11® - Laminar Research. All Rights Reserved.
- X-Plane 12® - Laminar Research. All Rights Reserved.

2. Introduction

Welcome,

This is considered to be manual for FS Logbook Editor software, which should answer some of your questions, like: What the hell is FS Logbook Editor? Hope it contains a good ratio of boring readings, pictures and useful information.

2.1. What is FS Logbook Editor?

FS Logbook Editor is small program to manage Flight Simulator logbook. Simply saying, that means opening of logbook, editing, adding and deleting records and of course saving. Aside from those basic functions, the program can also join two or more records into one, append one logbook to another or merge two logbooks.

FS Logbook Editor was primarily designed for Flight Simulator X logbook – so it has the same limitations. But also logbooks from Flight Simulator 2004, Prepar3D®[1] and Prepar3D® v2[2] can be opened and/or saved to.

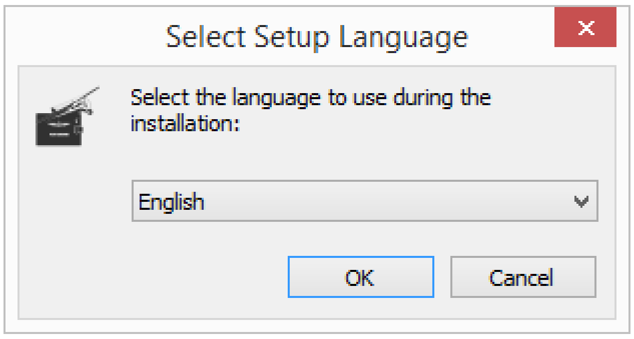
2.2. System Requirements

FS Logbook Editor requires Java Run-time Environment 1.7 or newer. To save you from trouble, the Java Run-time Environment is packed with program itself, so you don't have to care to get one. If you concern about minimum system specifications, there are no specifications. I believe no average computer of today should have any problem running FS Logbook Editor (considering the fact that it handles some flight simulator software – why would you use FS Logbook Editor without simulator anyway?).

2.3. Welcome to my PC (Installation)

FS Logbook Editor gets installed via standard installation wizard probably familiar to you to some extent from another software installers. All you have to do is follow the wizard instructions, choose installation folder (if you don't like the default choice), choose whether to create desktop icon or not and let the installer install the program for you.

The first dialog the installer will show is the language selection dialog.


Select Setup Language

Select the language to use during the installation:

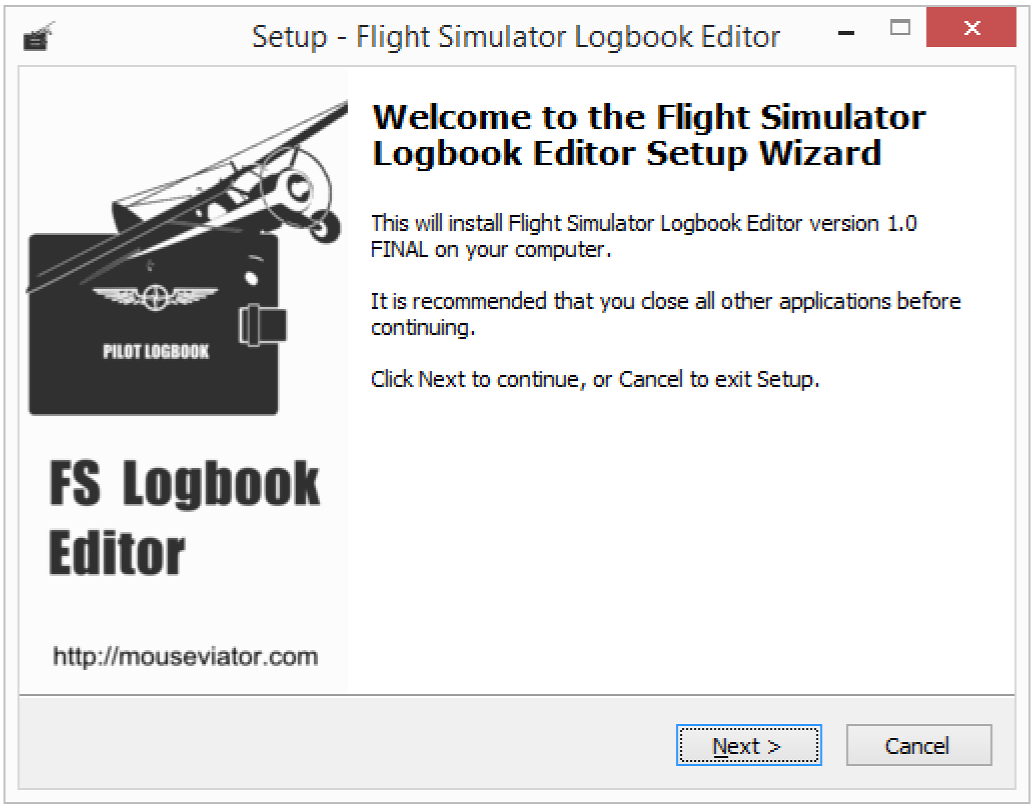
English

Picture 1: Setup language selection dialog

1 As long as the logbook file format stays the same as FSX's.
2 As long as the logbook file format stays the same as FSX's.

Choose one of supported languages which you are most familiar with. This selection is only for installer language, it has nothing to do with languages FS Logbook Editor is able to communicate with you.

The next window should be the welcome screen:



Welcome to the Flight Simulator Logbook Editor Setup Wizard

This will install Flight Simulator Logbook Editor version 1.0 FINAL on your computer.

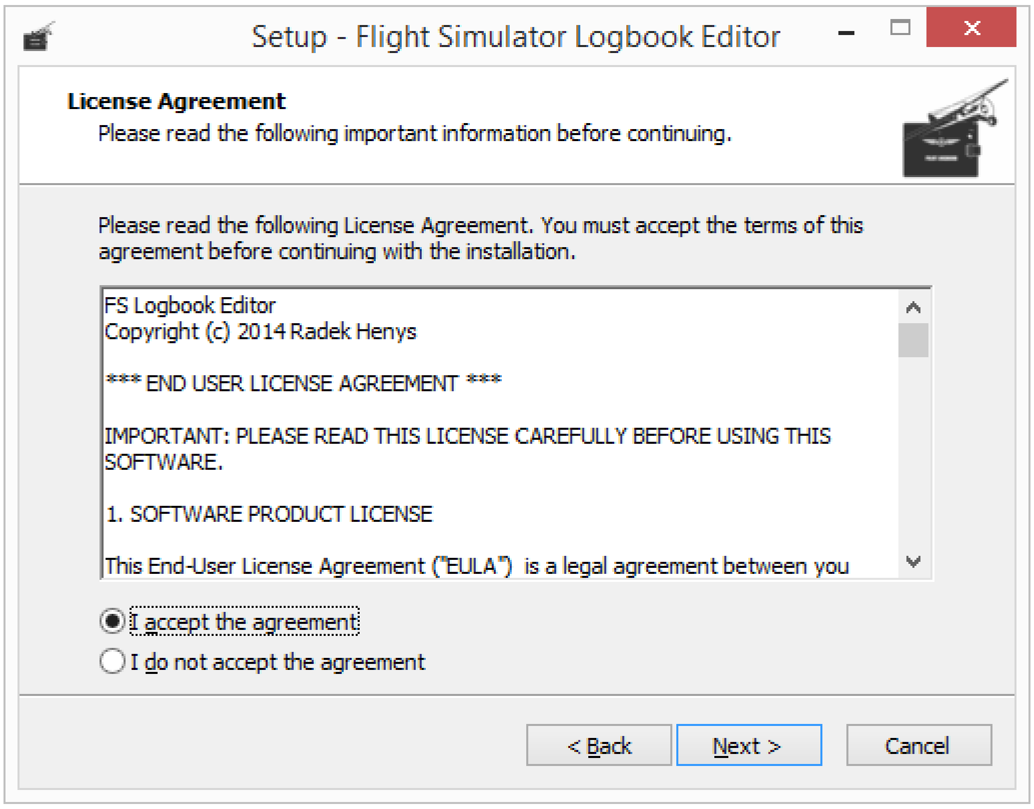
It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Picture 2: Setup welcome screen

On the next screen you must accept FS Logbook Editor End User License Agreement (EULA) in order to continue setup.

<content follows on next page>



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

FS Logbook Editor
Copyright (c) 2014 Radek Henys

*** END USER LICENSE AGREEMENT ***

IMPORTANT: PLEASE READ THIS LICENSE CAREFULLY BEFORE USING THIS SOFTWARE.

1. SOFTWARE PRODUCT LICENSE

This End-User License Agreement ("EULA") is a legal agreement between you

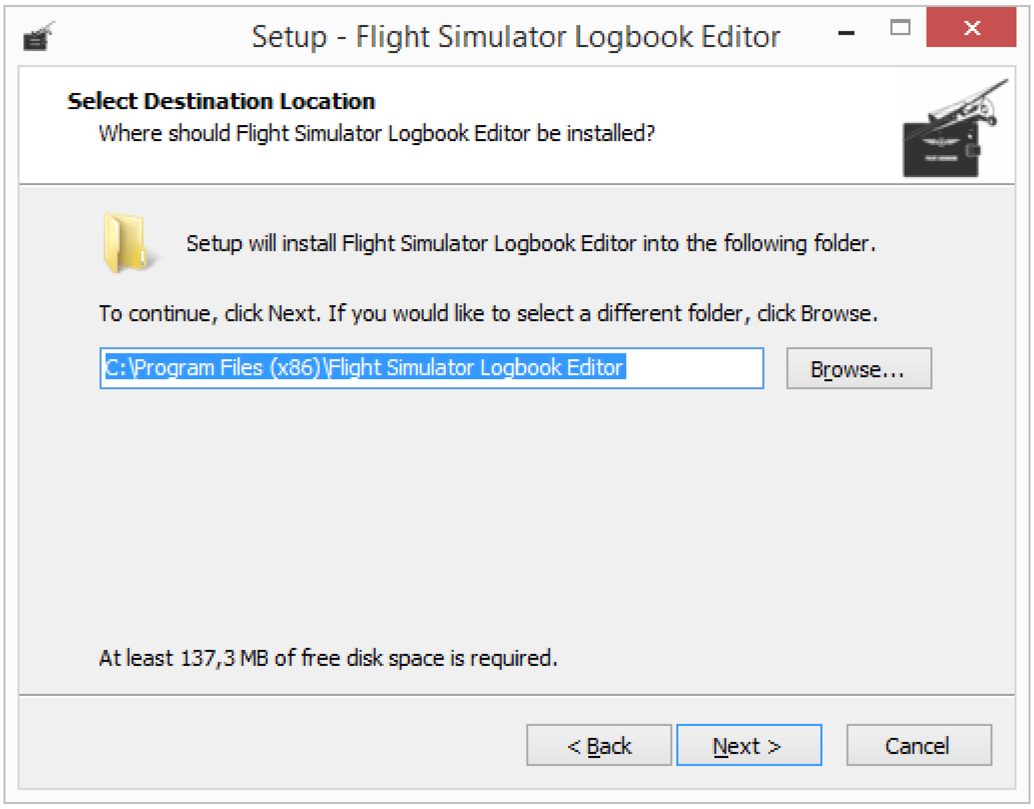
[✓ I accept the agreement]
I do not accept the agreement

Picture 3: End User License Agreement dialog

Worth mentioning is that you should actually read it, because by choosing accept option and clicking the Next button you hereby agree that you have done so. Well, not everyone reads EULA for every software he or she installs, but it is solely your responsibility to know what you agree with by continuing with the installation.

In the next step, you can select which folder to install FS Logbook Editor into. Make your choice or leave the default one:

<content follows on next page>


Setup - Flight Simulator Logbook Editor

Select Destination Location

Where should Flight Simulator Logbook Editor be installed?

Setup will install Flight Simulator Logbook Editor into the following folder.

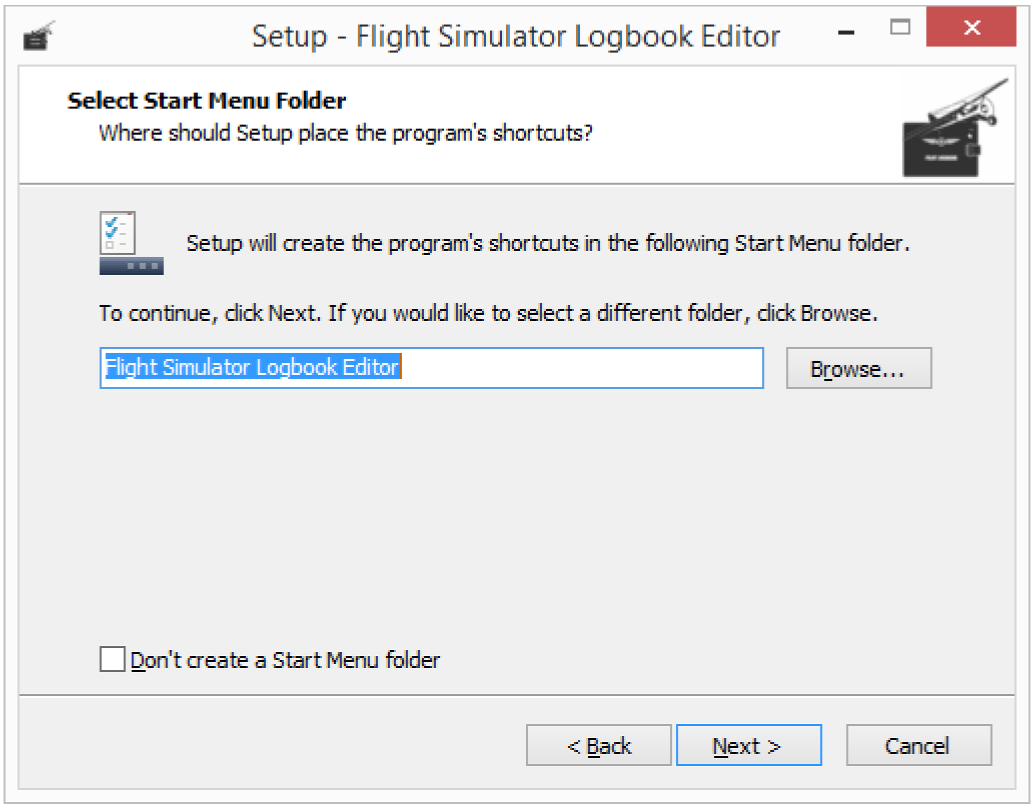
To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files (x86)\Flight Simulator Logbook Editor

At least 137,3 MB of free disk space is required.

Picture 4: Installation folder selection

Following window informs you that a folder (or group) with program shortcuts will be created in start menu. You can define your own folder or choose not to create the folder at all (no program shortcuts in start menu then):


Setup - Flight Simulator Logbook Editor

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

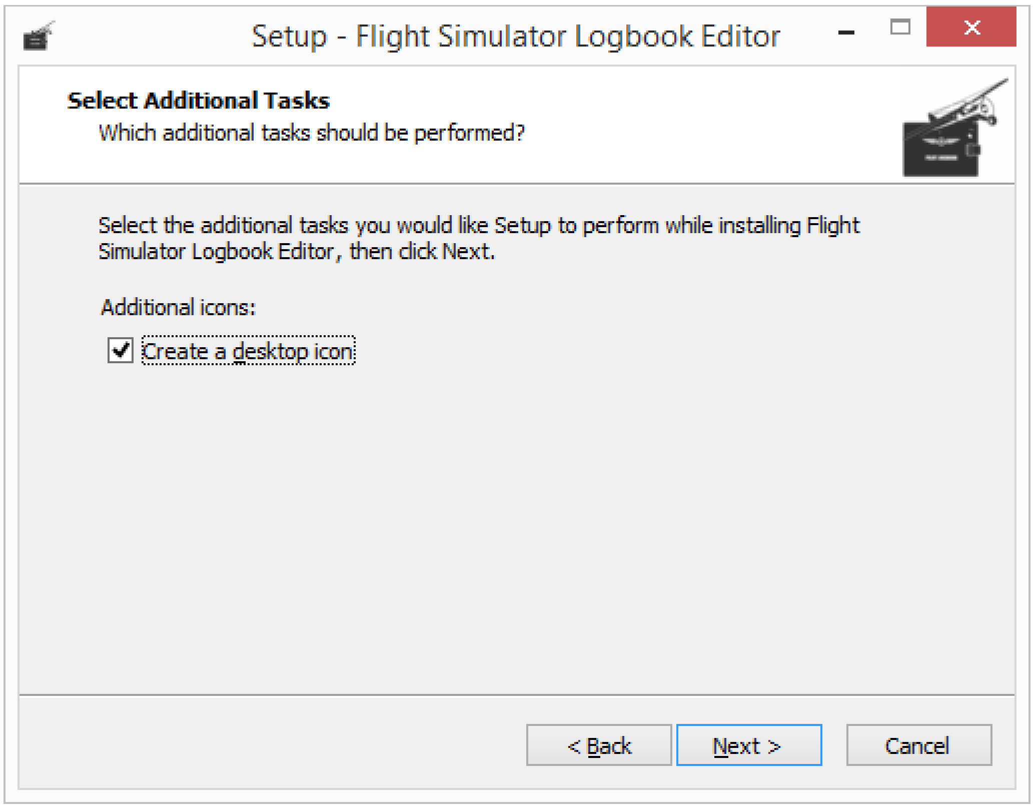
To continue, click Next. If you would like to select a different folder, click Browse.

Flight Simulator Logbook Editor

☐ Don't create a Start Menu folder

Picture 5: Start Menu folder selection

If you do not want a shortcut to be placed on your desktop, there is an option to cancel:


Setup - Flight Simulator Logbook Editor

Select Additional Tasks

Which additional tasks should be performed?

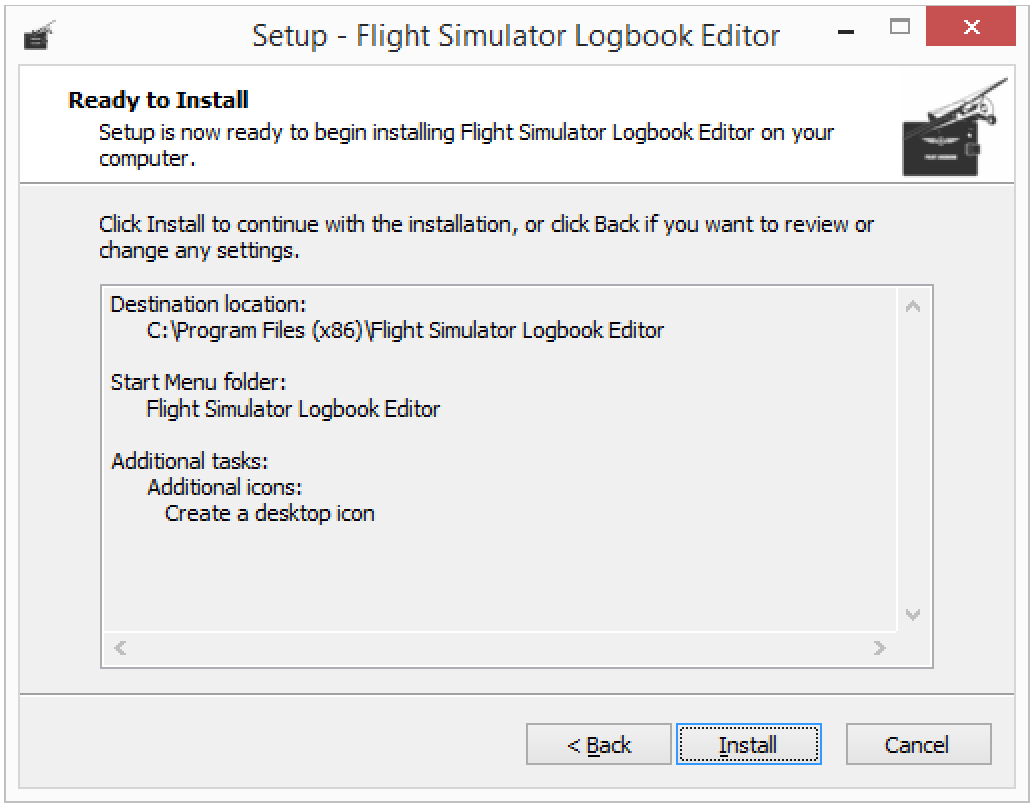
Select the additional tasks you would like Setup to perform while installing Flight Simulator Logbook Editor, then click Next.

Additional icons:

- [✓] Create a desktop icon

Picture 6: Select whether to create desktop icon or not

Before you finally install FS Logbook Editor, there is a summary window listing the options you have chosen within preceding steps:


Setup - Flight Simulator Logbook Editor

Ready to Install

Setup is now ready to begin installing Flight Simulator Logbook Editor on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

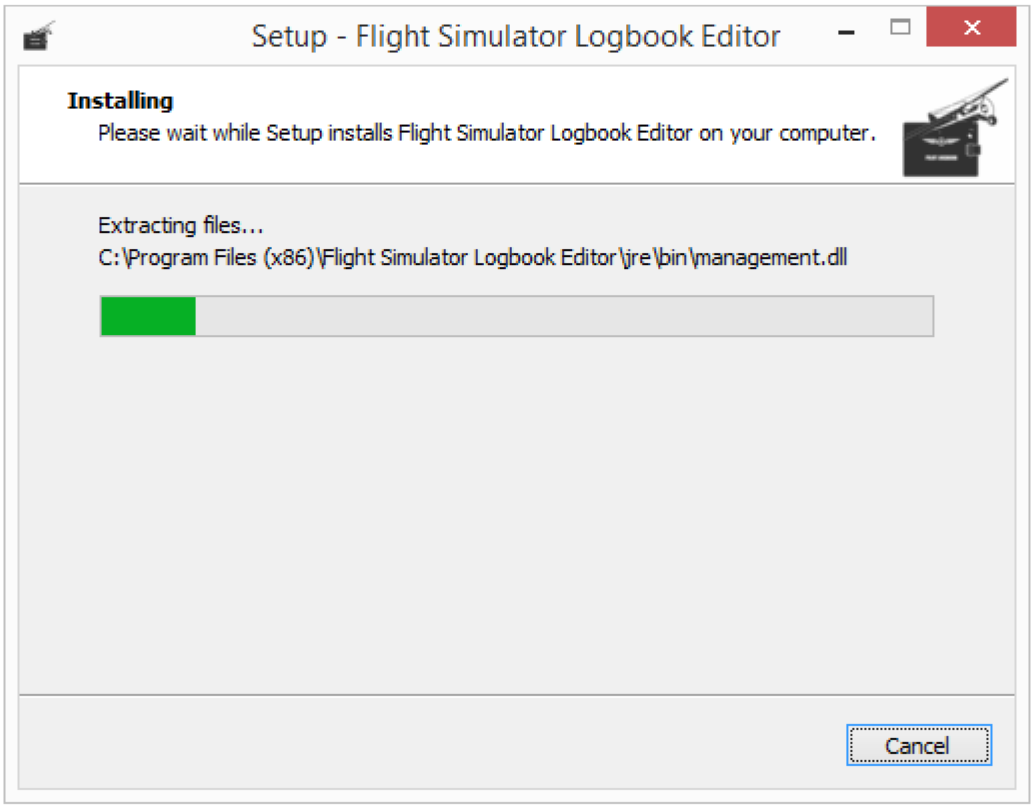
```
Destination location:
    C:\Program Files (x86)\Flight Simulator Logbook Editor

Start Menu folder:
    Flight Simulator Logbook Editor

Additional tasks:
    Additional icons:
      Create a desktop icon
```

Picture 7: Installation summary

Finally, wait a moment before FS Logbook Editor is installed:


Setup - Flight Simulator Logbook Editor

Installing

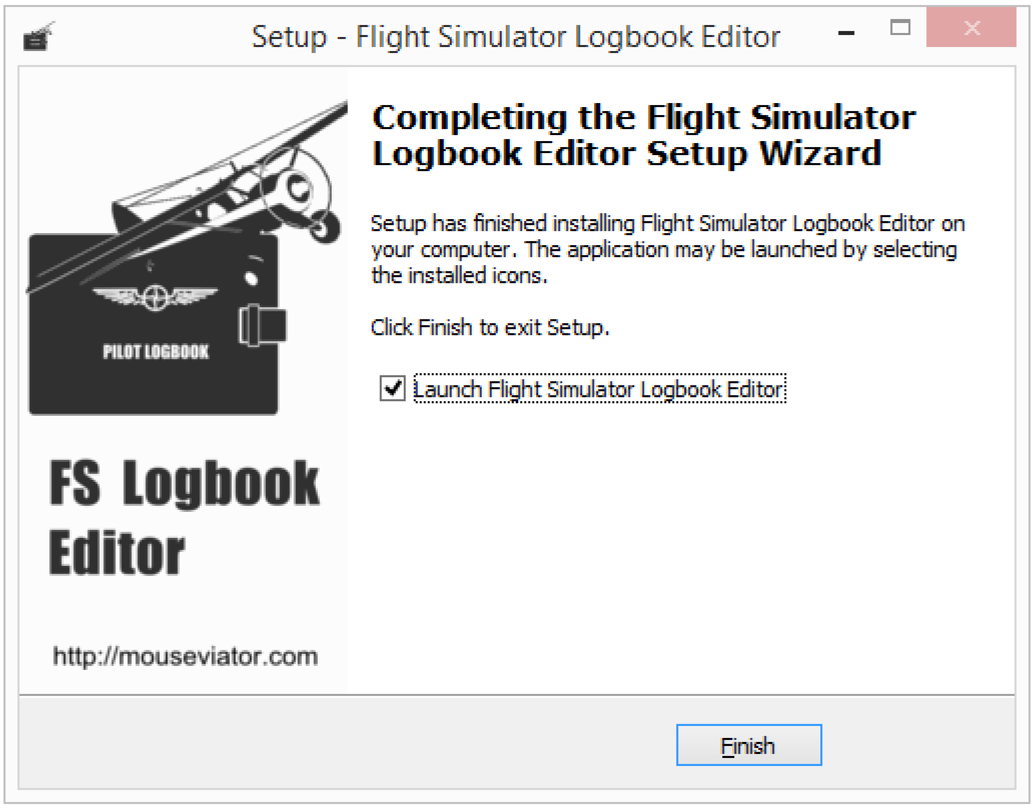
Please wait while Setup installs Flight Simulator Logbook Editor on your computer.

Extracting files...

C:\Program Files (x86)\Flight Simulator Logbook Editor\jre\bin\management.dll

Picture 8: Installation progress

The last window the installer will bother you with is ... the final window. There is only Finish button and you can choose to start FS Logbook Editor right away, or not.


Setup - Flight Simulator Logbook Editor

Completing the Flight Simulator Logbook Editor Setup Wizard

Setup has finished installing Flight Simulator Logbook Editor on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

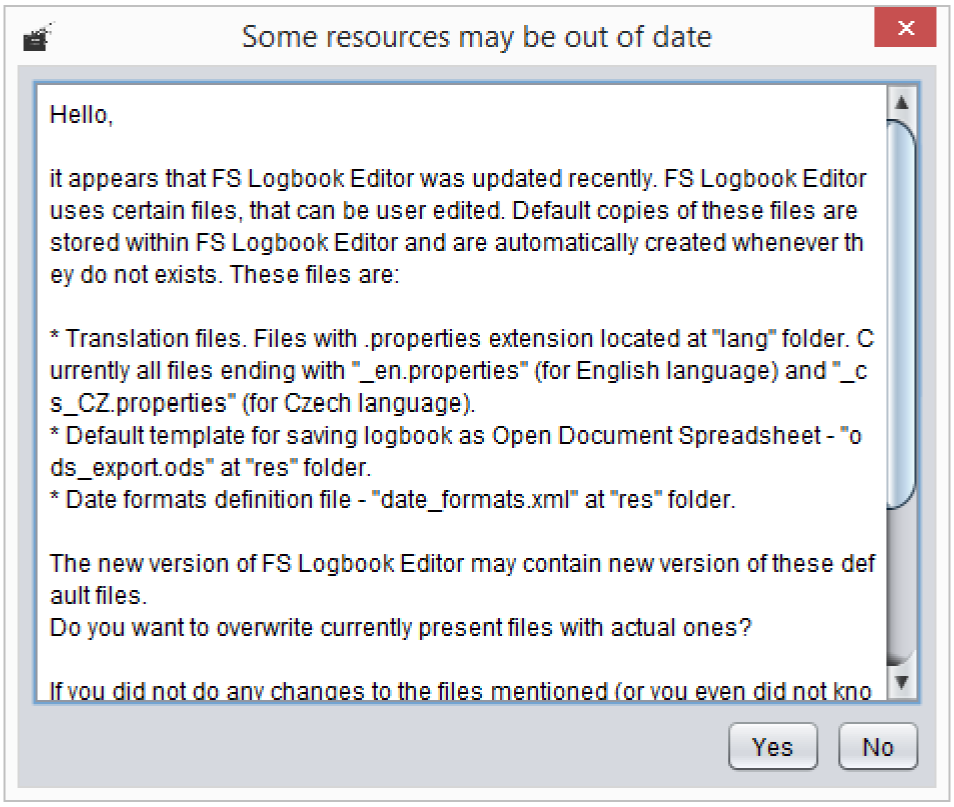
✓ Launch Flight Simulator Logbook Editor

Picture 9: Final installer window

2.4. Again in my PC (Updating)

If there is an update to the FS Logbook Editor, it should come as a new installer. Follow the same steps as in the case of installation (previous chapter). Make sure FS Logbook Editor is closed before installing newer version and let the new version install over the old version. Next time FS Logbook Editor starts, it should detect that it was updated and display certain notifications about it.

First, a dialog saying that some resources may be outdated should be displayed to you:



Hello,

it appears that FS Logbook Editor was updated recently. FS Logbook Editor uses certain files, that can be user edited. Default copies of these files are stored within FS Logbook Editor and are automatically created whenever th ey do not exists. These files are:

* Translation files. Files with .properties extension located at "lang" folder. C urrently all files ending with "_en.properties" (for English language) and "_c s_CZ.properties" (for Czech language).
* Default template for saving logbook as Open Document Spreadsheet - "o ds_export.ods" at "res" folder.
* Date formats definition file - "date_formats.xml" at "res" folder.

The new version of FS Logbook Editor may contain new version of these def ault files.
Do you want to overwrite currently present files with actual ones?

If you did not do any changes to the files mentioned (or you even did not kno

Picture 10: Resources update dialog

FS Logbook Editor uses certain files, that can be user edited. Default copies of these files are stored within FS Logbook Editor and are automatically created whenever they do not exists. Most of them are related to ***Customizing – a bit advanced topic***. The files are:

- **Translation files.** Files with .**properties** extension located at **lang** folder. Currently all files ending with *"_en.properties"* (for English language) and *"_cs_CZ.properties"* (for Czech language). See ***Translating FS Logbook Editor*** for more details.
- **Default template for saving logbook as Open Document Spreadsheet** - *"ods_export.ods"* at **res** folder. See ***Customizing ods template*** for more details.
- **Default template for saving logbook as Excel Spreadsheet** - *"xls_export.xls"* and *"xlsx_export.xlsx"* at **res** folder. See ***Load / save logbook from / as Excel spreadsheet*** for more details.
- **Date formats definition file** - *"date_formats.xml"* at **res** folder. See ***Customizing date formats*** for more details.
- **Default pilot statistics templates** (*.txt files) in **ps_templates** folder. The default templates are:
  - Flight_Times_CSV_en.txt
  - Letove_casy_CSV_cs_CZ.txt

  - Line_Style_en.txt
  - Radkovy_vypis_cs_CZ.txt
  - Table_Style_en.txt
  - Tabulkovy_vypis_cs_CZ.txt
  - Takeoffs_and_Landings_CSV_en.txt
  - Vzlety_a_pristani_CSV_cs_CZ.txt
- **Default pilot certifications definitions**, stored at FS Logbook Editor program folder:
  - pilot_certifications_en.xml
  - pilot_certifications_cs_CZ.xml
- **Default user aircraft categories definitions**, stored at FS Logbook Editor program folder:
  - user_aircraft_categories_en.xml
  - user_aircraft_categories_cs_CZ.xml
- Files in **res\kml_icons** – icons used when exporting logbook to KML file.
- Files in **res\map_icons** – icons used for displaying airports at map at Map Viewer window.
- **Makerwys.exe** and **LorbySceneryExport.exe** in **res** folder. Used when importing FSX/P3D airport data.
- Files in **filters** folder. The xml files storing default, last used and optionally user filters to filter logbook records.
- Files in **res/fshub** folder. This folder contains files related to the functionality of importing flights from FsHub. The **templates** sub-folder contains default comment templates.
- Files in **res/msfs** folder. This folder contains files related to the functionality of importing flights from MSFS. The **templates** sub-folder contains default comment templates.
- Files in **res/lnm** folder. This folder contains files related to the functionality of importing LittleNavMap CSV files. The **templates** sub-folder contains default comment templates.
- Files in **res/jasperreports** folder. This folder contains default templates for generating JasperReports.

The new version of FS Logbook Editor may contain new version of these default files and the dialog asks you if you want to overwrite currently present files with actual ones?

If you did not do any changes to the files mentioned (or you even did not know they are there), click **Yes**.

If you did any changes to the files mentioned and you do not want to lost them – click **No**. If you want to compare what has changed, move the files to somewhere else (for FS Logbook Editor this is the same like deleting them). FS Logbook Editor will recreate all missing files next time it starts.

FS Logbook Editor will then continue starting as usual. As a last step, a change log should be displayed.

2.5. Get OFF my PC (Uninstallation)

It is easy to get rid of Flight Simulator Logbook Editor. You can uninstall it simply by going to **Control Panel → Programs → Uninstall a program**, then selecting Flight Simulator Logbook Editor and click on **Uninstall** button. Or you should find a shortcut for uninstaller program in Flight Simulator Logbook Editor start menu folder (if you did not prevent it's creation during installation, See: ***Picture 5: Start Menu folder selection***). The procedure is the same as with any other program.

However, not all files will be removed. Flight Simulator Logbook Editor creates couple of files which it requires for proper functioning. These files are created with first execution of the program or whenever they are missing. Most of these files are those that allow customizations to be made and therefore the decision to delete them is left over to you to do it manually. All of these files are created within Flight Simulator Logbook Editor program folder. Ok, now what are those files:

- **settings.properties** – program settings.
- Files with **.log** extension – log files. See: ***Log files*** for more details.
- Files in **res** folder with **.xml** or **.ods** extension – ODS templates and XML configuration files (date formats fur now). See: ***Customizing ods template*** and ***Customizing date formats*** for more details.
- Files in **res\kml_icons** – icons used when exporting logbook to KML file.
- Files in **res\map_icons** – icons used for displaying airports at map at Map Viewer window.
- Files in **res\fshub** – files related to functionality to import flights from FsHub.
- Files in **res\msfs** – files related to functionality to import flights from MSFS.
- Files in **res\lnm** – files related to functionality to import flights from LittleNavMap CSV.
- Files in **res\jaspereports** – files related to generating JasperReports.
- **Makerwys.exe** and **LorbySceneryExport.exe** in **res** folder. Used when importing FSX/P3D airport data.
- **mapviewer_chache** folder – used as cache folder when caching is enabled for Map Viewer (which it is by default).
- Files in **filters** folder. The xml files storing default, last used and optionally user filters to filter logbook records.
- Files in **lang** folder having **.properties** extension - language files used for program localization. See: ***Translating FS Logbook Editor*** for more details.
- **personal_minimums.xml** file in FS Logbook Editor installation folder. Personal minimums definitions are stored in this file.
- **XML files** in FS Logbook Editor installation folder which stores user aircraft categories and pilot certifications. See: ***Pilot certifications*** and ***User aircraft categories*** for more information about pilot certifications and user aircraft categories. These are the default files:
  - pilot_certifications_en.xml,
  - pilot_certifications_cs_CZ.xml
  - user_aircraft_categories_en.xml
  - user_aircraft_categories_cs_CZ.xml
- Pilot statistics templates in **ps_templates** folder. See: ***Writing custom pilot statistics template*** to get more information about pilot statistics templates. These are the default templates:
  - Flight_Times_CSV_en.txt
  - Letove_casy_CSV_cs_CZ.txt
  - Line_Style_en.txt
  - Radkovy_vypis_cs_CZ.txt
  - Table_Style_en.txt
  - Tabulkovy_vypis_cs_CZ.txt
  - Takeoffs_and_Landings_CSV_en.txt
  - Vzlety_a_pristani_CSV_cs_CZ.txt

You can delete all those files if you did not make any changes to them (or if you wish, of course). There should be no other files left in the program folder, unless someone other put them in. So if there is nothing you want to keep in the Flight Simulator Logbook Editor program folder, you can delete the whole folder.

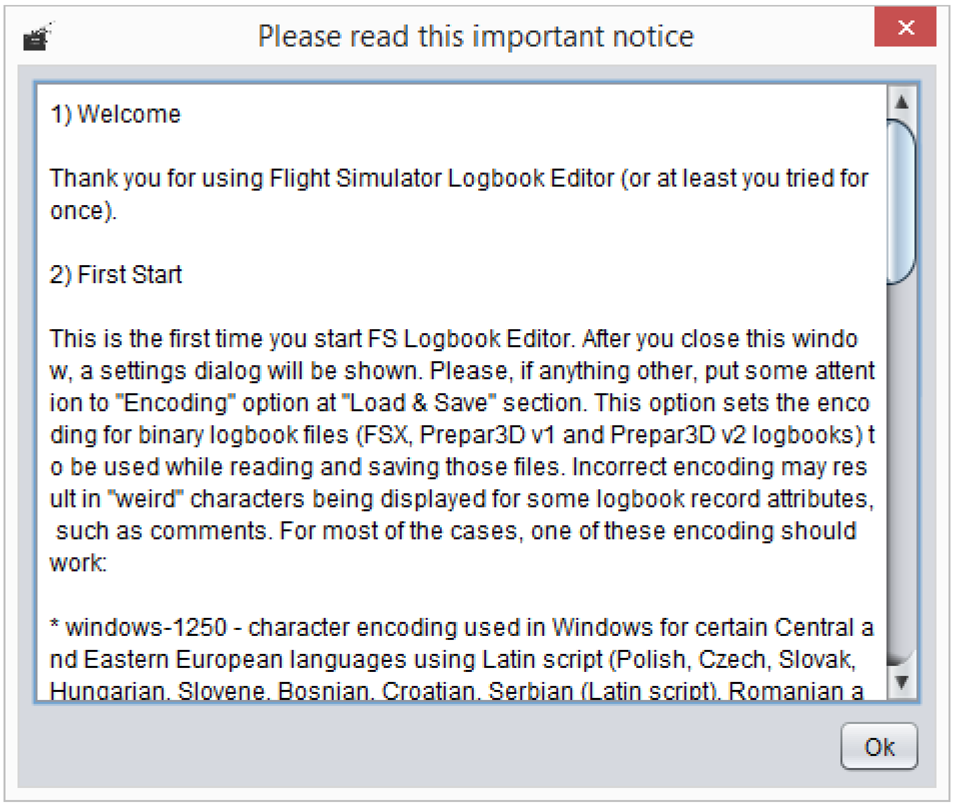
Just to note, the Flight Simulator Logbook Editor program folder is the folder you selected during installation (See ***Picture 4: Installation folder selection***). If you left the default one, it will be: *"C:\Program Files (x86)\Flight Simulator Logbook Editor"* or *"C:\Program Files\Flight Simulator Logbook Editor"*. This means you can delete the *"Flight Simulator Logbook Editor"* folder if you do not need anything it contains. **DO NOT** try to delete the whole parent "Program Files" thing. This would result in a "coffee break" with your IT specialist. And the experience is such that happy moments of that kind always happens when you want them the least.

2.6. Support & Updates

Any support and/or program updates (if any) you should find either at developer's website: http://mouseviator.com, product website: http://mouseviator.com/pc-creations/fs-logbook-editor/ or forums: http://forums.mouseviator.com and respective pages therein. Please note that there is no warranty of any kind for any updates and/or support services shall be provided to you by developer.

2.7. First start

When you start the application for the first time, you will be presented with welcome dialog containing some important information. Please read this notice before you continue on. The welcome dialog will be shown only the first time you start the application (unless you have DEMO version which shows it at every start).



1) Welcome

Thank you for using Flight Simulator Logbook Editor (or at least you tried for once).

2) First Start

This is the first time you start FS Logbook Editor. After you close this windo w, a settings dialog will be shown. Please, if anything other, put some attent ion to "Encoding" option at "Load & Save" section. This option sets the enco ding for binary logbook files (FSX, Prepar3D v1 and Prepar3D v2 logbooks) t o be used while reading and saving those files. Incorrect encoding may res ult in "weird" characters being displayed for some logbook record attributes, such as comments. For most of the cases, one of these encoding should work:

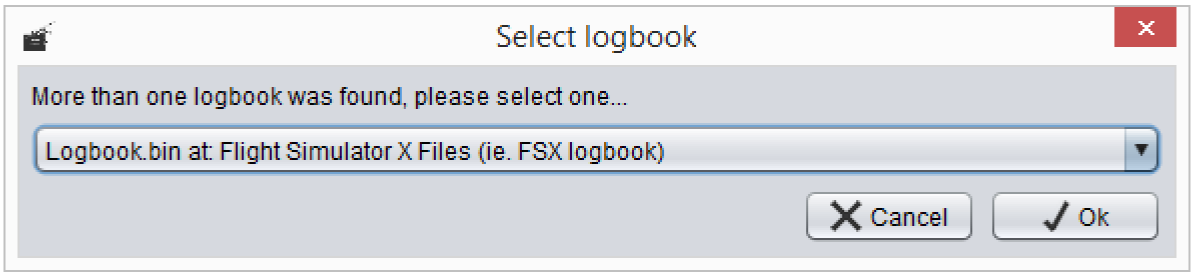
* windows-1250 - character encoding used in Windows for certain Central a nd Eastern European languages using Latin script (Polish, Czech, Slovak, Hungarian, Slovene, Bosnian, Croatian, Serbian (Latin script), Romanian a

Picture 11: FS Logbook Editor - Welcome dialog

After you click **Ok** button, a settings dialog will be opened. You can adjust some settings before you open the very first logbook. Please see ***Settings*** the for details. I recommend you to give some attention to **Encoding** option at ***Load & Save*** section. Having this option **set correctly is essential** to display all characters from loaded logbook correctly, especially in comments. However, the default **windows-1252** encoding should work just fine for most languages using Latin alphabet. When you are done with settings, close the dialog with **Ok** button.

If you left the Automatically load logbook file option enabled in the settings, FS Logbook Editor will try to automatically load logbook for one of supported flight simulators. Multiple scenarios may follow depending on your settings and the amount of logbook files found.

- If more than one logbook file is found, the selection dialog will pop up allowing you to choose one to open:


Select logbook

More than one logbook was found, please select one...

Logbook.bin at: Flight Simulator X Files (ie. FSX logbook)

Picture 12: Dialog to select one of logbook files.

- If only one logbook is found, it is loaded right away.
- If no logbook is found and the **Show Open Dialog at start up (If Logbook is not found automatically)** option at ***Load & Save*** section is enabled, an open file dialog is shown allowing you to select logbook to open manually. Otherwise, no logbook is loaded.

2.8. News in version 1.86

Version 1.86 brings some just small fixes and some additions. Writing about fixes, the input of airport codes and number of landings at ***Edit logbook record dialog*** was behaving sort of wonky. Now it should be more user friendly.

Another thing fixed is that the dialog to select logbook was displayed even if only one logbook was found. This happened in the case there was only one recent file and no other logbook files were found.

An option to to only allow Logbook.bin files in automatic logbook find selection. It was added to the ***Load & Save*** section of ***Settings*** and it is enabled by default. So, files like wxstationlist.bin will no longer be listed as logbook files.

A new feature called "**File Parameters Cache**" or "**FPC**" has been added to minimize the amount of dialog windows you have to click through while opening/saving logbook files. FS Logbook Editor will remember selected file parameters (per file) and thus will not display the dialog to specify them next time you open the same file. This behavior can be overridden by a "Shift" key without having to disable the function. Read more in the ***Load & Save*** section of ***Settings*** chapter.

Added German and French translation. But note that they are straight machine translations of language files. They were not human created/corrected like the English and Czech ones, so they might be buggy, incorrect, wonky, funny …

Added one command line paramater, which increases the number of command line parameters supported by FS Logbook Editor to one :) Read more about that in the new ***Command line parameters*** chapter.

Lastly, the ... biggest change, I guess. From now on, FS Logbook Editor becomes freeware. Or, as I named it officially (I think I saw it somewhere else already, so not really my genuine idea), an Old-School Freeware Edition. That means, no online accounts needed, no registrations, no 3 billion of clicks to download (hope we can stay at 3 maximum :)). Just download it and use it, forever or as long as it is able to work.

2.9. News in version 1.85

Version 1.85 brings some additions and some small fixes that I discovered during the implementation of those additions :)

Make sure to look at the ***User aircraft categories*** chapter again, as every category has now a checkbox for multi-engine and a selection of aircraft type. These options define whether the category contains multi-engine aircraft and what aircraft type are the aircraft within the category. Also, every aircraft within the category has those options too. The category defines those values for all aircraft within the category, but you can override it for specific aircraft within the category. These information than can be used during import from FsHub, from MSFS and/or when loading LittleNavMap CSV file. See the chapters: ***Importing flights from FSHub***, ***Importing flights from Microsoft Flight Simulator 2020*** and ***Loading logbook from LittleNavMap CSV file***.

A support for reading and writing LittleNavMap CSV file has been added. So logbook data can be imported/exported between FS Logbook Editor and LittleNavMap. See the ***Load / save logbook from / as LittleNavMap CSV file*** chapter. The implementation of this feature also led to the usage of 3rd party library for working with CSV files, leading to slight rewrite of code for loading / saving logbook from/to regular CSV file. So revisit the ***Load / save logbook from / as CSV file*** chapter too.

Another bigger change is that FS Logbook Editor is no longer limited to display just one flight on the map. Now you can select any amount of flights in the logbook table view and click on **View on map** button. All selected flights will be displayed in slightly different map viewer window, where you can select between them. The selected one gets highlighted, but you still also see all other flights. Read about that in the updated ***Viewing flight on map*** chapter.

2.10. News in version 1.80

Version 1.80 brings a lot of new functions. I was planning to release it as “Christmass” present, but, you see, it came out at March…

I was about to close the feature addition to version 1.80, when I discovered a way to at least **import logbook records from Microsoft Flight Simulator 2020**. Not sure if it ever will be possible to modify them or export logbook records from FS Logbook Editor to MSFS, but, at least we have something now, I think. The flights can be exported using system clipboard or using network communication. For both of these, an add-on package: “**Mouseviator Enhanced Logbook**” is required. Read more about the import in the ***Importing flights from Microsoft Flight Simulator 2020*** chapter, and in the ***Settings*** → ***MSFS***.

Another cool new feature is an import of flight from **FsHub**. FsHub.io is a wonderful, free, online platform for monitoring flights and flight performance. If you do not know it, I highly suggest to take a look at it. You can read about this new function at ***Importing flights from FSHub*** chapter and at ***Settings*** → ***FsHub*** chapter.

The ***Pilot Statistics*** chapter has been updated. ***User aircraft categories*** have new check-box, **Multi-engine** which defines whether the respective category contains multi-engine aircraft. Import of flights from FsHub, MSFS and loading of X-Plane logbook can benefit from that information, matching aircraft names against the user aircraft categories to find out if the aircraft used on flight is multi-engine or not (none of the mentioned sources contain this information by default). Pilot statistics can now be saved also in **PDF**, **ODT** and **DOCX** format. And there is also a new button to open selected pilot statistics template in embed text editor (See ***Little More Than Dumb Text Editor***).

There is a new menu: ***Tools*** which contains just one item for now – ***Little More Than Dumb Text Editor*** – which is smart text editor, with syntax highlighting, auto-complete and help features. It should made editing of pilot statistics templates, new comment templates for FsHub and MSFS import, and other files, easier.

File formats from which FS Logbook Editor can load logbook , or to which it can save it has been expanded in version 1.8. The ***File*** chapter has been updated at respective places describing the new file formats. What are the new formats, you ask? Well, finally, FS Logbook Editor can now also load the logbook from CSV file, apart from just saving. The save process has to be slightly updated too. Read about both in the ***Load / save logbook from / as CSV file*** chapter. Another newly supported file format are Excel spreadsheets, both, the old XLS format and also the newer XLSX format. Read more at the ***Load / save logbook from / as Excel spreadsheet*** chapter. The last new format is JSON format. If that sounds unfamiliar to you, than you probably can ignore it. It is being used for data exchange between various software (services). The ones who know what it is for, will sure know how to handle it. It is in short described in the ***Load / save logbook from / as JSON file*** chapter.

2.11. News in version 1.65

Well, I skipped some news paragraphs for sure :) But, what brings version 1.65? It brings formal compatibility with Prepar3D v5. FS Logbook Editor was able to read v5 logbook file even before the update, because it has the same format as in v4. But airport data import did not work without workarounds like copying actual version of MakeRwys and LorbySceneryExport to the simulator folder manually and running FS Logbook Editor with admin privileges. None of this is required with this update and FS Logbook Editor will take case of necessary actions for you. At least it should :) This required new **Try to run MakeRwys with elevated privileges** option at the import dialog. Read the ***Importing airport data for FS2004 / FSX / Prepar3D*** chapter again.

An option to change Look and Feel of FS Logbook Editor was added to the ***General*** section of the ***Settings***. This allows you to "dress" FS Logbook Editor like classic Windows application, it is something like using themes…

There is a new section – **Font Options** in the ***General*** section of the ***Settings***. It allows you to change some program fonts. Hopefully, this will help to adjust previously constant-sized fonts for better readability at high resolution displays. Fonts can also be changed for odd/even/special rows for all the tables. Read again the ***General*** chapter of the ***Settings*** to find out how to do it. In short, it is being done via the components to define colors.

One last major change is, that this version was compiled with Java 11. Unfortunately, that effectively means the end of support for Windows XP, as the last version supported by this system was 8.

There were also some changes "under the hood", which you may or may not notice. At the ***Settings*** dialog, in different sections, you can start various tasks which may take some time. Like importing airport data or automatic search for simulators. It is now possible to close the dialog while those tasks are running (actually, one at a time can run). The respective sections of settings will stay locked until the task is not done. This is true for all tasks that somehow manipulate the airport database. (Any you run from ***Database***, ***Airport Data*** and ***Simulators*** section)

2.12. News in 1.50 version

The big...ok maybe not so big, but visible change is an addition of Map viewer. You can now view the flights (if you have airport data loaded so the airport coordinates are known) right within FS Logbook Editor. Just select a flight and click the **View on map** button. Read about that in the ***Viewing flight on map*** chapter.

Next little addition is that you can show/hide columns from the main logbook table. You can choose which columns to show/hide at the ***General*** tab of the ***Settings***.

When you export logbook to ODS or CSV files, you can choose which record attributes (columns) to export. FS Logbook Editor now remembers your selection for both file formats and will offer the same selection next time you do the export, so you don't need to re-select it every time.

FS Logbook Editor version 1.4 brought the feature of displaying airport data and flight distances. For this a database is needed to store the airport and simulators data. Only SQLite was formally supported even thought the interface was prepared to support other database systems too. Well the code wasn't completely prepared. FS Logbook Editor 1.50 has this support completed. Now you can use also **MySQL**, **MariaDB** and **Microsoft SQL Server** database engines to store flight simulators and airport data. FS Logbook Editor has been tested with all these database engines (MySQL v5.7, MariaDB v10.2 and Microsoft SQL Server Express 2014 (version 2008 and above is required)). See the updated ***Database*** chapter.

The encoding option has been added to the dialog that shows before importing X-Plane and Navigraph airport data. This is for the case the source files are not encoded in UTF-8, which FS Logbook Editor expects them to be by default. The images and description has been updated in the ***Importing airport data from X-Plane*** and the ***Importing airport data from Navigraph data cycle*** chapter.

The new feature to filter logbook records has been added. It is described in the ***Filtering logbook records*** chapter. The ***Editor*** chapter has been updated with description of the new buttons added at Editor tab. The ***Load & Save*** chapter has been updated with description of filters auto save and load option.

2.13. News in 1.42 version

FS Logbook Editor v 1.42 brings support for Prepar3D v4 logbook files, in which the text information are saved in 16bit Unicode encoding. It might have look that the Unicode support was there already, but it was...only half. Kind of did not take the 2 bytes per character thing in account. This change of encoding resulted in redesign of the dialog that was displayed during saving FSX/P3D binary logbook files. This dialog is now displayed also while opening a binary logbook and requires you to select version of flight simulator the file is targeted for (or from) and also the encoding. The encoding selection was moved here from the Load and Save section of Settings, so you do not have to go there every time you load logbook from different P3D version. The encoding option was removed from the General section of Load and Save panel and was separated into two options. Now FS2004 logbook files and X-Plane logbook files have its own encoding settings.

So from now, please pay attention on what simulator version and encoding you select while

opening or saving FSX/P3D binary logbook, so that you are not surprised, why the logbook does not work in the simulator. The dialog will remember the selected encoding for each simulator version, for open and save action separately, and will pre-select the last used one every time you change the simulator version in the dialog.

See the updated ***Load & Save*** chapter and the ***File*** chapter.

2.14. News in 1.4 version

The first new feature is that FS Logbook Editor now can compute flight distance for certain flights (logbook record). But to be able to do this, it needs some information about the starting, destination and en-route airports. Such as airport longitude and latitude. Where to get this data? Well, FS Logbook Editor is an editor of flight simulators logbook files, so the simulators itself comes in mind. And so they are. FS Logbook Editor reads airport data from supported simulators. Because FS Logbook Editor is not oriented on one simulator, it allows you to import airport data for as many simulators as you define and then switch between them. **So the flight simulators becomes Airport data sources or providers**. There is a new list box named **Airport Data Provider** on the ***Editor*** tab, which allows you to select simulator (Airport Data Provider) whose data will be used to compute flight distances for currently opened logbook.

While we import longitudes and latitudes of airport, why not to import other data, such as airport name, city, country, state? Of course FS Logbook Editor does and those also change when you select another Airport Data Provider.

There are 3 new sections in the ***Settings*** related to this new functionality. The ***Database*** section, which you may ignore, it is for advanced users. The ***Simulators*** section, where you can define your flight simulators (which will be used as Airport Data Sources) and ***Airport Data*** section, where you can import airport data for defined simulators.

When we know airport longitude and latitude, we can **export logbook to KML** file to see flights in Google Earth. The KML file export is another new function. See the updated ***Load & Save*** chapter to see how you can customize KML export.

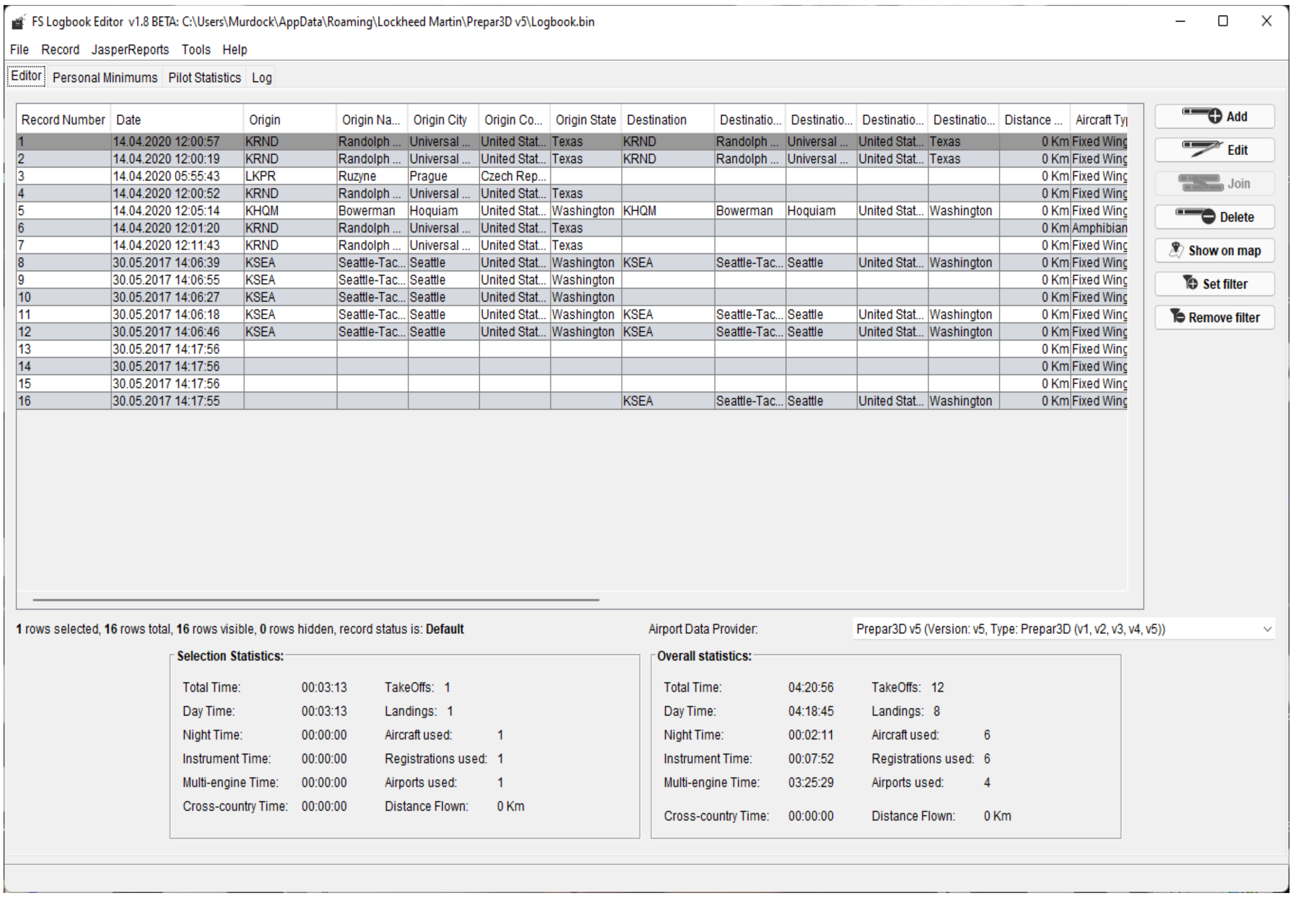
Pilot statistics have also been upgraded with the new information available. Now you can enhance templates by these new statistics: Distance flown per aircraft / User aircraft category, Takeoffs and landings per city, state and/or country. The new statistics template tags for these new statistics types has of course been added to the ***Writing custom pilot statistics template*** chapter.

A **Save Selected As**… menu option has been added to File menu, and it does what it says (allows you to export only selected records from logbook to any of supported file formats). Don’t know why it wasn’t there earlier.

3. The Program

This section contains description of program user interface and its usage. The main program window contains three tabs, that are discussed in the following chapters.

3.1. Editor



Record Number	Date	Origin	Origin Na...	Origin City	Origin Co...	Origin State	Destination	Destinatio...	Destinatio...	Destinatio...	Destinatio...	Distance ...	Aircraft Ty
1	14.04.2020 12:00:57	KRND	Randolph ...	Universal ...	United Stat...	Texas	KRND	Randolph ...	Universal ...	United Stat...	Texas	0 Km	Fixed Wing
2	14.04.2020 12:00:19	KRND	Randolph ...	Universal ...	United Stat...	Texas	KRND	Randolph ...	Universal ...	United Stat...	Texas	0 Km	Fixed Wing
3	14.04.2020 05:55:43	LKPR	Ruzyne	Prague	Czech Rep...							0 Km	Fixed Wing
4	14.04.2020 12:00:52	KRND	Randolph ...	Universal ...	United Stat...	Texas						0 Km	Fixed Wing
5	14.04.2020 12:05:14	KHQM	Bowerman	Hoquiam	United Stat...	Washington	KHQM	Bowerman	Hoquiam	United Stat...	Washington	0 Km	Fixed Wing
6	14.04.2020 12:01:20	KRND	Randolph ...	Universal ...	United Stat...	Texas						0 Km	Amphibian
7	14.04.2020 12:11:43	KRND	Randolph ...	Universal ...	United Stat...	Texas						0 Km	Fixed Wing
8	30.05.2017 14:06:39	KSEA	Seattle-Tac...	Seattle	United Stat...	Washington	KSEA	Seattle-Tac...	Seattle	United Stat...	Washington	0 Km	Fixed Wing
9	30.05.2017 14:06:55	KSEA	Seattle-Tac...	Seattle	United Stat...	Washington						0 Km	Fixed Wing
10	30.05.2017 14:06:27	KSEA	Seattle-Tac...	Seattle	United Stat...	Washington						0 Km	Fixed Wing
11	30.05.2017 14:06:18	KSEA	Seattle-Tac...	Seattle	United Stat...	Washington	KSEA	Seattle-Tac...	Seattle	United Stat...	Washington	0 Km	Fixed Wing
12	30.05.2017 14:06:46	KSEA	Seattle-Tac...	Seattle	United Stat...	Washington	KSEA	Seattle-Tac...	Seattle	United Stat...	Washington	0 Km	Fixed Wing
13	30.05.2017 14:17:56											0 Km	Fixed Wing
14	30.05.2017 14:17:56											0 Km	Fixed Wing
15	30.05.2017 14:17:56											0 Km	Fixed Wing
16	30.05.2017 14:17:55							KSEA	Seattle-Tac...	Seattle	United Stat...	Washington	0 Km	Fixed Wing

1 rows selected, **16** rows total, **16** rows visible, **0** rows hidden, record status is: **Default**

Airport Data Provider: Prepar3D v5 (Version: v5, Type: Prepar3D (v1, v2, v3, v4, v5))

Selection Statistics:

Total Time:	00:03:13	TakeOffs:	1
Day Time:	00:03:13	Landings:	1
Night Time:	00:00:00	Aircraft used:	1
Instrument Time:	00:00:00	Registrations used:	1
Multi-engine Time:	00:00:00	Airports used:	1
Cross-country Time:	00:00:00	Distance Flown:	0 Km

Overall statistics:

Total Time:	04:20:56	TakeOffs:	12
Day Time:	04:18:45	Landings:	8
Night Time:	00:02:11	Aircraft used:	6
Instrument Time:	00:07:52	Registrations used:	6
Multi-engine Time:	03:25:29	Airports used:	4
Cross-country Time:	00:00:00	Distance Flown:	0 Km

Picture 13: Main program tab to work with logbook records

The Editor tab is the first you will see after the program starts and you will spend most of the time here. It is quite simple. The majority of display contains table with logbook records, either manually created or loaded from logbook file. Right to this table are four buttons and below this table you will find simple selection and overall statistics.

Add button will add new record at the end of the table and will open Edit logbook record dialog to edit record parameters. See the ***Edit logbook record dialog*** in the following section.

Edit button will open currently selected record in the Edit Logbook record dialog so you can change record attributes. See the ***Edit logbook record dialog*** in the following section. This button is disabled when more than one record is selected.

Join button will join two or more records into one record. This can be useful when you do some test flights (like trying out new scenery, airplane, tweaking etc.). Those would probably be short flights with some landing. If you do no want to delete those records, but you don´t like the “messy” feel of them, you can join them into one. So there will be only one record preserving your landings and flight time. All flight times of records that are being joined are summed, one airport is chosen as departure, one as destination and the others are added to airport list. Some information may be lost by this action (like aircraft description and model if you join record with different aircraft model and/or description), but the resultant flight times and number of landings will be the same as for joined records. This action is only available when two or more records are selected.

Delete button will delete currently selected record(s).

All the actions provided by above buttons are also available in the ***Record*** menu.

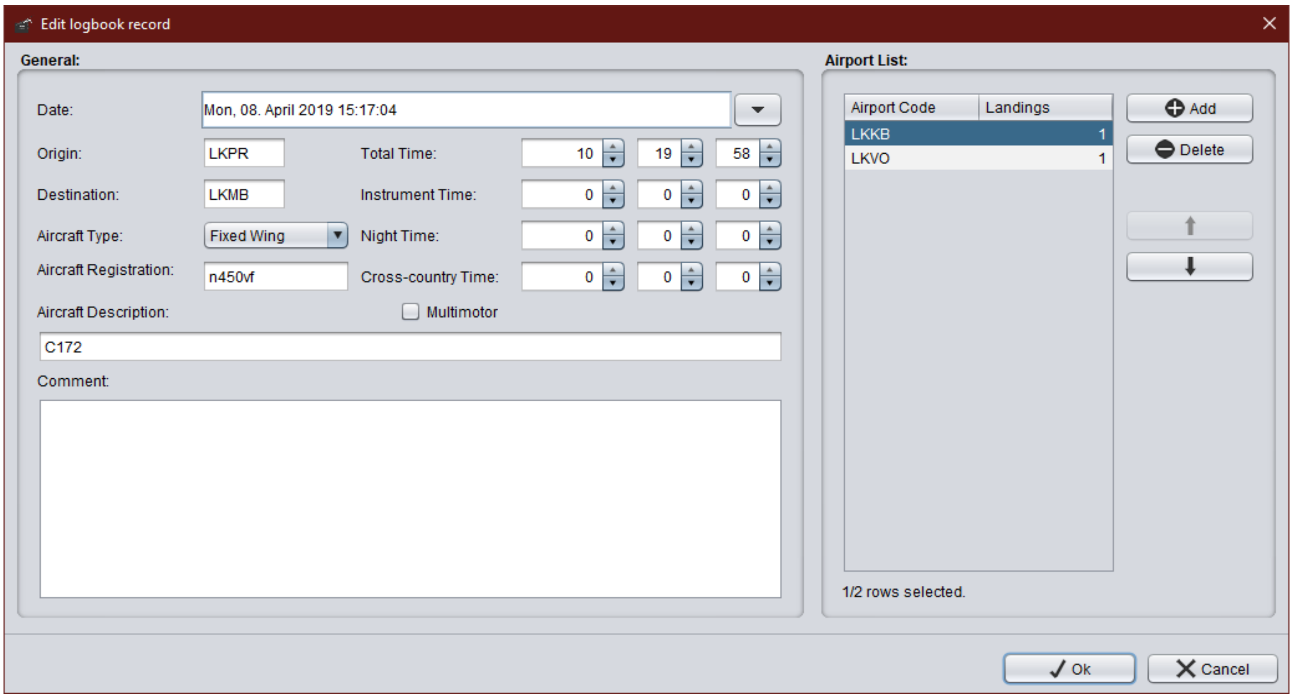
Show on map button will display currently selected flight on map. This button is disabled if no airport data are loaded or if the flight does not contain any data that would put at least one point on the map. See ***Viewing flight on map*** and ***Airport Data*** for more details.

The **Set filter** and **Remove filter** buttons allows to filter logbook records. This is described in the ***Filtering logbook records*** chapter.

Airport Data Provider list allows you to select which data provider (data source) is used to display airport information, such as airport name, city, country and state. This also affects the computation of flight distances, as not all data providers will have the same data. Airport data provider is the source of the mentioned data, it is Flight Simulator X, X-Plane or other supported simulator. See the

Selection statistics and **Overall statistics** shows basic statistics for selected record(s) and whole logbook respectively. These are just basic summaries. The overall ones you can see in the simulator too. Please note that number of takeoffs is only informative as this cannot be calculated precisely from the data contained in the logbook.

3.2. Edit logbook record dialog



Date:	Mon, 08. April 2019 15:17:04				
Origin:	LKPR	Total Time:	10	19	58
Destination:	LKMB	Instrument Time:	0	0	0
Aircraft Type:	Fixed Wing	Night Time:	0	0	0
Aircraft Registration:	n450vf	Cross-country Time:	0	0	0

Aircraft Description: ☐ Multimotor

C172

Comment:

Airport Code	Landings
LKKB	1
LKVO	1

1/2 rows selected.

Picture 14: The dialog to edit logbook record attributes

The Edit Logbook dialog is displayed after adding new logbook record or when editing existing logbook record. Here you can edit all attributes that are supported by flight simulator logbook[3].Those are:

- **Date** – date and time of flight.
- **Origin** – ICAO code of origin airport.
- **Destination** – ICAO code of destination airport (or the airport last landed at during flight).
- **Aircraft Type** – one of four aircraft types – Glider, Fixed Wing, Amphibian or Rotor (Helicopter).
- **Aircraft Registration** – aircraft registration number (for example: N-450VF).
- **Aircraft Description** – something more about the aircraft, like detailed name (for example: Bonanza A36)
- **Comment** – flight comment.
- **Total Time** – total flight time. First field for hours, next for minutes, last for seconds.
- **Instrument Time** – time of flight that was flown in IMC conditions. First field for hours, next for minutes, last for seconds.
- **Night Time** – time of flight, that was flown during night time. First field for hours, next for minutes, last for seconds.
- **Multimotor** – whether the aircraft flown on this flight had more than one engine or not.
- **Airport List** – See ***Airport List*** below.

On the bottom of the window there are two buttons – **Ok** and **Cancel**. You may guess what are they for. Clicking the **Cancel** button you discard everything you just have input in this dialog and close it. By clicking **Ok**, your changes will be saved to actually edited record (existing or new one) and the dialog will close. Please note that in some cases you may not be able to close the dialog by **Ok** button and save changes. This is when there are some errors in the data input, like if you set, for example, night time greater than total time etc. But don't worry, the application will inform you of such errors and will ask you to correct them. When there are no errors in the data input, the dialog will close with no more complaints after clicking **Ok**.

3.2.1. Airport List

The list contains all airports that you have landed at during flight. Each row in this list (table) contains two values. ICAO code of airport that you landed at and number of landings at this airport. The airport ICAO code must be 1 to 4 characters long and may contain only numbers and/or alphabet characters. New records will have "UNMD" airport ICAO code. It is just default constant used by FS Logbook Editor, not anything official. The number of landings have to be between 1 and 255.

To **add** record, click the **Add** button and than write the ICAO code of airport into the corresponding Airport Code cell and number of landings into the corresponding Landings cell. To accept the value, click another cell or press Enter on your keyboard.

To **edit** record, simply click the cell containing value you want to change and write new value. To accept the value, click another cell or press Enter on your keyboard.

To **delete** one or more records, select the record(s) and click **Delete** button.

Since FS Logbook Editor version 1.60 there are two new buttons with the up (↑) and down (↓) arrows on them. These two buttons move an item(s) in the airport list up and down. To move one or more item(s) in the airport list, select the respective records and press the button with the ↑ arrow. The selected items will move up in the list by one position. This means, those airports were visited

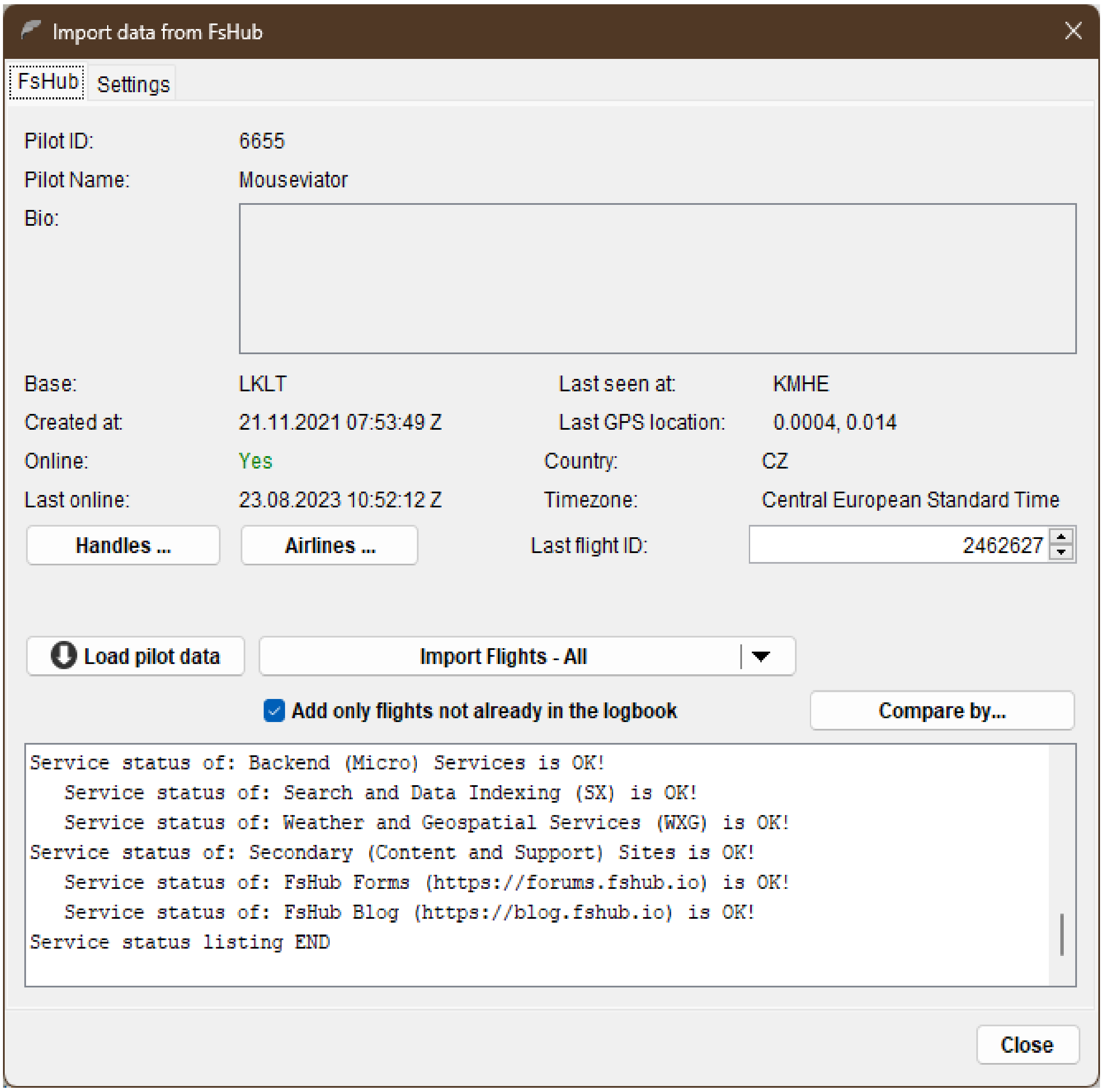
3 FSX (primarily designed for), Prepar3D and Prepar3D v2-v5 (as long as the format is the same as for FSX)

sooner in the flight. If you want to move some airport so that it was visited later in the flight, just select the respective item and press the ↓button. The respective airport list item(s) will be moved down in the list (towards the bottom). Please note that if the movement is not possible, the respective button will turn gray. For example, if you select the last item in the airport list, the ↓button will become unavailable, since it is not possible to move the last item further down.

3.3. Importing flights from FSHub

FSHub is wonderful, free, online flight tracking and performance monitoring platform. It has also great API – simply saying, it was not that hard to implement the import of flights from FSHub into FS Logbook Editor – so you can, for example, backup your flights, generate ***Pilot Statistics*** or ***JasperReports*** and some other stuff that FS Logbook Editor allows you to do…

The import is available via the **File** → **Import From …** → **FsHub** menu. It will bring up the following dialog:



Pilot ID:	6655		
Pilot Name:	Mouseviator		
Bio:			
Base:	LKLT	Last seen at:	KMHE
Created at:	21.11.2021 07:53:49 Z	Last GPS location:	0.0004, 0.014
Online:	Yes	Country:	CZ
Last online:	23.08.2023 10:52:12 Z	Timezone:	Central European Standard Time
		Last flight ID:	2462627

☑ Add only flights not already in the logbook

```
Service status of: Backend (Micro) Services is OK!
   Service status of: Search and Data Indexing (SX) is OK!
   Service status of: Weather and Geospatial Services (WXG) is OK!
Service status of: Secondary (Content and Support) Sites is OK!
   Service status of: FsHub Forms (https://forums.fshub.io) is OK!
   Service status of: FsHub Blog (https://blog.fshub.io) is OK!
Service status listing END
```

Picture 15: FSHub Import dialog

When you first open the dialog, the buttons will be disabled, unless you have already configured the settings. You can configure the settings at the **Settings** tab in the dialog, or in the ***Settings*** dialog.

If FSHub settings are correctly configured, the FSHub import dialog might look similar to the picture shown, where it was checking the status of FsHub services and updating data about pilot upon dialog opening.

The upper part of the dialog shows data about the pilot loaded from FsHub, such as Pilot ID, name, bio, base (the base airport), current location, etc… The lower part contains a „log“, where you can find out about the result and progress of various operations.

Clicking the **Load pilot data** button, these information gest updates – so, nothing will happen if they did not change. But the log below will show that FS Logbook Editor indeed made the request and whether it succeded or not.

To the right of the Load pilot data button is split button, that actually hides 4 buttons to import flights from FsHub. These buttons are:

- **Import Flights – All** – Will import all your flights from FsHub and will append them to currently opened logbook. If no logbook is opened, will create new one.
- **Import Flights – All into new logbook** – Will import all your flight from FsHub into new logbook. **Note that any currently opened logbook will be just closed without saving!**
- **Import Flights – from last flight** – Will import all your flights since last import. Each time you import flights, FS Logbook Editor will remember the ID if the last flight imported. You can see that in the **Last Flight ID**. So next time, you do not need to do the whole import again. This to save the server from un-necessary requests. You have the option to override the ID (going before the last flight) by changing it in the spinner, but it will apply only to the next import (as it will be automatically changed by that import). The flights will be appended to currently opened logbook or new one will be created if none is opened.
- **Import Flights – from last flight into new logbook** – works like the previous button, except that existing logbook, if any opened, will be closed. **Note that any currently opened logbook will be just closed without saving!**

The import of flights from FSHub is pretty straight forward, nothing hard, I think. The **Settings** tab contains exactly the same as the FsHub section in the ***Settings*** (read about it in that chapter).

The **Add only flights not already in the logbook** option will make sure (if checked) that you have no doubled flights in the logbook, in case you added them previously. Using the **Compare by…** button you can modify which attributes will be used for comparison and therefore for the decision whether the flight being added is duplicate or not.

FS Logbook Editor version 1.85 adds two new buttons – **Handles…** and **Airlines…** .These will display the data about pilot handles – which are basically links pilot profiles on IVAO, Vatsim and other that the pilot entered in the profile. **Airlines...** button will open the list of airlines pilot is member of. Both are pretty simple dialogs to just read the information.

Oh, and yes, this is a one-way functionality. FS Logbook Editor does not support uploading the flights back to FsHub.

3.4. Importing flights from Microsoft Flight Simulator 2020

The crucial condition for the import of flights from Microsoft Flight Simulator to work is to have the "**Mouseviator Enhanced Logbook**" add-on package installed within the simulator. As the flights are stored within your Xbox account, there is still no way to access them directly by 3rd party app, like FS Logbook Editor, but a way was found to at least import them from the "logbook" view within the simulator. The flights can be imported in two ways:

- Using clipboard
- Using network

At the Settings tab, various settings related to the import can be changed. Refer to ***Settings*** → ***MSFS***, as the MSFS section of the settings is the same you will find at **Settings** tab.

In FS Logbook Editor version 1.81, a new checkbox and button was added to the MSFS import dialog that affect both import methods.

The **Add only flights not already in the logbook** option will make sure (if checked) that you have no doubled flights in the logbook, in case you added them previously. Using the **Compare by…** button you can modify which attributes will be used for comparison and therefore for the decision whether the flight being added is duplicate or not.

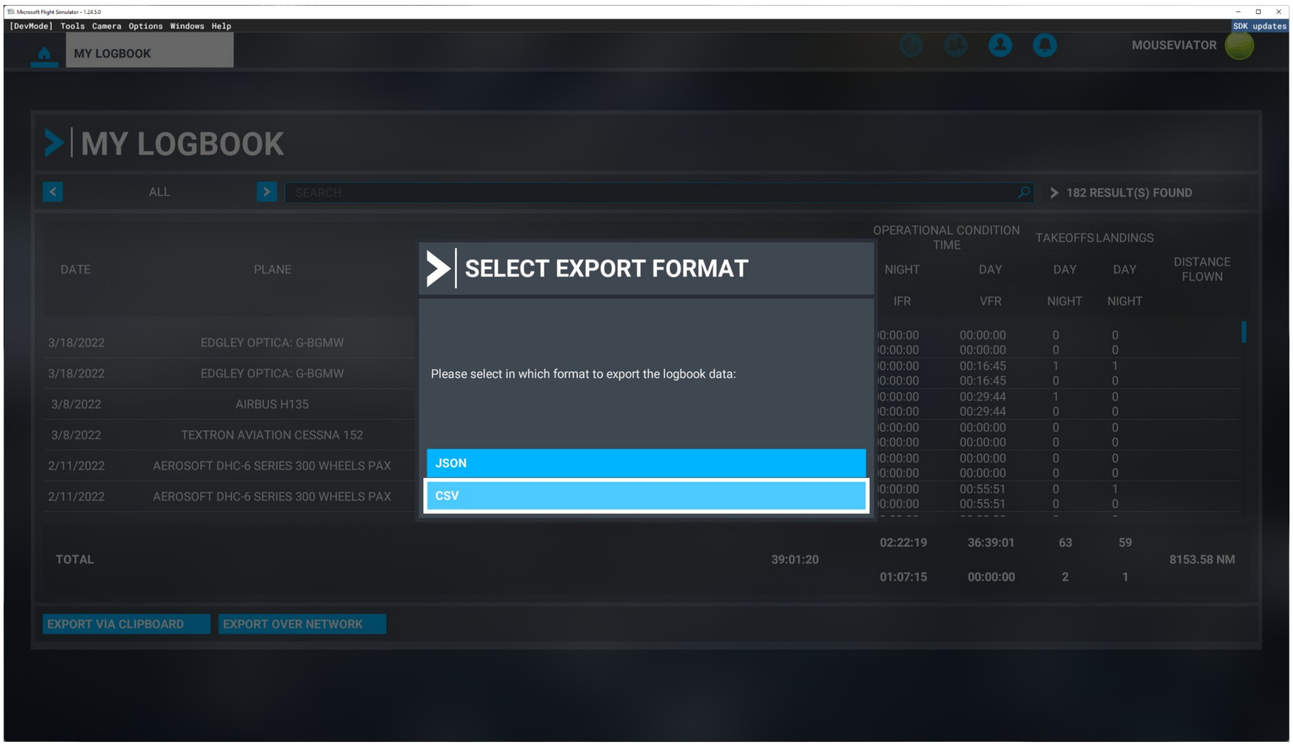
3.4.1. Import using clipboard

Using this method, the data are copied to clipboard using the **Export via Clipboard** button within Microsoft Flight Simulator logbook page. Once you click the button, a popup will show where you need to select export format. Options are – **JSON** and **CSV**. Both are good for computer processing, but CSV is more like „table" format and will use less memory in clipboard. Also, if you try to paste CSV into programs like Excel or LibreOffice Calc, they should recognize it, let you specify CSV import options, which, if set correctly, than should lead to the logbook data being inserted in those programs in tabular format (everything in its own respective column).

FS Logbook Editor of course can read the data from clipboard in both mentioned formats. Just click the **Import from clipboard** or **Import from clipboatd – in new logbook** button. If FS Logbook Editor can access the clipboard and if it will find valid data there, the flights will start to fill the logbook table at the ***Editor***. If you use the **Import from clipboatd – in new logbook** button, **any currently opened logbook will be just closed without saving!** If you enabled the **Monitor clipboard** option at ***Settings*** → ***MSFS***, FS Logbook Editor should detect the flights in clipboard automatically once you click the **Export via Clipboard** in MSFS and selecting either **JSON** or **CSV** export option. In this automatic mode, the insertion works like the **Import from clipboard** button – all flights will be added to current logbook.

Note that this approach was not tested on some really big amount of flights. As all the flights are copied in one step, there is chance for some kind of failure in the case of big amount of data. This method should be considered backup and you should use the network method wherever possible.

<content follows on next page>



MY LOGBOOK

182 RESULT(S) FOUND

SELECT EXPORT FORMAT

Please select in which format to export the logbook data:

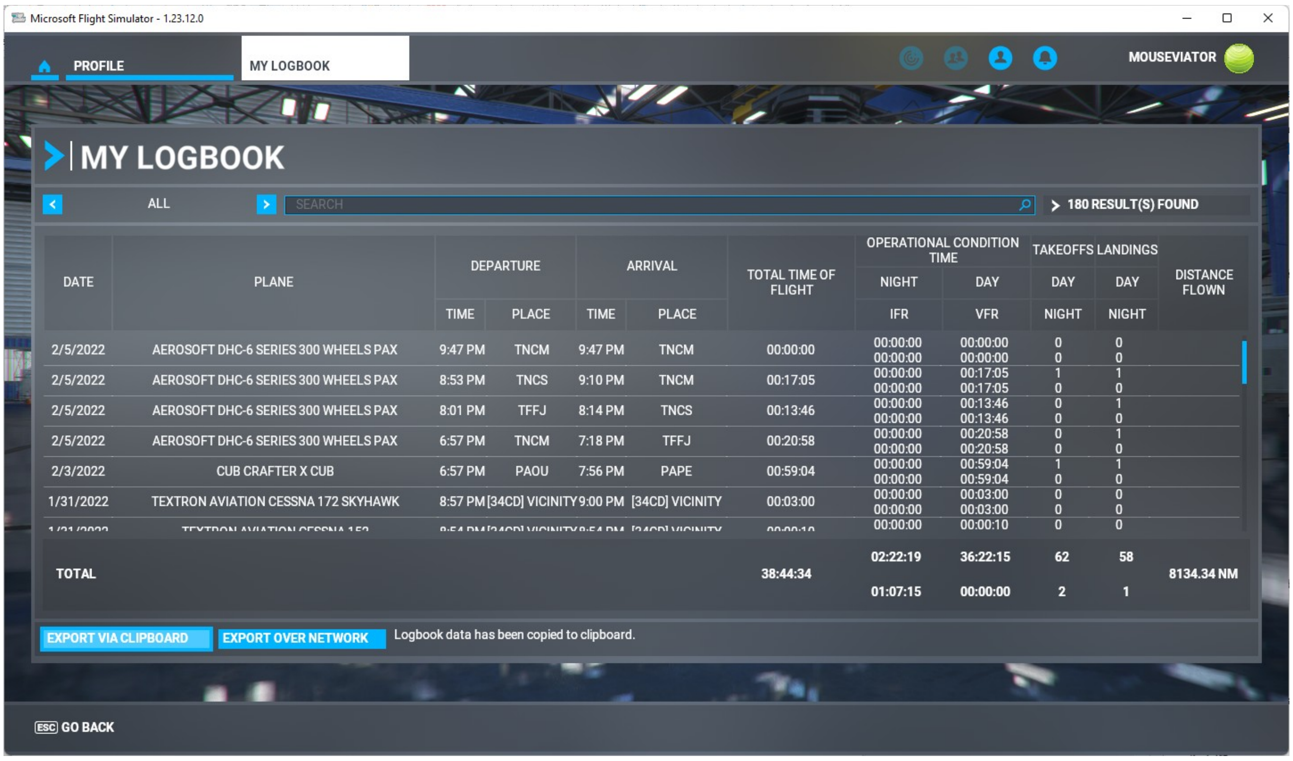
JSON

CSV

DATE	PLANE	OPERATIONAL CONDITION TIME NIGHT / IFR	DAY / VFR	TAKEOFFS DAY / NIGHT	LANDINGS DAY / NIGHT	DISTANCE FLOWN
3/18/2022	EDGLEY OPTICA: G-BGMW	0:00:00 0:00:00	00:00:00 00:00:00	0 0	0 0	
3/18/2022	EDGLEY OPTICA: G-BGMW	0:00:00 0:00:00	00:16:45 00:16:45	1 0	1 0	
3/8/2022	AIRBUS H135	0:00:00 0:00:00	00:29:44 00:29:44	1 0	0 0	
3/8/2022	TEXTRON AVIATION CESSNA 152	0:00:00 0:00:00	00:00:00 00:00:00	0 0	0 0	
2/11/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	0:00:00 0:00:00	00:00:00 00:00:00	0 0	0 0	
2/11/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	0:00:00 0:00:00	00:55:51 00:55:51	0 0	1 0	
TOTAL 39:01:20		02:22:19 01:07:15	36:39:01 00:00:00	63 2	59 1	8153.58 NM

Picture 16: Export format selection after clicking Export via clipboard button

The picture above shows the JSON and CSV export options. Once you click one of them, logbook data should be copied to clipboard in the selected format.



MY LOGBOOK

180 RESULT(S) FOUND

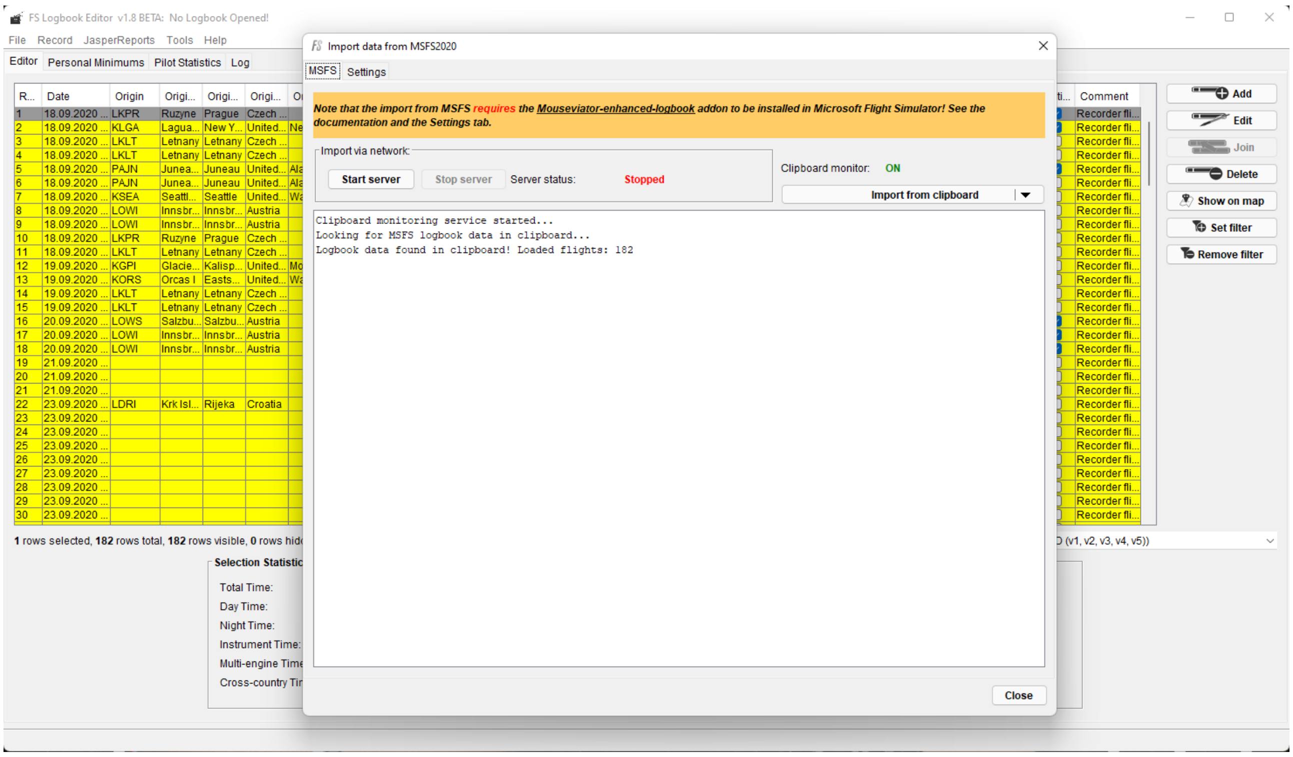
DATE	PLANE	DEPARTURE		ARRIVAL		TOTAL TIME OF FLIGHT	OPERATIONAL CONDITION TIME		TAKEOFFS	LANDINGS	DISTANCE FLOWN
		TIME	PLACE	TIME	PLACE		NIGHT / IFR	DAY / VFR	DAY / NIGHT	DAY / NIGHT	
2/5/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	9:47 PM	TNCM	9:47 PM	TNCM	00:00:00	00:00:00 00:00:00	00:00:00 00:00:00	0 0	0 0	
2/5/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	8:53 PM	TNCS	9:10 PM	TNCM	00:17:05	00:00:00 00:00:00	00:17:05 00:17:05	1 0	1 0	
2/5/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	8:01 PM	TFFJ	8:14 PM	TNCS	00:13:46	00:00:00 00:00:00	00:13:46 00:13:46	0 0	1 0	
2/5/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	6:57 PM	TNCM	7:18 PM	TFFJ	00:20:58	00:00:00 00:00:00	00:20:58 00:20:58	0 0	1 0	
2/3/2022	CUB CRAFTER X CUB	6:57 PM	PAOU	7:56 PM	PAPE	00:59:04	00:00:00 00:00:00	00:59:04 00:59:04	1 0	1 0	
1/31/2022	TEXTRON AVIATION CESSNA 172 SKYHAWK	8:57 PM	[34CD] VICINITY	9:00 PM	[34CD] VICINITY	00:03:00	00:00:00 00:00:00	00:03:00 00:03:00	0 0	0 0	
							00:00:00	00:00:10	0	0	
TOTAL						38:44:34	02:22:19 01:07:15	36:22:15 00:00:00	62 2	58 1	8134.34 NM

Logbook data has been copied to clipboard.

Picture 17: MSFS Logbook page with Mouseviator Enhanced Logbook add-on package, after clicking the Export via clipboard and selecting JSON or CSV option

The above picture shows My Logbook **page within Microsoft Flight Simulator with Mouseviator Enhanced Logbook** add-on package installed. The picture was taken right after clicking the **Export via clipboard** button. You can notice, that you will be given an information whether the operation succeeded or failed by a text message to the right of the buttons – this time, it was Ok, the message says: *Logbook data has been copied to clipboard.*

You can switch to FS Logbook Editor now, to opened MSFS Import dialog and click the **Import from clipboard** or **Import from clipboatd – in new logbook** button. The result should be similar like on the following picture. The log text area at the import dialog will tell you if and how many flights was found in the clipboard and the logbook table on the *Editor* tab on main FS Logbook Editor window should fill with these flights. They will be colored yellow – as imported records.



Note that the import from MSFS requires the Mouseviator-enhanced-logbook addon to be installed in Microsoft Flight Simulator! See the documentation and the Settings tab.

Server status: Stopped

Clipboard monitor: ON

```
Clipboard monitoring service started...
Looking for MSFS logbook data in clipboard...
Logbook data found in clipboard! Loaded flights: 182
```

R...	Date	Origin	Origi...	Origi...	Origi...	O	Comment
1	18.09.2020 ...	LKPR	Ruzyne	Prague	Czech ...		Recorder fli...
2	18.09.2020 ...	KLGA	Lagua...	New Y...	United...	Ne	Recorder fli...
3	18.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		Recorder fli...
4	18.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		Recorder fli...
5	18.09.2020 ...	PAJN	Junea...	Juneau	United...	Ala	Recorder fli...
6	18.09.2020 ...	PAJN	Junea...	Juneau	United...	Ala	Recorder fli...
7	18.09.2020 ...	KSEA	Seattl...	Seattle	United...	Wa	Recorder fli...
8	18.09.2020 ...	LOWI	Innsbr...	Innsbr...	Austria		Recorder fli...
9	18.09.2020 ...	LOWI	Innsbr...	Innsbr...	Austria		Recorder fli...
10	18.09.2020 ...	LKPR	Ruzyne	Prague	Czech ...		Recorder fli...
11	18.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		Recorder fli...
12	19.09.2020 ...	KGPI	Glacie...	Kalisp...	United...	Mo	Recorder fli...
13	19.09.2020 ...	KORS	Orcas I	Easts...	United...	Wa	Recorder fli...
14	19.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		Recorder fli...
15	19.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		Recorder fli...
16	20.09.2020 ...	LOWS	Salzbu...	Salzbu...	Austria		Recorder fli...
17	20.09.2020 ...	LOWI	Innsbr...	Innsbr...	Austria		Recorder fli...
18	20.09.2020 ...	LOWI	Innsbr...	Innsbr...	Austria		Recorder fli...
19	21.09.2020 ...						Recorder fli...
20	21.09.2020 ...						Recorder fli...
21	21.09.2020 ...						Recorder fli...
22	23.09.2020 ...	LDRI	Krk Isl...	Rijeka	Croatia		Recorder fli...
23	23.09.2020 ...						Recorder fli...
24	23.09.2020 ...						Recorder fli...
25	23.09.2020 ...						Recorder fli...
26	23.09.2020 ...						Recorder fli...
27	23.09.2020 ...						Recorder fli...
28	23.09.2020 ...						Recorder fli...
29	23.09.2020 ...						Recorder fli...
30	23.09.2020 ...						Recorder fli...

1 rows selected, **182** rows total, **182** rows visible, **0** rows hid

D (v1, v2, v3, v4, v5))

Selection Statistic

Total Time:
Day Time:
Night Time:
Instrument Time:
Multi-engine Time
Cross-country Ti

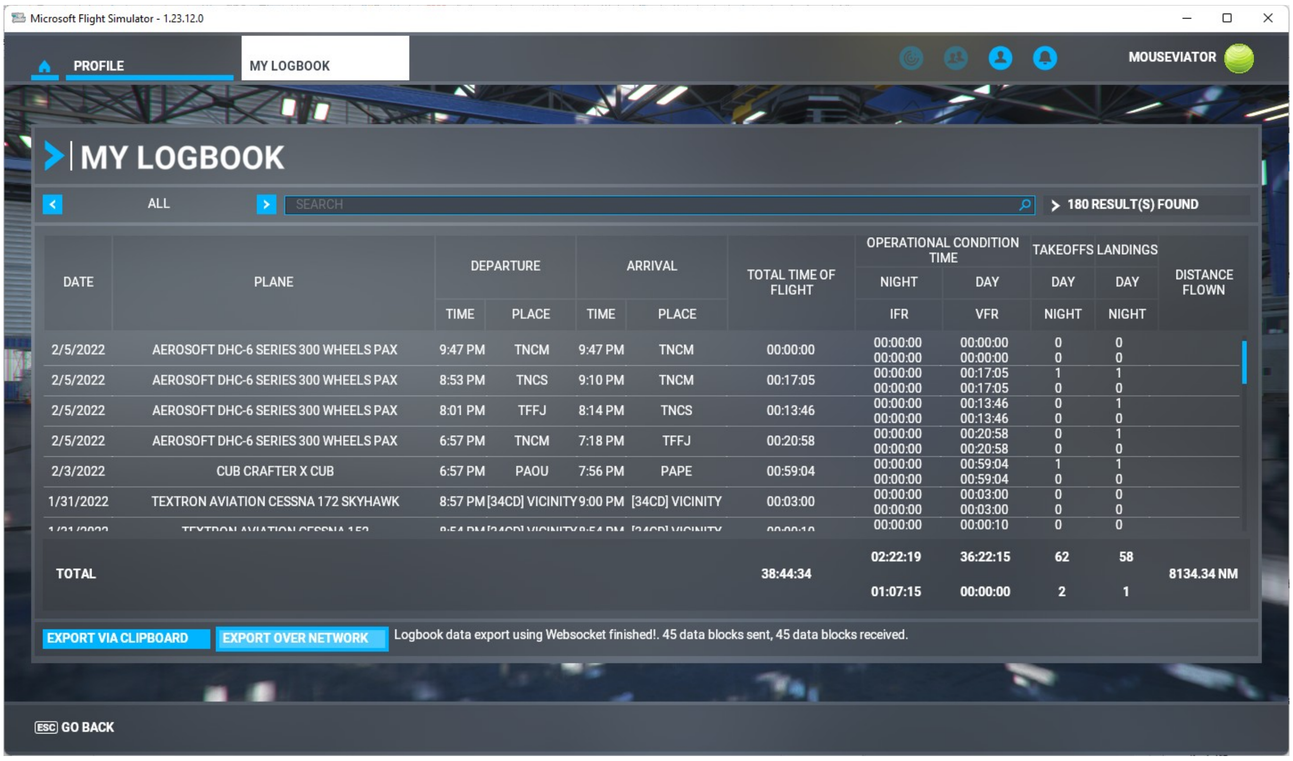
Picture 18: MSFS logbook data imported using clipboard

3.4.2. Import using network

The import of flights from Microsoft Flight Simulator using network method is two step process, similary like in the case of clipboard import. You need to start the FS Logbook Editor embed HTTP server first, using the **Start server** button. Or it may be started automatically upon opening the MSFS Import dialog if enabled to do so (See *MSFS* settings section).

The second step is to press the **Export over network** button at My Logook page within Microsoft Flight Simulator. You need to have the **Mouseviator Enhanced Logbook** add-on package installed in order for that button to be there. It should look like on the following picture:

<content follows on next page>



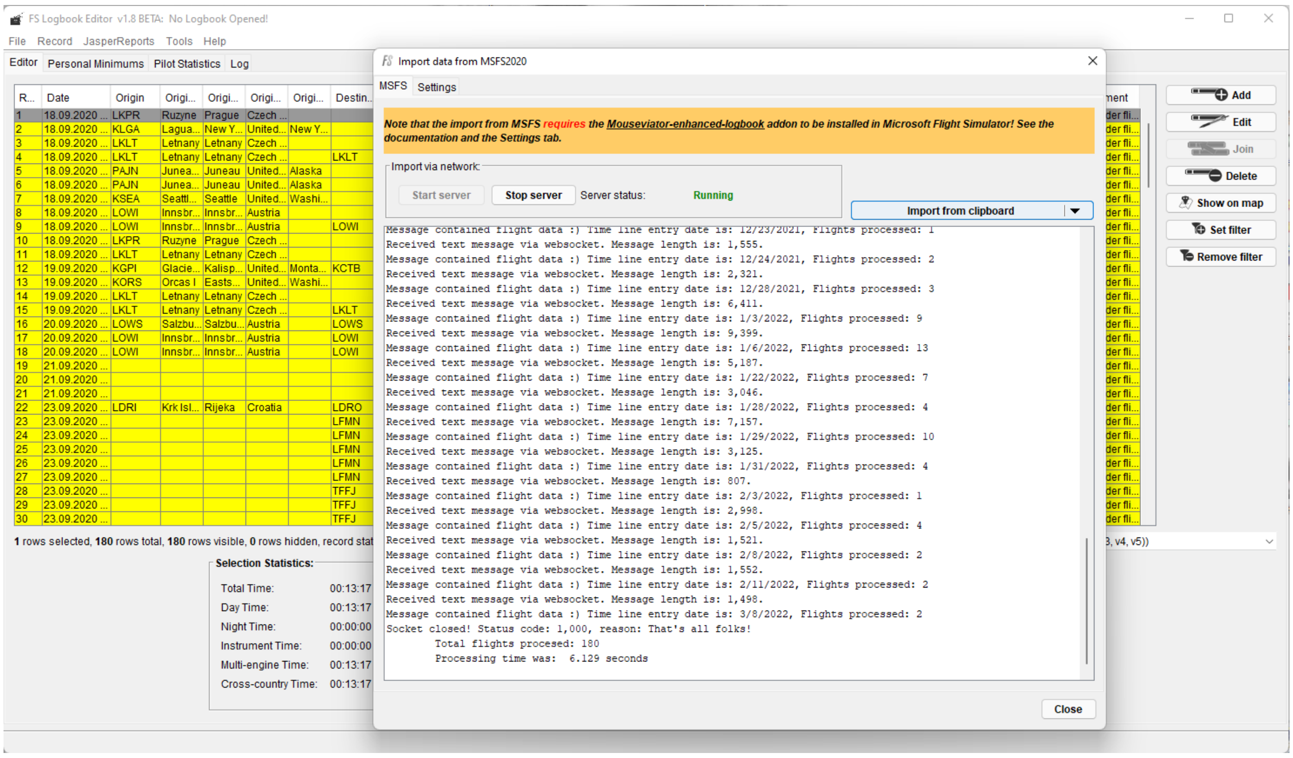
DATE	PLANE	DEPARTURE TIME	DEPARTURE PLACE	ARRIVAL TIME	ARRIVAL PLACE	TOTAL TIME OF FLIGHT	OPERATIONAL CONDITION TIME NIGHT / IFR	OPERATIONAL CONDITION TIME DAY / VFR	TAKEOFFS DAY / NIGHT	LANDINGS DAY / NIGHT	DISTANCE FLOWN
2/5/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	9:47 PM	TNCM	9:47 PM	TNCM	00:00:00	00:00:00 00:00:00	00:00:00 00:00:00	0 0	0 0	
2/5/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	8:53 PM	TNCS	9:10 PM	TNCM	00:17:05	00:00:00 00:00:00	00:17:05 00:17:05	1 0	1 0	
2/5/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	8:01 PM	TFFJ	8:14 PM	TNCS	00:13:46	00:00:00 00:00:00	00:13:46 00:13:46	0 0	1 0	
2/5/2022	AEROSOFT DHC-6 SERIES 300 WHEELS PAX	6:57 PM	TNCM	7:18 PM	TFFJ	00:20:58	00:00:00 00:00:00	00:20:58 00:20:58	0 0	1 0	
2/3/2022	CUB CRAFTER X CUB	6:57 PM	PAOU	7:56 PM	PAPE	00:59:04	00:00:00 00:00:00	00:59:04 00:59:04	1 0	1 0	
1/31/2022	TEXTRON AVIATION CESSNA 172 SKYHAWK	8:57 PM	[34CD] VICINITY	9:00 PM	[34CD] VICINITY	00:03:00	00:00:00 00:00:00	00:03:00 00:03:00	0 0	0 0	
							00:00:00	00:00:10	0	0	
TOTAL						38:44:34	02:22:19 01:07:15	36:22:15 00:00:00	62 2	58 1	8134.34 NM

EXPORT VIA CLIPBOARD | EXPORT OVER NETWORK | Logbook data export using Websocket finished!. 45 data blocks sent, 45 data blocks received.

ESC GO BACK

Picture 19: MSFS Logbook page with Mouseviator Enhanced Logbook add-on package, after clicking the Export over network button

After you click the **Export over network** button, FS Logbook Editor should start filling up with flight records, logging all of that to the log view at the MSFS Import dialog. Just like on the following picture:



R...	Date	Origin	Origi...	Origi...	Origi...	Origi...	Destin...
1	18.09.2020 ...	LKPR	Ruzyne	Prague	Czech ...		
2	18.09.2020 ...	KLGA	Lagua...	New Y...	United...	New Y...	
3	18.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		
4	18.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		LKLT
5	18.09.2020 ...	PAJN	Junea...	Juneau	United...	Alaska	
6	18.09.2020 ...	PAJN	Junea...	Juneau	United...	Alaska	
7	18.09.2020 ...	KSEA	Seattl...	Seattle	United...	Washi...	
8	18.09.2020 ...	LOWI	Innsbr...	Innsbr...	Austria		
9	18.09.2020 ...	LOWI	Innsbr...	Innsbr...	Austria		LOWI
10	18.09.2020 ...	LKPR	Ruzyne	Prague	Czech ...		
11	18.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		
12	19.09.2020 ...	KGPI	Glacie...	Kalisp...	United...	Monta...	KCTB
13	19.09.2020 ...	KORS	Orcas I	Easts...	United...	Washi...	
14	19.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		
15	19.09.2020 ...	LKLT	Letnany	Letnany	Czech ...		LKLT
16	20.09.2020 ...	LOWS	Salzbu...	Salzbu...	Austria		LOWS
17	20.09.2020 ...	LOWI	Innsbr...	Innsbr...	Austria		LOWI
18	20.09.2020 ...	LOWI	Innsbr...	Innsbr...	Austria		LOWI
19	21.09.2020 ...						
20	21.09.2020 ...						
21	21.09.2020 ...						
22	23.09.2020 ...	LDRI	Krk Isl...	Rijeka	Croatia		LDRO
23	23.09.2020 ...						LFMN
24	23.09.2020 ...						LFMN
25	23.09.2020 ...						LFMN
26	23.09.2020 ...						LFMN
27	23.09.2020 ...						LFMN
28	23.09.2020 ...						TFFJ
29	23.09.2020 ...						TFFJ
30	23.09.2020 ...						TFFJ

1 rows selected, **180** rows total, **180** rows visible, **0** rows hidden, record stat

Selection Statistics:

Total Time:	00:13:17
Day Time:	00:13:17
Night Time:	00:00:00
Instrument Time:	00:00:00
Multi-engine Time:	00:13:17
Cross-country Time:	00:13:17

Note that the import from MSFS requires the Mouseviator-enhanced-logbook addon to be installed in Microsoft Flight Simulator! See the documentation and the Settings tab.

Import via network: Start server | Stop server | Server status: Running

Import from clipboard

```
Message contained flight data :) Time line entry date is: 12/23/2021, Flights processed: 1
Received text message via websocket. Message length is: 1,555.
Message contained flight data :) Time line entry date is: 12/24/2021, Flights processed: 2
Received text message via websocket. Message length is: 2,321.
Message contained flight data :) Time line entry date is: 12/28/2021, Flights processed: 3
Received text message via websocket. Message length is: 6,411.
Message contained flight data :) Time line entry date is: 1/3/2022, Flights processed: 9
Received text message via websocket. Message length is: 9,399.
Message contained flight data :) Time line entry date is: 1/6/2022, Flights processed: 13
Received text message via websocket. Message length is: 5,187.
Message contained flight data :) Time line entry date is: 1/22/2022, Flights processed: 7
Received text message via websocket. Message length is: 3,046.
Message contained flight data :) Time line entry date is: 1/28/2022, Flights processed: 4
Received text message via websocket. Message length is: 7,157.
Message contained flight data :) Time line entry date is: 1/29/2022, Flights processed: 10
Received text message via websocket. Message length is: 3,125.
Message contained flight data :) Time line entry date is: 1/31/2022, Flights processed: 4
Received text message via websocket. Message length is: 807.
Message contained flight data :) Time line entry date is: 2/3/2022, Flights processed: 1
Received text message via websocket. Message length is: 2,998.
Message contained flight data :) Time line entry date is: 2/5/2022, Flights processed: 4
Received text message via websocket. Message length is: 1,521.
Message contained flight data :) Time line entry date is: 2/8/2022, Flights processed: 2
Received text message via websocket. Message length is: 1,552.
Message contained flight data :) Time line entry date is: 2/11/2022, Flights processed: 2
Received text message via websocket. Message length is: 1,498.
Message contained flight data :) Time line entry date is: 3/8/2022, Flights processed: 2
Socket closed! Status code: 1,000, reason: That's all folks!
        Total flights proceed: 180
        Processing time was:  6.129 seconds
```

Picture 20: MSFS logbook data imported using network

The progress of the import – what is happening is being written on the sending side – messages being displayed to the right of the **Export over network** button in MSFS, and also in the log view at MSFS Import dialog (receiving side, FS Logbook Editor embed HTTP server). No other action than starting the server is needed in FS Logbook Editor. The Mouseviator Enhanced Logbook add-on packages initiates and terminates the data transfer, FS Logbook Editor just receives, if it has what. The imported flights will be colored yellow – as imported records.

3.5. Load / save logbook from / as CSV file

FS Logbook Editor could save logbook as CSV for some time now, but since version 1.8, it can also load logbook from CSV file. While the dialog to save logbook as CSV slightly changed, let's describe the loading process first.

3.5.1. Loading logbook from CSV file

To load logbook from CSV file, use standard **Open**, **Merge** or **Append** command from ***File*** menu and then select desired CSV file. FS Logbook Editor will load a couple of lines (how many can be configured in ***Settings*** → ***Load & Save***) from the CSV file and will display the ***Picture 21: CSV Import dialog***.

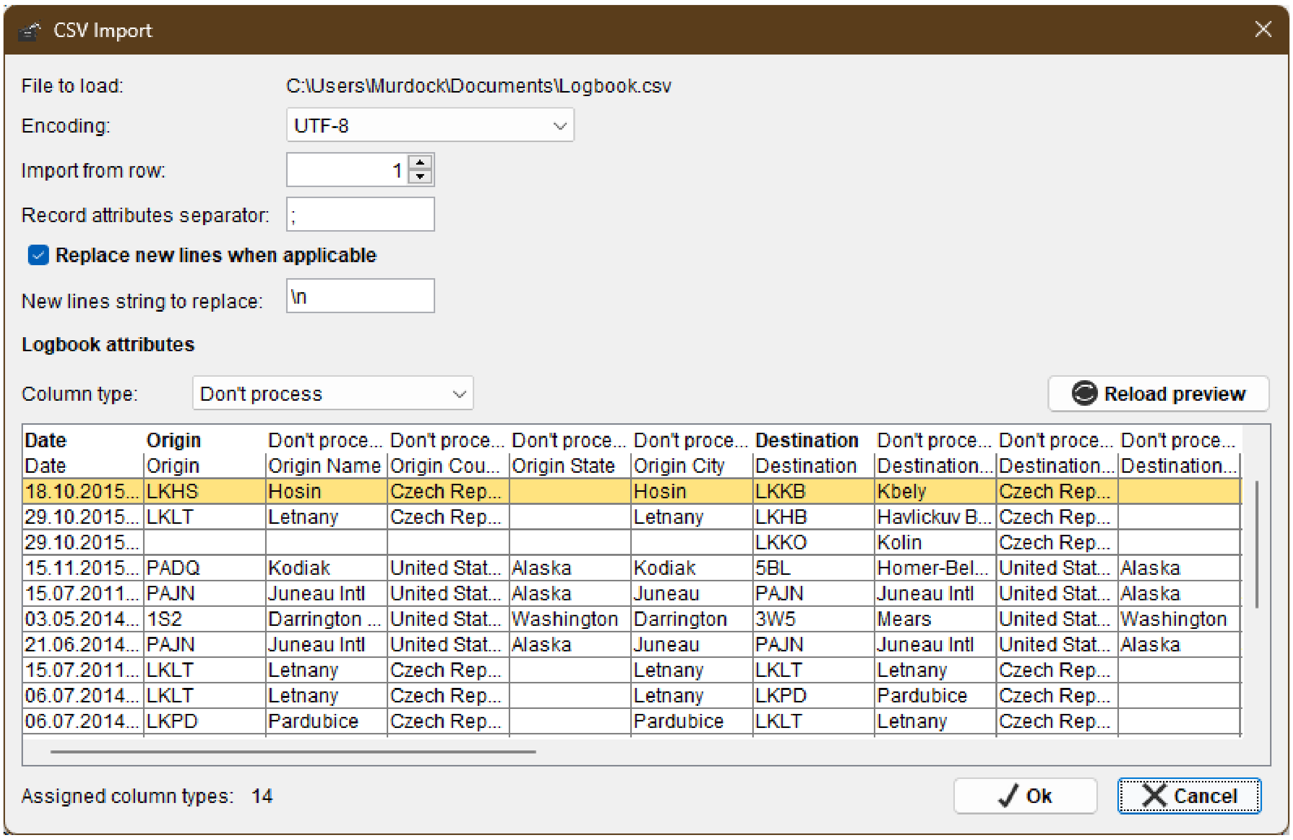
There are numerous settings to play with. The first line displays the path to the file being loaded. The **Encoding** combo box allows you to select the encoding of the file. The default, UTF-8 should work well in most cases, but you can select any.

The **Import from row:** option tells FS Logbook Editor from which line to start the import. In many cases, the first line in CSV file will contain column headers, in which there are no valid logbook data so there is no reason in trying to import that. FS Logbook Editor will try to automatically detect if the first line of the CSV file is the header line and adjust the **Import from row** value accordingly.

The **Record attribute separator:** is the character that separates logbook record attributes (columns in CSV file format) from each other. By default, this is the **;** character. If you change this character, than you need to press the **Reload preview** button to see what difference this makes in how FS Logbook Editor "sees" the columns.

The **~~Replace new lines with~~:** is FS Logbook Editor specific option. You can specify this also while saving logbook as CSV file. You can specify this also while saving logbook as CSV file. Some CSV file readers may not support multi-line column values in the CSV file (FS Logbook Editor did not until version 1.85). This might happen especially with flight comments and may break the loading or cause the file to be loaded incorrectly. Thus, while saving logbook to CSV file, FS Logbook Editor converts all multi-line text (mostly comments) to single-line by replacing new lines with special character/string. This option is to specify the character/string used for the conversion, so that FS Logbook Editor can convert single-line text back to multi-line.

In FS Logbook Editor version 1.85 the option was changed to – **Replace new line when applicable** and **New lines string to replace**. It works as before but only when the option is selected. The reason is that now an external, more robust CSV library is used to read CSV files and a CSV file with multi-line column values will no longer surprise it.


CSV Import

File to load: C:\Users\Murdock\Documents\Logbook.csv

Encoding: UTF-8

Import from row: 1

Record attributes separator: ;

☑ Replace new lines when applicable

New lines string to replace: \n

Logbook attributes

Column type: Don't process

Reload preview

Date	Origin	Don't proce...	Don't proce...	Don't proce...	Don't proce...	Destination	Don't proce...	Don't proce...	Don't proce...
Date	Origin	Origin Name	Origin Cou...	Origin State	Origin City	Destination	Destination...	Destination...	Destination...
18.10.2015...	LKHS	Hosin	Czech Rep...		Hosin	LKKB	Kbely	Czech Rep...	
29.10.2015...	LKLT	Letnany	Czech Rep...		Letnany	LKHB	Havlickuv B...	Czech Rep...	
29.10.2015...						LKKO	Kolin	Czech Rep...	
15.11.2015...	PADQ	Kodiak	United Stat...	Alaska	Kodiak	5BL	Homer-Bel...	United Stat...	Alaska
15.07.2011...	PAJN	Juneau Intl	United Stat...	Alaska	Juneau	PAJN	Juneau Intl	United Stat...	Alaska
03.05.2014...	1S2	Darrington ...	United Stat...	Washington	Darrington	3W5	Mears	United Stat...	Washington
21.06.2014...	PAJN	Juneau Intl	United Stat...	Alaska	Juneau	PAJN	Juneau Intl	United Stat...	Alaska
15.07.2011...	LKLT	Letnany	Czech Rep...		Letnany	LKLT	Letnany	Czech Rep...	
06.07.2014...	LKLT	Letnany	Czech Rep...		Letnany	LKPD	Pardubice	Czech Rep...	
06.07.2014...	LKPD	Pardubice	Czech Rep...		Pardubice	LKLT	Letnany	Czech Rep...	

Assigned column types: 14

Ok Cancel

Picture 21: CSV Import dialog

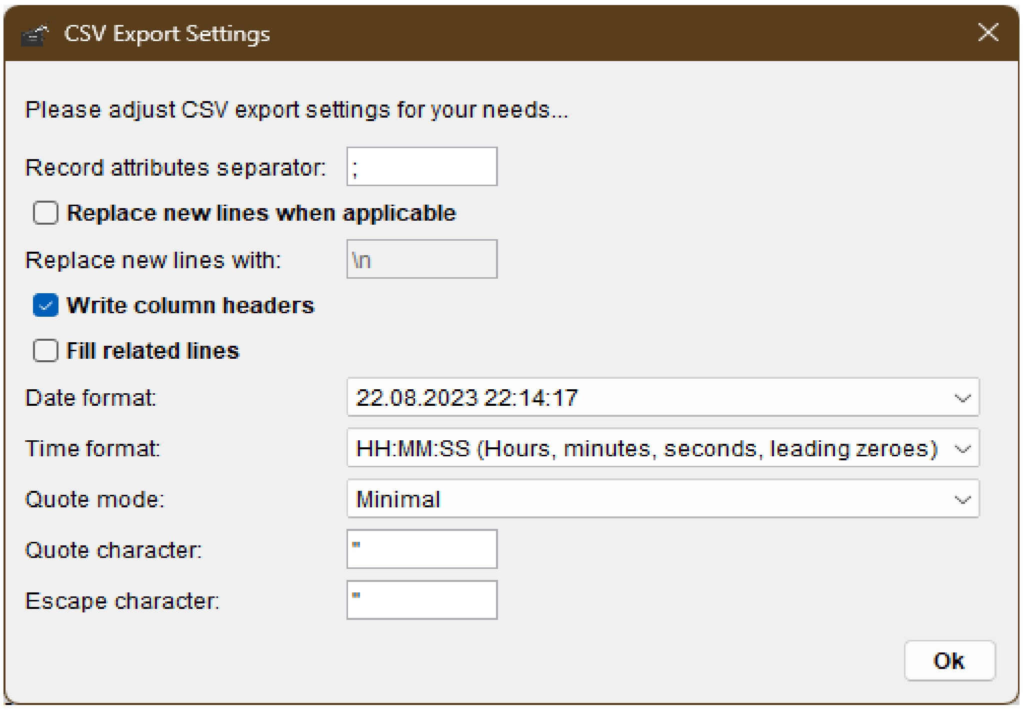
Below the **Logbook attributes** you can see the preview of the CSV file and how FS Logbook Editor "sees" the file using the Record attribute separator to split the lines into columns (record attributes). The **Column type** selection allows you to set the type of respective columns. To do that, just click on the column in the preview and select column type (what attribute) this column contains. If you created the CSV file using FS Logbook Editor, it will be able to find-out most (usually all) columns by itself. The columns marked as **Don't process** will be ignored.

If you click the **Ok** button, FS Logbook Editor will try to read the logbook from CSV file. If you click **Cancel**, the whole opening operation will be canceled.

3.5.2. Saving logbook into CSV file

To save logbook into CSV file, use the **Save** or **Save As** action from the ***File*** menu and select CSV as desired file type. The first dialog that will be shown is a dialog for selection what record attributes you want to export. Select the attributes you want and click **Ok** (there is no other button on this dialog :)). The next dialog will be the ***Picture 22: CSV Export Settings dialog***.

There are various options to fiddle with. The **Record attribute separator:** is character that will be used to separate each record attribute (like flight date, time etc.). In CSV, every record (flight) must be saved in one line, and the columns (record attributes) must be separated by some special character, so that it would be possible to decode this single-line information later. By default, the **;** is most commonly used.


CSV Export Settings

Please adjust CSV export settings for your needs...

Record attributes separator: ;

☐ Replace new lines when applicable

Replace new lines with: \n

☑ Write column headers

☐ Fill related lines

Date format: 22.08.2023 22:14:17

Time format: HH:MM:SS (Hours, minutes, seconds, leading zeroes)

Quote mode: Minimal

Quote character: "

Escape character: "

Picture 22: CSV Export Settings dialog

The **~~Replace new lines with~~:** is FS Logbook Editor specific option. You can specify this also while loading logbook from CSV file. For the CSV file to be valid, one record – one flight from logbook, must be saved at one line in CSV file. The flight comments may and often are multi-line. This would break the CSV file. Thus, while saving logbook to CSV file, FS Logbook Editor converts all multi-line text (mostly comments) to single-line by replacing new lines with special character/string. This option is to specify the character/string used for the conversion.

As in the case of importing CSV file, in FS Logbook Editor the option was changed to – **Replace new lines when applicable** and **Replace new lines with**. It works as before, but will replace new lines only when enabled. The reason is that version 1.85 uses new library to work with CSV files, which is more robust and have no problems with multi-line column values. It can read them and save them correctly.

Write column headers – will print the record attribute names (column headers) to the first line of the CSV file.

Fill related lines – In the case the logbook flight record has more than one entry in the airport list (See ***Airport List***), FS Logbook Editor will save this flight record in multiple lines in the CSV file, where each line will have the respective airport ICAO code and number of landings in the Airport list column. If this option is enabled, FS Logbook Editor will reprint other record attributes, such as date, times etc. to those additional lines too. If disabled, only the first line of the record will contain these information.

The **Date format** is simply a selection for the date format you wish to use for the export, as well as **Time format** is.

Quote Mode, **Quote character** and **Escape character** are options added in FS Logbook Editor version 1.85 and are related to the new CSV library that is being used to work with CSV files. The **Quote mode** defines how column values are quoted inside the CSV file. Quote character, as you might guess, tells what character to use for quoting. By default it is the " character. Usually, string values are being quoted. Like, should the value of example comment field be: ***Hello***, it would be quoted in the file like ***"Hello"***.

Quote mode can be:

- **Quote all fields** – all columns in the CSV file will be quoted.
- **All non-empty fields** – all columns that actually have some value will be quoted.
- **Minimal** – Only columns that require quotes will be quoted. Usually it will be fields that contain some special character, like the character that is being used as **Record attribute separator**.
- **All non-numeric fields** – also self explaining all non-numeric fields will be quoted. Note that date and time values may not be treated as numbers, so more columns than expected might get quoted.
- **None** – none of the fields will be quoted.

The **Escape character** is the character that will be used to escape special characters inside column values if **Quote mode** is set to **None**. For example, if column value contains Record attributes separator character – ***;***, and the Escape character is ", it will be printed as ***";*** in the CSV file.

These options are there just to allow you to setup the output as you like it. Most CSV readers should be able to read resulting CSV files without issues, including FS Logbook Editor.

Note that is **Quote Mode** is **not None**, the **Quote character** must be set, cannot be empty. Likewise, if **Quote mode** is **None**, the **Escape character** must be set, cannot be empty.

3.6. Load / save logbook from / as LittleNavMap CSV file

FS Logbook Editor version 1.85 has special handler to load/save LittleNavMap specific CSV file. So flights can be imported from LittleNavMap to FS Logbook Editor and vice versa. Of course, both programs support different set of data, so there will be some loss of information about the flights.

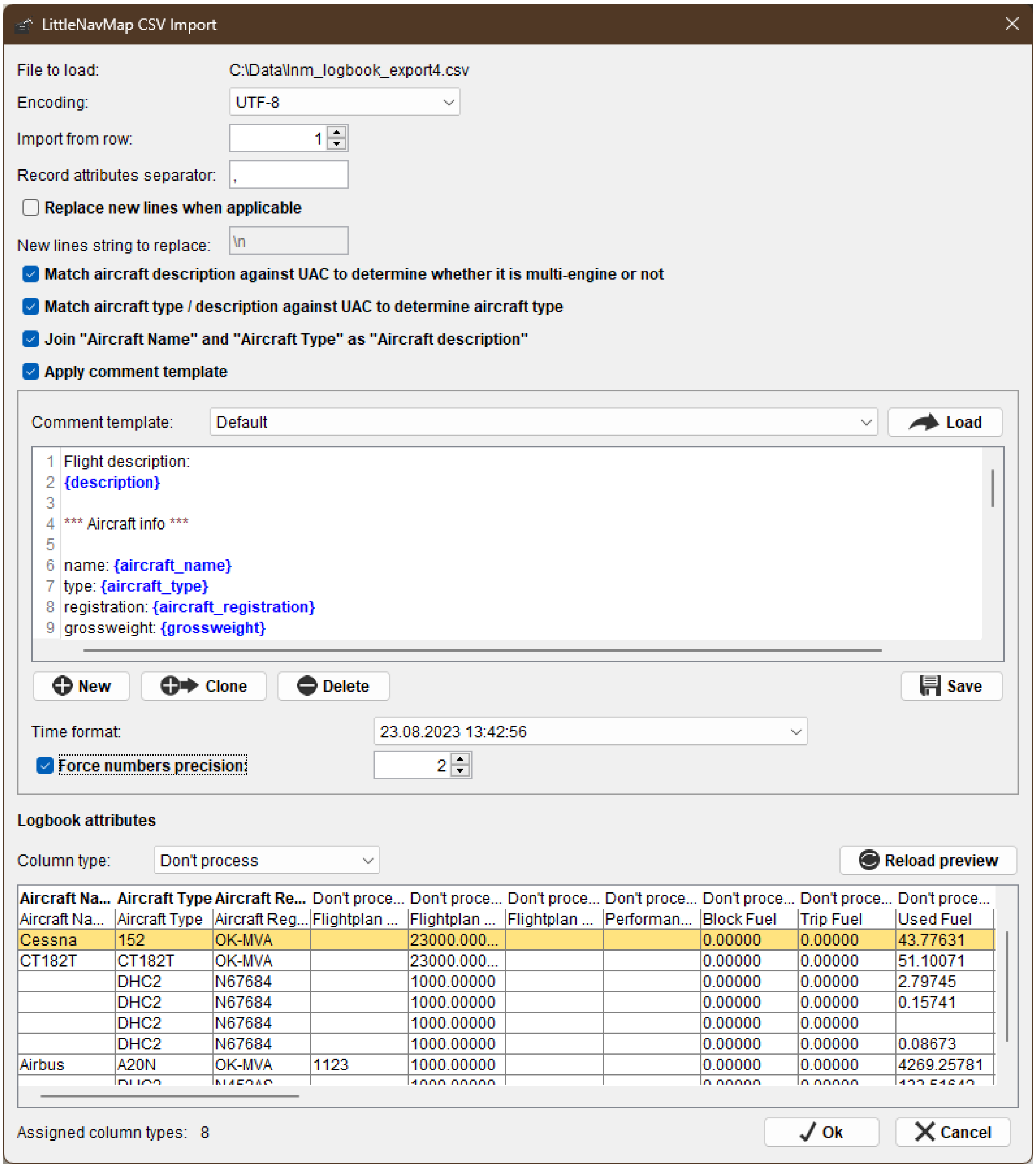
3.6.1. Loading logbook from LittleNavMap CSV file

To load logbook from CSV file, use standard **Open**, **Merge** or **Append** command from ***File*** menu and then select desired CSV file. FS Logbook Editor will load a couple of lines (how many can be configured in ***Settings*** → ***Load & Save***) from the CSV file and will display the ***Picture 23: LittleNavMap CSV Import dialog***.

There are numerous settings to play with. The first line displays the path to the file being loaded. The **Encoding** combo box allows you to select the encoding of the file. The default, UTF-8 should work well in most cases, but you can select any.

The **Import from row:** option tells FS Logbook Editor from which line to start the import. In many cases, the first line in CSV file will contain column headers, in which there are no valid logbook data so there is no reason in trying to import that. FS Logbook Editor will try to automatically detect if the first line of the CSV file is the header line and adjust the **Import from row** value accordingly.

The **Record attribute separator:** is the character that separates logbook record attributes (columns in CSV file format) from each other. By default, this is the **;** character. If you change this character, than you need to press the **Reload preview** button to see what difference this makes in how FS Logbook Editor "sees" the columns.


LittleNavMap CSV Import

File to load: C:\Data\lnm_logbook_export4.csv

Encoding: UTF-8

Import from row: 1

Record attributes separator: ,

☐ Replace new lines when applicable

New lines string to replace: \n

☑ Match aircraft description against UAC to determine whether it is multi-engine or not

☑ Match aircraft type / description against UAC to determine aircraft type

☑ Join "Aircraft Name" and "Aircraft Type" as "Aircraft description"

☑ Apply comment template

Comment template: Default

```
Flight description:
{description}

*** Aircraft info ***

name: {aircraft_name}
type: {aircraft_type}
registration: {aircraft_registration}
grossweight: {grossweight}
```

Time format: 23.08.2023 13:42:56

☑ Force numbers precision: 2

Logbook attributes

Column type: Don't process

Aircraft Na...	Aircraft Type	Aircraft Re...	Don't proce...	Don't proce...	Don't proce...	Don't proce...	Don't proce...	Don't proce...	Don't proce...
Aircraft Na...	Aircraft Type	Aircraft Reg...	Flightplan ...	Flightplan ...	Flightplan ...	Performan...	Block Fuel	Trip Fuel	Used Fuel
Cessna	152	OK-MVA		23000.000...			0.00000	0.00000	43.77631
CT182T	CT182T	OK-MVA		23000.000...			0.00000	0.00000	51.10071
	DHC2	N67684		1000.00000			0.00000	0.00000	2.79745
	DHC2	N67684		1000.00000			0.00000	0.00000	0.15741
	DHC2	N67684		1000.00000			0.00000	0.00000	
	DHC2	N67684		1000.00000			0.00000	0.00000	0.08673
Airbus	A20N	OK-MVA	1123	1000.00000			0.00000	0.00000	4269.25781

Assigned column types: 8

Picture 23: LittleNavMap CSV Import dialog

Replace new line when applicable and **New lines string to replace** are FS Logbook Editor specific options. You can specify this also while saving logbook as CSV file. Some CSV file readers may not support multi-line column values in the CSV file (FS Logbook Editor did not until version 1.85). This might happen especially with flight comments and may break the loading or cause the file to be loaded incorrectly. If enabled, FS Logbook Editor converts all multi-line text (mostly comments) to single-line by replacing new lines with special character/string. This option is to specify the character/string used for the conversion, so that FS Logbook Editor can convert single-line text back to multi-line.

Try to match aircraft description against UAC to determine whether it is multi-engine or not – LittleNavMap flight records does not contain information whether the used aircraft was multi-engine or not, so this information, that FS Logbook Editor supports, would always be false. When this option is enabled, FS Logbook Editor will try to find out aircraft name in user aircraft categories (See ***User aircraft categories***). User aircraft categories, since FS Logbook Editor version 1.8, have new property that specifies that the aircraft category contains multi-engine aircraft. Since FS Logbook Editor version 1.85, this property can be set for each aircraft within the category individually. This way, if set properly, this information can be determined automatically during import.

Try to match aircraft type / description against UAC to determine aircraft type - LittleNavMap flight records have Aircraft Type column, but it does not correspond to aircraft types supported by FS Logbook Editor so it would always be the default – Fixed Wing. When this option is enabled, FS Logbook Editor will try to find out aircraft type in user aircraft categories (See ***User aircraft categories***) based on information in Aircraft Type column from the CSV file, or from aircraft description if it fails on aircraft type. User aircraft categories, since FS Logbook Editor version 1.85, have new property that specifies what aircraft type the category contains. On category level, is is sort of default value for all aircraft within the category, but each aircraft can override it. This way, if set properly, this information can be determined automatically during import.

Join "Aircraft Name" and "Aircraft Type" as "Aircraft description" – From FS Logbook Editor perspective, if the CSV file was generated by logbook from LittleNavMap, the aircraft description in FS Logbook Editor is more sensible to be constructed by joining aircraft name and aircraft type columns from the CSV file. May not be true if the CSV file was generated by other program, like FS Logbook Editor itself, thus, there is this option for you to choose.

Apply comment template - Flights imported from LittleNavMap CSV file contains a lot of additional information for which there are no „columns" in FS Logbook Editor. I did not want to just discard these information and the solutions seemed to be, to allow you to insert them into the flight comment. As usually, I like the things to be configurable, so the solution is to use the „templates" – so everyone you can insert information that you are interested in in the format you want. That is what is gonna happen if this option is enabled. There is also „simple" editor, that allows to **Load** existing templates, edit and **Save** them, create **New** ones, and **Clone** existing ones. I believe the GUI is quite self-explanatory, so try it out. The big white place is a text editor to write/edit the templates. It is a "smart" editor. If you **press Ctrl+Space**, it will show list of keywords supported (auto-completion window), along with short explanantory of their meaning. The keywords are enclosed in { and }. Keyword is special word that will be replaced by FS Logbook Editor by respective information from LittleNavMap CSV file. For example, the **{departure_ident}** keyword will be replaced by the ICAO code of departure airport of the respective flight. FS Logbook Editor contains two default templates – **Default_en** and **Default_cs_CZ** – one is in English, the other in Czech. Note that they are just named like this, **they will not switch automatically** if you change FS Logbook Editor language. The active template, that will be used during import (if enabled), is the one selected in the list. These default templates also contain all possible keywords! They are stored in the **res/lnm/templates** (as all other templates you create here). If you delete the default templates, FS Logbook Editor will recreate them on next start, just like other crucial files.

In the comment template section there are other options to modify. The **Time format** allows you to override the format of time values printed in the comment template. You can choose from predefined one or choose **Do not change** to keep the time values exactly as they are in the CSV file. The similar way you can modify number values. Just make sure the **Force numbers precision** is selected and choose desired numbers precision in the spinner next to the check box. If unchecked, raw values from CSV file will be used. Those might have like 6 decimal places and not everyone may like it.

The last is the **Logbook attributes** section with the preview of CSV file contents and how FS Logbook Editor "sees" the file using the Record attribute separator to split the lines into columns (record attributes). The **Column type** selection allows you to set the type of respective columns. In this dialog it is however slightly different than what was described in the loading of regular CSV file. LittleNavMap CSV file has some columns, that do not 100% assignable to FS Logbook Editor logbook record attributes. So, rather than assigning the type of FS Logbook attribute, you are actually assigning LittleNavMap CSV columns. FS Logbook Editor will try to assign the correct column types and your job is just to look it out and make sure it is correct. Maybe do some corrections. FS Logbook Editor will than transform those into respective logbook record attributes.

If you click on the column in the preview and than hover the mouse over the selected column in the **Column type** combo box, a tool-tip text will be shown with description of what will happen with that value.

You may notice that there is less columns to assign than with regular CSV import and most columns will be set to **Don't process**. Information from almost all those columns can be inserted to comment using comment template.

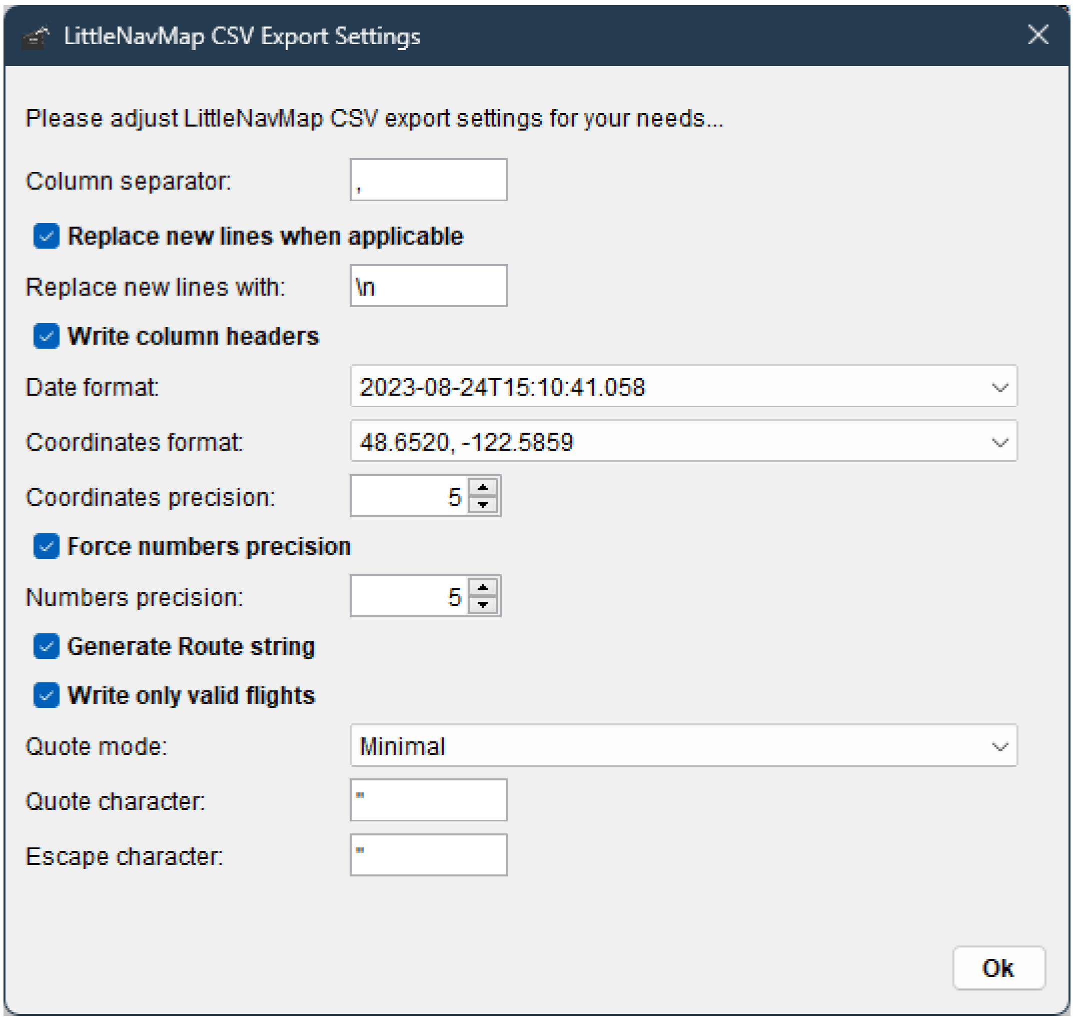
If you click the **Ok** button, FS Logbook Editor will try to read the logbook from LittleNavMap CSV file using the specified settings. If you click **Cancel**, the whole opening operation will be canceled.

3.6.2. Saving logbook into LittleNavMap CSV file

To save logbook into CSV file, use the **Save** or **Save As** action from the ***File*** menu and select LittleNavMap CSV as desired file type. The first dialog that will be shown is a dialog for selection what record attributes you want to export. Select the attributes you want and click **Ok** (there is no other button on this dialog :)). The next dialog will be the ***Picture 24: LittleNavMap CSV Export settings dialog***. Most settings in the dialog are there just because I don't like to hard-code anything without the option to change anything. Most of the settings will be just fine in their default values, but still you can adjust them.

There are various options to fiddle with. The **Record attribute separator:** is character that will be used to separate each record attribute (like flight date, time etc.). In CSV, every column (record attribute) must be separated by some special character, so that it would be possible to distinguish between them when reading. By default, LittleNavMap uses **,** as separator.

<content follows on next page>



Please adjust LittleNavMap CSV export settings for your needs...

Column separator:	,
✓ Replace new lines when applicable	
Replace new lines with:	\n
✓ Write column headers	
Date format:	2023-08-24T15:10:41.058
Coordinates format:	48.6520, -122.5859
Coordinates precision:	5
✓ Force numbers precision	
Numbers precision:	5
✓ Generate Route string	
✓ Write only valid flights	
Quote mode:	Minimal
Quote character:	"
Escape character:	"

Picture 24: LittleNavMap CSV Export settings dialog

Replace new line when applicable and **New lines string to replace** are FS Logbook Editor specific options. You can specify this also while reading logbook from CSV file. Some CSV file readers may not support multi-line column values in the CSV file (FS Logbook Editor did not until version 1.85). This might happen especially with flight comments and may break the loading or cause the file to be loaded incorrectly. If enabled, FS Logbook Editor converts all multi-line text (mostly comments) to single-line by replacing new lines with special character/string defined in this option. The same should be set when importing CSV file back with FS Logbook Editor.

Write column headers – whether or not to write first line with column headers to the resulting CSV file. I think LittleNavMap will protest if column headers are not present, but in case that changes…

Date format – allows you to choose desired format of date values. I think LittleNavMap supports only the *yyyy-MM-dd'T'HH:mm:ss.SSS* date format, but still you can choose another one. Just in case I am wrong.

Coordinates format – the format of geographic coordinate values. LittleNavMap uses the decimal format with 5 decimal places precision – that is set by the **Coordinates precision**. Still, this may change so you can choose different ones if you want to try.

Force numbers precision – When enabled, the amount of decimal places for numerical values will be forced to the value set by **Numbers precision**. If disabled, the amount of decimal places will be as needed.

Generate Route string – LittleNavMap CSV file has a column named "Route string", which should contain description of route flown or sort of flight plan. If enabled, FS Logbook Editor will generate the route string from departure, destination and en-route airports. So the resulting route string will be for example: LKLT – LKMB – LKLT.

Write only valid flights – LittleNavMap only supports flights that have departure and/or destination airport filled. FS Logbook Editor flight record can exist without either of those. If enabled, FS Logbook Editor will export only flights that meet that restriction – you should keep this enabled. But again, just in case this changes, the option is there to disable this behavior.

Quote Mode, **Quote character** and **Escape character** are related to the CSV library that is being used to work with CSV files. The **Quote mode** defines how column values are quoted inside the CSV file. Quote character, as you might guess, tells what character to use for quoting. By default it is the **"**character. Usually, string values are being quoted. Like, should the value of example comment field be: ***Hello***, it would be quoted in the file like ***"Hello"***.

Quote mode can be:

- **Quote all fields** – all columns in the CSV file will be quoted.
- **All non-empty fields** – all columns that actually have some value will be quoted.
- **Minimal** – Only columns that require quotes will be quoted. Usually it will be fields that contain some special character, like the character that is being used as **Record attribute separator**.
- **All non-numeric fields** – also self explaining all non-numeric fields will be quoted. Note that date and time values may not be treated as numbers, so more columns than expected might get quoted.
- **None** – none of the fields will be quoted.

The **Escape character** is the character that will be used to escape special characters inside column values if **Quote mode** is set to **None**. For example, if column value contains Record attributes separator character –**;** and the Escape character is ", it will be printed as "; in the CSV file.

These options are there just to allow you to setup the output as you like it. Most CSV readers should be able to read resulting CSV files without issues, including FS Logbook Editor.

3.7. Load / save logbook from / as Excel spreadsheet

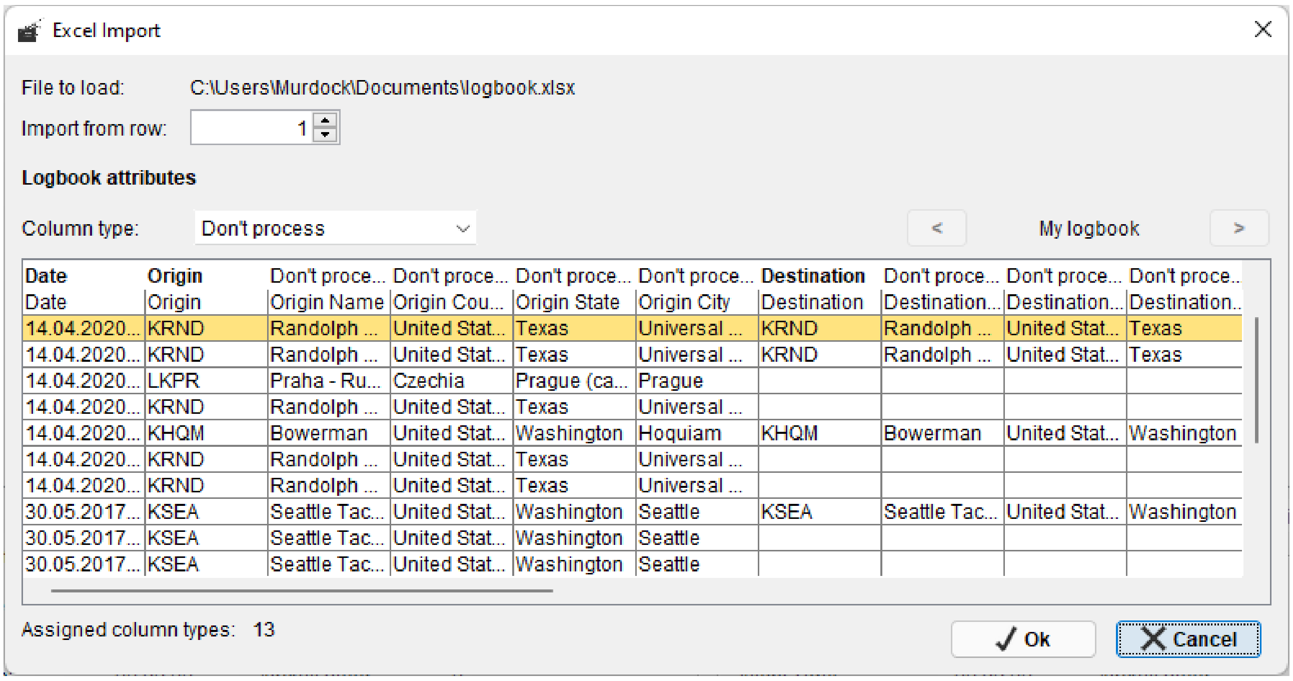
Since FS Logbook Editor version 1.8 it is possible to load/save logbook to/from Excel spreadsheet.

3.7.1. Loading logbook from Excel spreadsheet

To load logbook from Excel spreadsheet file, use standard **Open**, **Merge** or **Append** command from ***File*** menu and then select desired Excel file. FS Logbook Editor will load a couple of lines (how many can be configured in ***Settings*** → ***Load & Save***) from the Excel spreadsheet file and will display the ***Picture 25: Excel Import dialog***.

The first line displays the path to the file being loaded.

The **Import from row:** option tells FS Logbook Editor from which line to start the import. The logbook may not start at the first line or it may contain column headers, in which there are no valid logbook data so there is no reason in trying to import that. FS Logbook Editor will try to automatically detect if the first line of the Excel file is the header line and adjust the **Import from row** value accordingly.


Excel Import

File to load: C:\Users\Murdock\Documents\logbook.xlsx

Import from row: 1

Logbook attributes

Column type: Don't process

< My logbook >

Date	**Origin**	Don't proce...	Don't proce...	Don't proce...	Don't proce...	**Destination**	Don't proce...	Don't proce...	Don't proce..
Date	Origin	Origin Name	Origin Cou...	Origin State	Origin City	Destination	Destination...	Destination...	Destination..
14.04.2020...	KRND	Randolph ...	United Stat...	Texas	Universal ...	KRND	Randolph ...	United Stat...	Texas
14.04.2020...	KRND	Randolph ...	United Stat...	Texas	Universal ...	KRND	Randolph ...	United Stat...	Texas
14.04.2020...	LKPR	Praha - Ru...	Czechia	Prague (ca...	Prague				
14.04.2020...	KRND	Randolph ...	United Stat...	Texas	Universal ...				
14.04.2020...	KHQM	Bowerman	United Stat...	Washington	Hoquiam	KHQM	Bowerman	United Stat...	Washington
14.04.2020...	KRND	Randolph ...	United Stat...	Texas	Universal ...				
14.04.2020...	KRND	Randolph ...	United Stat...	Texas	Universal ...				
30.05.2017...	KSEA	Seattle Tac...	United Stat...	Washington	Seattle	KSEA	Seattle Tac...	United Stat...	Washington
30.05.2017...	KSEA	Seattle Tac...	United Stat...	Washington	Seattle				
30.05.2017...	KSEA	Seattle Tac...	United Stat...	Washington	Seattle				

Assigned column types: 13

Picture 25: Excel Import dialog

Below the **Logbook attributes** you can see the preview of the Excel spreadsheet file. The **Column type** selection allows you to set the type of respective columns. To do that, just click on the column in the preview and select column type (what attribute) this column contains. If you created the Excel file using FS Logbook Editor, it will be able to find-out most (usually all) columns by itself. The columns marked as **Don't process** will be ignored. To the right and above the preview table you will see the < and > buttons with text in between them. The text is the name of currently viewed sheet from the Excel file. If the Excel file contains more than one sheet, you can navigate between them using the < and > buttons. Only the currently selected (viewed) sheet will be used for loading logbook after clicking the **Ok** button.

If you click the **Ok** button, FS Logbook Editor will try to read the logbook from CSV file. If you click **Cancel**, the whole opening operation will be canceled.

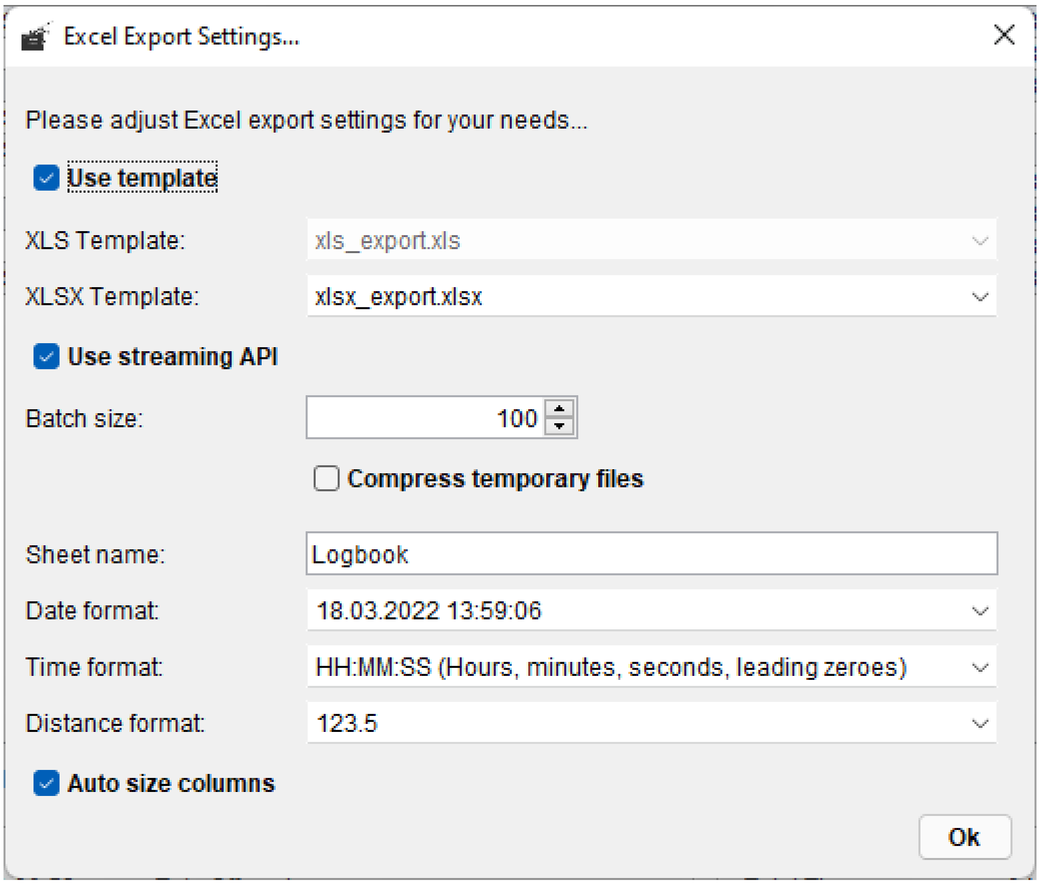
Note 1: The loading of the preview may take significant time if the Excel file is big, because even since the preview shows just couple of lines, it must process the whole file to do that! Yes, the Excel file is actually being read twice! Once for the preview, second time for the actual loading. But there is no other way, the building of preview to define which column is which is required step. FS Logbook Editor will display a "patience message" while loading the preview.

Note 2: FS Logbook Editor only load the actual data from the Excel file! No formatting info or anything else. For that reason, after opening logbook from Excel spreadsheet, the **Save** action in the ***File*** menu will remain **disabled**, so you can't accidentally overwrite the source file.

3.7.2. Saving logbook as Excel spreadsheet

To save logbook as Excel spreadsheet file, use the **Save** (will be disabled if you opened logbook from Excel file) or **Save As** action from the ***File*** menu and select Excel as desired file type. The first dialog that will be shown is a dialog for selection what record attributes you want to export. Select the attributes you want and click **Ok** (there is no other button on this dialog :)). The next dialog will be the *Picture 26: Excel Export Settings dialog*.

Let's try to bring some meaning into this slightly complex looking dialog. The **Use template** check box will enable the option to use template for creation of target Excel file. The **XLS Template** and **XLSX Template** combo boxes will be enabled based on whether you are saving the logbook as older **.xls** file or newer **.xlsx** file respectively. FS Logbook Editor comes with one default template:


Excel Export Settings...

Please adjust Excel export settings for your needs...

☑ Use template

XLS Template: xls_export.xls

XLSX Template: xlsx_export.xlsx

☑ Use streaming API

Batch size: 100

☐ Compress temporary files

Sheet name: Logbook

Date format: 18.03.2022 13:59:06

Time format: HH:MM:SS (Hours, minutes, seconds, leading zeroes)

Distance format: 123.5

☑ Auto size columns

Picture 26: Excel Export Settings dialog

xls_export.xls for XLS files, and **xlsx_export.xlsx** for XLSX files. These templates are stored within the **res** folder in the FS Logbook Editor install or data folder. The advantage of using templates is that you can modify styles that will be applied to various cells (columns) and records (lines) and get nicely formatted Excel file right away after export. The style names are the same as the export to ODS uses. Look at the ***Customizing ods template***. With the Excel spreadsheet templates it works exactly the same way like with the .ods template.

Use streaming API checkbox, **Batch size** and **Compress temporary files** checkbox will only be available while saving logbook as XLSX file. Normally, before the logbook can be saved to Excel file, FS Logbook Editor (the included library to work with Excel files) needs to create the whole document in computer memory. This can lead to really big resources (mainly memory) usage in the case of big logbook. For example, I was trying to save a logbook with roughly 81000 records and the memory required to build the Excel file in memory was about 2-3GB. Not mentioning that FS Logbook Editor was quite slow with that amount of data and it took quite some time. While using streaming API, the data are being saved to the file in smaller parts. How big? That is what the **Batch size** is for. If set for example to 100, like on the picture, the data will be saved to the file after each 100 lines are build in memory. This approach needs far less memory and is also faster. Behind the scenes, the streaming functionality uses temporary files. To lower the disk usage, an option to

Compress temporary files is available there. Compression on the other hand, makes the processing longer. The downside of using the streaming API is that summary line that FS Logbook inserts as the last line in the Excel file, will not contain formulas, but hardly computed values and that the streaming API is only supported for the newer XLSX Excel file format, not the older XLS format.

The rest of the available options are not that complex. The **Sheet name** will be the sheet name of sheet created in the Excel file for the logbook. The **Date format**, **Time format** and **Distance format** allows you to select the format of date, time and distance values. The **auto size columns** option, when enabled, will make sure the columns have optimal size for their content to fit in.

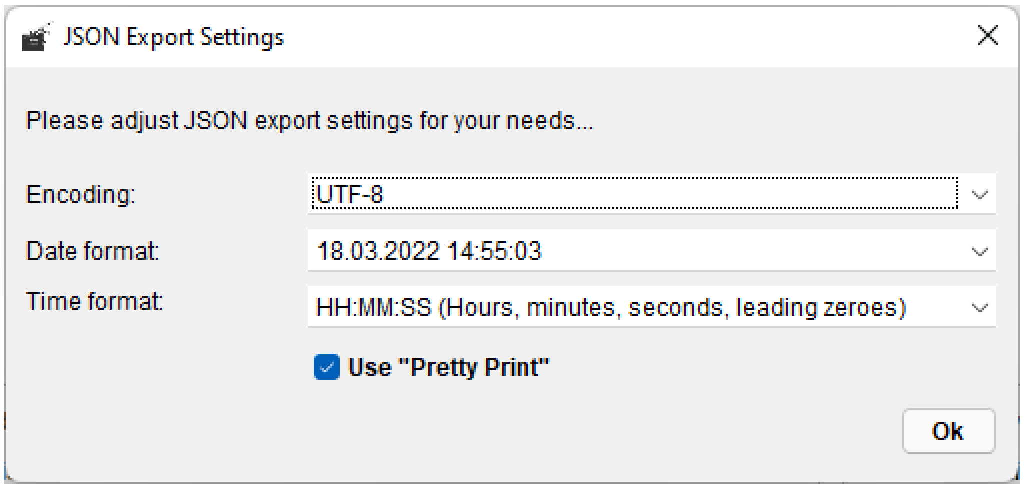
The dialog will remember last selected options, separately for XLS and XLSX file types.

3.8. Load / save logbook from / as JSON file

The JSON is open standard file format for data interchange that uses human readable text … if that does not ring any bells for you, just skip this…

The load process is simple, no questions asked. As long as the file format is what FS Logbook Editor expects, there will be no issues.

While saving logbook to JSON format, the following dialog will be shown:


JSON Export Settings

Please adjust JSON export settings for your needs...

Encoding: UTF-8

Date format: 18.03.2022 14:55:03

Time format: HH:MM:SS (Hours, minutes, seconds, leading zeroes)

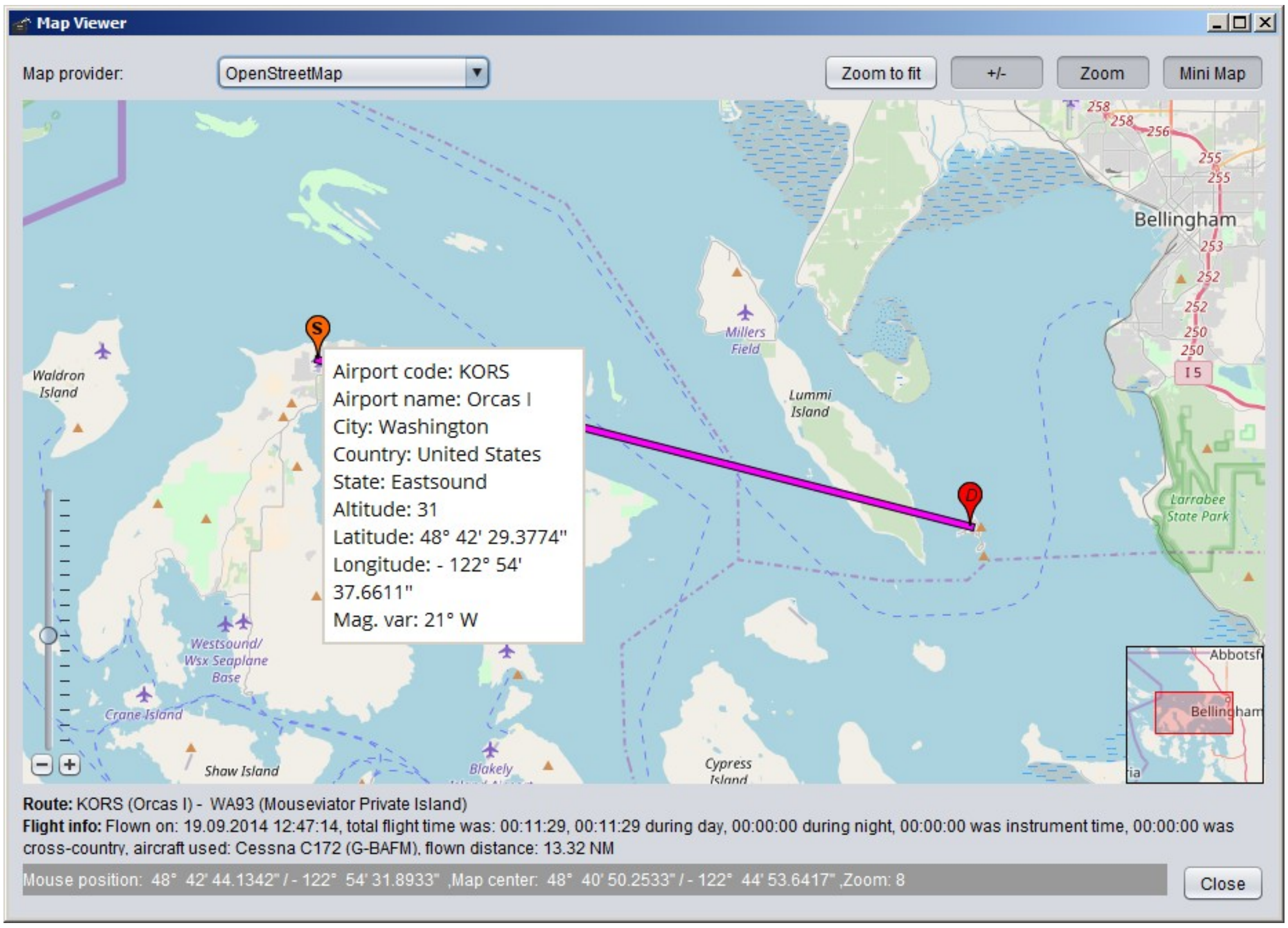
[✓] Use "Pretty Print"

Picture 27: The JSON Export Settings dialog

File **encoding** can be selected. The default **UTF-8** should be enough most of the times. Since JSON is text based format, you can select the format to format date values (**Date format**) and time values (**Time format**). You can also enable **Use “Pretty Print”** which makes the resulting file more easily readable for humans, but at the cost of bigger size.

3.9. Viewing flight on map

When you select a flight in the logbook table at Editor tab which has at least one airport specified, you can view that flight route on the map. Just click **View on map** button. You must have some airport data imported for this feature to work (airport coordinates would come handy when we want to display the flight route on the map). See the ***Simulators*** and ***Airport Data*** chapters for more details about the airport data subject. When these conditions are met, you can see the flight route of your flight such as the one on the following picture:



Airport code: KORS
Airport name: Orcas I
City: Washington
Country: United States
State: Eastsound
Altitude: 31
Latitude: 48° 42' 29.3774"
Longitude: - 122° 54' 37.6611"
Mag. var: 21° W

Route: KORS (Orcas I) - WA93 (Mouseviator Private Island)
Flight info: Flown on: 19.09.2014 12:47:14, total flight time was: 00:11:29, 00:11:29 during day, 00:00:00 during night, 00:00:00 was instrument time, 00:00:00 was cross-country, aircraft used: Cessna C172 (G-BAFM), flown distance: 13.32 NM

Mouse position: 48° 42' 44.1342" / - 122° 54' 31.8933" ,Map center: 48° 40' 50.2533" / - 122° 44' 53.6417" ,Zoom: 8

Picture 28: Map Viewer window showing selected flight route

The Map Viewer window is rather simple. Most of its area is occupied by the map view. You can control the map view as any other map application. Use your mouse to drag the map and mouse wheel to zoom in/out. You can zoom also using the zoom slider or zoom buttons at left bottom edge, provided they are displayed. Move the mouse over the airport (route point) to view the information about it. The appearance and format of latitude, longitude and magnetic variation can be set at ***Map Viewer*** section of ***Settings***.

Above the map view, there is a list labeled **Map provider:**. It allows you to switch between available map providers. I think there is no need to comment on this one much more, just look to see which ones are there. On the right side, there are four buttons:

- **Zoom to fit** – will center the map on the flight route and will find the maximum zoom at which you can see the whole route with about 30% margin of the map area on the sides.
- **+/-** - will show/hide the zoom buttons at the left bottom edge of the map. This is a switch button. When it is gray (pushed down), the buttons will be visible. When it is not pressed, the buttons will be hidden. Changing this will also change the same option at ***Map Viewer*** section of ***Settings***.
- **Zoom** – will show/hide the zoom slider at the left bottom edge of the map. This is a switch button. When it is gray (pushed down), the slider will be visible. When it is not pressed, the slider will be hidden. Changing this will also change the same option at ***Map Viewer*** section of ***Settings***.
- **Mini Map** – will show/hide the mini map at the right bottom corner of the map. This is a switch button. When it is gray (pushed down), the mini map will be visible. When it is not pressed, the mini map will be hidden. Changing this will also change the same option at

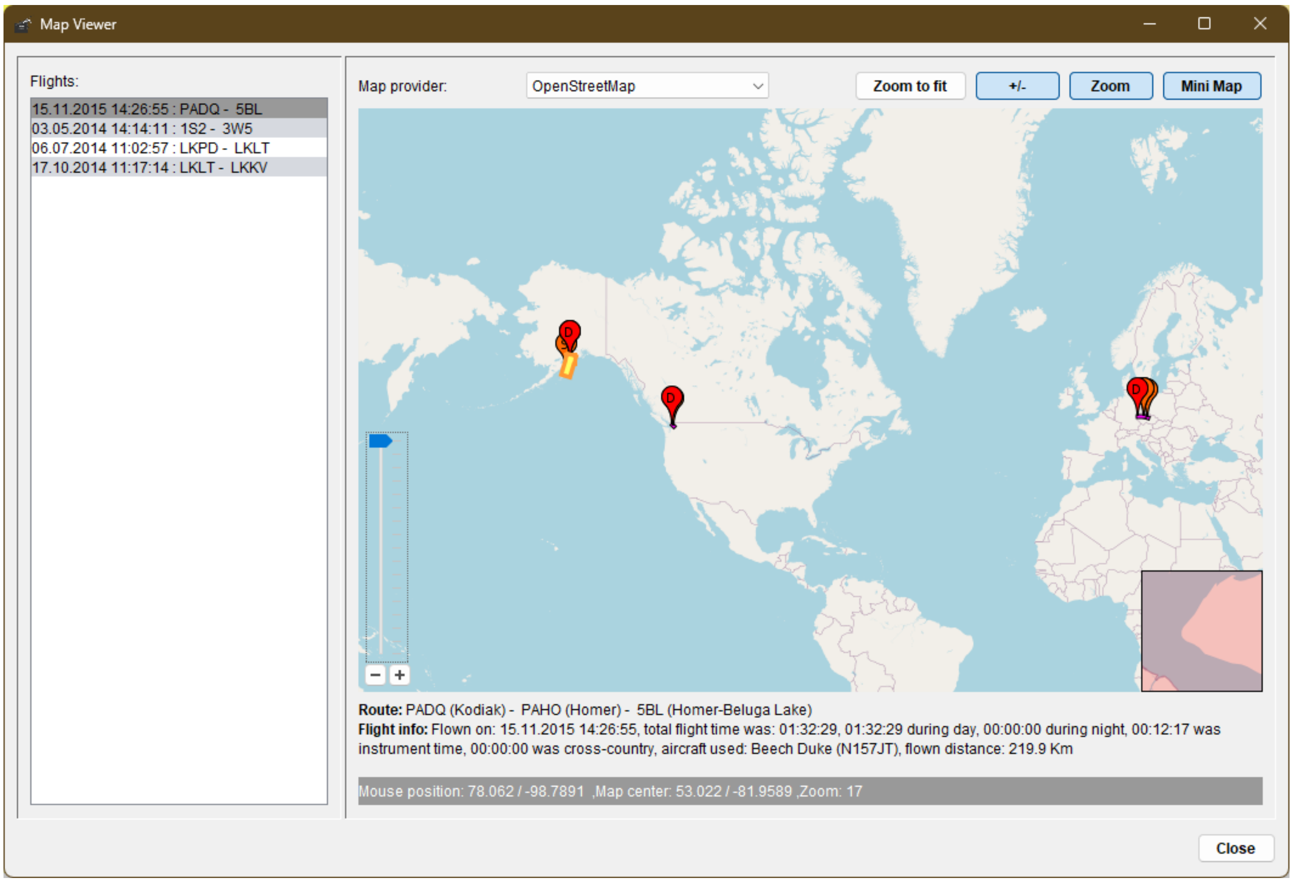
Map Viewer section of *Settings*.

Below the map view, there are two information labels. One named **Route**, which should contain textual representation of flight route. The other is **Flight info**, which should contain information about the flight, such as date of flight, total time etc. Basically most of the information you can see in the logbook table for respective flight.

Below those, there is a gray information box. It shows the coordinates of map center, current mouse position and actual zoom level. The format in which geographic coordinates are shown can be set at *Map Viewer* section of *Settings*.

The last control here is a **Close** button at the right bottom corner of the window. It just closes this window. Please note that the position, size, selected map provider and status of **+/-**, **Zoom** and **Mini Map** buttons is being saved every time you close the window. So every time you open the window it will be opened in the state you closed it.

Since FS Logbook Editor version 1.85, it is possible to display more than one flight. The **Show on map** button will no longer be disabled when you select more than one flight in the logbook table at the *Editor* tab. Only, a slightly different map viewer dialog will be displayed:



Flights:
15.11.2015 14:26:55 : PADQ - 5BL
03.05.2014 14:14:11 : 1S2 - 3W5
06.07.2014 11:02:57 : LKPD - LKLT
17.10.2014 11:17:14 : LKLT - LKKV

Route: PADQ (Kodiak) - PAHO (Homer) - 5BL (Homer-Beluga Lake)
Flight info: Flown on: 15.11.2015 14:26:55, total flight time was: 01:32:29, 01:32:29 during day, 00:00:00 during night, 00:12:17 was instrument time, 00:00:00 was cross-country, aircraft used: Beech Duke (N157JT), flown distance: 219.9 Km

Mouse position: 78.062 / -98.7891 ,Map center: 53.022 / -81.9589 ,Zoom: 17

Picture 29: Multi-flight Map Viewer dialog

The list on the left contains all flights selected before clicking the **Show on map** button. When the window opens, the view will adjust to display all selected flights. This window works almost the same like in the case of displaying a single flight, with just a minor differences. If you click on the flight in the list, it will be highlighted and centered on the map at current zoom level (zoom level will not be changed). If you double click at the flight in the list, the zoom will be changed to best fit the selected flight. The **Zoom to fit** button works as in the case of single flight if you left click on it (zoom to best fit selected flight, so, the same like double click on flight in the list), but if you right click on it, the map zoom will reset to best fit all flights.

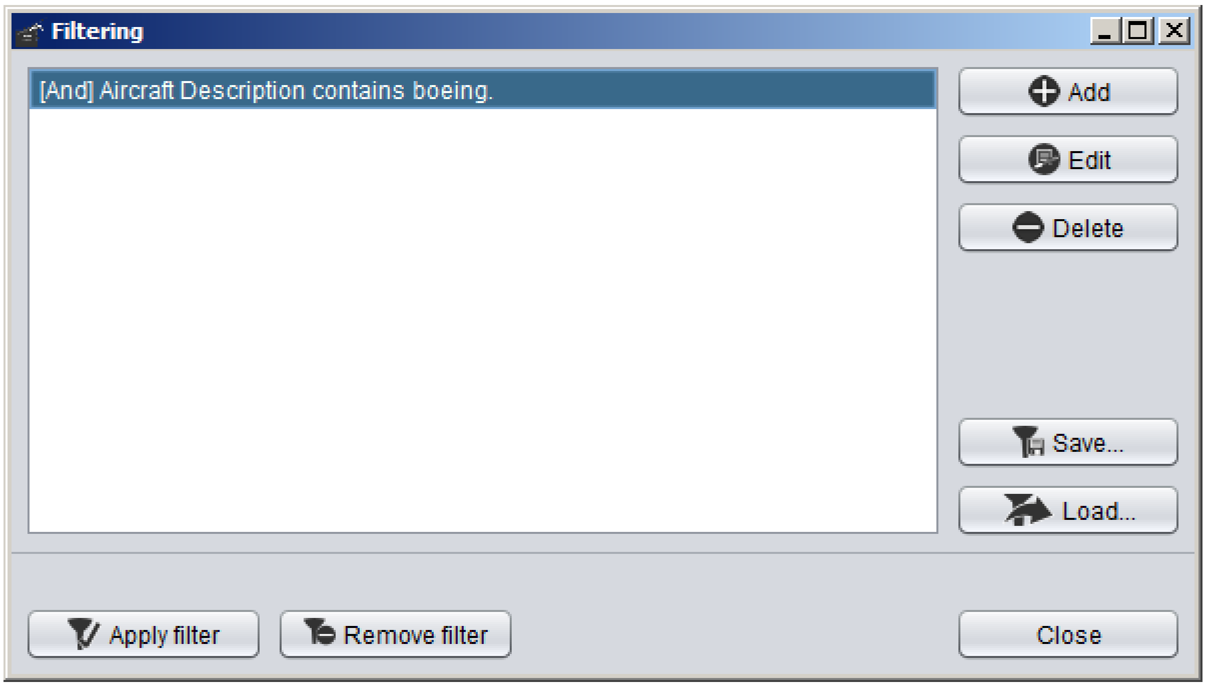
There are also new options at the ***Map Viewer*** section of ***Settings*** that allows you to define the look of the line of selected flight.

3.10. Filtering logbook records

In FS Logbook Editor version 1.50 two new buttons were added to the ***Editor*** tab. These buttons are **Set filter** and **Remove filter** button. See ***Picture 13: Main program tab to work with logbook records*** to see where they are. These two buttons allows you to filter what logbook records are currently shown.

Let's start with the easier button, which is the **Remove filter** button. If there is any filter active, clicking this button will cancel the filter and the table showing logbook records will again show all logbook records. When no filter is active, nothing will happen (actually it does remove the filter also, but since there was no one, it has no effect).

The **Set filter** is little more complex. When you click it, the following window will be shown:

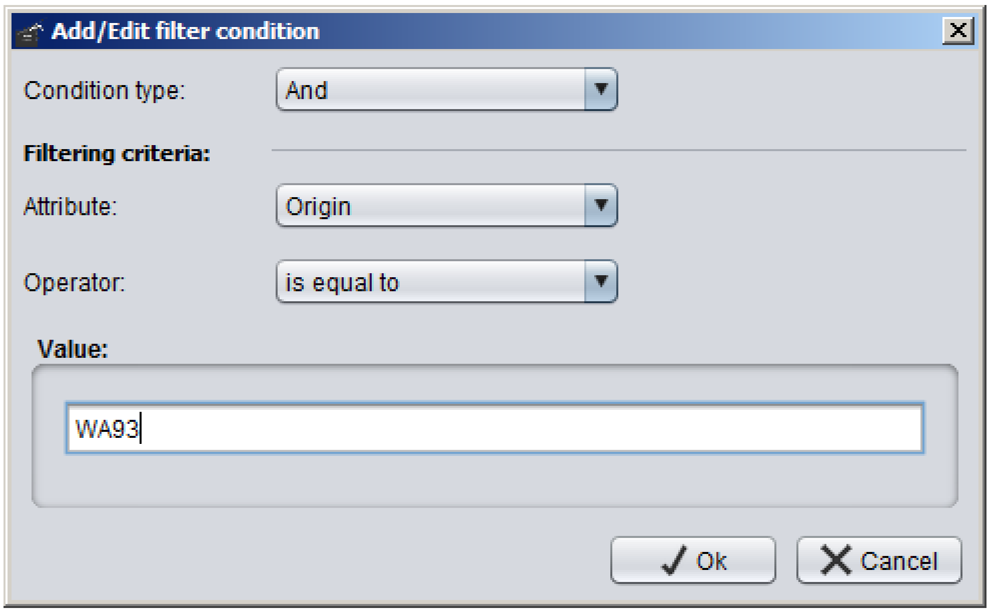


[And] Aircraft Description contains boeing.

Picture 30: Window to define filtering conditions

In this window, you can define pretty complex filtering conditions. The one the image says to show only records whose aircraft description contains the word boeing. There is a couple of buttons on this window, let's describe what they do.

The **Add** button will open another dialog:


Add/Edit filter condition

Condition type: And

Filtering criteria:

Attribute: Origin

Operator: is equal to

Value:

WA93

Picture 31: Add/Edit filtering condition dialog

On this dialog you actually define or edit the filtering condition. The first thing to choose is **Condition Type**. There are only two types – **And** and **Or**. This comes in play when you have more than one condition defined and it tells FS Logbook Editor how to combine them. Like if we would add this one the picture above to the one shown on previous picture, it would mean that only records which have the word boeing in aircraft description and WA93 as origin airport should be shown when applying this filter (and that should be none – WA93 is too small for Boeing :)).

The next thing to choose is an **Attribute** which this condition will operate with. You can choose from many attribute shown in the table view, like **Origin**, **Destination** and even the ones like **Flight distance**. Note on **Flight distance** – there is no unit to choose, it always compares with actually selected flight distance unit in settings. See the **Distance unit** settings at ***General*** in ***Settings***. Any time you change the attribute, the list of available **operators** will change and also the **Value** input will change. Not all attributes supports the same operators and values. Some expects date values, some text etc. But it is nothing complex, just try it and see.

Third option to choose is an **Operator**. It defines the relationship between attribute and **value**. There are operators like **is equal to**, **is not equal to**, **is greater than** etc. I think they are all self explanatory.

The last option is the **Value**. Enter your desired value that should be compared with **attribute** by **operator**.

To summarize the example on the picture above, it says that this filtering condition should be combined in an **And** manner with any other conditions. And the condition will match logbook records that have WA93 as origin airport.

Click the **Ok** button to close the dialog and add the filtering condition to the list on Filtering window. Clicking the **Cancel** button will close dialog without adding the filtering condition to the list on Filtering window.

Back to the Filtering window. The **Edit** button will display the same dialog as the **Add** button, only filtering condition selected in the list of conditions will be pre-selected in the dialog allowing you to edit it. To edit the condition, continue like if adding the new one.

The next button is the **Delete** button. It will delete selected filtering condition(s) from the list. It will not reapply the filter, if there is any. You have to do this yourself using the **Apply filter** button.

The **Save...** button allows you to save currently defined filtering conditions to xml file, so you can load them quickly next time you need them. It will open standard Save dialog, save the file where you wish (preferably where you will find it later).

The **Load...** button will open standard Open file dialog, allowing you to select any previously saved filtering conditions xml file. If that file is successfully loaded, the loaded filtering conditions will replace the ones actually in the list. There is no warning of saving current conditions, so think about saving them before loading any other. It will also will not apply the conditions, you have to do it yourself using the **Apply filter** button.

The **Apply filter** button will apply the currently defined filtering conditions to the logbook table at ***Editor*** tab. So the table should get immediately filtered.

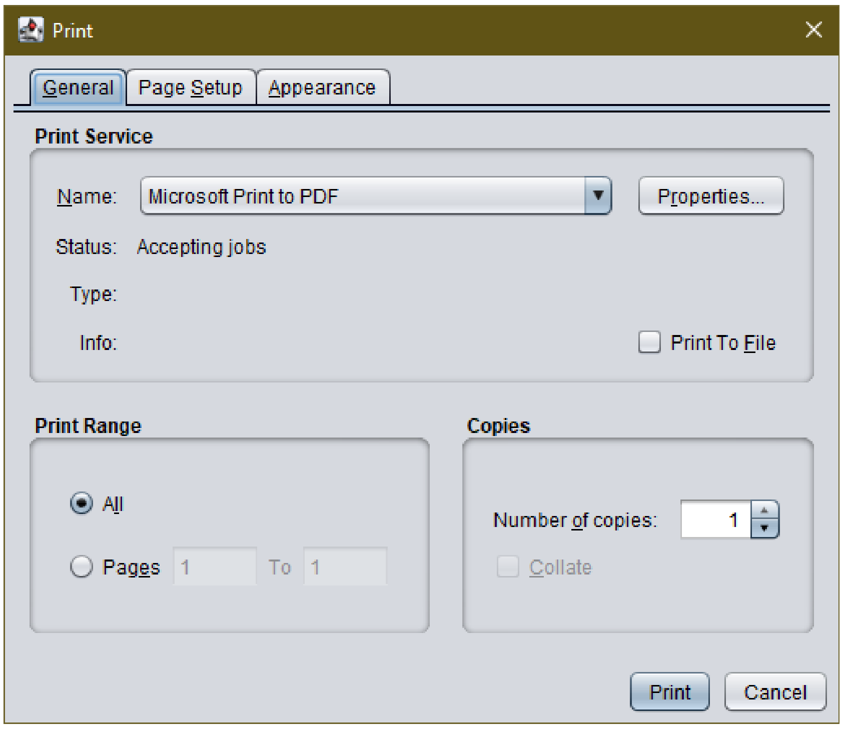
Remove filter button will cancel the filter and the logbook table will again show all the records.

The **Close** button will close the window, again without applying the filter.

To summarize, the filtering windows serves to manage filtering conditions. If you want to apply them (the filter), you have to do it using the **Apply filter** button. To remove the filter, click the **Remove filter** button, which is next to the **Apply filter** button and on the right side of the logbook table at ***Editor*** tab.

3.11. Printing logbook table

FS Logbook Editor version 1.60 added support for basic print of logbook table. That is, the table will get printed as seen in the program. You can print logbook table by clicking the **Print logbook table…** command in the **File** menu. Depending of what options you defined at **Printing options** of ***General*** section of ***Settings***, the system print dialog may show:



Print Service

Name: Microsoft Print to PDF

Status: Accepting jobs

Type:

Info:

☐ Print To File

Print Range

● All

○ Pages 1 To 1

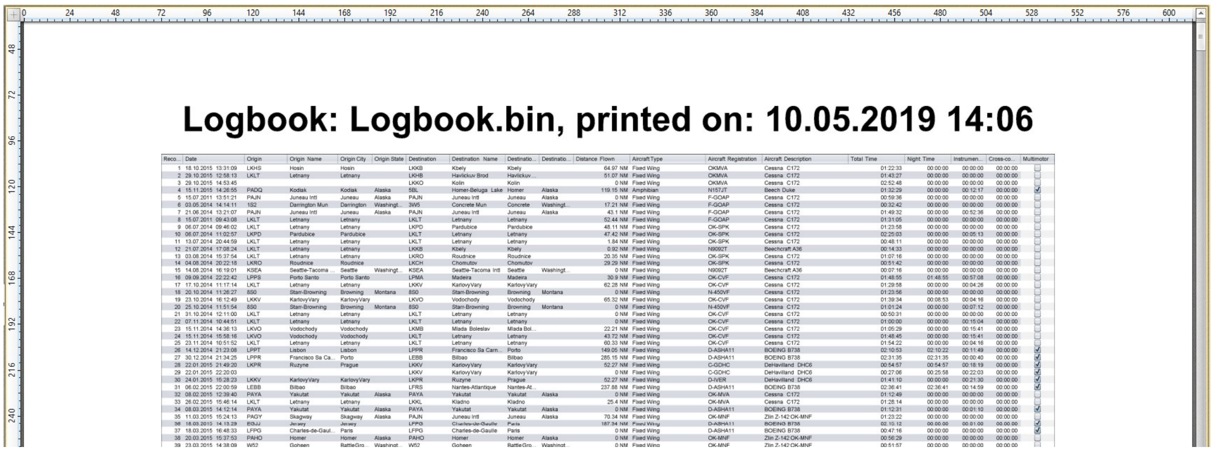
Copies

Number of copies: 1

☐ Collate

Picture 32: System print dialog

In the system print dialog you can select printer, define paper and other options, just as you are used when printing from any other program. If the option to display this dialog is turned off in the Settings, the table will be printed on default printer with default settings (which may vary for each computer). The printed table may look like the one below, depending on options selected in the print dialog and in the **Print options**.

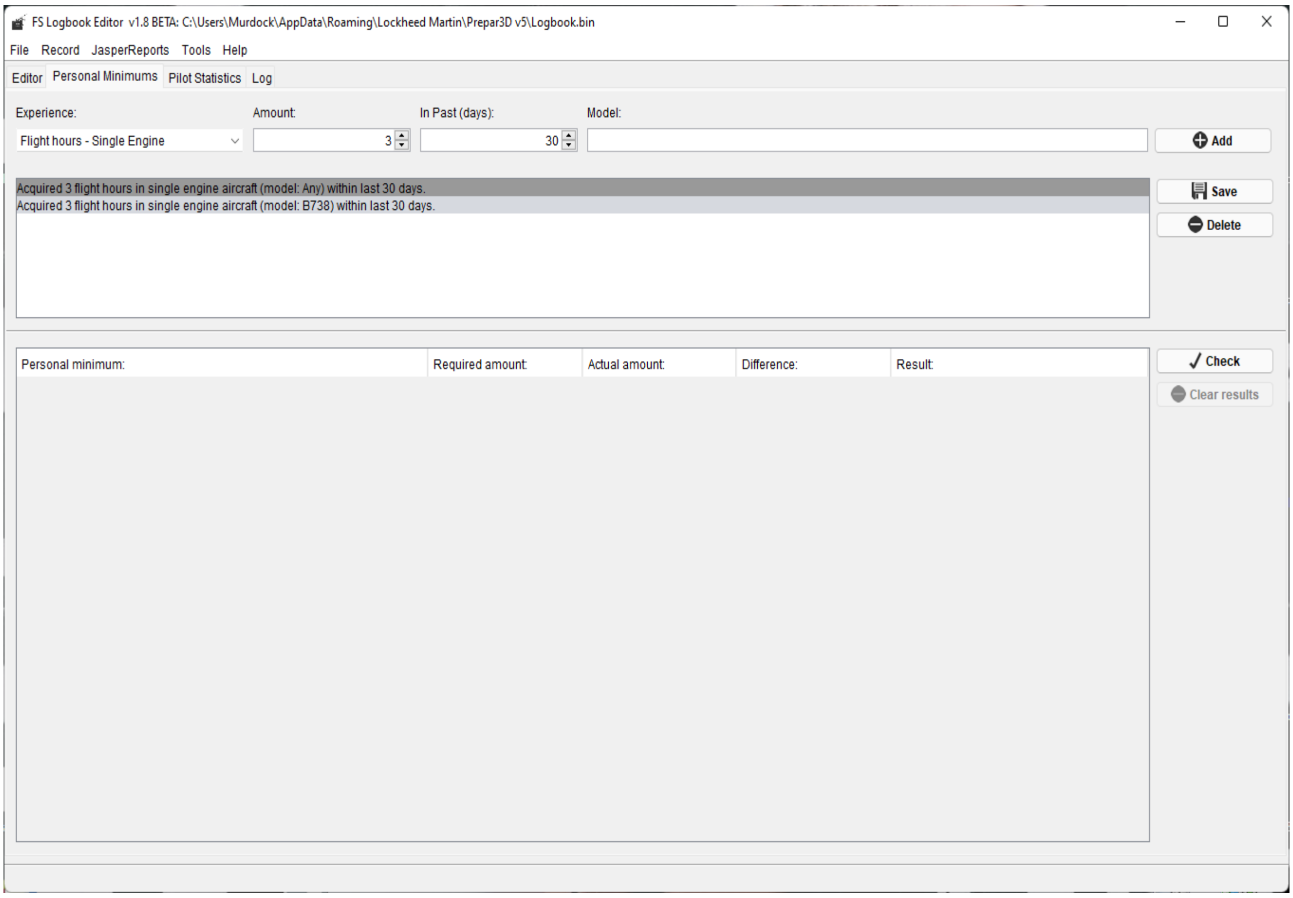


Logbook: Logbook.bin, printed on: 10.05.2019 14:06

Reco...	Date	Origin	Origin Name	Origin City	Origin State	Destination	Destination Name	Destinatio...	Destinatio...	Distance Flown	Aircraft Type	Aircraft Registration	Aircraft Description	Total Time	Night Time	Instrumen...	Cross-co...	Multimotor
1	18.10.2015 13:31:09	LKHS	Hosin	Hosin		LKKB	Kbely	Kbely		64.97 NM	Fixed Wing	OKMVA	Cessna C172	01:22:33	00:00:00	00:00:00	00:00:00	☐
2	29.10.2015 12:58:13	LKLT	Letnany	Letnany		LKHB	Havlickuv Brod	Havlickuv...		51.07 NM	Fixed Wing	OKMVA	Cessna C172	01:43:27	00:00:00	00:00:00	00:00:00	☐
3	29.10.2015 14:53:45					LKKO	Kolin	Kolin		0 NM	Fixed Wing	OKMVA	Cessna C172	02:52:48	00:00:00	00:00:00	00:00:00	☐
4	15.11.2015 14:26:55	PADQ	Kodiak	Kodiak	Alaska	5BL	Homer-Beluga Lake	Homer	Alaska	119.15 NM	Amphibian	N157JT	Beech Duke	01:32:29	00:00:00	00:12:17	00:00:00	✓
5	15.07.2011 13:51:21	PAJN	Juneau Intl	Juneau	Alaska	PAJN	Juneau Intl	Juneau	Alaska	0 NM	Fixed Wing	F-GOAP	Cessna C172	00:59:36	00:00:00	00:00:00	00:00:00	☐
6	03.05.2014 14:14:11	1S2	Darrington Mun	Darrington	Washingt...	3W5	Concrete Mun	Concrete	Washingt...	17.21 NM	Fixed Wing	F-GOAP	Cessna C172	00:32:42	00:00:00	00:00:00	00:00:00	☐
7	21.06.2014 13:21:07	PAJN	Juneau Intl	Juneau	Alaska	PAJN	Juneau Intl	Juneau	Alaska	43.1 NM	Fixed Wing	F-GOAP	Cessna C172	01:49:32	00:00:00	00:52:36	00:00:00	☐
8	15.07.2011 09:43:08	LKLT	Letnany	Letnany		LKLT	Letnany	Letnany		52.44 NM	Fixed Wing	F-GOAP	Cessna C172	01:31:05	00:00:00	00:00:00	00:00:00	☐
9	06.07.2014 09:46:02	LKLT	Letnany	Letnany		LKPD	Pardubice	Pardubice		48.11 NM	Fixed Wing	OK-SPK	Cessna C172	01:23:58	00:00:00	00:00:00	00:00:00	☐
10	06.07.2014 11:02:57	LKPD	Pardubice	Pardubice		LKLT	Letnany	Letnany		47.42 NM	Fixed Wing	OK-SPK	Cessna C172	02:25:03	00:00:00	00:05:13	00:00:00	☐
11	13.07.2014 20:44:59	LKLT	Letnany	Letnany		LKLT	Letnany	Letnany		1.84 NM	Fixed Wing	OK-SPK	Cessna C172	00:48:11	00:00:00	00:00:00	00:00:00	☐
12	21.07.2014 17:08:24	LKLT	Letnany	Letnany		LKKB	Kbely	Kbely		0.92 NM	Fixed Wing	N9092T	Beechcraft A36	00:14:33	00:00:00	00:00:00	00:00:00	☐
13	03.08.2014 15:37:54	LKLT	Letnany	Letnany		LKRO	Roudnice	Roudnice		20.35 NM	Fixed Wing	OK-SPK	Cessna C172	01:07:16	00:00:00	00:00:00	00:00:00	☐
14	04.08.2014 20:22:18	LKRO	Roudnice	Roudnice		LKCH	Chomutov	Chomutov		29.29 NM	Fixed Wing	OK-SPK	Cessna C172	00:51:42	00:00:00	00:00:00	00:00:00	☐
15	14.08.2014 16:19:01	KSEA	Seattle-Tacoma ...	Seattle	Washingt...	KSEA	Seattle-Tacoma Intl	Seattle	Washingt...	0 NM	Fixed Wing	N9092T	Beechcraft A36	00:07:16	00:00:00	00:00:00	00:00:00	☐
16	09.09.2014 22:22:42	LPPS	Porto Santo	Porto Santo		LPMA	Madeira	Madeira		30.9 NM	Fixed Wing	OK-CVF	Cessna C172	01:48:55	01:48:55	00:57:08	00:00:00	☐
17	17.10.2014 11:17:14	LKLT	Letnany	Letnany		LKKV	KarlovyVary	KarlovyVary		62.28 NM	Fixed Wing	OK-CVF	Cessna C172	01:29:58	00:00:00	00:04:26	00:00:00	☐
18	20.10.2014 11:26:27	8S0	Starr-Browning	Browning	Montana	8S0	Starr-Browning	Browning	Montana	0 NM	Fixed Wing	N-450VF	Cessna C172	01:23:56	00:00:00	00:00:00	00:00:00	☐
19	23.10.2014 16:12:49	LKKV	KarlovyVary	KarlovyVary		LKVO	Vodochody	Vodochody		65.32 NM	Fixed Wing	OK-CVF	Cessna C172	01:39:34	00:08:53	00:04:16	00:00:00	☐
20	25.10.2014 11:51:54	8S0	Starr-Browning	Browning	Montana	8S0	Starr-Browning	Browning	Montana	0 NM	Fixed Wing	N-450VF	Cessna C172	01:01:24	00:00:00	00:07:12	00:00:00	☐
21	31.10.2014 12:11:00	LKLT	Letnany	Letnany		LKLT	Letnany	Letnany		0 NM	Fixed Wing	OK-CVF	Cessna C172	00:50:31	00:00:00	00:00:00	00:00:00	☐
22	07.11.2014 10:44:51	LKLT	Letnany	Letnany		LKLT	Letnany	Letnany		0 NM	Fixed Wing	OK-CVF	Cessna C172	01:00:00	00:00:00	00:15:04	00:00:00	☐
23	15.11.2014 14:36:13	LKVO	Vodochody	Vodochody		LKMB	Mlada Boleslav	Mlada Bol...		22.21 NM	Fixed Wing	OK-CVF	Cessna C172	01:05:29	00:00:00	00:15:41	00:00:00	☐
24	15.11.2014 15:58:16	LKVO	Vodochody	Vodochody		LKLT	Letnany	Letnany		43.72 NM	Fixed Wing	OK-CVF	Cessna C172	01:48:45	00:00:00	00:15:41	00:00:00	☐
25	23.11.2014 10:51:52	LKLT	Letnany	Letnany		LKLT	Letnany	Letnany		60.33 NM	Fixed Wing	OK-CVF	Cessna C172	01:54:22	00:00:00	00:04:16	00:00:00	☐
26	14.12.2014 21:23:08	LPPT	Lisbon	Lisbon		LPPR	Francisco Sa Carn...	Porto		149.05 NM	Fixed Wing	D-ASHA11	BOEING B738	02:10:53	02:10:22	00:11:49	00:00:00	✓
27	30.12.2014 21:34:25	LPPR	Francisco Sa Ca...	Porto		LEBB	Bilbao	Bilbao		285.15 NM	Fixed Wing	D-ASHA11	BOEING B738	02:31:35	02:31:35	00:00:40	00:00:00	✓
28	22.01.2015 21:49:20	LKPR	Ruzyne	Prague		LKKV	KarlovyVary	KarlovyVary		52.27 NM	Fixed Wing	C-GDHC	DeHavilland DHC6	00:54:57	00:54:57	00:18:19	00:00:00	✓
29	22.01.2015 22:20:03					LKKV	KarlovyVary	KarlovyVary		0 NM	Fixed Wing	C-GDHC	DeHavilland DHC6	00:27:06	00:25:58	00:22:03	00:00:00	✓
30	24.01.2015 15:28:23	LKKV	KarlovyVary	KarlovyVary		LKPR	Ruzyne	Prague		52.27 NM	Fixed Wing	D-IVER	DeHavilland DHC6	01:41:10	00:00:00	00:21:30	00:00:00	✓
31	06.02.2015 22:00:59	LEBB	Bilbao	Bilbao		LFRS	Nantes-Atlantique	Nantes-At...		237.88 NM	Fixed Wing	D-ASHA11	BOEING B738	02:36:41	02:36:41	00:14:59	00:00:00	✓
32	08.02.2015 12:39:40	PAYA	Yakutat	Yakutat	Alaska	PAYA	Yakutat	Yakutat	Alaska	0 NM	Fixed Wing	OK-MVA	Cessna C172	01:12:49	00:00:00	00:00:00	00:00:00	☐
33	26.02.2015 15:46:14	LKLT	Letnany	Letnany		LKKL	Kladno	Kladno		25.4 NM	Fixed Wing	OK-MVA	Cessna C172	01:28:14	00:00:00	00:00:00	00:00:00	☐
34	08.03.2015 14:12:14	PAYA	Yakutat	Yakutat	Alaska	PAYA	Yakutat	Yakutat	Alaska	0 NM	Fixed Wing	D-ASHA11	BOEING B738	01:12:31	00:00:00	00:01:10	00:00:00	✓
35	11.03.2015 15:24:13	PAGY	Skagway	Skagway	Alaska	PAJN	Juneau Intl	Juneau	Alaska	70.34 NM	Fixed Wing	OK-MNF	Zlin Z-142 OK-MNF	01:23:22	00:00:00	00:00:00	00:00:00	☐
36	18.03.2015 14:13:29	EGJJ	Jersey	Jersey		LFPG	Charles-de-Gaulle	Paris		187.34 NM	Fixed Wing	D-ASHA11	BOEING B738	02:10:12	00:00:00	00:01:00	00:00:00	✓
37	18.03.2015 16:48:33	LFPG	Charles-de-Gaul...	Paris		LFPG	Charles-de-Gaulle	Paris		0 NM	Fixed Wing	D-ASHA11	BOEING B738	00:47:16	00:00:00	00:00:00	00:00:00	✓
38	20.03.2015 15:37:53	PAHO	Homer	Homer	Alaska	PAHO	Homer	Homer	Alaska	0 NM	Fixed Wing	OK-MNF	Zlin Z-142 OK-MNF	00:56:29	00:00:00	00:00:00	00:00:00	☐
39	23.03.2015 14:38:09	W52	Goheen	BattleGro...	Washingt...	W52	Goheen	BattleGro...	Washingt...	0 NM	Fixed Wing	OK-MNF	Zlin Z-142 OK-MNF	00:51:57	00:00:00	00:00:00	00:00:00	☐

Picture 33: Logbook table printed on virtual PDF printer

3.12. Personal Minimums


FS Logbook Editor v1.8 BETA: C:\Users\Murdock\AppData\Roaming\Lockheed Martin\Prepar3D v5\Logbook.bin

File Record JasperReports Tools Help

Editor | Personal Minimums | Pilot Statistics | Log

Experience: Flight hours - Single Engine

Amount: 3

In Past (days): 30

Model:

Acquired 3 flight hours in single engine aircraft (model: Any) within last 30 days.
Acquired 3 flight hours in single engine aircraft (model: B738) within last 30 days.

Personal minimum:	Required amount:	Actual amount:	Difference:	Result:

Picture 34: Define and check your personal minimums at this tab

I believe not many of PC aviators are really watching their personal minimums. It can be quite boring and time consuming (well not that much to count a couple of numbers every week). But it is always more comfortable to do it by couple of clicks. Better by one-click. Well, at this tab, you can do it by one click (OK, after you define your personal minimums, which requires more clicks).

What are personal minimums? Well, lets say these are „restrictions“that real world pilots put on themselves to stay current, proficient and safe. And most of us virtual aviators want to fly as real as possible, so we should watch ourselves too. So personal minimum is some restriction, like: I should fly at least 3 hours in the night in multi-engine airplane every month. Than, if I will not comply with that, I should for example not fly at night with multi-engine airplane at night, but should rather acquire some hours in the day to „re-familiarize“ myself with flying multi-engine iron or do that night flight with an instructor.

To briefly describe this tab, in the topmost area, there are four fields – Experience, Amount, In Past (days) and Model. These are used to create and edit personal minimums. On the right of those fields is **Add** button to add personal minimum to the list of personal minimums, that is located below the mentioned fields. On the right, there is **Save** button to save changes after editing personal minimum and **Delete** button to delete one or more personal minimum(s). Down below is table that displays the result of personal minimums check, that is performed with **Check** button to the right of the table.

3.12.1. Add personal minimum

To define personal minimum, select experience you want to check from the Experience box. This box contains possible experience types that can be dig out from logbook. No all personal minimums can be defined, like weather related for example (there is no such information in the logbook). Currently supported experience types are various flight times and amount of landings. Than select experience amount. For flight hours experience, the number represents flight hours, for landings, it's number of landings. Next, select time interval in which to look. This value is in days. Last field enables you to select model of aircraft to limit this personal minimum on.

Probably not much clear from the above paragraph. Let's do an example. Say I want to check I did fly at least 3 hours in Cessna 172 in last 30 days. So, I will select **Flight hours – Single Engine** in the **Experience** box. Than, I select **3** in **Amount** field, for 3 hours. In the **In past (days)** field I will select **30**, for 30 days. And in the **Model** text box, I will type **C172** for Cessna 172 (the model information is matched against Aircraft description, so every logbook record containing C172 in Aircraft description attribute will be used by this personal minimum). Next, I click **Add** to add personal minimum in the list of personal minimums.

3.12.2. Change personal minimum

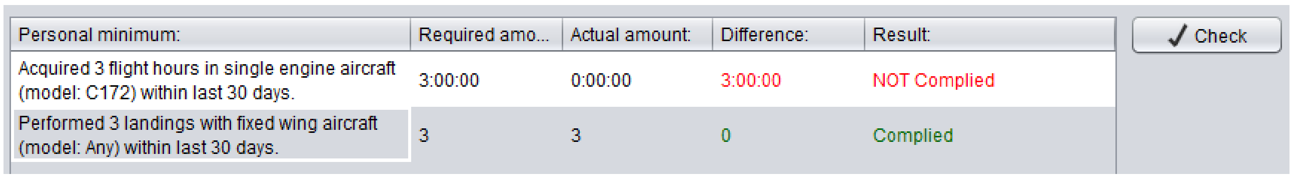
To change personal minimum, select personal minimum you want to change and set new values as if you were creating new personal minimum (using the fields above the list). When done, click **Save** and the changes made will be applied to selected personal minimum.

3.12.3. Delete personal minimum

To delete personal minimum(s), select the personal minimum(s) you want to delete and click **Delete**.

3.12.4. Checking personal minimums

To check personal minimums, click the **Check** button. This button is disabled if no personal minimums are defined or if no logbook is loaded or is empty. For each personal minimum, there will be one record in the results table. The rows show personal minimum description, the amount of experience required by the personal minimum for given time, the actual amount of experience gained in given time and the result. As you can expect, if the actual amount is greater than or equal to required amount, you complained with your personal minimum. The other case is, you know already...not complained. Both of the possibilities are shown on following picture:

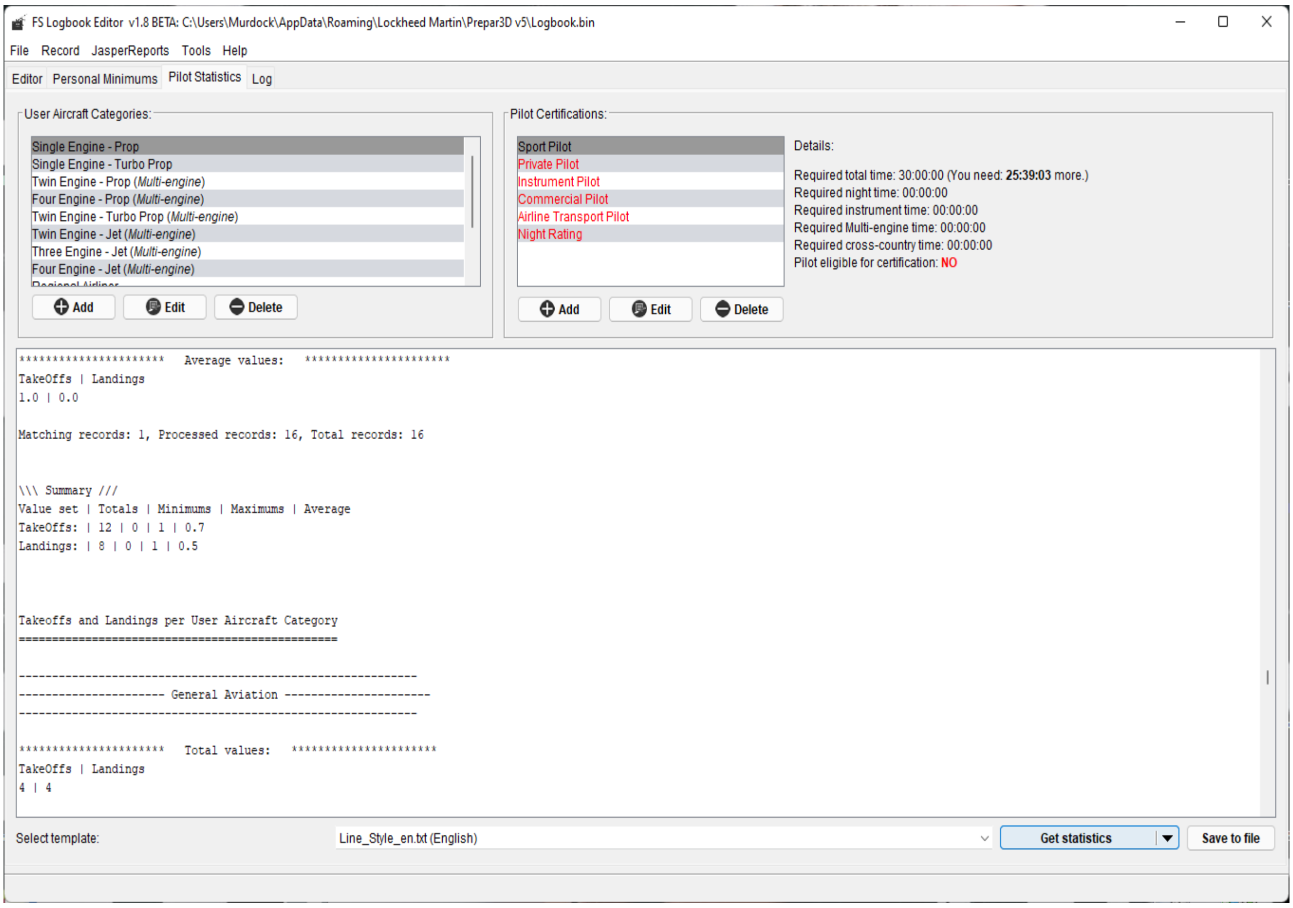


Personal minimum:	Required amo...	Actual amount:	Difference:	Result:
Acquired 3 flight hours in single engine aircraft (model: C172) within last 30 days.	3:00:00	0:00:00	3:00:00	NOT Complied
Performed 3 landings with fixed wing aircraft (model: Any) within last 30 days.	3	3	0	Complied

Picture 35: The result of personal minimums check

3.13. Pilot Statistics

Pilot statistics tab is a new tab introduced in FS Logbook Editor version 1.3. It allows you to dig out some interesting numbers (hopefully) out from your logbook just by pressing one button. It should be much simpler than exporting logbook to Open Document spreadsheet and computing them all manually, which is Let's see how the pilot statistics tab looks like:


FS Logbook Editor v1.8 BETA: C:\Users\Murdock\AppData\Roaming\Lockheed Martin\Prepar3D v5\Logbook.bin
File
Record
JasperReports
Tools
Help
Editor
Personal Minimums
Pilot Statistics
Log
User Aircraft Categories:
Single Engine - Prop
Single Engine - Turbo Prop
Twin Engine - Prop (Multi-engine)
Four Engine - Prop (Multi-engine)
Twin Engine - Turbo Prop (Multi-engine)
Twin Engine - Jet (Multi-engine)
Three Engine - Jet (Multi-engine)
Four Engine - Jet (Multi-engine)
Add
Edit
Delete
Pilot Certifications:
Sport Pilot
Private Pilot
Instrument Pilot
Commercial Pilot
Airline Transport Pilot
Night Rating
Add
Edit
Delete
Select template:
Line_Style_en.txt (English)
Get statistics
Save to file

Details:

Required total time: 30:00:00 (You need: **25:39:03** more.)
Required night time: 00:00:00
Required instrument time: 00:00:00
Required Multi-engine time: 00:00:00
Required cross-country time: 00:00:00
Pilot eligible for certification: NO

```
*********************   Average values:   *********************
TakeOffs | Landings
1.0 | 0.0

Matching records: 1, Processed records: 16, Total records: 16


\\\ Summary ///
Value set | Totals | Minimums | Maximums | Average
TakeOffs: | 12 | 0 | 1 | 0.7
Landings: | 8 | 0 | 1 | 0.5



Takeoffs and Landings per User Aircraft Category
================================================


------------------------------------------------------------
--------------------- General Aviation ---------------------
------------------------------------------------------------


*********************   Total values:   *********************
TakeOffs | Landings
4 | 4
```

Picture 36: Pilot statistics tab

Most part of the tab is occupied by text area that is used to display computed statistics. Below this text area is template selection box and two buttons – The split button, which actually contains the **Get statistics** button and **Open in editor...** button. The second button is **Save to file** button. The **Get statistics** button will compute the statistics and displays the result in the mentioned text area. The statistics result will be formatted using selected template. There are several default templates:

- **Line_Style_en.txt** – this template formats statistics as report. For example for every aircraft in flight times per aircraft statistics there will be a "section" holding total, minimum, maximum and average values.
- **Table_Style_en.txt** – this will generate standard table formatted statistics.
- **Flight_Times_CSV_en.txt** – this will generated comma separated values formatted flight times statistics to save the result as CSV file.
- **Takeoffs_and_Landings_CSV_en.txt** – this will generated comma separated values formatted takeoffs and landings statistics to save the result as CSV file.

There are also the same templates in Czech language.

The template also determines which of the supported statistics will be computed. FS Logbook

Editor only computed statistics that can be displayed using selected template. For the list of all supported statistics types, see: ***Supported statistics types***

If you to know more about pilot statistics templates, see: ***Writing custom pilot statistics template***

The **Open in editor...** button will open selected template in the ***Little More Than Dumb Text Editor***, which was added in FS Logbook Editor version 1.8. It allows for simple editing of the templates, as it supports highlighting of keywords, auto-completion and showing help.

The **Save to file** button, surprisingly, will open a save file dialog and to let you save the statistics result to the file of your choice. The saved file does not have to have .txt extension. It is purely up to you in which file and with what extension you save the statistics. For example, if you use one CSV templates, you can save the statistics as CSV file and open it in spreadsheet editor like Open Office or Libre Office.

Since FS Logbook Editor version 1.80, the save dialog also allows to save the pilot statistics as PDF document (.pdf), ODF Text document (.odt) and Word document – 2007 and newer (.docx).

Since FS Logbook Editor version 1.50, which added option to filter logbook records at the main view at ***Editor*** tab (See ***Filtering logbook records***), the **Get statistics will use just the filtered records, if there is filter active**. To get statistics for the whole logbook, there must be no filter active.

However, there is not just a button to calculate the statistics. In the upper part of the tab, there are two lists. The list on the left holds user aircraft categories and the list on the right holds pilot certifications.

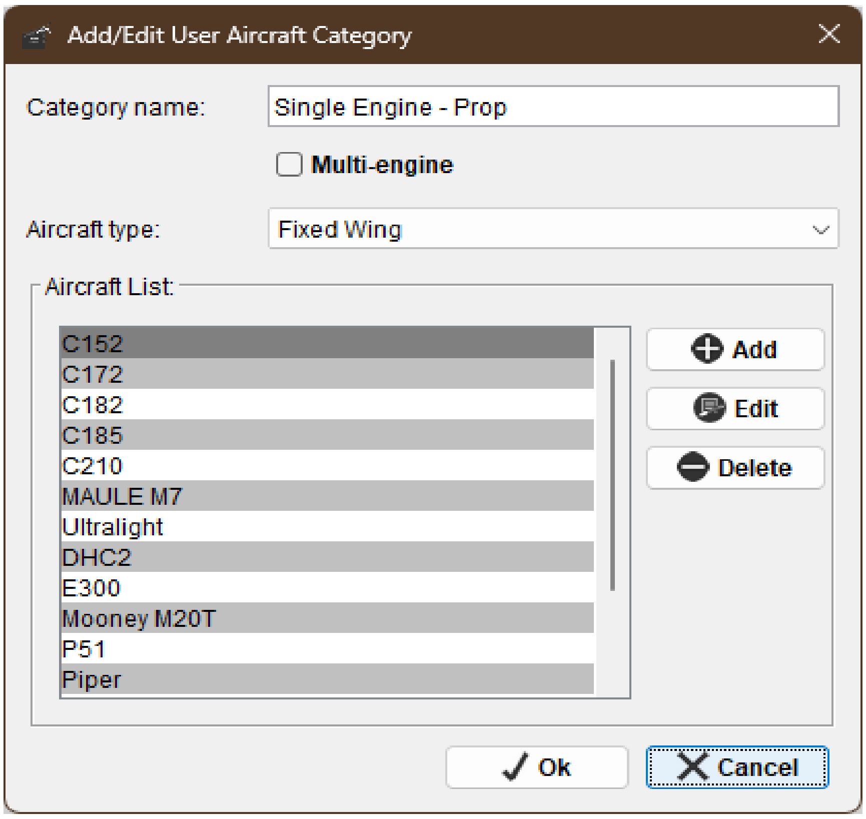
3.13.1. User aircraft categories

In this list you can define your own aircraft categories which will be used to categorize aircraft when computing pilot statistics. There is a standard set of buttons:

- **Add** – adds new user aircraft category.
- **Edit** – edits the selected user aircraft category.
- **Delete** – removes the selected user aircraft category/categories.

If you click to add or edit user aircraft category, a following dialog will open:

<content follows on next page>


Add/Edit User Aircraft Category

Category name: Single Engine - Prop

Multi-engine

Aircraft type: Fixed Wing

Aircraft List:

C152
C172
C182
C185
C210
MAULE M7
Ultralight
DHC2
E300
Mooney M20T
P51
Piper

Picture 37: User aircraft category add/edit dialog

It is a simple dialog where you can select user aircraft category name and add matching aircraft names to the list of aircraft that should fall under this category. Only partial aircraft name is required here. When pilot statistics categorizes aircraft into user aircraft category, it will try to match any aircraft from respective user aircraft category against aircraft description for logbook record being processed. If you look for example into "Single Engine – Prop" category, you will see that first aircraft in the list is "C172". Then, when FS Logbook Editor computes for example flight times per user aircraft category statistics, and finds logbook record with aircraft description "Cessna C172", it will match the "Single Engine – Prop" category. A single logbook record thus can match more than one user aircraft category.

You can add/edit/remove aircraft in the aircraft list using respective buttons.

The **multi-engine** checkbox was added in FS Logbook Editor version 1.8. It allows you to mark the user aircraft category as containing multi-engine aircraft. This was added because of new functionality of importing flights from FsHub (See ***Importing flights from FSHub***) and Microsoft Flight Simulator 2020 (See ***Importing flights from Microsoft Flight Simulator 2020***). Neither of these has the information of whether the aircraft used is multi-engine or not, so to not completely loose this information, FS Logbook Editor can match aircraft names against user aircraft categories during the import to find out whether they are multi-engine or not. Assuming of course that user aircraft categories are correctly marked and this behavior is enabled for the import (which it is by default). Since FS Logbook Editor version 1.85, this property can be also set for each aircraft within the category individually, and thus overriding the value of the category.

The **Aircraft type** combo box was added in FS Logbook Editor version 1.85. The reason this was added is the same as with multi-engine checkbox. The flights imported from FsHub (See ***Importing flights from FSHub***) and Microsoft Flight Simulator 2020 (See ***Importing flights from Microsoft Flight Simulator 2020***) does not contain any information about aircraft type used for the flight so the value in FS Logbook Editor would always be the default – Fixed Wing. Using this option, FS

Logbook Editor can match aircraft names against user aircraft categories during the import to try to determine aircraft type that FS Logbook Editor supports. Assuming of course that aircraft type is correctly set for user aircraft categories and this behavior is enabled for the import (which it is by default). This property can be also set for each aircraft within the category individually, and thus overriding the value of the category.

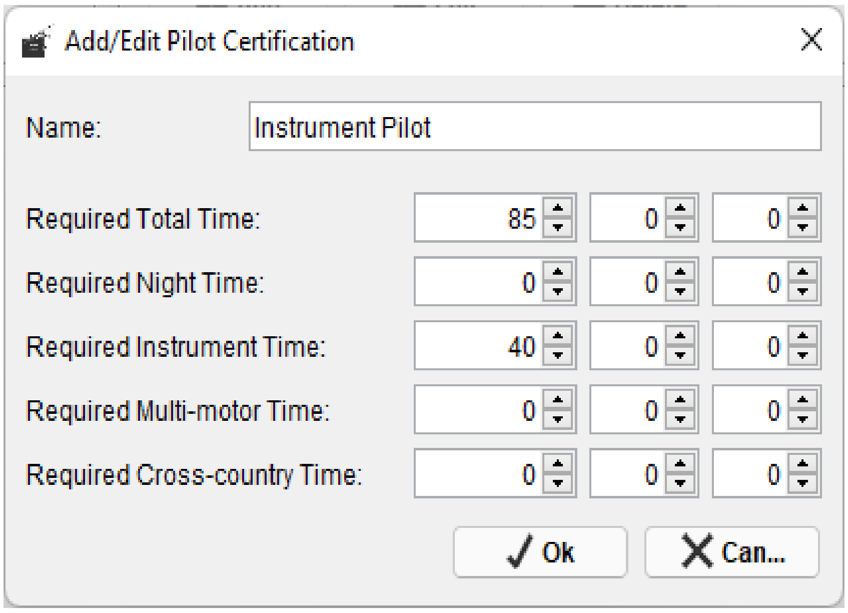
3.13.2. Pilot certifications

The real world aviation is all about papers such as most other business. You need a license (the paper) which tells that you can fly and what you can fly. It is called pilot certificate (the process to get it is pilot certification) and of course it is more complex than I just have written. Mostly, the pilot certification is based on flight time you have earned. The more you have, the better toys you can fly. But is is always not just about flight time, such things as number of engines of the aircraft or aircraft type also matters. In FS Logbook Editor however, this is simplified only to flight time. Basic set of pilot certifications is included so you can quickly see what certification you are eligible for based on your logged flight time.

You customize the list of pilot certifications using standard set of buttons:

- **Add** – adds new pilot certification.
- **Edit** – edits the selected pilot certification.
- **Delete** – removes the selected pilot certification/certifications.

If you click to add or edit user pilot certification, a following dialog will open:



Name: Instrument Pilot

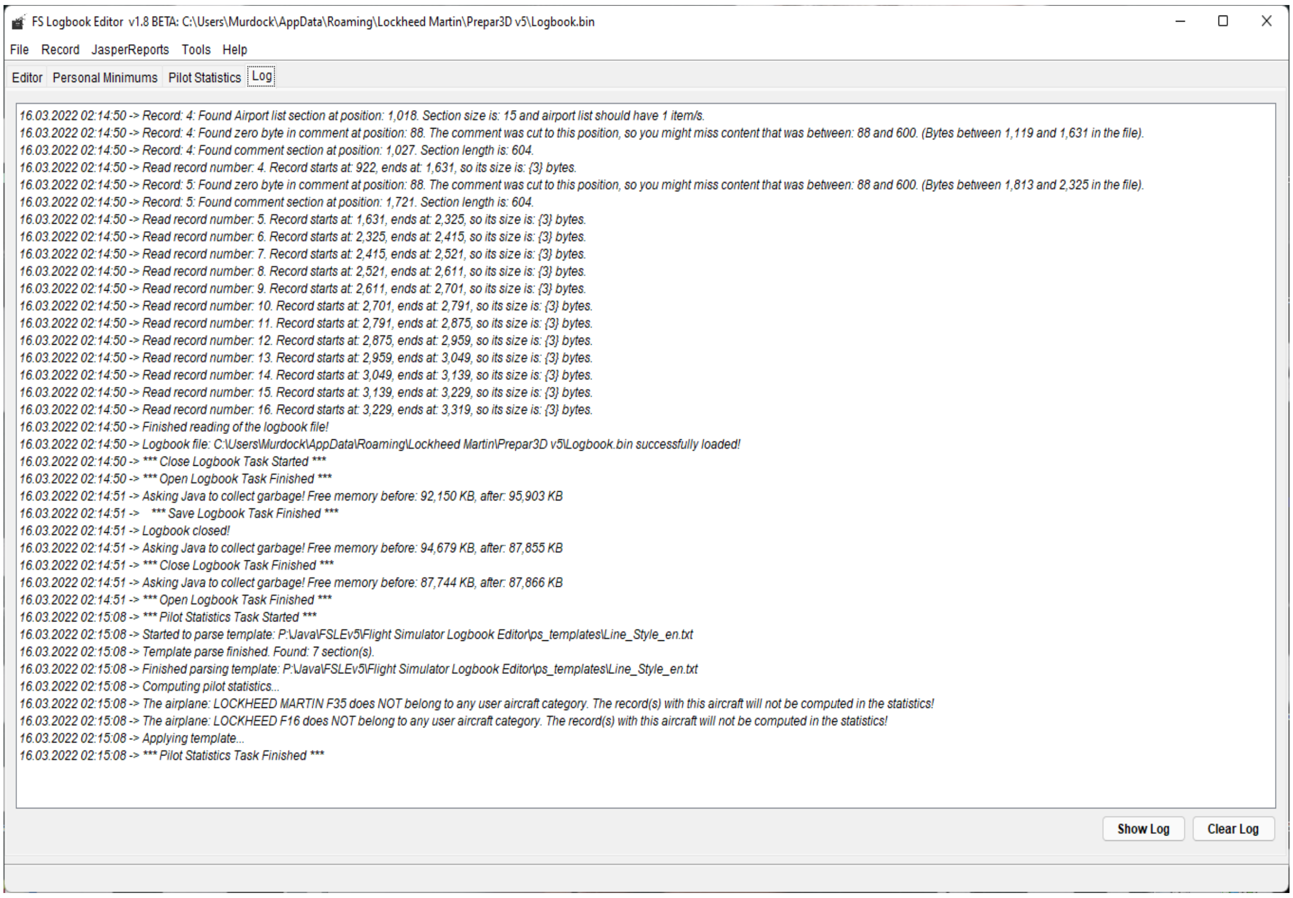
Required Total Time:	85	0	0
Required Night Time:	0	0	0
Required Instrument Time:	40	0	0
Required Multi-motor Time:	0	0	0
Required Cross-country Time:	0	0	0

Picture 38: Pilot certification add/edit dialog

In the Add/Edit Pilot Certification dialog you can define flight times required for respective pilot certifications.

The details section to the right of the pilot certifications list shows the required flight times for selected pilot certification and the difference between required flight time and your logged flight time.

Please note that some flight times in the predefined pilot certifications does not correspond exactly with the real world criteria. For example, Instrument pilot certification has about 84 flight hours of total time required and 40 hours of instrument time. In the real world, to get Instrument rating, you need these 40 hours of instrument flight time. But you cannot do that before you get Private Pilot license, for which you need 45 flight hours in minimum. So, to be sure you have Private Pilot License in simulator, the predefined Instrument pilot license required you to have 85 hours of total flight time (45 of PPL + another 40 flying instrument training) + 40 of actual instrument flight time.


FS Logbook Editor v1.8 BETA: C:\Users\Murdock\AppData\Roaming\Lockheed Martin\Prepar3D v5\Logbook.bin
File
Record
JasperReports
Tools
Help
Editor
Personal Minimums
Pilot Statistics
Log
Show Log
Clear Log

```
16.03.2022 02:14:50 -> Record: 4: Found Airport list section at position: 1,018. Section size is: 15 and airport list should have 1 item/s.
16.03.2022 02:14:50 -> Record: 4: Found zero byte in comment at position: 88. The comment was cut to this position, so you might miss content that was between: 88 and 600. (Bytes between 1,119 and 1,631 in the file).
16.03.2022 02:14:50 -> Record: 4: Found comment section at position: 1,027. Section length is: 604.
16.03.2022 02:14:50 -> Read record number: 4. Record starts at: 922, ends at: 1,631, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Record: 5: Found zero byte in comment at position: 88. The comment was cut to this position, so you might miss content that was between: 88 and 600. (Bytes between 1,813 and 2,325 in the file).
16.03.2022 02:14:50 -> Record: 5: Found comment section at position: 1,721. Section length is: 604.
16.03.2022 02:14:50 -> Read record number: 5. Record starts at: 1,631, ends at: 2,325, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 6. Record starts at: 2,325, ends at: 2,415, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 7. Record starts at: 2,415, ends at: 2,521, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 8. Record starts at: 2,521, ends at: 2,611, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 9. Record starts at: 2,611, ends at: 2,701, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 10. Record starts at: 2,701, ends at: 2,791, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 11. Record starts at: 2,791, ends at: 2,875, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 12. Record starts at: 2,875, ends at: 2,959, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 13. Record starts at: 2,959, ends at: 3,049, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 14. Record starts at: 3,049, ends at: 3,139, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 15. Record starts at: 3,139, ends at: 3,229, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Read record number: 16. Record starts at: 3,229, ends at: 3,319, so its size is: {3} bytes.
16.03.2022 02:14:50 -> Finished reading of the logbook file!
16.03.2022 02:14:50 -> Logbook file: C:\Users\Murdock\AppData\Roaming\Lockheed Martin\Prepar3D v5\Logbook.bin successfully loaded!
16.03.2022 02:14:50 -> *** Close Logbook Task Started ***
16.03.2022 02:14:50 -> *** Open Logbook Task Finished ***
16.03.2022 02:14:51 -> Asking Java to collect garbage! Free memory before: 92,150 KB, after: 95,903 KB
16.03.2022 02:14:51 ->    *** Save Logbook Task Finished ***
16.03.2022 02:14:51 -> Logbook closed!
16.03.2022 02:14:51 -> Asking Java to collect garbage! Free memory before: 94,679 KB, after: 87,855 KB
16.03.2022 02:14:51 -> *** Close Logbook Task Finished ***
16.03.2022 02:14:51 -> Asking Java to collect garbage! Free memory before: 87,744 KB, after: 87,866 KB
16.03.2022 02:14:51 -> *** Open Logbook Task Finished ***
16.03.2022 02:15:08 -> *** Pilot Statistics Task Started ***
16.03.2022 02:15:08 -> Started to parse template: P:\Java\FSLEv5\Flight Simulator Logbook Editor\ps_templates\Line_Style_en.txt
16.03.2022 02:15:08 -> Template parse finished. Found: 7 section(s).
16.03.2022 02:15:08 -> Finished parsing template: P:\Java\FSLEv5\Flight Simulator Logbook Editor\ps_templates\Line_Style_en.txt
16.03.2022 02:15:08 -> Computing pilot statistics...
16.03.2022 02:15:08 -> The airplane: LOCKHEED MARTIN F35 does NOT belong to any user aircraft category. The record(s) with this aircraft will not be computed in the statistics!
16.03.2022 02:15:08 -> The airplane: LOCKHEED F16 does NOT belong to any user aircraft category. The record(s) with this aircraft will not be computed in the statistics!
16.03.2022 02:15:08 -> Applying template...
16.03.2022 02:15:08 -> *** Pilot Statistics Task Finished ***
```

Picture 39: Log messages display on this tab

3.14. Log

On this tab you can read various messages informing you about what the program does or did. Those can be useful for troubleshooting. Please note that this tab only displays limited amount of messages, like last 100 messages. But every message is logged also in log file that resides in the program directory. For every application session (every run of the program), new log file is generated. The number of this files can also be limited, to keep only last 5 of them for example. Please see the Logging section in Settings for limiting the amount of log messages displayed at Log tab or amount of kept log files.

The **Show Log** button will open actual log file in the operating system default text editor, like Notepad, so you can see all messages generated since program start.

The **Clear Log** button will clear the log view on this tab. The log file is not altered, only the program view of log messages is cleared.

3.14.1. Log files

As mentioned above, all program messages are logged into log file. These log files are stored in the root of the program folder and are named like this:

yyyyMMddHHmmss.log

All those characters are replaced with numbers and represents date and time when the program was started. The name of log file may be for example this: *20131215090045.log*. And the characters means this:

- yyyy – is actual year (2013 in the example).
- MM – is actual month (12 in the example).
- dd – is actual day (15 in the example).
- HH – is actual hour (09 in the example).
- mm – is actual minute (00 in the example)
- ss – is actual second (45 in the example).

Knowing this, we can say that log file with the name: *20131215090045.log* was created for program session (execution), that started at 09:00:45 at 15.12.2013.

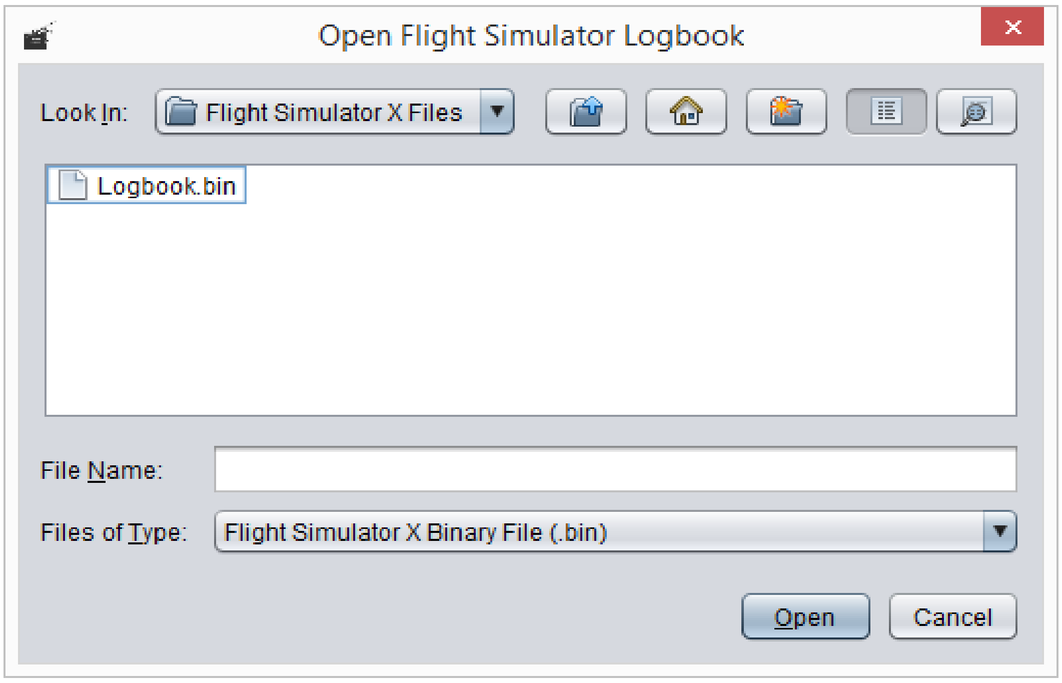
While this may be absolutely unnecessary information for you as a user, this file can be helpful for developer in the case of issues with the program.

4. Menus

4.1. File

This menu contains actions related to operations with logbook files – for opening, saving etc.

- **Open Logbook...** – This action brings up standard open dialog, where you can select logbook file to open:



Picture 40: Dialog to choose logbook file to open

By choosing file type, you can filter the files displayed. You can select from:

1. **Flight Simulator X Binary File (.bin)** – to open logbook from Flight Simulator X (and Prepar3D and Prepar3D v2/v3/v4/v5 as long as the file format does not change)
2. **Flight Simulator 2004 Logbook (.log)** – to open Flight Simulator 2004 logbook.
3. **X-Plane 10/11 Logbook (.txt)** – top open X-Plane 10/11 logbook file.
4. **FS Logbook Editor XML file (.fslex)** – to open FS Logbook Editor native XML logbook. This file format can store all information supported by FS Logbook Editor.
5. **Excel (2007 and newer) (.xlsx)** – to open logbook stored in Excel spreadsheet. This was added in FS Logbook Editor version 1.8. Read more about it in the ***Loading logbook from Excel spreadsheet*** chapter.
6. **Excel (97-2003) (.xls)** - to open logbook stored in Excel (legacy format) spreadsheet. This was added in FS Logbook Editor version 1.8. Read more about it in the ***Loading logbook from Excel spreadsheet*** chapter.
7. **Comma separated values (.csv)** - to open logbook stored in CSV file. This was added in FS Logbook Editor version 1.8. Read more about it in the ***Loading logbook from CSV file*** chapter.
8. **JSON file (.json)** – to open logbook stored as JSON. This was added in FS Logbook Editor version 1.8. The JSON format is a common format to interchange data between various programs. You can read a little bit more here: ***Load / save logbook***

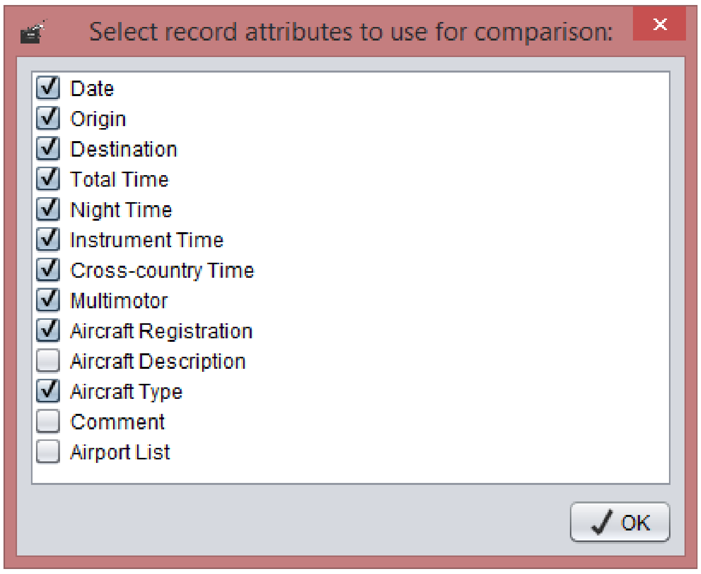
from / as JSON file.

9. **All Files** – to see all files in the directory. Please note that if you select file with extension other than *.log* or *.bin*, FS Logbook Editor will try to open that file as if it was Flight Simulator X Binary file. So, unless the file really is Flight Simulator X logbook, the results are unexpected!

Than you only need to click Open button to load selected file. The table on the *Editor* tab should fill with logbook records loaded from file, unless there is some serious error.

In the case of errors, you should be notified by message dialog with more detailed information in the *Log.*

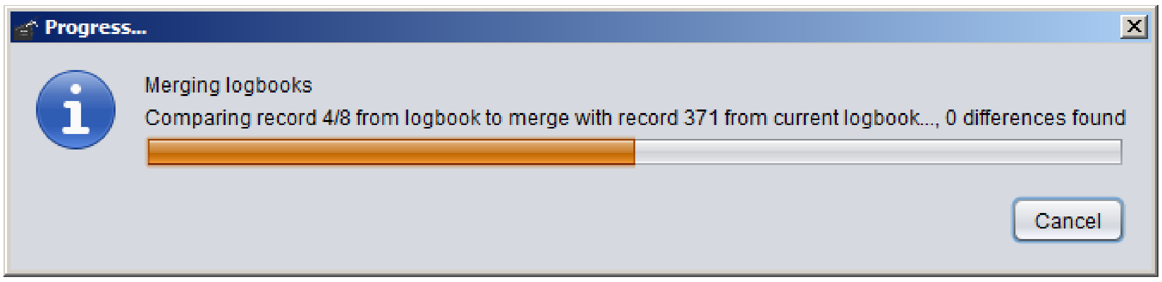
- **Open Logbook (Auto find...)** – will perform the same search as if the *Automatically load logbook file* option in *Settings* → *Load & Save* section is enabled. (Actually, this option must be enabled for this menu item to be available). Useful when you're not so sure where are the logbook files stored.
- **Recent Files...** – this menu will show last 5 logbook files opened. You can simply open the file by clicking one of the menu items. The same files will be shown also in the dialog that is displayed when *Automatically load logbook file* option is enabled. This menu will be hidden if no list of recent files is stored.
- **New Logbook** – Choosing this action, new logbook with one record will be created. If you have some logbook currently opened, you will be asked to save it. As a new logbook is automatically created with one new record, *Edit logbook record dialog* will pop-up to edit that new record.
- **Append...** – Click this action if you want to append logbook to your currently opened logbook. Same open dialog as in *New Logbook* action will show up allowing you to select logbook to append. If the selected logbook is successfully loaded, all its records will be appended to currently opened logbook. You will find appended records at the end of logbook table at *Editor* tab highlighted in color for imported records.
- **Merge With...** – Merge operation will merge currently opened logbook with another logbook. Only the records that are not already present at currently opened logbook are added from within the logbook you selected to merge with opened logbook. First of all, a standard open dialog will show up allowing you to select logbook to merge opened logbook with. If that logbook is successfully loaded, you will see following dialog:


Select record attributes to use for comparison:

- [✓] Date
- [✓] Origin
- [✓] Destination
- [✓] Total Time
- [✓] Night Time
- [✓] Instrument Time
- [✓] Cross-country Time
- [✓] Multimotor
- [✓] Aircraft Registration
- [ ] Aircraft Description
- [✓] Aircraft Type
- [ ] Comment
- [ ] Airport List

Picture 41: Dialog to choose logbook record attributes to compare records

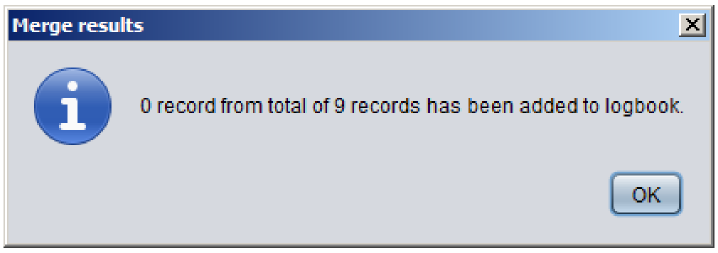
In this dialog you can select which record attributes to use to compare records – how to tell if records are equal. The records are than compared and the ones that are not contained in opened logbook are added. This operation may take considerable time depending on the size (number of records) of both logbooks. This is why you may see following dialog:



Merging logbooks
Comparing record 4/8 from logbook to merge with record 371 from current logbook..., 0 differences found

Picture 42: Dialog showing progress of logbook merge operation

You can cancel the merging operation by **Cancel** button if it takes too long. Finally, you will be given result of the merge operation, like this one:



0 record from total of 9 records has been added to logbook.

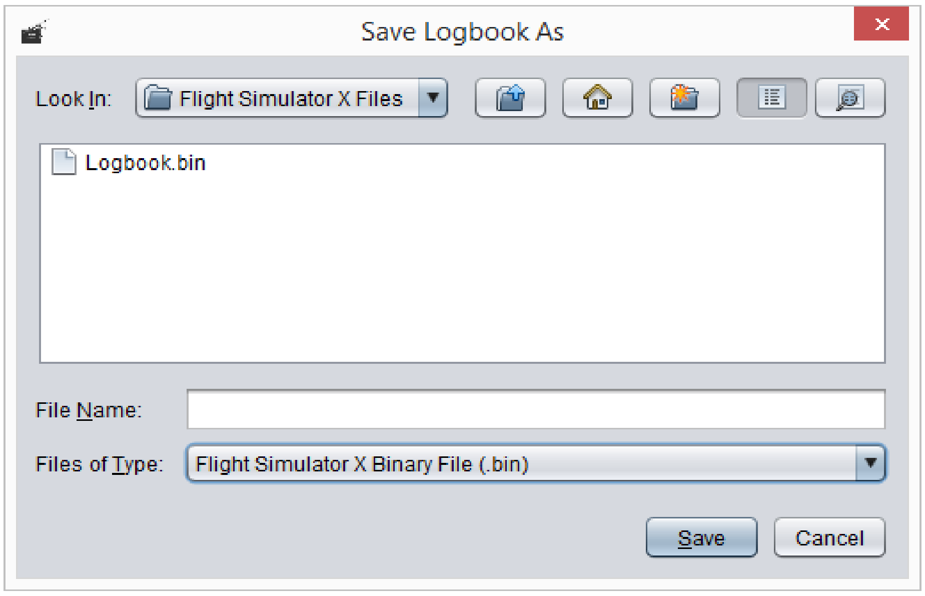
Picture 43: Dialog showing result of logbook merge operation

You will find added records at the end of logbook table at ***Editor*** tab highlighted in color for imported records.

- **Save...** – Does what it says, it saves opened logbook. If you opened logbook from file, it is saved to the same file it was opened from. If opened/edited logbook is a new logbook, the name of the file is not known, so the save file dialog is displayed, as in the case of ***Save As*** action. See its description below for more details.

  Once the file is saved using this action, the name of the file is remembered and you will not be asked it again when saving using this action (logbook will be saved directly to that file without prompt!).
- **Save As...** – If you want to save logbook to another file and/or format than the one it was loaded from, use this action. Standard save dialog will pop-up, asking you to select file to save the logbook to:

<content follows on next page>


Save Logbook As

Look In: Flight Simulator X Files

Logbook.bin

File Name:

Files of Type: Flight Simulator X Binary File (.bin)

Picture 44: Dialog to choose file to save logbook to

You can select file name and type (format) of the file. Currently, these options for file type (format) are supported:

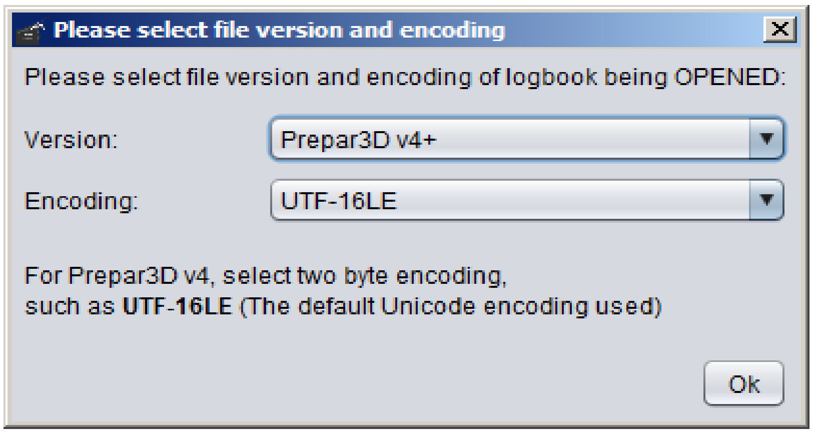
1. **Flight Simulator X Binary file (.bin)** – saves the logbook as Flight Simulator X (as long as Prepar3D and Prepar3D v2 uses the same logbook file format, you can use this file type for this simulators too) binary logbook.

   A Note on Prepar3D v2.2+: Prepar3D v2.2 and above saves comments slightly differently. As a result, a comment saved to logbook by FSX is NOT visible in Prepar3D v2.2+ and not surprisingly, comment saved to logbook file by Preapr3D v2.2+ is NOT visible in FSX logbook view. All other information should load correctly by both simulators and thus no modifications are needed to interchange the logbook files between simulators. I am not sure whether the difference in saving comments in both simulators is intentional or whether it is a bug. However, FS Logbook Editor version 1.15+ will ask you for which simulator is the logbook targeted when you will save to binary logbook file format. The dialog that will show up was extended in version 1.42, and now looks like the one on ***Picture 45: Dialog to select target simulator***.

   A Note on Prepar3D v4+: - Prepar3D v4 and above changed the encoding of texts in the logbook to 16bit Unicode. This resulted in redesign of the dialog that was shown before saving FSX/P3D logbook file. It was extended and is now displayed also while by saving the logbook. You can see the dialog on ***Picture 45: Dialog to select target simulator***. Now you need to select version of the simulator the file is targeted for and also file encoding. The encoding was moved there so you do not have to go to Settings and change it every time you want to load Prepar3D v4 logbook and them some of the previous version. The dialog your remember selected encoding for each supported simulator version, for open and save action separately. The last used encoding will be pre-selected every time you change simulator version.

   Unfortunately, the very first release of Prepar3D v4 had a bug and airport codes were stored incomplete – there were only 2 chars instead of 4. Hotfix 1 fixed this, but only for the new records, not the ones already stored by the previous version of Prepar3D.

This is why loading some Prepar3D v4 logbook files, you may see some records with incomplete airport codes and marked as corrupted. Just fix the missing data and save the logbook, it will be saved correctly.


Please select file version and encoding

Please select file version and encoding of logbook being OPENED:

Version: Prepar3D v4+

Encoding: UTF-16LE

For Prepar3D v4, select two byte encoding,
such as **UTF-16LE** (The default Unicode encoding used)

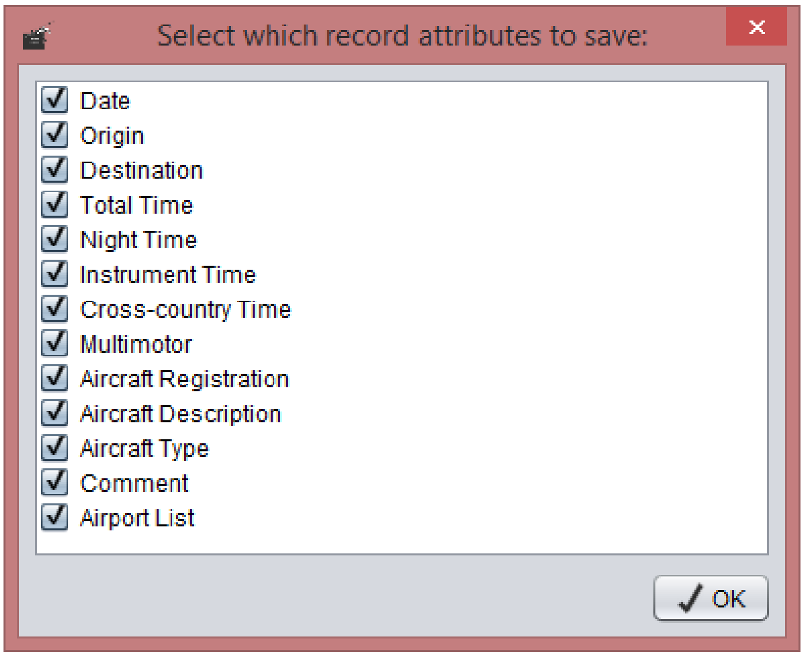
Picture 45: Dialog to select target simulator

2. **Flight Simulator 2004 Logbook (.log)** – saves the logbook as Flight Simulator 2004 text logbook file.
3. **X-Plane 10/11 Logbook (.log)** – saves the logbook as X-Plane 10/11 text logbook.
4. **FS Logbook Editor XML file (.fslex)** – saves the logbook as FS Logbook Editor XML file.
5. **Comma-Separated Values (.csv)** – exports logbook as comma separated values file. This file format can be used to import data into spreadsheet editor or for further processing. The resultant file is simple text file, where each line contains one logbook record where each record attribute is separated by special character - "**;**" by default. This character can be set in Settings at ***Load & Save*** section. On each save, a CSV Export Settings dialog will be displayed, allowing to modify the settings. Read more in the ***Saving logbook into CSV file*** chapter.
6. **Open Document Spreadsheet (.ods)** – exports logbook as spreadsheet that can be opened using Open Office or Libre Office. The spreadsheet is generated using the template that is saved at **res** folder, which is located at program folder. By default, there is only one template – *ods_export.ods*, but there may be more of them. If so, you can select the one to use in Settings at ***Load & Save*** section. For more details, see: ***Customizing ods template***.
7. **Keyhole Markup Language (KML) – to view in Google Earth** – exports flights from logbook to Google Earth KML file. You can see your trips around the world. At least one Airport Data Provider must be available, otherwise the resulting file will be empty. Also, there must be at least two airports in the flight record in order for the record to be exported. See the ***News in 1.4 version*** chapter and the ***Simulators*** and ***Airport Data*** chapters under ***Settings*** chapter.
8. **Excel (2007 and later) (.xlsx)** – to store logbook as Excel spreadsheet. This was added in FS Logbook Editor version 1.8. Read more about it in the ***Saving logbook as Excel spreadsheet*** chapter.
9. **Excel (97 – 2003) (.xls)** – to store logbook as Excel (legacy format) spreadsheet. This was added in FS Logbook Editor version 1.8. Read more about it in the ***Saving logbook as Excel spreadsheet*** chapter.

10. **JSON file (.json)** – to store logbook as JSON to file. This was added in FS Logbook Editor version 1.8. The JSON format is a common format to interchange data between various programs. You can read a little bit more here: ***Load / save logbook from / as JSON file.***
11. **All Files** – using this option, you can specify whatever extension you want the logbook file to have, however, logbook will be saved as Flight Simulator X binary logbook anyway, just with your extension (unless you type one of the extensions from previous file types).

When saving using this action, the file you choose to save to is NOT remembered. This action works like Export, or save and forget if you want.

If you save the logbook as **.csv** or **.ods**, you will be also prompted to choose which attributes of logbook record to save – you can save just some of them:


Select which record attributes to save:

- ✓ Date
- ✓ Origin
- ✓ Destination
- ✓ Total Time
- ✓ Night Time
- ✓ Instrument Time
- ✓ Cross-country Time
- ✓ Multimotor
- ✓ Aircraft Registration
- ✓ Aircraft Description
- ✓ Aircraft Type
- ✓ Comment
- ✓ Airport List

OK

Picture 46: Dialog to choose logbook record attributes to be saved

- **Save Selected As…** - the same as Save As option, but it will save only selected records on the Editor tab. At least one record must be selected for this option to become active.
- **Save filtered As…** the same as Save As option, but it will only save records that are currently visible in case any filter is active. This option is only available when the main table showing logbook records is filtered. See ***Filtering logbook records*** for more details about filtering records.
- **Print logbook table… -** Allows to print the logbook table as seen in the program. See ***Error: Reference source not found*** chapter and read about the **Print options** the ***General*** of ***Settings***.
- **Settings...** – Opens the settings window. See ***Settings*** chapter for more details.
- **Close File** – Closes currently opened logbook. If logbook is not saved, you will be asked whether to save it or not.
- **Exit** – Terminates the program. If logbook is opened and not saved, you will be asked whether to save it or not.

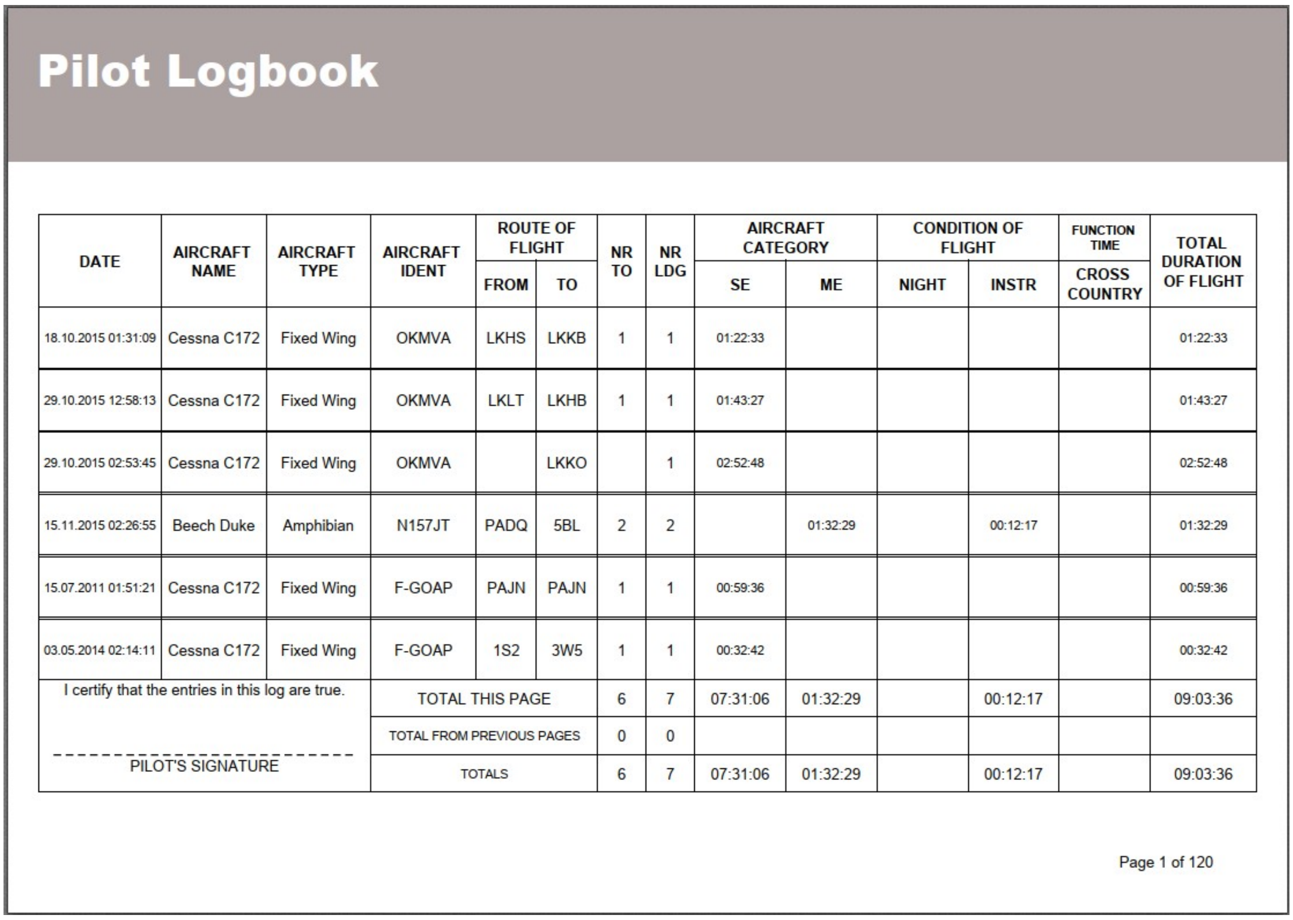
4.2. Record

Record menu contains actions that can be performed with logbook records. These are mostly the same actions you can perform by corresponding buttons at ***Editor*** tab.

- **Add** – Adds new record to the end of table displaying logbook records. See ***Editor***.
- **Edit** – Opens the ***Edit logbook record dialog*** to edit selected record. See ***Editor***.
- **Join** – joins two or more records into one record. See ***Editor***.
- **Delete** – deletes selected record(s). See ***Editor***.
- **Clear status** – clears record status. Record have status, so the program can display various records (news records, corrupted records, imported records) by different color. Clearing the status will cause the record to be displayed in color for odd or even record respectively.

4.3. JasperReports

For a while I have been thinking about a way to get nice looking output from FS Logbook Editor. If it would be possible to generate PDF, table that looks like my ordinary paper pilot logbook - book. Well, there are some ways and one of them is JasperReports framework. What is it? Will try to explain simply. JasperReports is a tool which allows you to create pixel perfect documents with almost any content. Well, actually, you create template, which is then, filled by data. In our case, with flight data by FS Logbook Editor using the JasperReports framework. For example, the output may look like on the picture below:


Pilot Logbook

DATE	AIRCRAFT NAME	AIRCRAFT TYPE	AIRCRAFT IDENT	ROUTE OF FLIGHT		NR TO	NR LDG	AIRCRAFT CATEGORY		CONDITION OF FLIGHT		FUNCTION TIME	TOTAL DURATION OF FLIGHT
				FROM	TO			SE	ME	NIGHT	INSTR	CROSS COUNTRY	
18.10.2015 01:31:09	Cessna C172	Fixed Wing	OKMVA	LKHS	LKKB	1	1	01:22:33					01:22:33
29.10.2015 12:58:13	Cessna C172	Fixed Wing	OKMVA	LKLT	LKHB	1	1	01:43:27					01:43:27
29.10.2015 02:53:45	Cessna C172	Fixed Wing	OKMVA		LKKO		1	02:52:48					02:52:48
15.11.2015 02:26:55	Beech Duke	Amphibian	N157JT	PADQ	5BL	2	2		01:32:29		00:12:17		01:32:29
15.07.2011 01:51:21	Cessna C172	Fixed Wing	F-GOAP	PAJN	PAJN	1	1	00:59:36					00:59:36
03.05.2014 02:14:11	Cessna C172	Fixed Wing	F-GOAP	1S2	3W5	1	1	00:32:42					00:32:42
I certify that the entries in this log are true.			TOTAL THIS PAGE			6	7	07:31:06	01:32:29		00:12:17		09:03:36
			TOTAL FROM PREVIOUS PAGES			0	0						
PILOT'S SIGNATURE			TOTALS			6	7	07:31:06	01:32:29		00:12:17		09:03:36

Page 1 of 120

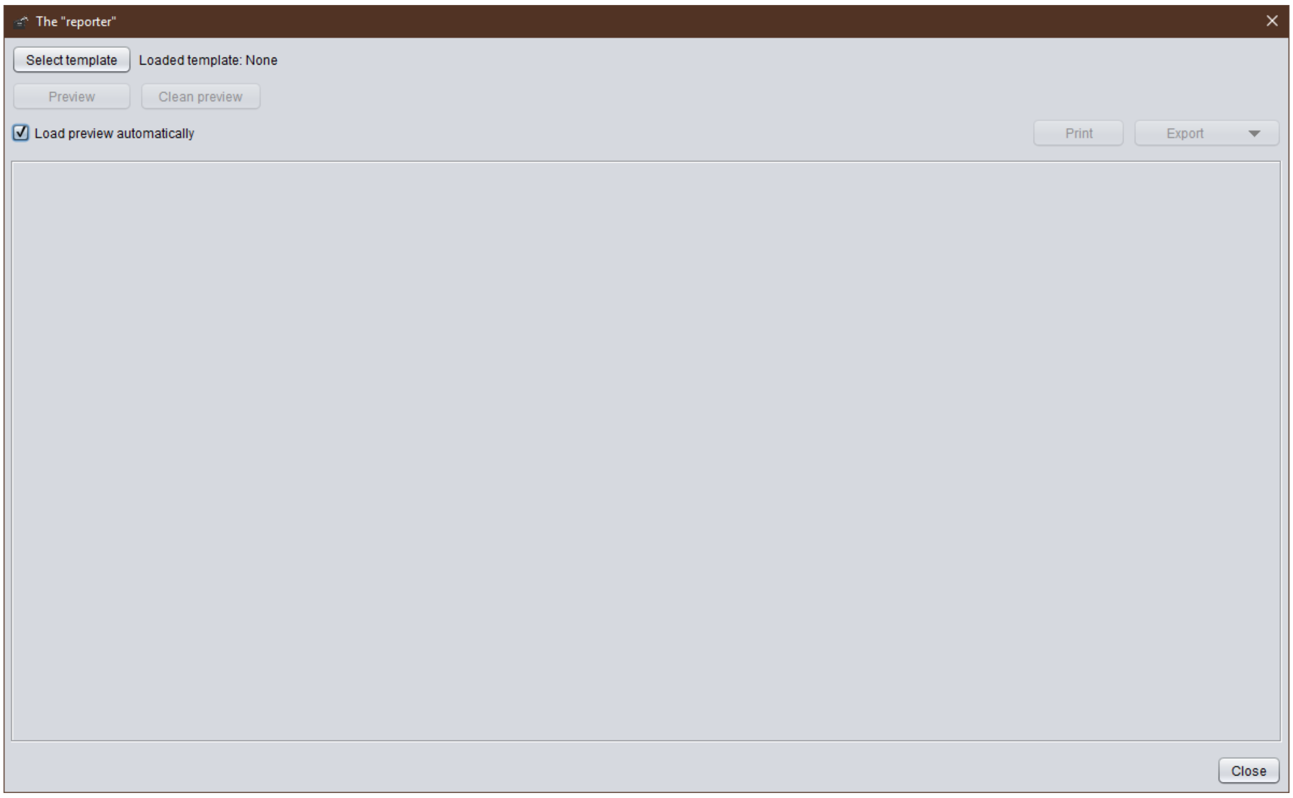
Picture 47: A logbook (report) generated using JasperReports

These are several commands available in the JasperReport menu:

- **Logbook report ...** – will open dialog to generate JasperReport from the whole logbook.
- **Logbook report (filtered) ...** – will open a dialog to generate JasperReports from the filtered logbook records. A filter must be active. See ***Filtering logbook records*** for more details.
- **Clean preview** – will clean the preview at JasperReports dialog. Same as pushing the **Clean preview** button at the same dialog. See next chapter for more details.

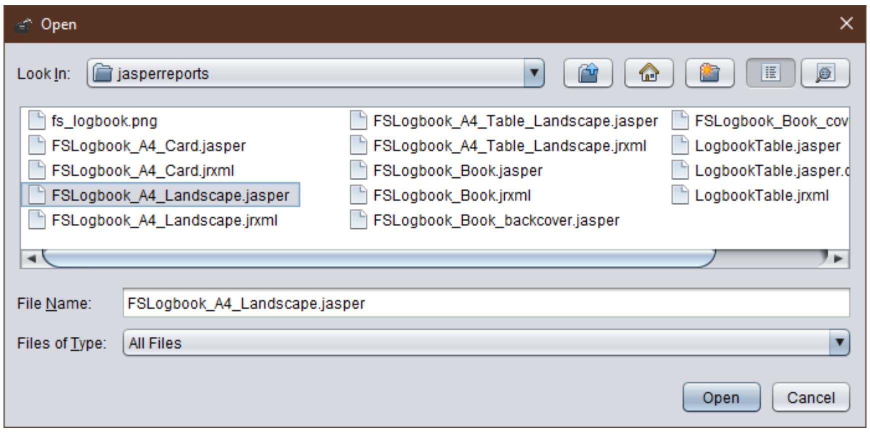
4.3.1. How to make the report?

Generating the report in FS Logbook Editor is quite simple. Just decide, whether you want to create the report from the whole logbook, or just from the filtered records (if there is a filter set. See the ***Filtering logbook records*** for more details). To create report from the whole logbook, just select the **JasperReports → Logbook report…** from the menu. To create report just from the filtered records, use the **JasperReports → Logbook report (filtered)…** menu command. Of course, if no logbook is opened or no filter applied, the respective commands will be grayed out. A dialog window like the one on the next picture will be displayed:



Picture 48: JasperReports dialog

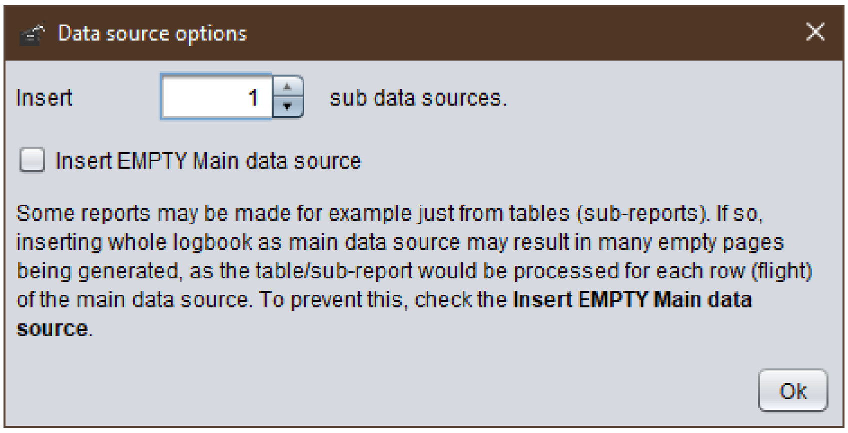
As you can see on the dialog, while no template is selected, the only available button is **Select template** button. As written in the introductory paragraph, JasperReports generates output from template, so first thing we need to do, is to select a template. So click the **Select template** button, which will open standard file selection dialog. The template can be stored anywhere. FS Logbook Editor will remember last location you used and will try to open it again next time. By default, this location will be the **res\jasperreports** folder under the FS Logbook Editor installation folder. Some default templates should be there. The templates are files with the **.jrxml** or **.jasper** extension. It is advised to use the **.jasper** files, as these are "compiled" templates and thus are faster to load. The **Load preview automatically** check box should be selected by default (FS Logbook Editor will remember whether it is selected or not). If it is selected, a preview of filled template will be generated right upon selecting one. Otherwise, you have to do one more click and generate the preview by **Preview** button, which will become available after selecting a template.



Picture 49: JasperReports template selection

There is also a **Clean preview** button, which will close the preview. This button was added due to the fact, that the preview of some templates can be very demanding on computer resources. While testing a template with graphs, with one complex one, it took tens of seconds before the preview window got rendered and was almost impossible to move between pages. For the cases like this, use the Clean preview button which will close the preview and should resume normal operation of FS Logbook Editor. The same action can be called from the JasperReports menu in the main FS Logbook Editor window.

Each template file needs at least one data source – so that it has something to gather data from. It can also have many sub-data sources. These are generally used by tables or sub-templates within the template and are usually passed as parameters to the template. A "master" data source – the logbook itself – is inserted automatically by FS Logbook Editor. FS Logbook Editor will also read template parameters in order to determine how many sub-data sources are needed by the template (if they are defined as parameters with the name **subDataSource1**...**subDataSourceN**) and it will select the required amount in the **Data source options** dialog below. If this fails, the last used value will be selected. Yes, I could keep that to happen automatically and save you from the following dialog, but there is one more option and ... I think it is good to have a manual override just in case the automation fails. By default (and for most pre-made templates), one sub-data source is enough. So the following dialog, you can accept just by clicking **Ok**, or you can adjust the amount of sub-data sources inserted.

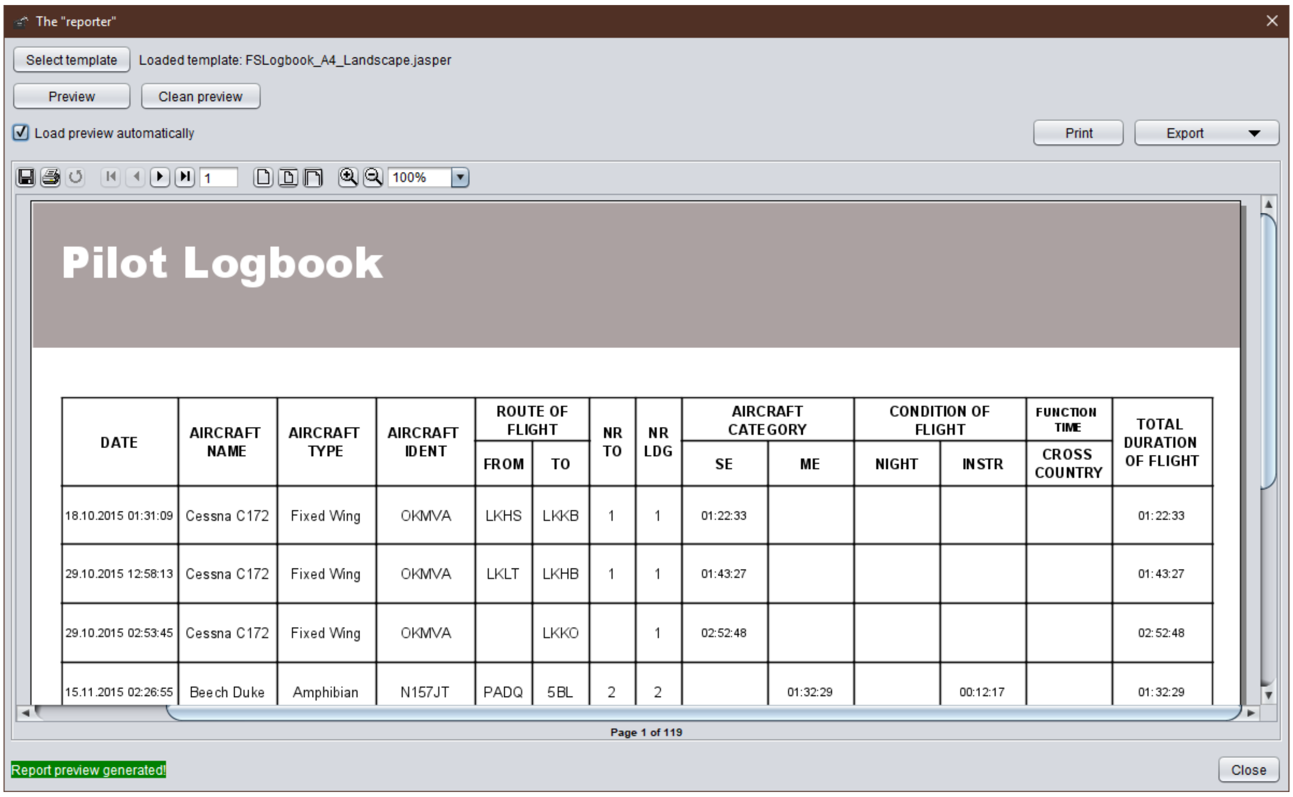


Some reports may be made for example just from tables (sub-reports). If so, inserting whole logbook as main data source may result in many empty pages being generated, as the table/sub-report would be processed for each row (flight) of the main data source. To prevent this, check the **Insert EMPTY Main data source**.

Picture 50: A dialog to select amount of sub-data sources

There is also an **Insert EMPTY Main data source** checkbox option. It is supposed for this option to be off most of the time. What it is there for? Well, some JasperReports templates may be made out of table or another elements, which gathers data from those sub-data source. Let‘s say that you want to generate a report from logbook with 900 flights. If the whole logbook is inserted as main data source in the template that uses table for the purpose of displaying the logbook, the table would be processed 900 times. Depending on how the template is constructed, this will result in having 900 tables (all with 900 records), or one table and many empty pages. To prevent this, check the **Insert EMPTY Main data source**. When checked, FS Logbook Editor will insert an empty data source with one empty record (so the table or another element is processed once). Of course, this makes sense if you specify to insert at least one sub-data source, so the table or another elements using them have data to build on. FS Logbook Editor will remember whether you check/uncheck this so you do not have to set it all the time. So, if you get empty report, try to uncheck this option. If on the other side you get report with many blank pages or repeated elements, try to check this option.

After you click Ok, FS Logbook Editor will work for a while (depending on the size of your logbook). And if all goes Ok, the dialog with the preview generated may look like the one below:



The "reporter"
Select template
Loaded template: FSLogbook_A4_Landscape.jasper
Preview
Clean preview
Load preview automatically
Print
Export
100%
Close

Pilot Logbook

DATE	AIRCRAFT NAME	AIRCRAFT TYPE	AIRCRAFT IDENT	ROUTE OF FLIGHT		NR TO	NR LDG	AIRCRAFT CATEGORY		CONDITION OF FLIGHT		FUNCTION TIME	TOTAL DURATION OF FLIGHT
				FROM	TO			SE	ME	NIGHT	INSTR	CROSS COUNTRY	
18.10.2015 01:31:09	Cessna C172	Fixed Wing	OKMVA	LKHS	LKKB	1	1	01:22:33					01:22:33
29.10.2015 12:58:13	Cessna C172	Fixed Wing	OKMVA	LKLT	LKHB	1	1	01:43:27					01:43:27
29.10.2015 02:53:45	Cessna C172	Fixed Wing	OKMVA		LKKO		1	02:52:48					02:52:48
15.11.2015 02:26:55	Beech Duke	Amphibian	N157JT	PADQ	5BL	2	2		01:32:29		00:12:17		01:32:29

Page 1 of 119

Report preview generated!

Picture 51: JasperReport preview generated

When a template is loaded, the **Print** and **Export** buttons will become available also.

With the **Print** button, as you might guess, you can print the on any printer (physical or virtual). This is useful for example for printing into PDF on virtual printer. With the **Export** button, you can export the report into many formats:

- **.jrprint** – A JasperReports output format. Can be printed by other software.
- **HTML** – Will export as HTML page.
- **RTF** – Will export Rich Text Document (similar to Word)
- **XLS** – Will export as Excel spreadsheet.

- **XLSX** – Will export as Excel (2007 and newer) spreadsheet.
- **DOCX** – Will export as Word document.
- **ODS** – Will export as Open Document Spreadsheet (for Open Office or Libre Office)
- **ODT** – Will export as Open Document (for Open Office or Libre Office)
- **CSV** – Will export as Comma separated values document.
- **XML** – Will export XML document.
- **PDF** – Will export to PDF.

In both cases, the dialog to enter amount of inserted sub-data sources is displayed. (See ***Picture 50: A dialog to select amount of sub-data sources***). Than, depending on whether you are printing or exporting, you will be asked to select a printer (and print options) – if printing, and file name to store to – if exporting. These steps are pretty straight forward I believe and does not differ from printing or saving/exporting from any other program.

Please note that the print/export may not always look the same as in the preview. There are simply too many variables involved - the capabilities of the selected file format / printer, the printer driver itself, operating system, the JasperReports library... many things that FS Logbook Editor cannot affect.

The last button on this dialog is the **Close** button and as the caption says, it will close the dialog.

4.4. Tools

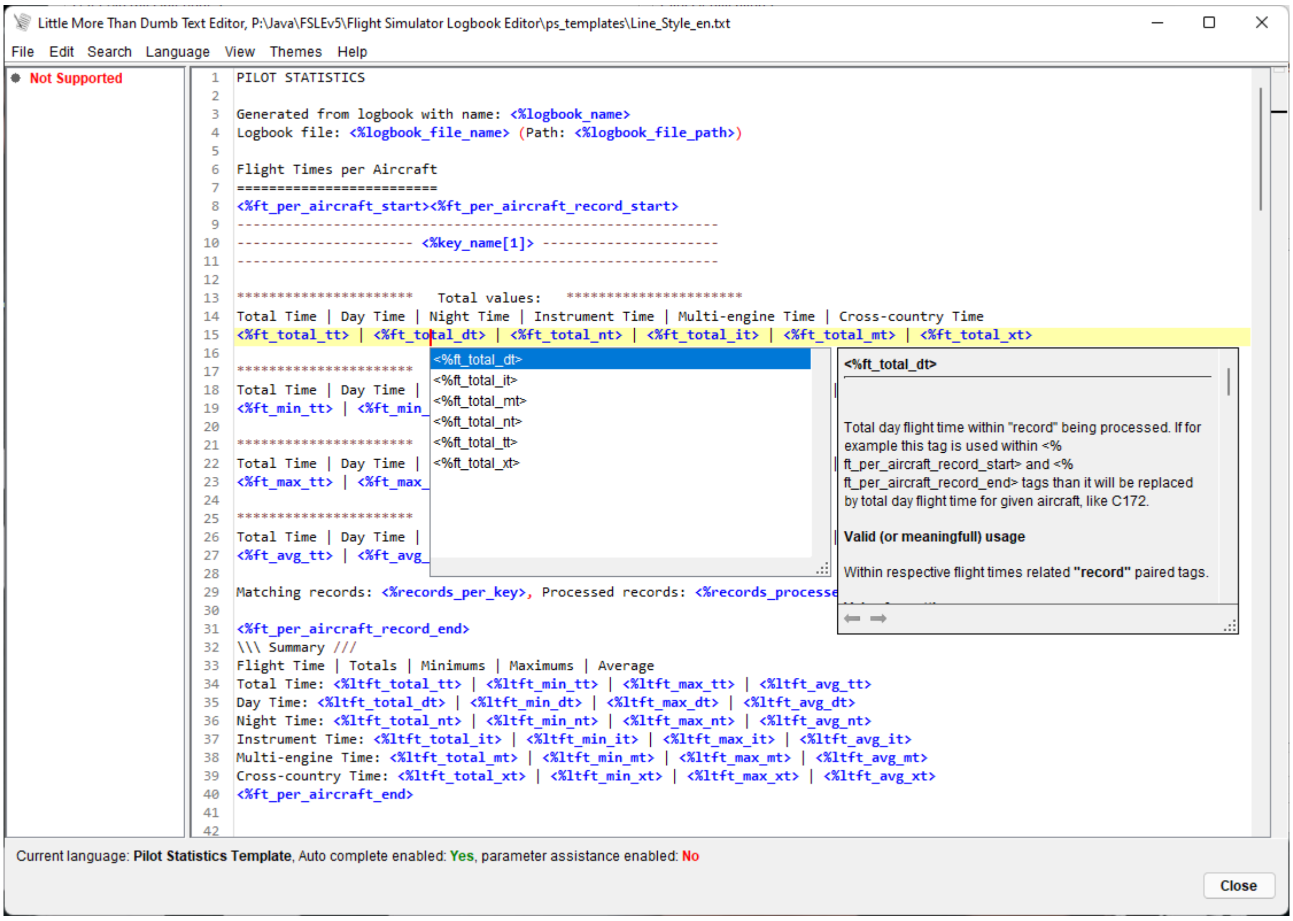
In FS Logbook Editor version **1.8**, the Tools menu has been added. It contains only one item – a **Little More Than Dumb Text Editor** …

4.4.1. Little More Than Dumb Text Editor

Ok, what is that? Well, as the title says, it is a text editor, integrated within FS Logbook Editor and with some smart features like syntax highlighting and auto-completion for various languages. The main reason for this editor was to provide some more friendly way to write pilot statistics templates (see ***Writing custom pilot statistics template***), as the editor supports all the keywords, can highlight them and also provide some quick help for their usage. The same is true for new comment templates, introduced with FS Logbook version version 1.8, which can be used when importing flights from FsHub and/or MSFS. The other is … just a showcase of what the wonderful libraries the editor is build upon can do – I left most options enabled, so you can edit other types of files too.

So, no need to describe it in much detail… just try it out.

<content follows on next page>



Little More Than Dumb Text Editor, P:\Java\FSLEv5\Flight Simulator Logbook Editor\ps_templates\Line_Style_en.txt
File
Edit
Search
Language
View
Themes
Help
Not Supported
Close

```
PILOT STATISTICS

Generated from logbook with name: <%logbook_name>
Logbook file: <%logbook_file_name> (Path: <%logbook_file_path>)

Flight Times per Aircraft
=========================
<%ft_per_aircraft_start><%ft_per_aircraft_record_start>
------------------------------------------------------------
-------------------- <%key_name[1]> --------------------
------------------------------------------------------------

**********************   Total values:   **********************
Total Time | Day Time | Night Time | Instrument Time | Multi-engine Time | Cross-country Time
<%ft_total_tt> | <%ft_total_dt> | <%ft_total_nt> | <%ft_total_it> | <%ft_total_mt> | <%ft_total_xt>

**********************
Total Time | Day Time |
<%ft_min_tt> | <%ft_min_

**********************
Total Time | Day Time |
<%ft_max_tt> | <%ft_max_

**********************
Total Time | Day Time |
<%ft_avg_tt> | <%ft_avg_

Matching records: <%records_per_key>, Processed records: <%records_processe

<%ft_per_aircraft_record_end>
\\\ Summary ///
Flight Time | Totals | Minimums | Maximums | Average
Total Time: <%ltft_total_tt> | <%ltft_min_tt> | <%ltft_max_tt> | <%ltft_avg_tt>
Day Time: <%ltft_total_dt> | <%ltft_min_dt> | <%ltft_max_dt> | <%ltft_avg_dt>
Night Time: <%ltft_total_nt> | <%ltft_min_nt> | <%ltft_max_nt> | <%ltft_avg_nt>
Instrument Time: <%ltft_total_it> | <%ltft_min_it> | <%ltft_max_it> | <%ltft_avg_it>
Multi-engine Time: <%ltft_total_mt> | <%ltft_min_mt> | <%ltft_max_mt> | <%ltft_avg_mt>
Cross-country Time: <%ltft_total_xt> | <%ltft_min_xt> | <%ltft_max_xt> | <%ltft_avg_xt>
<%ft_per_aircraft_end>
```

```
<%ft_total_dt>
<%ft_total_it>
<%ft_total_mt>
<%ft_total_nt>
<%ft_total_tt>
<%ft_total_xt>
```

<%ft_total_dt>

Total day flight time within "record" being processed. If for example this tag is used within <%ft_per_aircraft_record_start> and <%ft_per_aircraft_record_end> tags than it will be replaced by total day flight time for given aircraft, like C172.

Valid (or meaningfull) usage

Within respective flight times related **"record"** paired tags.

Current language: **Pilot Statistics Template**, Auto complete enabled: Yes, parameter assistance enabled: No

Picture 52: Little More Than Bumb Text Editor - with loaded pilot statistics template

4.5. Help

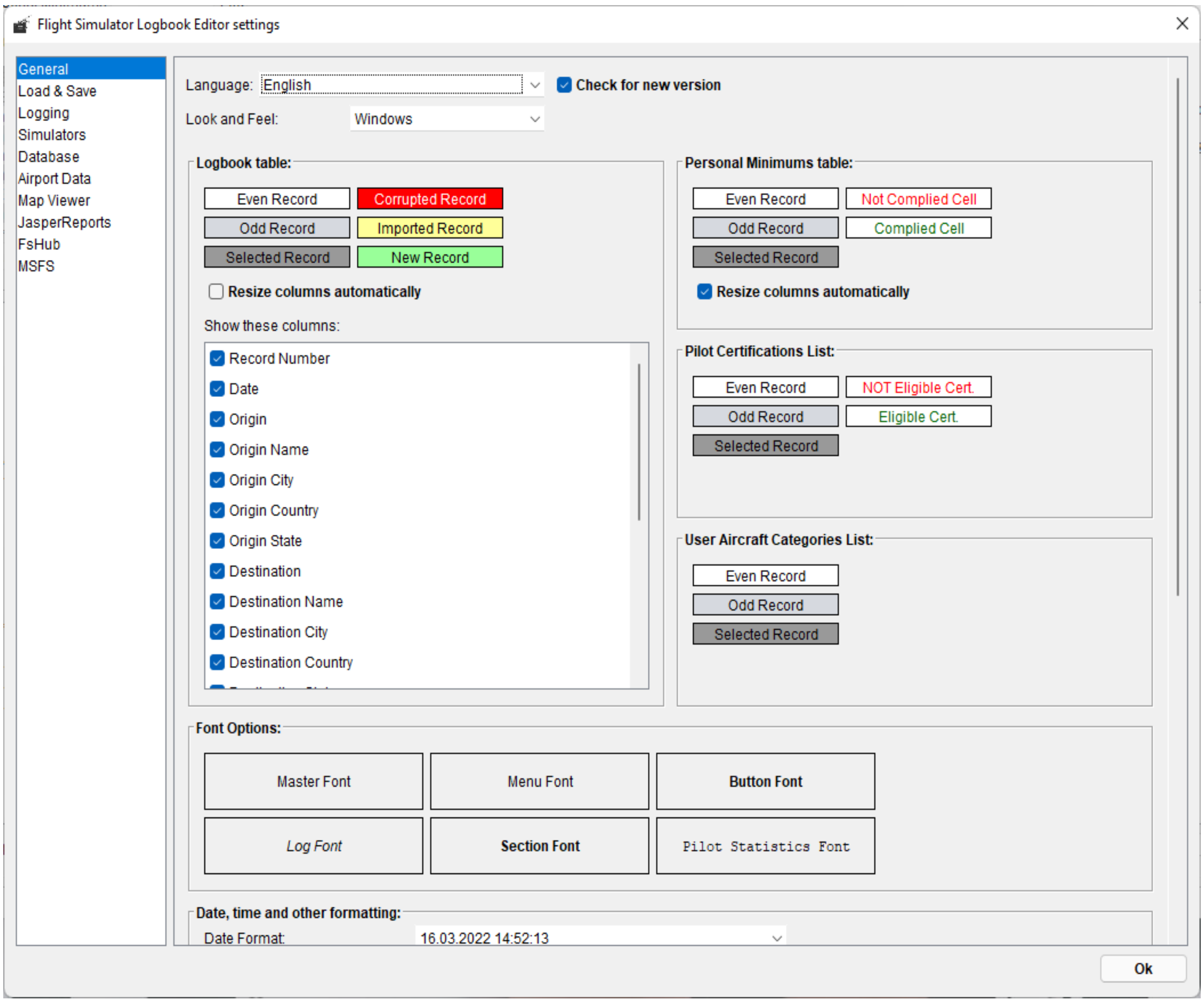
Help menu contains following options:

- **About** – displays about dialog with basic information about program. Since FS Logbook Editor version 1.31, a simple online version check is performed when you open the dialog. The result of the check is displayed next to the information about current version. The result may be:
  - **N/A** – when the version check failed for any reason (no internet connection etc.)
  - **Latest version** – when you have the latest version.
  - **<Some number> is available** – tells you that newer version is available and what version it is.
- **License** – displays FS Logbook Editor EULA.
- **Help** – displays this document.

5. Settings

The settings window contains ~~three~~ ten sections (as for now). More on those in the following chapters.

5.1. General



Language: English

[✓] Check for new version

Look and Feel: Windows

Logbook table: [ ] Resize columns automatically

Personal Minimums table: [✓] Resize columns automatically

Show these columns:

- [✓] Record Number
- [✓] Date
- [✓] Origin
- [✓] Origin Name
- [✓] Origin City
- [✓] Origin Country
- [✓] Origin State
- [✓] Destination
- [✓] Destination Name
- [✓] Destination City
- [✓] Destination Country

Date Format: 16.03.2022 14:52:13

Picture 53: General section of Settings

General options allows to set the following:

- **Language** – allows you to switch to one of supported program translations. The language will change immediately after you make your selection.
- **Check for new version** – will perform the check if there is new version of FS Logbook Editor. If there is, an information message will be shown. (Added in version 1.6).
- **Look and Feel** – allows to change the Look and Feel. It does, what it says it does - changes the look and feel of the whole application. For example, you can set that FS Logbook Editor will look like standard Windows application. There should be couple of options, but the amount may vary on each system, so I will not describe all the options . Some like Metal, Nimbus (the default one), Windows .... should be available. Changing the value should

update the whole GUI and it might take a little time on some systems.... **Please note also, that it might not work on all dialogs / windows (like message boxes, file selection dialogs....) right away.** It might require to restart FS Logbook Editor and even then some message boxes / dialogs might look "not completely switched". (Added in version 1.65)

Logbook table group allows to change some options related to table displaying logbook records, located at ***Editor*** tab, such as:

- **Even record** – font, background and foreground color of even records.
- **Odd record** – font, background and foreground color of odd records.
- **Selected record** – font, background and foreground color of selected record.
- **Corrupted record** – font, background and foreground color of corrupted record. Corrupted record is record that triggered some errors during loading of logbook, so it may not be complete or correct.
- **Imported record** – font, background and foreground color of record that was added using ***Append*** or ***Merge With*** menu actions.
- **New record** – font, background and foreground color that was added using Add button at ***Editor*** tab or Add menu action at ***Record*** menu.

Setting the background and foreground color works the same for all above. ~~Left-click the sample cell to set background color, right-click the sample cell to set foreground color.~~ Since version 1.65, use left mouse click to change the font, middle mouse click to change background color and right mouse click to change foreground color. Standard dialog to choose font/color will show up, select font/color, click **Ok** and you are done. The change will be previewed in the sample cell immediately.

- **Date format** – date format in which to display date and time of the flight in the table at ***Editor*** tab. These date formats are defined at *date_formats.xml* file that is located at **res** folder, that in turn is at program folder. For more details, see: ***Customizing date formats***.
- **Resize columns automatically** – this will enable/disable automatic resizing of columns in the table displaying logbook at Editor tab. If you disable automatic resizing, you can set the width of each column by yourself to the size you like. But the size will not change if the size of FS Logbook Editor window changes (and thus the size of the table).

Personal Minimums table group, grouping pretty much the same options as Logbook table group, but those are for Personal Minimums check results table at ***Personal Minimums*** tab. You can set:

- **Even record** – font, background and foreground color of even records.
- **Odd record** – font, background and foreground color of odd records.
- **Selected record** – font, background and foreground color of selected record.
- **Not Complied Cell** – foreground color of the cell of personal minimum that you did not comply with. This one does not support setting background color, it will always have background color of odd, even or selected row.
- **Complied Cell** – foreground color of the cell of personal minimum that you complied with. This one does not support setting background color, it will always have background color of odd, even or selected row.

Setting the background and foreground color works the same for all above. ~~Left-click the sample cell to set background color, right-click the sample cell to set foreground color.~~ Since version 1.65, use left mouse click to change the font, middle mouse click to change background color and right mouse click to change foreground color. Standard dialog to choose font/color will show up, select font/color, click **Ok** and you are done. The change will be previewed in the sample cell immediately. The font cannot be changed for “Complied” and “Not Complied” rows – because than, the different font for odd, even, selected record would have no effect.

- **Resize columns automatically** – this will enable/disable automatic resizing of columns in the table displaying personal minimums check results at Personal Minimums tab. If you disable automatic resizing, you can set the width of each column by yourself to the size you like. But the size will not change if the size of FS Logbook Editor window changes (and thus the size of the table).

User Aircraft Categories List group allows you to change font and colors for displaying items in the user aircraft categories list at Pilot statistics tab:

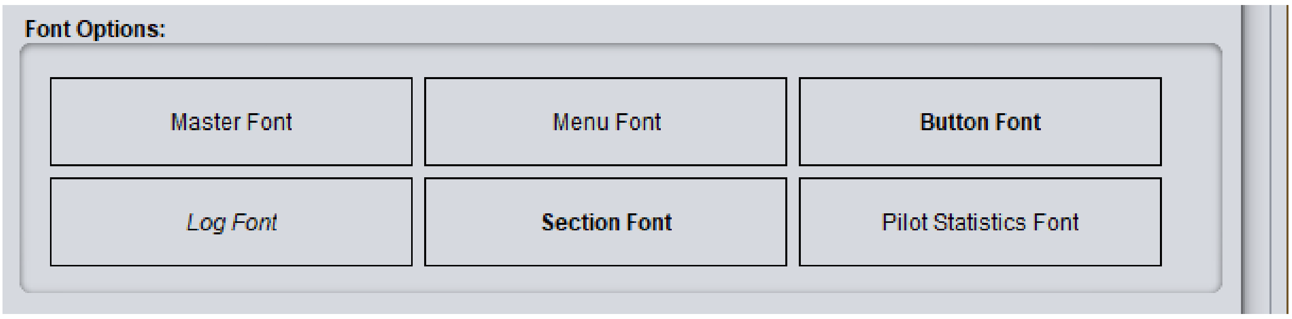
- **Even record** – font, background and foreground color of even records.
- **Odd record** – font, background and foreground color of odd records.
- **Selected record** – font, background and foreground color of selected record.

Pilot Certifications List group allows you to change font and colors for displaying items in the pilot certifications list at Pilot statistics tab:

- **Even record** – font, background and foreground color of even records.
- **Odd record** – font, background and foreground color of odd records.
- **Selected record** – font, background and foreground color of selected record.
- **NOT Eligible Cert.** – color of pilot certification that you are NOT eligible for.
- **Eligible Cert.** – color of pilot certification that you are eligible for.

Font Options

Font options were added in FS Logbook Editor version 1.65.



Picture 54: Font Options

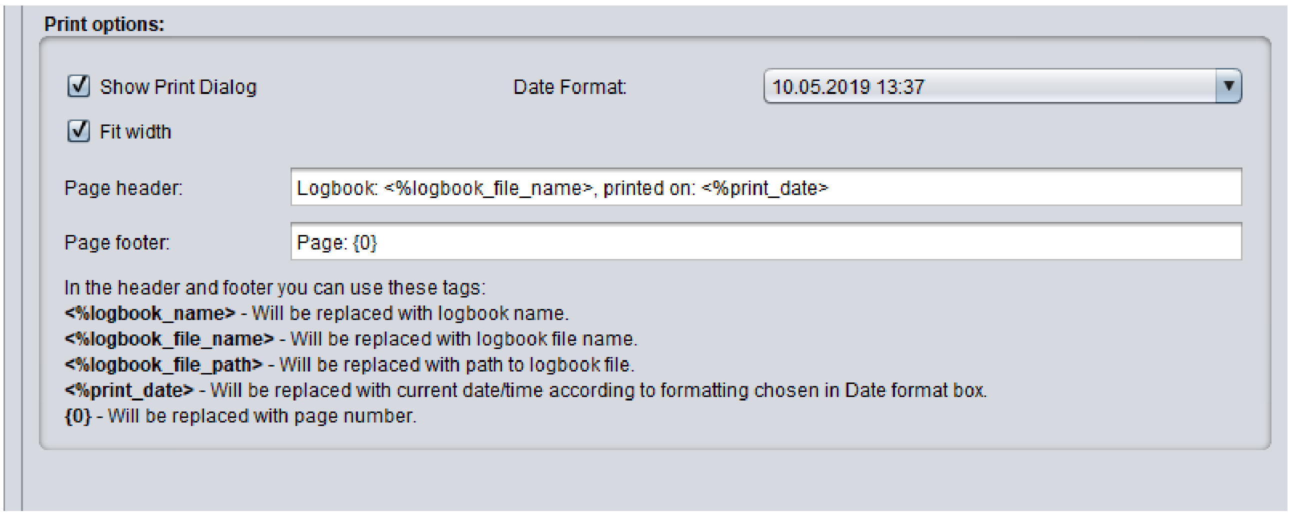
You can set the following fonts:

- **Master font** – the font that is used throughout the application. Ie - almost all text that you see will use this font.
- **Menu font** – the font that will be used in the ***Menus***.
- **Button font** – the font for buttons.
- **Log Font** – the font that will be used for log messages at ***Log*** tab.
- **Section font** – the font that will be used to distinguish some sections of setting and also will be used for the Selection and Overall statistics titles.
- **Pilot Statistics Font** – the font that will be used for showing pilot statistics at the ***Pilot Statistics*** tab.

All these six controls works the same way. Use left mouse button to select the font (standard font selection dialog will be used). Use right mouse click to reset the font to the default value. Note that font for logbook table and other tables can be changed using controls for selecting colors (described above). **Note that some parts of the GUI may not react to the font change immediately (like message box dialogs, file selection dialogs...).** It might require to restart FS Logbook Editor and even than some components may ignore it, no matter how I tried to force them all :(

Date, time and other formatting option group allows you to modify how certain values are formatted within FS Logbook Editor:

- **Date format** – sets how the date of flight is formatted on ***Editor*** tab in the main FS Logbook Editor window. The format of date when exporting logbook to various formats may be different (file format-specific).
- **Time format** – sets how the flight times are displayed throughout the application. This applies to Editor tab, Pilot statistics and other places. The format of time when exporting logbook to various formats may be different (file format-specific).
- **Distance unit** – sets what unit is used to display computed flight distances throughout the application..
- **Distance format** – sets format in which the computed flight distances are displayed throughout the application.



Print options:

[✓] Show Print Dialog
Date Format: 10.05.2019 13:37
[✓] Fit width
Page header: Logbook: <%logbook_file_name>, printed on: <%print_date>
Page footer: Page: {0}

In the header and footer you can use these tags:
<%logbook_name> - Will be replaced with logbook name.
<%logbook_file_name> - Will be replaced with logbook file name.
<%logbook_file_path> - Will be replaced with path to logbook file.
<%print_date> - Will be replaced with current date/time according to formatting chosen in Date format box.
{0} - Will be replaced with page number.

Picture 55: Print options

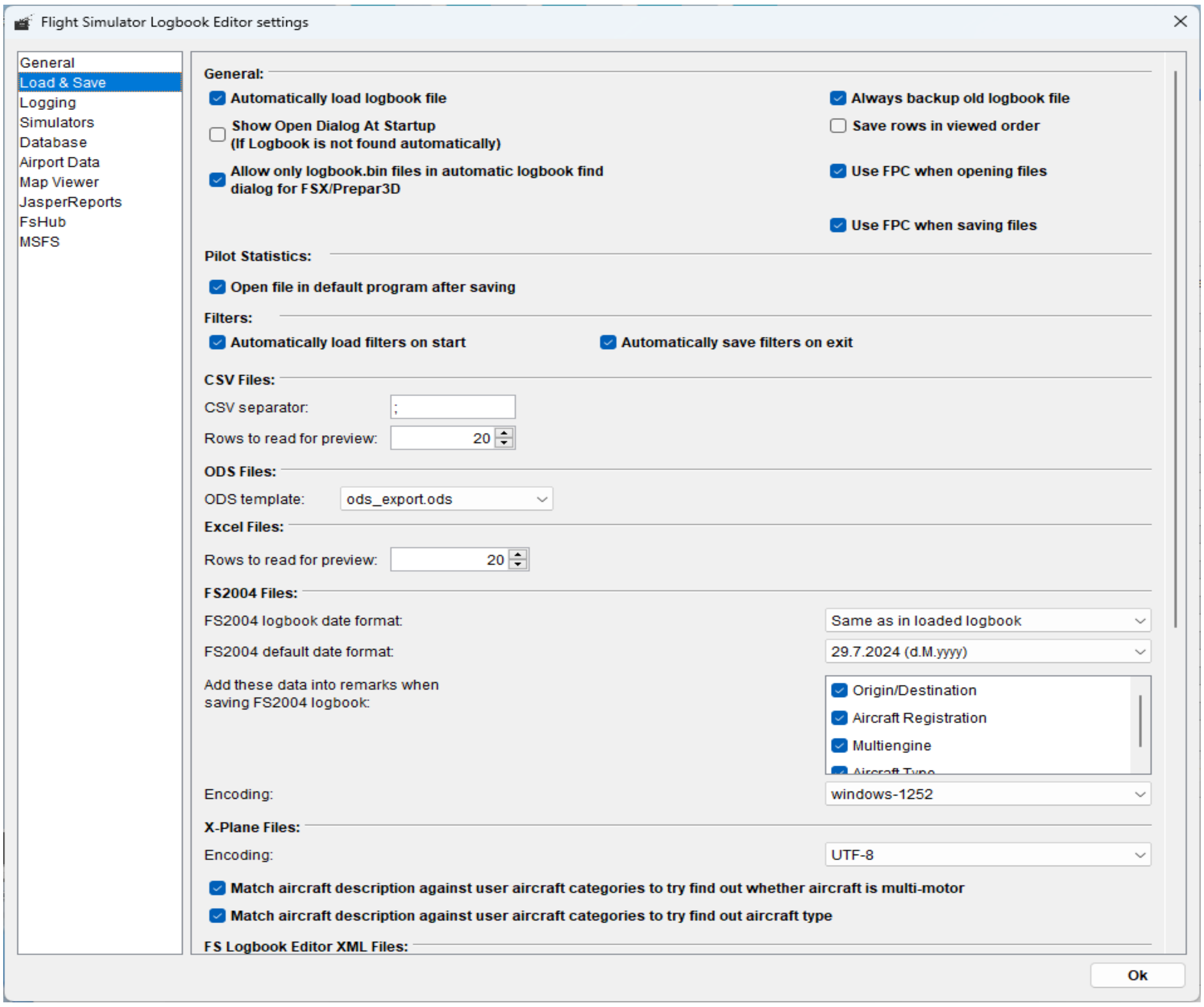
Picture 55: Print options shows new printing options added in FS Logbook Editor version 1.60, which supports basic print of the logbook table as viewed in the program. The options are:

- **Show print dialog** – will show system print dialog while printing the logbook table. It will allow you to select printer, paper etc. When this option is off, your default printer with default options will be used.
- **Fit width** – Determines whether the printed table should fit width of the paper or not. If disabled, the table may span over multiple papers.
- **Date Format** – The format of date if it is printed in the header / footer of each page.
- **Page header and page footer** – Those two text fields allows you to define custom text to be printed in the header / footer of each page. You can use some variables which will be replaced by respective values. Those variables are:
  - **<%logbook_name>** – The name of the logbook. Since for now, only FS2004 logbook supports the name of the logbook, for all others this will be the default value STANDARD PILOT LOG.
  - **<%logbook_file_name>** – The name of the logbook file (For example: Logbook.bin).
  - **<%logbook_file_path>** – The full path to the logbook file. Note that it may easily

overflow the header / footer area.

- **<%print_date>** – Date of printing the table. It will be formatted as specified by **Date Format** option.
- **{0}** – Will be replaced by page number. Note that it is not possible to tell how many pages there will be before the print, that is why there is no variable for total page count.

5.2. Load & Save


Flight Simulator Logbook Editor settings

General
Load & Save
Logging
Simulators
Database
Airport Data
Map Viewer
JasperReports
FsHub
MSFS

General:
- [✓] Automatically load logbook file
- [✓] Always backup old logbook file
- [ ] Show Open Dialog At Startup (If Logbook is not found automatically)
- [ ] Save rows in viewed order
- [✓] Allow only logbook.bin files in automatic logbook find dialog for FSX/Prepar3D
- [✓] Use FPC when opening files
- [✓] Use FPC when saving files

Pilot Statistics:
- [✓] Open file in default program after saving

Filters:
- [✓] Automatically load filters on start
- [✓] Automatically save filters on exit

CSV Files:
CSV separator: ;
Rows to read for preview: 20

ODS Files:
ODS template: ods_export.ods

Excel Files:
Rows to read for preview: 20

FS2004 Files:
FS2004 logbook date format: Same as in loaded logbook
FS2004 default date format: 29.7.2024 (d.M.yyyy)
Add these data into remarks when saving FS2004 logbook:
- [✓] Origin/Destination
- [✓] Aircraft Registration
- [✓] Multiengine
- [✓] Aircraft Type
Encoding: windows-1252

X-Plane Files:
Encoding: UTF-8
- [✓] Match aircraft description against user aircraft categories to try find out whether aircraft is multi-motor
- [✓] Match aircraft description against user aircraft categories to try find out aircraft type

FS Logbook Editor XML Files:

Picture 56: Load & Save section of Settings

This section hides some options related to load and save operations and different file formats supported. The General section have the following options:

- **Automatically load logbook file** – if this option is checked, FS Logbook Editor will try to find your default logbook file in your profile folder. If you have more than one flight simulator installed (Flight Simulator 2004 and X for example) more than one logbook may be found. In this case, you will be given an option to select one to load. This automatic search only happens after program start, the ***Open Logbook*** menu action does not trigger it. Since FS Logbook Editor version 1.31, you can trigger this action also with ***File*** → ***Open Logbook (Auto find...)*** action. For that menu item to be available, this option must be enabled. This option is on by default.
- **Show Open Dialog at Startup (If Logbook is not found automatically)** – if this option

and **Automatically load logbook file** option is checked and no logbook is found, then standard open file dialog is displayed after program start, as in the case of ***Open Logbook*** menu action. Otherwise, no logbook is opened. This option is on by default.

- **Allow only logbook.bin files in automatic logbook find dialog for FSX/Prepar3D** – this option was added in version 1.86. It is enabled by default and will only allow *logbook.bin* files for logbook files that are automatically found for FSX/Prepar3D. So files like *wxstationlist.bin* etc. should no longer be listed as logbook files.
- **Always backup old logbook file** – if checked, a backup copy of every logbook that is opened either automatically or by ***Open Logbook*** menu action is automatically created.

  The backup files are created at the same folder in which the file being opened is. The name of backup files is combined from the name of file being opened, time stamp and **.bak** extension. For example, if opening logbook of Flight Simulator X (Logbook.bin file), the name of backup may look like this:

  logbook.101113_213311.bin.bak

  The time stamp says, that the file was created at 10.11.2013 at 21:33:11. This option is on by default and I recommend you keep it on.
- **Save rows in viewed order** – enabling this option will force FS Logbook Editor to save records to output file in the same order as viewed at FS Logbook Editor. In FS Logbook Editor you can sort records by clicking on respective column headers. For example, if you click Date column header, the records will be sorted by date. In fact however, it is just the "view", what gets sorted, not the logbook records themselves. When you save the logbook file, the records will be saved in the order they were loaded from opened logbook. If you want them to be saved in the order as you see them, enable this option. This is useful if the simulator does not support sorting of logbook records and therefore may display them in wrong order (usually in the order the records are stored in).
- **~~Logbook encoding~~** – removed in version 1.42. The description of encoding has been moved to ***Note on Encoding*** chapter.
- **Use FPC when opening files** – added in version 1.86. FPC stands for *"File Parameters Cache"* and it is new feature to minimize the amount of dialogs you have to click through while opening or saving files. Simply saying (writing), FS Logbook Editor will now remember options (file parameters) you selected when opening certain files. For example, an encoding and file version in the case of FSX/Prepar3D logbook. And if this option is enabled, it will not ask for them the next you open the same file and will use the remembered values and will skip the dialog entirely, saving you from that click.
- **Use FPC when saving files** – the same as above, but for saving the files.

I hope that **"Use FPC"** options will save you time and will serve to your liking's. I can imagine however, a situation might arise here and there when you would rather not to skip the file parameters dialog when opening/saving the file. And so that you do not have to always go to the settings to disable/enable these options, all commands that opens/saves files now support an override of this function using "**Shift**" key. For example, hold **Shift** before clicking on the **Open Logbook...** command in ***File*** menu, and the file parameters dialog (if selected logbook file format needs some) will be displayed regardless of this settings.

The commands that support the "Shift" key are all located in the ***File*** menu and are the:

- Open Logbook…
- Open Logbook (Auto find) …

- Recent files …
- Append …
- Merge with …
- Save …
- Save As …
- Save Selected As …
- Save Filtered As ...

Open file in default program after saving in the **Pilot statistics** section is pretty self explanatory I think. If enabled, when you generate a pilot statistics at ***Pilot Statistics*** tab and save it using the **Save to file** button, it will try to open the saved file with the program that is registered to open that kind of files on your system. Like Notepad for text files, mostly.

A **Filters** have two options. This section was added in FS Logbook Editor version 1.50.

- **Automatically load filters on start** – when checked, FS Logbook Editor will try to load last filters stored in **last_filters.xml** file, which is being saved to **filters** folder within FS Logbook Editor installation folder (if **Automatically save filters on exit** is checked).
- **Automatically save filters on exit** – if checked and at least one filtering condition is defined in the list at Filtering window (See ***Filtering logbook records*** for more details), FS Logbook Editor will save these filtering conditions to the **last_filters.xml** file in the **filters** folder within FS Logbook Editor installation folder.

A **CSV Files** have one option, which may be override during export:

- **CSV Separator** – default character to be used as separator of logbook record attributes when saving to Comma-separated values (.csv) file format. See saving to ***Comma-Separated Values (.csv)***.
- **Rows to read for preview** – defines how many rows for previewing the CSV file while loading logbook from CSV file. Read more in the ***Loading logbook from CSV file*** chapter. This was added in FS Logbook Editor version 1.8.

An **ODS Files** also have one export option:

- **ODS template** – you can choose template to use when exporting to Open Document Spreadsheet (.ods) file format. See saving to ***Open Document Spreadsheet (.ods)*** and ***Customizing ods template***.

Excel Files have one option:

- **Rows to read for preview** – defines how many rows for previewing the Excel file while loading logbook from CSV file. Read more in the ***Loading logbook from Excel spreadsheet*** chapter. This Excel section was added in FS Logbook Editor version 1.8. There are more options to set, but they are being displayed before loading/saving the file.

Flight Simulator 2004 logbook files have the following options:

- **FS 2004 logbook date format** – this option lets you select which format of date will be used when saving logbook to Flight Simulator 2004 text format (.log). Flight Simulator 2004 saves the date in the short date format as set in your regional settings in Windows. The Flight Simulator 2004 logbook being loaded may contain date information in any of these formats in order the date to be successfully recognized. If you want to save the date in the same format as was present in loaded logbook, select **Same as in loaded logbook** option. If you select any other option – specific date format – it will be used every time saving logbook as Flight Simulator 2004 logbook.

- **FS 2004 default date format** – this option has the same meaning as the **FS2004 logbook date format** above. But this one will be used when date format is not known – as is in the case you are saving Flight Simulator X binary logbook (.bin) or new logbook as Flight Simulator 2004 text logbook (.log). Under these circumstances, the date format was not loaded from logbook and the **FS2004 logbook date format** option does not apply. This option can only be set if **FS2004 logbook date format** option is set to: Same as in loaded logbook, otherwise **FS2004 logbook date format** option will override this one.
- **Add these data into remarks when saving FS2004 logbook** – Flight Simulator 2004 logbook only saves date of flight, aircraft description, remarks (comment) and flight times into logbook. Mostly, the remarks section is not used. Using this option, you can make use of this often unused space. You can select which information should FS Logbook Editor save into remarks when saving logbook as Flight Simulator 2004 text logbook (.log).

  Not all information are supported, like list of airports that you landed on during flight, as the remarks are limited to 24 chars in Flight Simulator 2004. If you select all supported options here, it will fill almost whole remark for the given flight.

  When loading Flight Simulator 2004 logbook, the remarks in each record are searched for these information and loaded into respective logbook record attribute (Even if nothing is selected here).

  In the logbook itself, the remarks may look like this:

  KSEA/PAKT:D-ASHA/M/2

  First is the pair of origin and destination airport. -> **KSEA/PAKT** here. If only origin airport has been entered, there would be just KSEA. If only destination airport has been entered, only /PAKT would have been present.

  Aircraft registration always starts with "**:**" character -> **D-ASHA** here.

  Multi-motor mark, whether the aircraft flown had 2 or more engines - **/M**. If single engine aircraft, there would be **/S**.

  Last is aircraft type, that is:

  - /1 - for Glider
  - /2 - for Fixed Wing
  - /3 - for Amphibian
  - /4 - for Rotorcraft

  Not all of these information may be present. They are only inserted if there is enough room for them and the comment, if any, has always highest priority. If comment is also present, it will start with "**;**" character. The remark can then look like this: *":N450VF;circuit training"*. The "**;**" character will not be there should the remarks contain only comment and none of the information mentioned above.

  Please note, that if you only select to store the **Origin/Destination**, it may happen that origin airport will end up in comment. This will happen if you set only origin airport for the logbook record and no comment. Then the remark will be only like: *"KSEA"* and it will be treated as comment.
- **Encoding** – an encoding that will be used to read FS2004 logbook files and to store them.

Default value is **windows-1252**. See the ***Note on Encoding*** chapter for more information about encoding.

X-Plane Files options, well, option was added in version 1.42 and it is:

- **Encoding** – an encoding that will be used to read FS2004 logbook files and to store them. Default value is **UTF-8**. See the ***Note on Encoding*** chapter for more information about encoding.

FS Logbook Editor XML Files also have one option:

- **Autosave when saving to other format** – this option under the **FS Logbook Editor XML Files** section was added to help preserve all the information you can edit with FS Logbook Editor. FS Logbook Editor XML file format is the only file format that supports all logbook record attributes (properties) you can edit with FS Logbook Editor. See ***Information supported by different logbook files*** chapter. Therefore, when you save logbook to any other format, you may easily loose some information, because it is easy to forgot that the specific information is not supported by selected file format. If you enable this option, FS Logbook Editor will save logbook also in FS Logbook Editor XML file format (.fslex extension) whenever you save logbook to other file format. The saved file will be in the same location with the same name (but with .fslex extension) as the logbook being originally save. Be aware that if there is already such a file, it will be overwritten without notice!

KML Export Settings have several options:

Since FS Logbook Editor v 1.4, export to KML file is possible, so you can view your flights in Google Earth (those flights for which airport data are available). That means, there must be at least two airports filled in the flight record, otherwise the flight will not be exported to KML file. For the purpose of KML export, FS Logbook editor distinguishes 3 types of airport - starting airport, en-route airport and destination airport. Then, when airport data is known, a route can be drawn between those airports. You can modify the look of all these four items by these options.

For each type of airport, you can choose the **icon**, **icon color** and the **scale**. But wait, the icon has its own color already painted in, what is the icon color for? Setting the icon color will make the icon look like if someone placed a colored transparent foil over it. The icon color will overlay the icon. You probably have to try it to see the result. To set the color, click in the black bordered rectangle with **left mouse button**. The standard color selection dialog will appear where you can mix color of your choice. If you do not want to use any color, click in the rectangle with **right mouse button**, which will reset the color to None (transparent).With scale, you can make the icon really big or small. The default value is the nature size of the icon, 100%.

The **Route settings** are little bit more tricky. You can select **Line** (route) **color** and **Line** (route) **width**. The color is pretty much self explanatory. Click in the rectangle to display the color selection dialog to mix different color. The line width determines the width of line (route). It is in pixels (the unit used in computer world, not much important for us). The point is, the bigger the value, the thicker the line will be. The important part is, that when viewing the resulting file in Google Earth, the width of the line will be the same throughout zooming.

But than there is Line outer width and Line outer color option. What is that good for? Well, the line outer color comes useful when you want the route to became something like road. You can for example set the line color to be gray, and the line outer color to be black. Than the resulting line will look like 3 lines drawn next to each other. The middle one will be gray and the ones on the edges will be black. Where it gets little more complicated is when it comes to Line outer width. Line outer width is defined as percentage of Line width. But also, in order the "outer line" to work, the Line width must be in meters. That is why, if you change Line outer width to non-zero value, the label of Line width will change to "Line width (meters)" and will be printed in red color to warn you that this value has now different meaning. The cool thing about "outer line" is that it looks cooler than simple one-color line. The downside is, that its width must be in meters. This has the downside effect, that the thickness of the line is always the same, whether you zoomed Google Earth to 5 meters or you are looking on the Earth from space. You probably won't be able to see your 5m wide flight route from space, right? You would have to come a lot closer. So, in order to enable the “outer line”, you must set Line outer width to non-zero value (and to zero value to disable it).

In FS Logbook Editor version 1.5 there are two more options – **Coordinates format:** and **Magnetic variation format:**. Both of these allows you to select format in which to display airport geographic coordinates and magnetic variation in airport description in the exported KML file. Both of these have also **Precision** setting, which applies to decimal parts of the respective formatting. For example, if you choose to the coordinates format to be in degrees decimal minutes – like this: 122° 35.42' , the precision of 2 for example, means that the minutes should be printed with maximum precision of two decimal places.

Well, experiment with the KML export settings to see what they do.

5.2.1. Note on Encoding

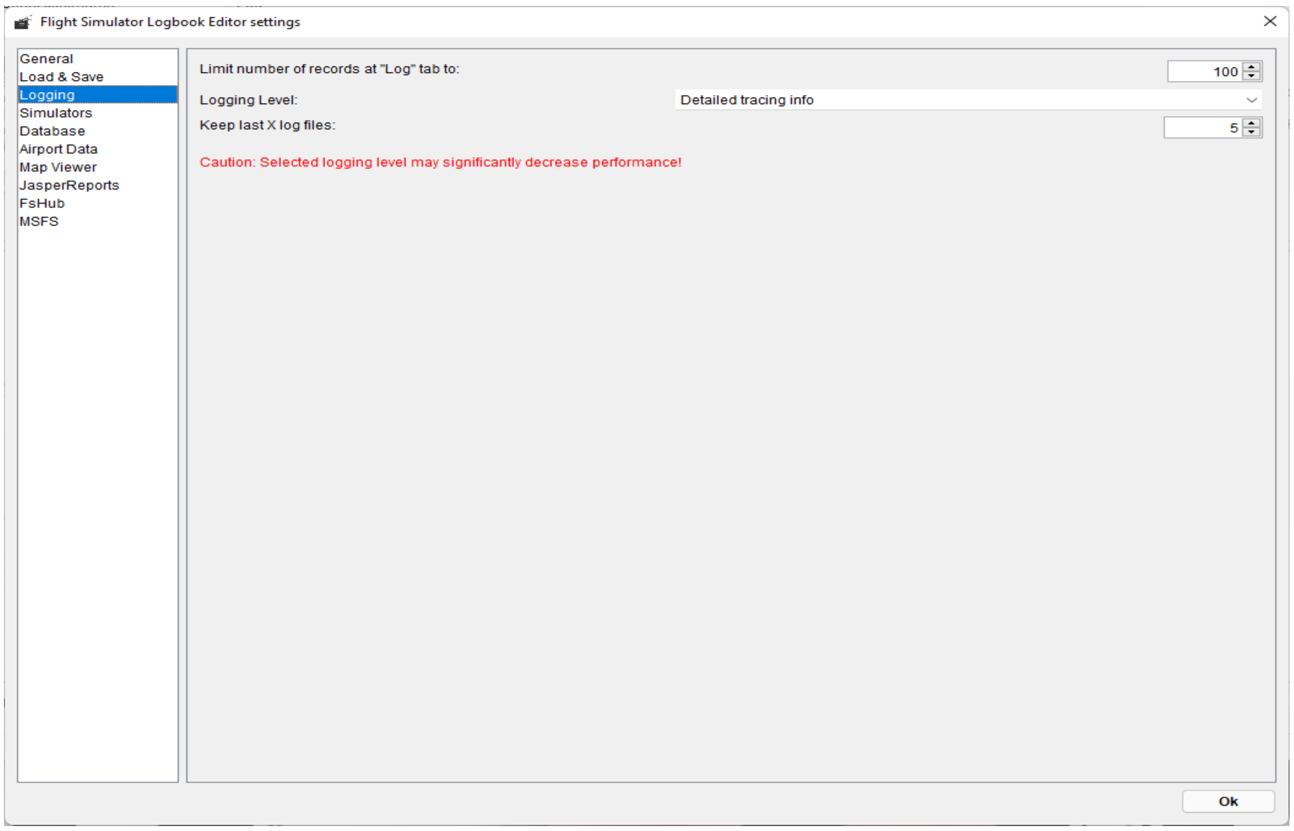
The encoding option, that was present in the Load and Save section of Settings, was there since some of the very first versions of FS Logbook Editor. It was removed in version 1.42, because it has been separated for each file format that needs it. The reason to know know the encoding is actually quite old, goes back to when someone decided we are not going to speak the same language all around the world. So we have multiple languages that may use different characters to express people thoughts. And for the computer to display these characters, all the different encoding exists – to have your flight comments correctly displayed. But, the design of some logbook file formats makes it practically impossible for FS Logbook editor to detect the encoding from logbook file automatically. That is the reason why there are options to set encoding for respective logbook file formats in the settings. If you flight comments are displayed incorrectly, the encoding settings will need to be changed. The respective encoding will be used while reading the logbook file and while saving to it (if not overridden during load or save). Encoding for FS2004 and X-Plane logbook files is set in the ***Load & Save*** section of ***Settings***. FSX/P3D encoding can be set while opening or saving the logbook file. Please note, that in the case of some encoding, the load or save may fail. If this happens, try another encoding as if the comments were displayed incorrectly. The encoding commonly used by Windows systems are:

- **windows-1250** – character encoding used in Windows for certain Central and Eastern European languages using Latin script (Polish, Czech, Slovak, Hungarian, Slovene, Bosnian, Croatian, Serbian (Latin script), Romanian and Albanian). May also be used with German.
- **windows-1251** – character encoding used for languages that use Cyrillic script such as Russian, Bulgarian, Serbian and Macedonian.
- **windows-1252** – character encoding of the Latin alphabet, for English versions of Windows (and some other western languages)
- **windows-1253** – character encoding used in Windows for modern Greek. It is not capable of supporting the older Greek.
- **windows-1254** – character encoding used in Windows for Turkish.
- **windows-1255** – character encoding used in Windows for Hebrew.

- **windows-1256** – character encoding used in Windows for Arabic (and possibly some other languages that use Arabic script, like Persian and Urdu).
- **windows-1257** – character encoding used in Windows for Estonian, Latvian and Lithuanian.
- **windows-1258** – character encoding used in Windows for Vietnamese texts.

By default, the **windows-1252** encoding is used by FS Logbook Editor for FS 2004 logbook files and FSX and P3D logbook files up to Prepar3D v3. For Prepar3D v4, the default encoding is **UTF16-LE**. For X-Plane logbook files, the default encoding is **UTF-8**. When you first start the application, settings dialog will appear allowing you to select encoding before the very first opening of any logbook.

5.3. Logging


Flight Simulator Logbook Editor settings

Limit number of records at "Log" tab to: 100

Logging Level: Detailed tracing info

Keep last X log files: 5

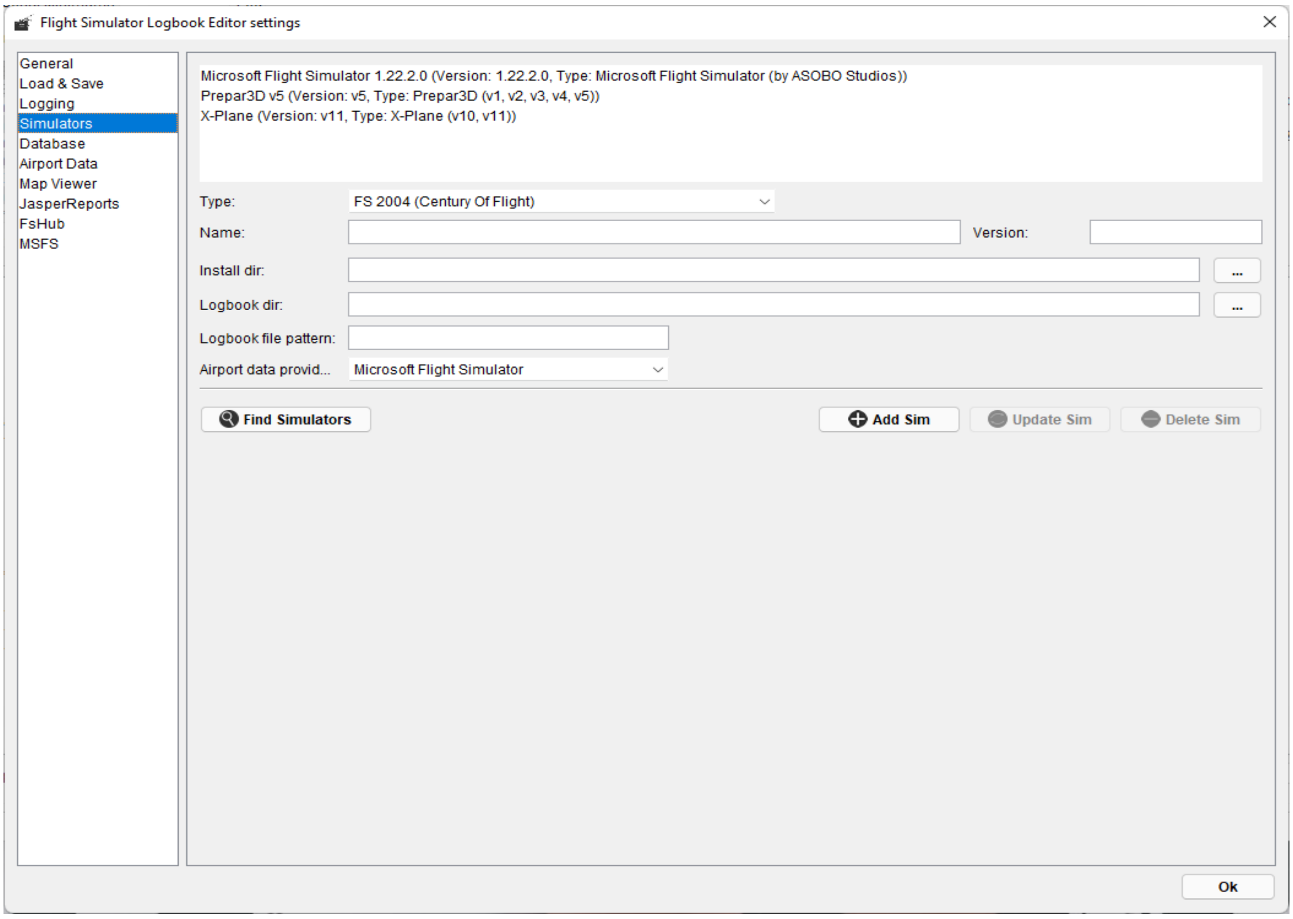
Caution: Selected logging level may significantly decrease performance!

Picture 57: Logging section of Settings

Here you can change some logging options, such as:

- **Limit number of records at "Log" tab to:** - here you can limit the amount of log messages that will be visible in the log view. Continuous printing of log messages into the log view would result in rising memory consumption which in turn may after some time slow down the program and the whole system, crashing it in the worst case. Default value is 100 – after 100 printed messages, the log view is cleared and another 100 messages has their space. This setting does not limit messages logged into log file, that is automatically created for each program session (execution). See ***Log files*** chapter for more info.
- **Logging level:** - sets the level of logging. This affects the amount and details of messages being logged and is a bit advanced feature reserved for troubleshooting. I recommend leave this on default value – Information messages. Some other value can result in a lot of log messages and can hit performance pretty hard.
- **Keep last X log files:** - As mentioned above and in the ***Log files*** chapter, log file is automatically created for each program session (execution). To keep program folder clean, the program deletes old log files before it is terminated keeping only the last ones – in the amount set here.

5.4. Simulators


Flight Simulator Logbook Editor settings

Microsoft Flight Simulator 1.22.2.0 (Version: 1.22.2.0, Type: Microsoft Flight Simulator (by ASOBO Studios))
Prepar3D v5 (Version: v5, Type: Prepar3D (v1, v2, v3, v4, v5))
X-Plane (Version: v11, Type: X-Plane (v10, v11))

Type: FS 2004 (Century Of Flight)
Name:
Version:
Install dir:
Logbook dir:
Logbook file pattern:
Airport data provid... Microsoft Flight Simulator

Picture 58: Simulators section of Settings

Please read the ***News in 1.4 version*** chapter for some introduction on what the hell is this for? Got it? Ok, so this is the place where you can define flight simulators you have. These flight simulators will serve as airport data providers / sources and also automatic search of flight simulator logbooks function uses them.

5.4.1. Fields description

Lets start from the top, there is a simulators listing – it contains all simulators you have defined. Below this listing, there are simulator properties:

- **Type** – supported flight simulator type. This is mainly for completeness. It is used by logbook auto-find function to decide where to look for flight simulator logbook files for specific flight simulator (The logbook auto-find function works even when no simulator is defined in this section). See the **Automatically load logbook file** option at ***General*** section of Settings for more details.
- **Name** – a name of the flight simulator. Does not have to be exact, it is just for you to be able to distinguish one simulator from another.
- **Version** – a version of the simulator. Does not have to be exact, nor a number (but can be a number). For example, for boxed edition of Flight Simulator X, I would type there “Boxed”, for various version of Prepar3D I would type “v1”, “v2” or “v3” respectively. **The important thing is**, that **Version** and **Airport data provider** forms and **unique flight simulator** for FS Logbook Editor (it will not allow you to create 2 flight simulators with the same Airport data provider and Version)!

- **Install dir** – a path to the directory where the simulator is installed. You can browse to it using the button with 3 dots on the right side of the input field.
- **Logbook dir** – a path to the directory where the simulator stores logbook files. You can browse to it using the button with 3 dots on the right side of the input field.
- **Logbook file pattern** – a file name pattern that will be used by FS Logbook Editor to filter out logbook files from **Logbook dir**. Should support standard windows wild-chars, such as *. For example, the default value for Flight Simulator X and Prepar3D logbook files is ***.bin,** so any file in **Logbook dir** having a **bin** extension will be considered a logbook file. FS Logbook Editor uses this in the logbook auto-find function and to filter files in the open / save dialog. See the **Automatically load logbook** option in ***Load & Save*** section of ***Settings*** and the **Open Logbook (Auto find...)** option in ***File*** menu.
- **Airport data provider** – select which airport data provider to use with this simulator. Well, the description here is little ambiguous, as for FS Logbook Editor, the simulator being defined is an airport data provider. But here, the provider means one of predefined function, that will be used to actually read the airport data (as this process is different for each simulator). The supported providers are:
  - **FS2004 / Flight Simulator X / Prepar3D** – this provider can read airport data from Flight Simulator 2004, Flight Simulator X or Prepar3D. Well, actually the MakeRwys utility from Peter Dowson does read the airport data and FS Logbook Editor just reads the generated file. This should work with Flight Simulator 2002 (any simulator supported by MakeRwys), but it was NOT tested.
  - **X-Plane** – This one supports reading of airport data from X-Plane 10/11 flight simulator. May work with previous versions too (NOT tested).
  - **Navigraph (AIRAC)** – this function / airport data provider can read airport data from Navigraph data cycle files.

That was the description of flight simulator attributes (those one important for FS Logbook Editor). Now, there are four buttons below these to work with them.

5.4.2. Adding flight simulator

To add a flight simulator, fill in fields described in the previous ***Fields description*** chapter and click **Add Sim** button. If any of your input values are wrong, you will be notified with a message box. Of course, if all the values are correct, the simulator will be added to the flight simulators listing at the top of the dialog. It will be also stored to the database, so it is there next time you launch FS Logbook Editor. A result message will appear below the buttons informing you whether the simulator was successfully saved or not. This message is green (everything is Ok) or red (it failed completely).

5.4.3. Updating flight simulator

Every time you select a flight simulator from the list, the fields below the list will be filled with values set for the selected flight simulator. If you want to change any of them, just change them. When edit all the values you want, click the **Update Sim** button and the actual values will be saved within the currently selected flight simulator. As in the case when adding flight simulator, a result message will be displayed below the buttons. It will be green (everything is Ok) or red (it failed completely).

5.4.4. Deleting flight simulator

To delete flight simulator, simply select it in the listing and click **Delete Sim** button. The simulator will be deleted from the listing and from the database. Also, any airport data imported using the deleted flight simulator (airport data provider) will also be deleted from the database. A result message will be displayed below the buttons. It will be green (everything is Ok) or red (it failed completely).

5.4.5. Automatically discovering flight simulators

The single button on the left, titled **Find Simulators** will trigger the function that will try to automatically find your flight simulators and filling the listing for you. All in single click. It does this by searching the registry, hard-drive and performing various tests whether the found installation is correct. As with almost everything that FS Logbook Editor does, you can find some messages in the ***Error: Reference source not found*** informing you about the search process.

5.4.6. Example – adding "dummy" flight simulator for importing Navigraph airport data

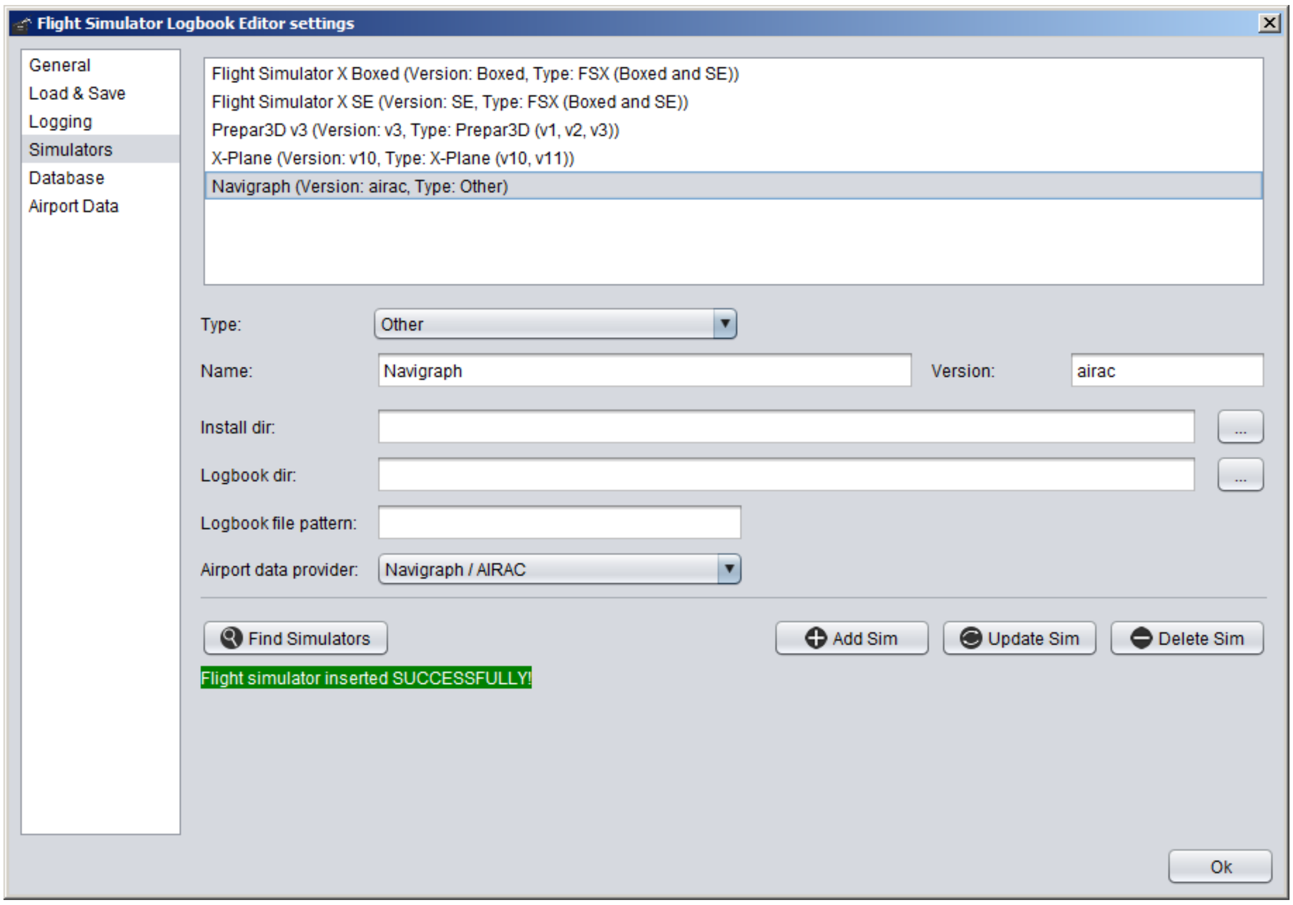
FS Logbook Editor can import airport data from Navigraph data cycle. But this kind of breaks concept that FS Logbook Editor imports airport data from "airport data providers", which are defined flight simulators. But Navigraph data are not flight simulator, they are just … data, that dozen of other flight simulator products use for similar purposes as FS Logbook Editor. So for FS Logbook Editor to be able to import airport data from Navigraph data cycle, you must define a dummy flight simulator, which will have **Airport data provider** set to **Navigraph**. Here is how to do it:

Fill the required fields:

- **Type**: Other
- **Name**: Navigraph
- **Version**: airac
- **Airport data provider**: Navigraph / AIRAC

You can fill other values in the Name and Version field, the important one is the Airport data provider. The following picture shows the fields filled and the simulator added:

<content follows on next page>


Flight Simulator Logbook Editor settings

Flight Simulator X Boxed (Version: Boxed, Type: FSX (Boxed and SE))
Flight Simulator X SE (Version: SE, Type: FSX (Boxed and SE))
Prepar3D v3 (Version: v3, Type: Prepar3D (v1, v2, v3))
X-Plane (Version: v10, Type: X-Plane (v10, v11))
Navigraph (Version: airac, Type: Other)

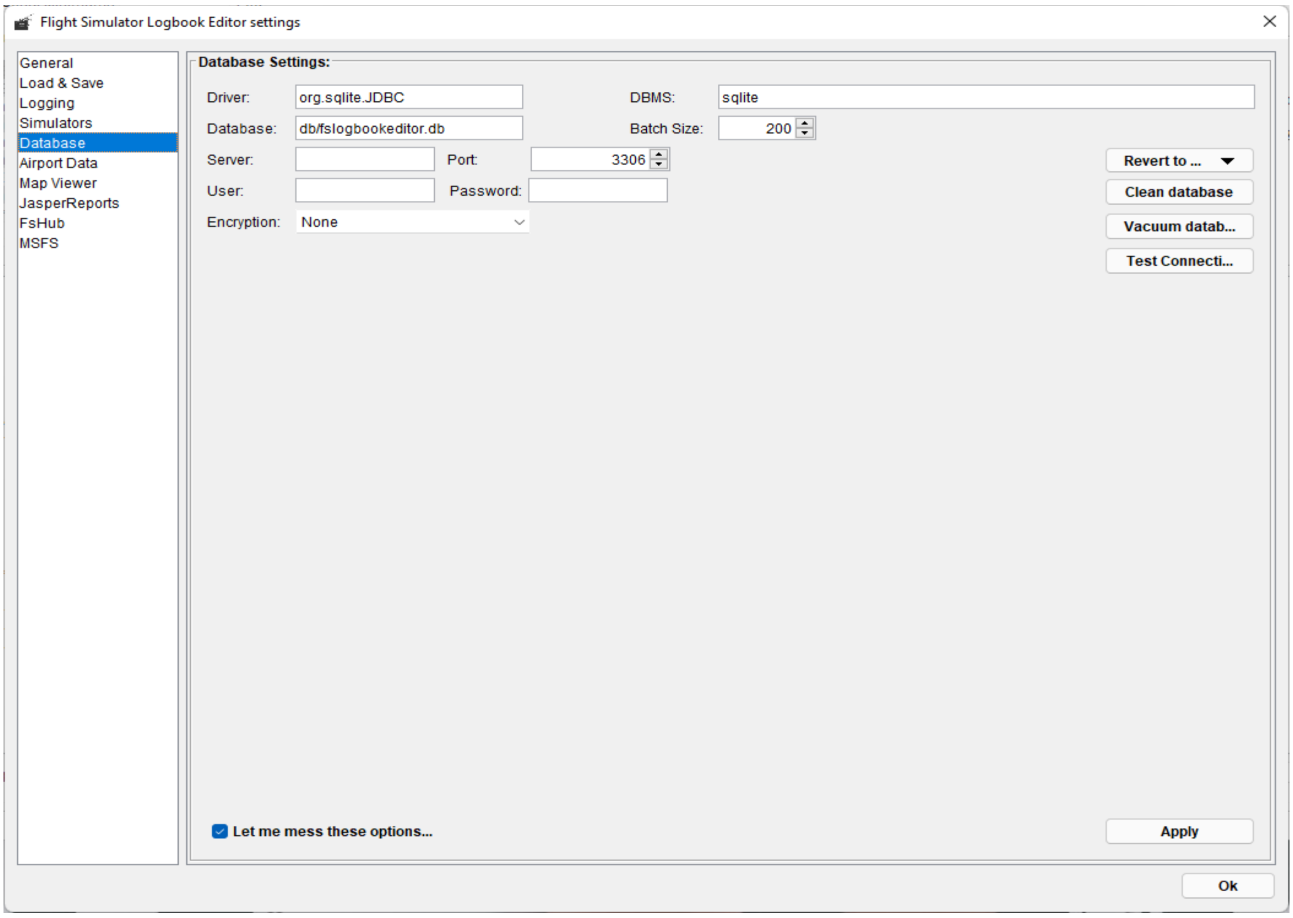
Type: Other
Name: Navigraph
Version: airac
Install dir:
Logbook dir:
Logbook file pattern:
Airport data provider: Navigraph / AIRAC

Flight simulator inserted SUCCESSFULLY!

Picture 59: Dummy "Navigraph" flight simulator created

That’s it. Now you can import Navigraph data at ***Airport Data*** section. See ***Importing airport data from Navigraph data cycle***.

5.5. Database


Flight Simulator Logbook Editor settings

Database Settings:

Driver: org.sqlite.JDBC
DBMS: sqlite
Database: db/fslogbookeditor.db
Batch Size: 200
Server:
Port: 3306
User:
Password:
Encryption: None

[✓] Let me mess these options...

Picture 60: The Database section of Settings

Most of the database section is for advanced users. You should leave this as it is. FS Logbook Editor stores airport data for different simulators in database and looks there for them when needed. The default database is created automatically and the options and buttons in this section are mainly for troubleshooting (while messing with these can cause the trouble at the first place). So I will describe only the buttons here. If you are about to edit the grayed values, you should be experienced enough to know what they mean anyway.

In FS Logbook Editor version 1.5 the database support has been completed to correctly support database systems other than the default SQLite. MySQL, MariaDB and Microsoft SQL Server (2008 and above) are now also supported[4]. So now it actually makes sense to play with these options if you want to store airport and simulators data in other database system. If you want to use MySQL, MariaDB or Microsoft SQL Server, you must create database for FS Logbook Editor and type it in the Database field. FS Logbook Editor will not create the database by itself. But that should be no problem. Since you decided to change these options, I am sure you have the respective SQL server running and configured (saying you should know what you are doing).

Please NOTE that FS Logbook Editor will not ask to confirm an action, after you click some button, the action will take place.

4 *FS Logbook Editor has been tested with MySQL v8, MariaDB v10.2 and Microsoft SQL Server Express 2019 at the time of releasing version 1.80*

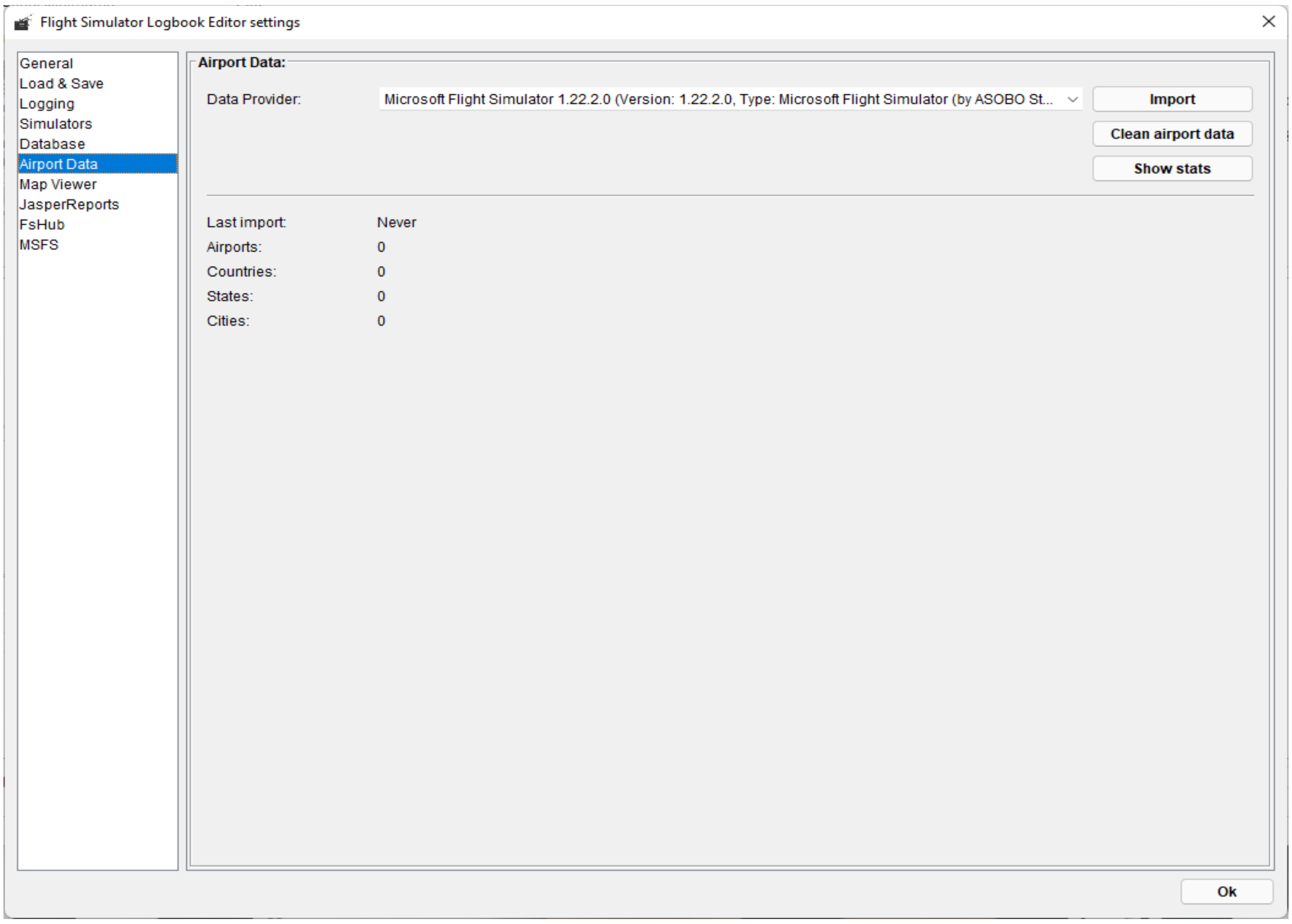
- **Let me mess these options…** - checking this check box will enable the grayed items for you to play with them.
- **Revert to defaults** – this will set all, by default grayed values, to their default values. **This has changed to list-button in FS Logbook version 1.5** and it allows you to select to fill in default values for supported database systems – SQLite, MySQL, MariaDB and Microsoft SQL. Just click the arrow and select which default values you want to restore.
- **Clean database** – Clicking this option will **CLEAN**, meaning **ERASE** all data from FS Logbook Editor database. All defined flight simulators from ***Simulators*** section, and all airport data will be **GONE!**
- **Vacuum database –** this will try to reduce the size of the database and optimize it. No data loss occurs by this operation. This should not be needed to perform, as FS Logbook Editor does this automatically after operations that are likely to create mess in the database, such as deleting.
- **Test connection** – this tries to connect to database defined by the options.
- **Apply** – After you click this option, FS Logbook Editor will try to connect to database using the newly specified options (if you changed some of them). If it succeeds, FS Logbook Editor will disconnect from the current database and reconnect to the newly specified one. It will also check if the database contains required tables and will offer to create them if they do not exists. You may need to redefine flight simulators and import airport data if the newly connected database is empty. From now on, the new database is in use.

Since FS Logbook Editor 1.5 you have to click the Apply button if you want to apply new database connection settings. If you just change the connection settings and close the Setting dialog, the changes will NOT be saved and the old setting will remain in use. But FS Logbook Editor will warn you if you change database connection settings and don't apply them.

Since FS Logbook Editor version 1.8, there is a new **Encryption** combo box. This allows to enable SSL/TLS encrypted connection to the database. Note that this function is kind of experimental – basic. It is not yet possible to specify certificates to trust, so, the server certificate is always trusted and is not being verified. This option has no effect when using SQLite database (default settings). There are 3 options to select from now:

- **None** – No encryption. SSL/TLS disabled.
- **If Available** – Works only with MySQL database. Will try to connect using encrypted connection, but will use un-encrypted connection if case of failure. For MariaDB and Microsoft SQL this is the same as the **Required** option.
- **Required** – Encrypted connection is required and FS Logbook Editor will fail connecting to the database if encrypted connection cannot be established.

5.6. Airport Data


Flight Simulator Logbook Editor settings

Airport Data:

Data Provider: Microsoft Flight Simulator 1.22.2.0 (Version: 1.22.2.0, Type: Microsoft Flight Simulator (by ASOBO St...

Last import: Never
Airports: 0
Countries: 0
States: 0
Cities: 0

Picture 61: The Airport Data section of Settings

Please read the ***News in 1.4 version*** chapter for some introduction on what the hell is this for? Also make sure you defined at least one flight simulator in the ***Simulators*** section, otherwise, you won't be able to do anything here.

Got it? Is there at least one item in the **Data Provider** selection? Ok, so you defined at least one flight simulator (called Data Provider here), so now you can import airport data from it. Then, FS Logbook Editor should be able to compute flight distances from your logbook, display things such as airport name and city and do a KML export. There are 3 buttons:

- **Import data** - the most important one. Clicking this button will start the import process for selected data provider (flight simulator). See the following sub-chapters which describes the process for some of supported flight simulators.
- **Clean airport data** – click this button if you want to delete airport data for selected data provider, for whatever reason.
- **Show stats** – this will show airport data statistics for selected data provider, such as number of airports, cities, countries and states in database as well as the date of last data import (for selected data provider).

Clicking any button will display a result message above the horizontal line (on the left of Show stats button). When the result is Ok, the message will be green, when there is a failure, the message will be red. And when the things go semi-bad, the message will be yellow, as warning. The message stays there until you perform some other action.

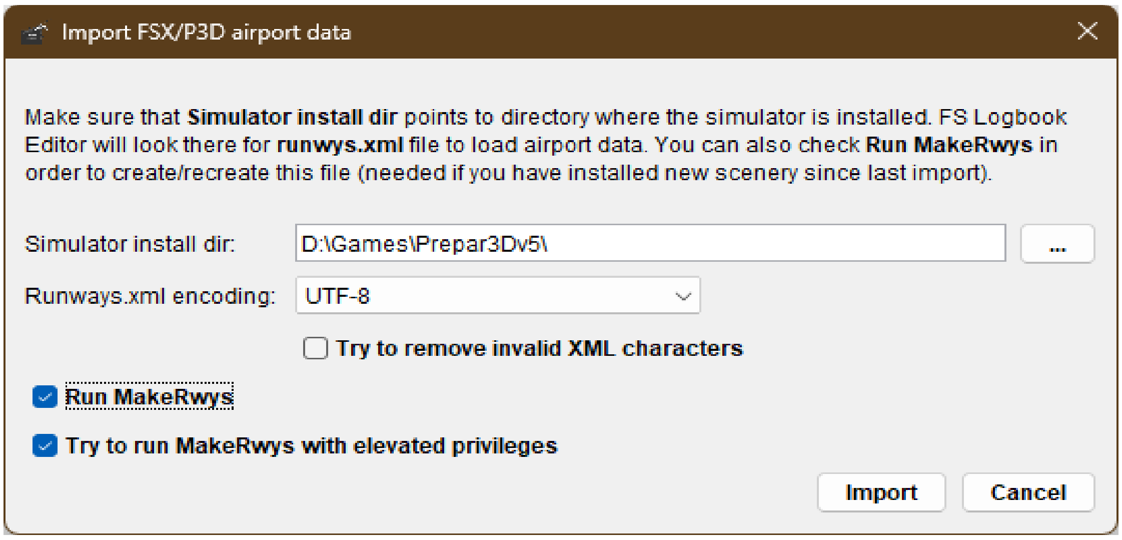
5.6.1. Importing airport data for FS2004 / FSX / Prepar3D

The import process for all these simulators is the same. FS Logbook Editor does not read airport data by itself, but uses **MakeRwys.exe** program, which is program from **Peter Dowson**, the author of many must-have flight-sim tools. FS Logbook reads **runways.xml** file which is generated by MakeRwys.exe and imports the information needed from that file.

Since Prepar3D v3, I think, certainly in v4 and v5 it is possible to also install sceneries via the Add-on method. Newer versions of MakeRwys.exe are distributed with new program – LorbySceneryExport.exe – which helps MakeRwys to scan all sceneries (included the ones installed as add-ons) and thus generate the complete runways.xml file. Since FS Logbook Editor v 1.65 this tool is also part of FS Logbook Editor distribution. LorbySceneryExport.exe is free tool written by Oliver Binder (Lorby-SI) – a developer of many other great free and payware tools for flight simulators. Check them out too ;)

This means, you must have a copy MakeRwys.exe and LorbySceneryExport.exe to generate complete runways.xml file, so FS Logbook Editor can import data. But don't worry, even if you don't have any of those tools nor the runways.xml file generated, FS Logbook Editor comes with a copy of MakeRwys.exe and LorbySceneryExport.exe (since version 1.65). It will copy both of these tools to the folder of selected flight simulator and instruct MakeRwys to build the runways.xml file if needed, or if you say (check) so.

Select the Data provider (flight simulator) you want to import airport data for and click **Import**, which will bring the following dialog on your screen:


Import FSX/P3D airport data

Make sure that **Simulator install dir** points to directory where the simulator is installed. FS Logbook Editor will look there for **runwys.xml** file to load airport data. You can also check **Run MakeRwys** in order to create/recreate this file (needed if you have installed new scenery since last import).

Simulator install dir: D:\Games\Prepar3Dv5\

Runways.xml encoding: UTF-8

☐ Try to remove invalid XML characters

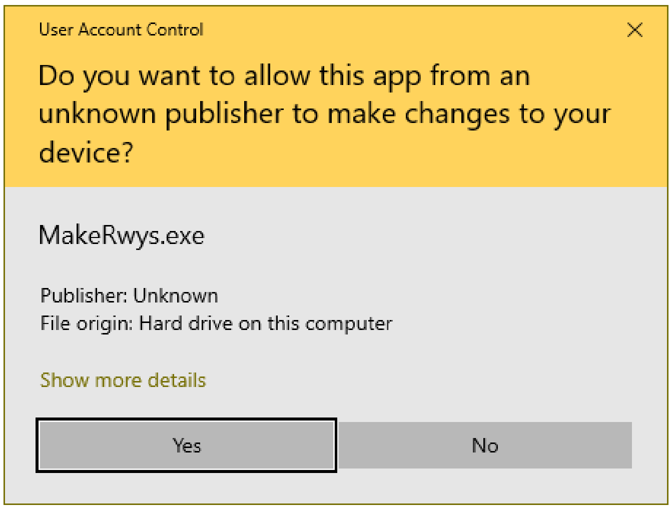
☑ Run MakeRwys

☑ Try to run MakeRwys with elevated privileges

Picture 62: FS2004 / FSX / Prepar3D airport data import dialog

The **Simulator install path** should show valid flight simulator installation directory (for FS2004, FSX or Prepar3D). Checking the **Run MakeRwys** check box will cause FS Logbook Editor to run the MakeRwys.exe tool before the import. This is important to check if you have installed/removed sceneries. The check box will be automatically checked if there is no **runways.xml** file in the simulator install path.

FS Logbook Editor version 1.85 introduced new option – **Try to remove invalid XML characters**. It may sometimes happen than the **runways.xml** contains invalid XML characters, which makes the file "invalid" and this leads to failure during import of airport data (even though the file is not really corrupted in the sense that the structure would be wrong). To allow FS Logbook Editor continue in the import, this option was added. So, if you enable it, FS Logbook Editor will try to remove those invalid characters to prevent failure of processing the **runways.xml** file. **But note that this can significantly slow down the import process. So you should use it only if the import with the option deselected fails.**



Do you want to allow this app from an unknown publisher to make changes to your device?

MakeRwys.exe

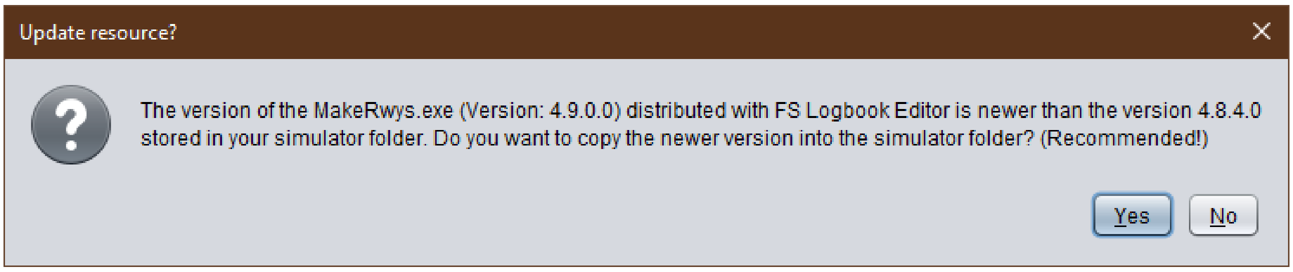
Publisher: Unknown
File origin: Hard drive on this computer

Show more details

Picture 63: UAC prompt to run MakeRwys

There is also a new option **Try to run MakeRwys with elevated privileges** (added in version 1.65). This option should be on every time the **Run MakeRwys** is on. The reason is that the **LorbySceneryExport.exe tool needs admin rights to be able to scan all of the add-on sceneries**. As MakeRwys is running LorbySceneryExport.exe (if it is present) before building the runways.xml file, running MakeRwys with admin rights makes sure that LorbySceneryExport will also run with admin rights, thus you will get the most complete runways.xml file and most complete airport database in FS Logbook Editor. If FS Logbook Editor will succeed with running the tools with admin rights, you will see the following prompt8yyyx**Yes** to allow MakeRwys run with admin rights. If you click No (by mistake or intentionally), MakeRwys will run with standard privileges. But in that case, it will fail running the LorbySceneryExport tool and the runways.xml will miss the sceneries installed using the Add-on method.

Since FS Logbook Editor version 1.65, the version of MakeRwys.exe and LorbySceneryExport.exe contained within the selected simulator folder is also being checked. If it is older than the versions distributed with FS Logbook Editor, you will be prompted if you want to update them (by simply overwriting them).



The version of the MakeRwys.exe (Version: 4.9.0.0) distributed with FS Logbook Editor is newer than the version 4.8.4.0 stored in your simulator folder. Do you want to copy the newer version into the simulator folder? (Recommended!)

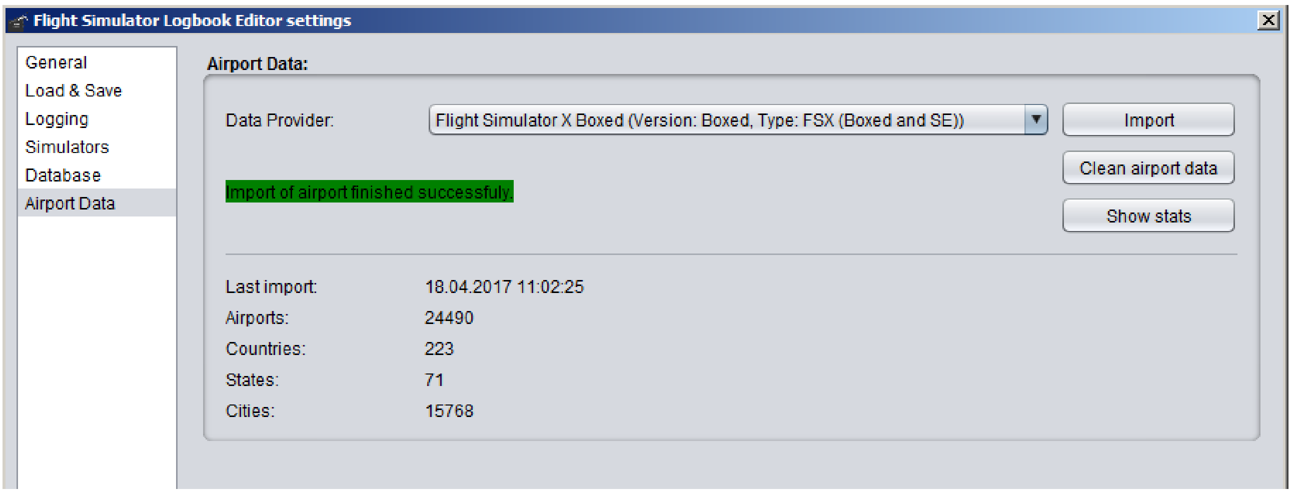
Picture 64: Question dialog about updating MakeRwys

It is recommended to update – clicking Yes button. But you can of course click No, if you keep some specific version for your specific reasons there.

You can also change **Runways.xml encoding** option. By default, there will be UTF-8, but this may differ on your Windows version due to the language you use. For western European languages, like on my Czech windows, this should be ISO-8859-1 to get the correct characters in airport names etc.

If you do not know how to determine the encoding of the file, you will have to do it by trial and error.

Click **Import** to start the import process. A hopefully positive import result should be shown when import is done, such as on the following picture:



Airport Data:

Data Provider: Flight Simulator X Boxed (Version: Boxed, Type: FSX (Boxed and SE))

Import of airport finished successfuly.

Last import:	18.04.2017 11:02:25
Airports:	24490
Countries:	223
States:	71
Cities:	15768

Picture 65: The result of FS2004/ FSX / Prepar3D airport data import

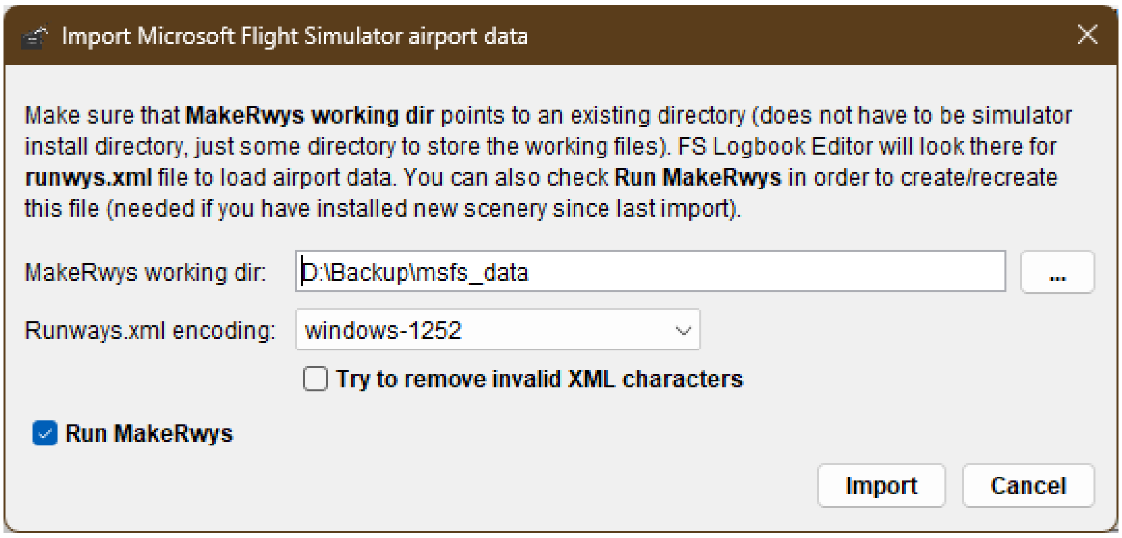
5.6.2. Importing airport data for Microsoft Flight Simulator

The import process for the Microsoft Flight Simulator simulators is almost the same as for FSX/Preapr3D. FS Logbook Editor does not read airport data by itself, but uses **MakeRwys.exe** program, which is program from **Peter Dowson**, the author of many must-have flight-sim tools. FS Logbook reads **runways.xml** file which is generated by MakeRwys.exe and imports the information needed from that file.

The difference between importing airport data from FXS/Prepar3D is that the MakeRwys.exe no does not need to be stored in MSFS install folder. It can stored in any folder and it will find the required folders and data by itself. Thus, the import dialog does not specify the **Simulation install dir**, but the **MakeRwys Working dir**. It may be any folder you like. Note that the generated files might get quite large (it was round 740MB on my system) and the scanning may take quite long (even 10 minutes or more), so be patient, while the dots are running, it is doing something.

If you have Microsoft Flight Simulator selected as airport data provider, click **Import**, which will bring the following dialog on your screen:

<content follows on next page>


Import Microsoft Flight Simulator airport data

Make sure that **MakeRwys working dir** points to an existing directory (does not have to be simulator install directory, just some directory to store the working files). FS Logbook Editor will look there for **runwys.xml** file to load airport data. You can also check **Run MakeRwys** in order to create/recreate this file (needed if you have installed new scenery since last import).

MakeRwys working dir: D:\Backup\msfs_data

Runways.xml encoding: windows-1252

☐ Try to remove invalid XML characters

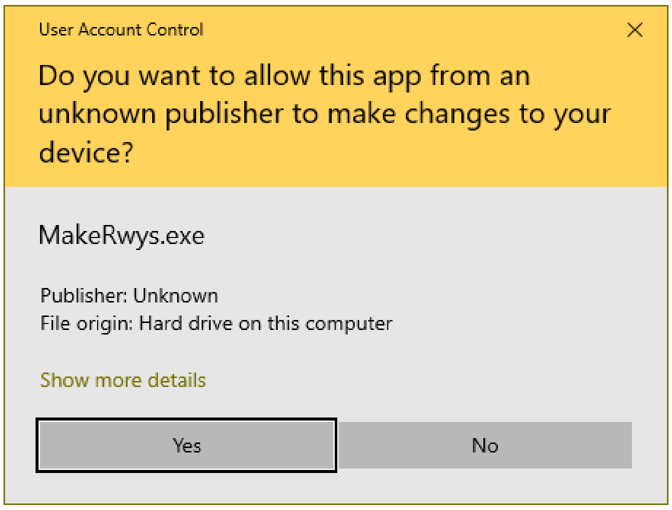
☑ Run MakeRwys

Picture 66: Microsoft Flight Simulator airport data import dialog

Checking the **Run MakeRwys** check box will cause FS Logbook Editor to run the MakeRwys.exe tool before the import. This is important to check if you have installed/removed sceneries. The check box will be automatically checked if there is no **runways.xml** file in the **MakeRwys working dir**.

FS Logbook Editor version 1.85 introduced new option – **Try to remove invalid XML characters**. It may sometimes happen than the **runways.xml** contains invalid XML characters, which makes the file "invalid" and this leads to failure during import of airport data (even though the file is not really corrupted in the sense that the structure would be wrong). To allow FS Logbook Editor continue in the import, this option was added. So, if you enable it, FS Logbook Editor will try to remove those invalid characters to prevent failure of processing the **runways.xml** file. **But note that this can significantly slow down the import process. So you should use it only if the import with the option deselected fails.**

The option to run MakeRwys with elevated permissions is missing on this dialog. It is mandatory to run it with elevated privileges in order to read MSFS airport data correctly. Thus, FS Logbook Editor will always try to run it with elevated privileges. If FS Logbook Editor succeeds with running the tools with admin rights, you will see the following prompt:


User Account Control

Do you want to allow this app from an unknown publisher to make changes to your device?

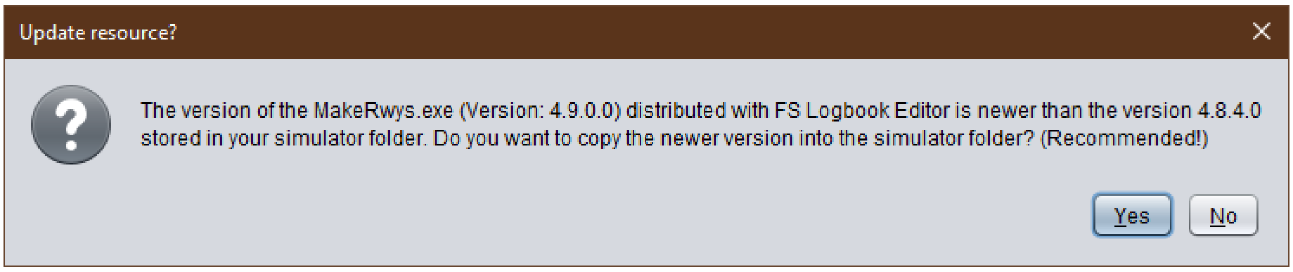
MakeRwys.exe

Publisher: Unknown
File origin: Hard drive on this computer

Picture 67: UAC prompt to run MakeRwys

Click **Yes** to allow MakeRwys run with admin rights. If you click No (by mistake or intentionally), MakeRwys will run with standard privileges. But in that case, it may, probably will fail reading MSFS airport data (or some of them).

FS Logbook Editor will also check the version of MakeRwys.exe contained within working directory and will offer to update it if the version distributed with FS Logbook Editor is newer (by simply overwriting it).


Update resource?

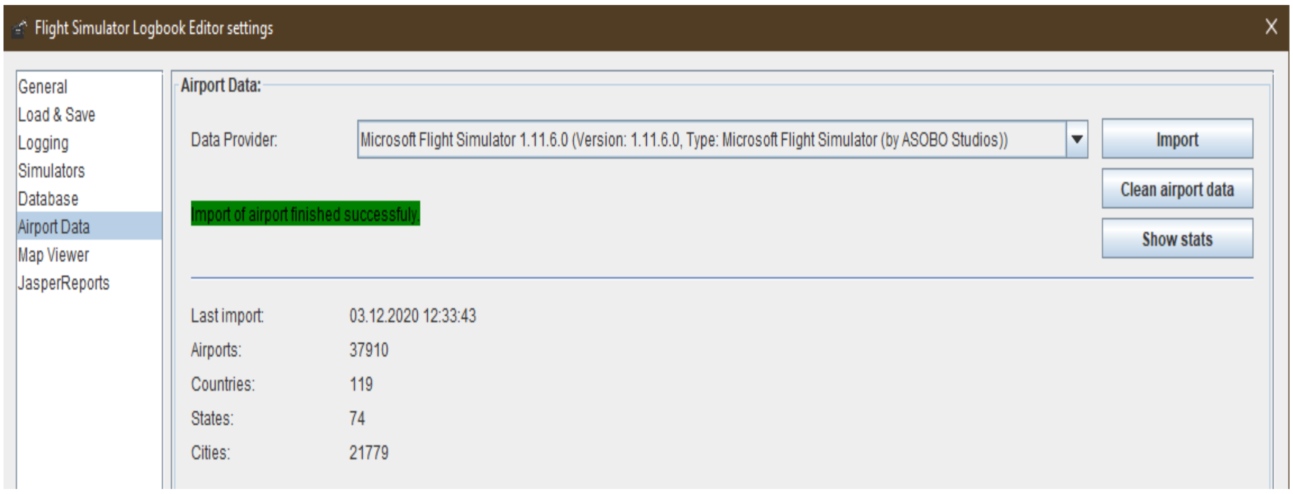
The version of the MakeRwys.exe (Version: 4.9.0.0) distributed with FS Logbook Editor is newer than the version 4.8.4.0 stored in your simulator folder. Do you want to copy the newer version into the simulator folder? (Recommended!)

Picture 68: Question dialog about updating MakeRwys

It is recommended to update – clicking Yes button. But you can of course click No, if you keep some specific version for your specific reasons there.

You can also change **Runways.xml encoding** option. By default, there will be UTF-8, but this may differ on your Windows version due to the language you use. For western European languages, like on my Czech windows, this should be ISO-8859-1 to get the correct characters in airport names etc. If you do not know how to determine the encoding of the file, you will have to do it by trial and error.

Click **Import** to start the import process. A hopefully positive import result should be shown when import is done, such as on the following picture:


Flight Simulator Logbook Editor settings

Airport Data:

Data Provider: Microsoft Flight Simulator 1.11.6.0 (Version: 1.11.6.0, Type: Microsoft Flight Simulator (by ASOBO Studios))

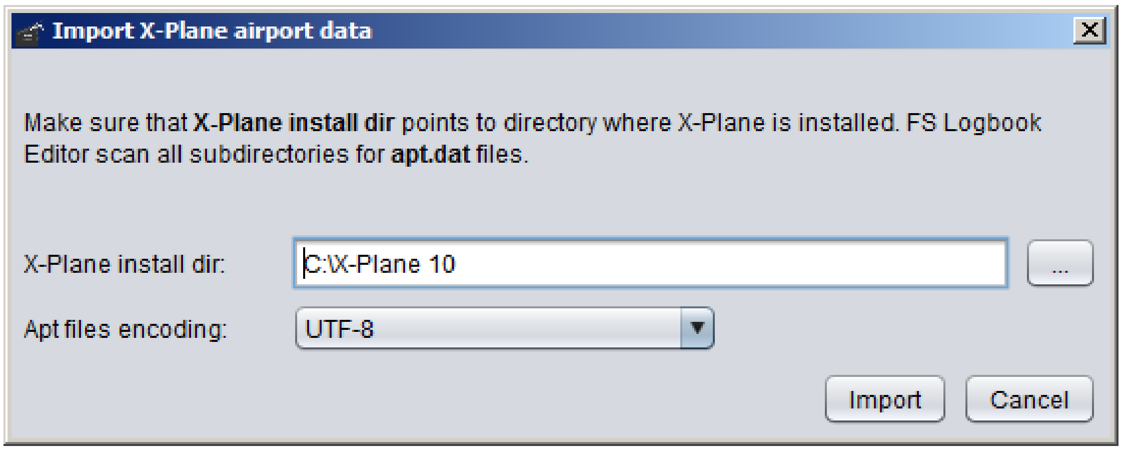
Import of airport finished successfuly

Last import:	03.12.2020 12:33:43
Airports:	37910
Countries:	119
States:	74
Cities:	21779

Picture 69: The result of Microsoft Flight Simulator airport data import

5.6.3. Importing airport data from X-Plane

To import airport data from X-Plane, you must have an X-Plane flight simulator defined. See the ***Simulators*** chapter for the instructions on how to do it. If you have on more, select the one you want to import airport data from from the **Data provider** selection and click **Import**. The following dialog should appear:


Import X-Plane airport data

Make sure that **X-Plane install dir** points to directory where X-Plane is installed. FS Logbook Editor scan all subdirectories for **apt.dat** files.

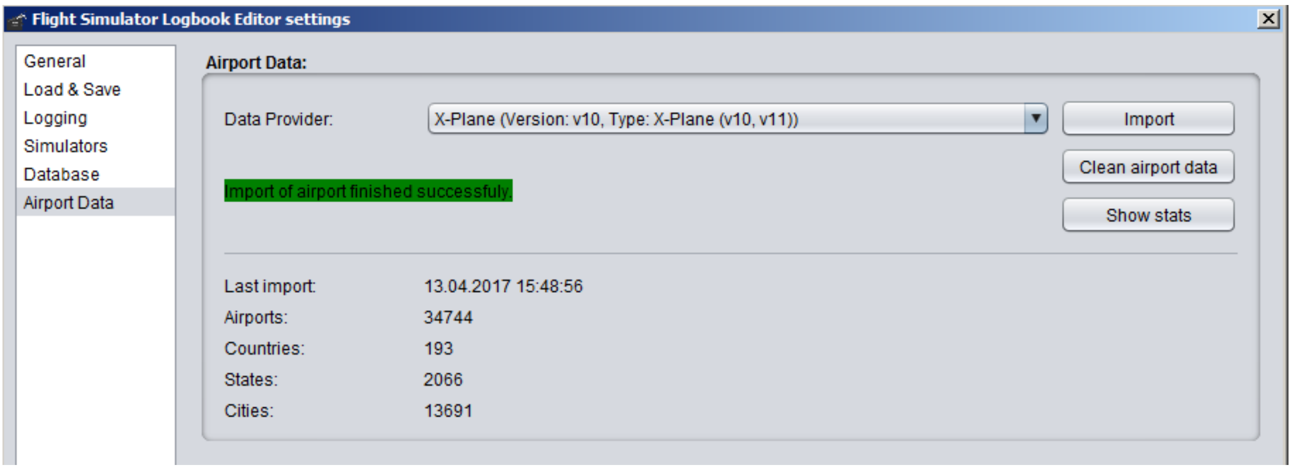
X-Plane install dir: C:\X-Plane 10

Apt files encoding: UTF-8

Picture 70: X-Plane airport data import dialog.

In this dialog you should only confirm that the **X-Plane install dir** is correct and click **Import**. You can of course modify the path to the X-Plane installation directory with the button with 3 dots. Note that this change will not be saved to the defined flight simulator, it will only be used for the import. You can also change the **Apt files encoding** option, which allows you to select different apt files encoding. UTF-8 is there by default and it should work in most cases. Change this to correct encoding in the case the files are encoded in different encoding (when it is, you probably get some weird characters in airport names etc.).

FS Logbook Editor will work for a while scanning X-Plane after you click the Import button. When it is finished, a hopefully positive result should be displayed:


Flight Simulator Logbook Editor settings

Airport Data:

Data Provider: X-Plane (Version: v10, Type: X-Plane (v10, v11))

Import of airport finished successfuly

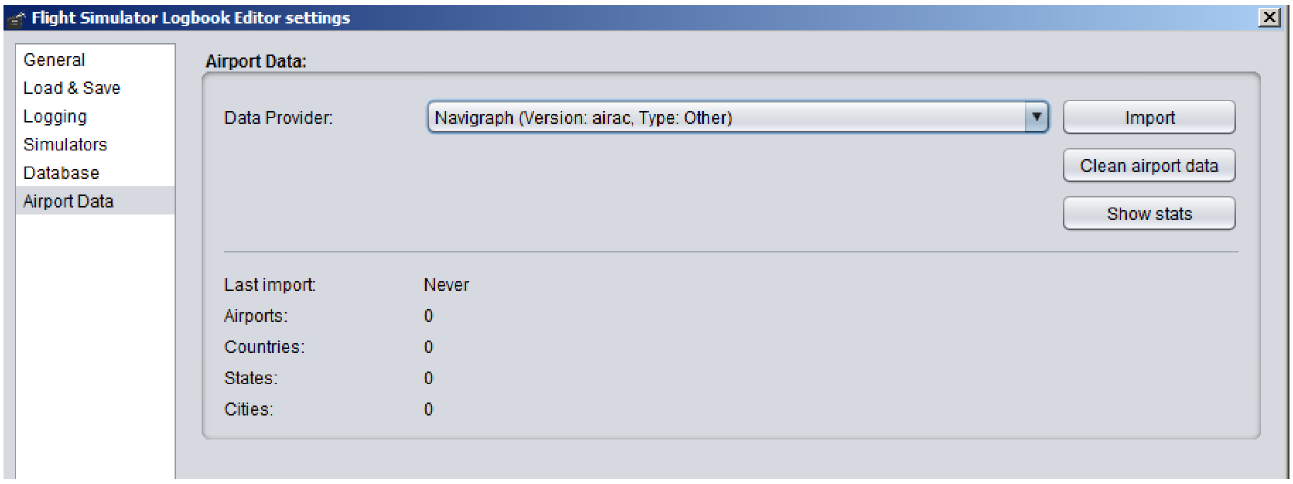
Last import:	13.04.2017 15:48:56
Airports:	34744
Countries:	193
States:	2066
Cities:	13691

Picture 71: The result of X-Plane airport data import

informing that the import has completed, as well as some statistics (such as last import data – which should be just now, count of airports, cities, states etc. imported).

5.6.4. Importing airport data from Navigraph data cycle

If you want the source of your airport data to be the Navigraph data cycle, let it be so. But first, you have to create a "dummy" flight simulator in the ***Simulators*** section. See the ***Example – adding "dummy" flight simulator for importing Navigraph airport data*** chapter. When done, select defined Navigraph data provider as on the picture below and click **Import**:



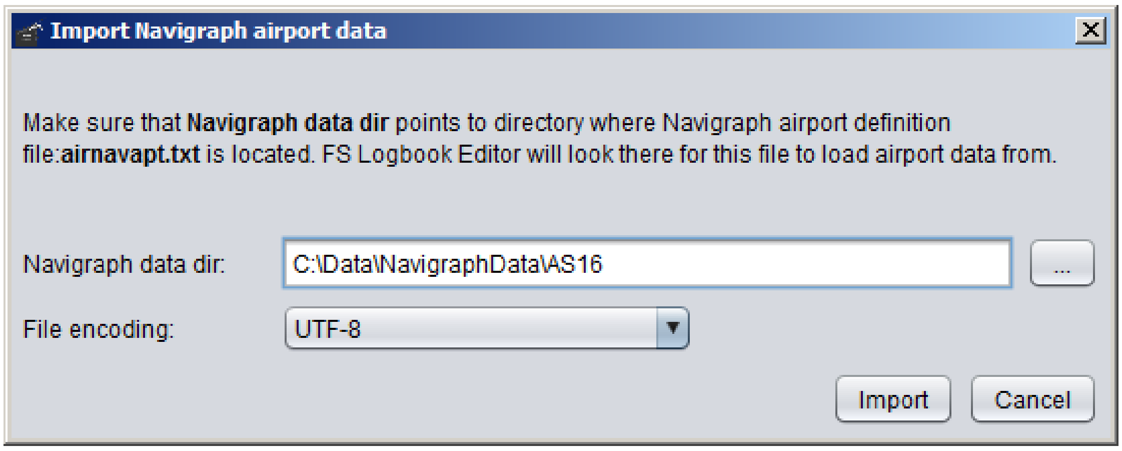
Airport Data:

Data Provider: Navigraph (Version: airac, Type: Other)

Last import: Never
Airports: 0
Countries: 0
States: 0
Cities: 0

Picture 72: Navigraph data provider selected for airport data import.

After you click import, following dialog will appear:



Make sure that **Navigraph data dir** points to directory where Navigraph airport definition file:**airnavapt.txt** is located. FS Logbook Editor will look there for this file to load airport data from.

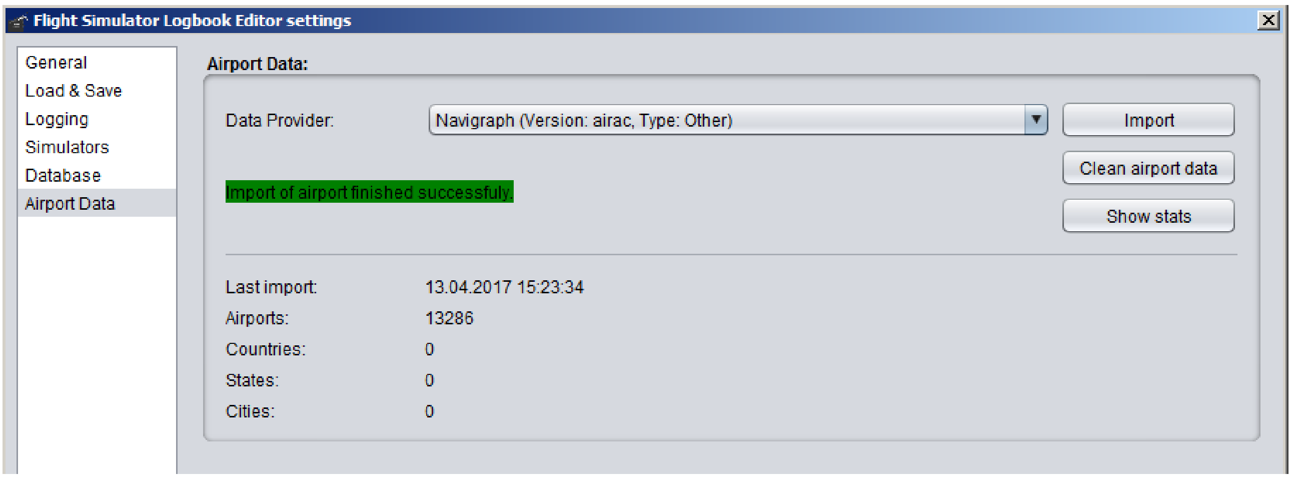
Navigraph data dir: C:\Data\NavigraphData\AS16

File encoding: UTF-8

Picture 73: Navigraph airport data import dialog

Use the button with 3 dots to browse for folder where you have stored your Navigraph data. The folder must contain the **WPNAVAPT.txt** file. It is the file FS Logbook Editor reads the airport data from. If it is not there, you will be notified… You can also change the **File encoding** option, which allows you to select different file encoding. UTF-8 is there by default and it should work in most cases. Change this to correct encoding in the case the **WPNAVAPT.txt** is encoded in different encoding (when it is, you probably get some weird characters in airport names etc.). Then just click **Import** to import the data, or **Cancel** if you changed your mind. After the import is finished, hope successfully, you should see some result like on the picture below:

<content follows on next page>



Airport Data:

Data Provider: Navigraph (Version: airac, Type: Other)

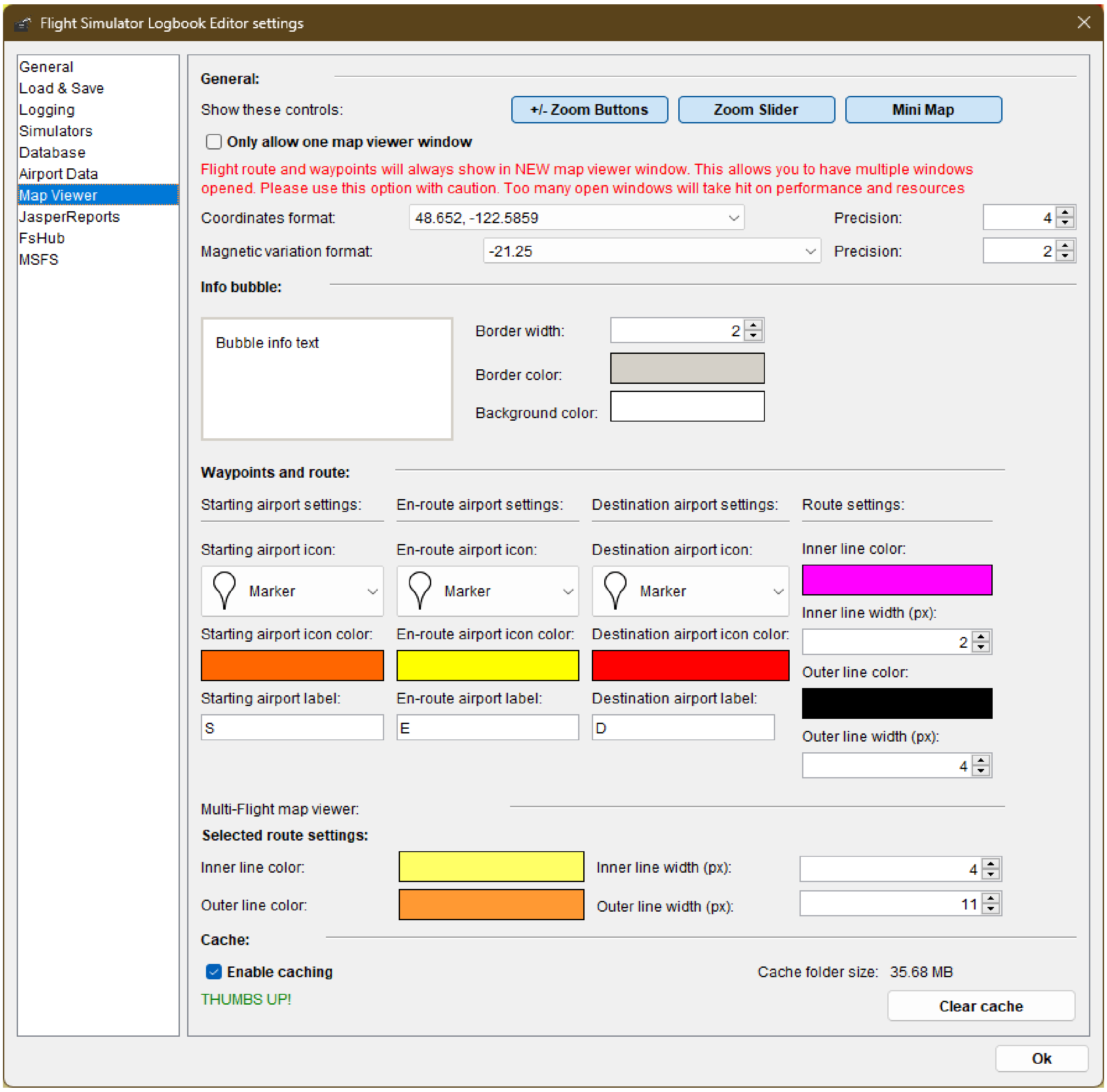
Import of airport finished successfuly.

Last import:	13.04.2017 15:23:34
Airports:	13286
Countries:	0
States:	0
Cities:	0

Picture 74: The result of Navigraph data import

After import is finished, green success message is shown as well as statistics with the last import date (which should be right now) and number of airports imported. Navigraph data does not contain information about airport city, country nor state, so these will read 0.

5.7. Map Viewer


Flight Simulator Logbook Editor settings

General
Load & Save
Logging
Simulators
Database
Airport Data
Map Viewer
JasperReports
FsHub
MSFS

General:

Show these controls: +/- Zoom Buttons | Zoom Slider | Mini Map

☐ Only allow one map viewer window

Flight route and waypoints will always show in NEW map viewer window. This allows you to have multiple windows opened. Please use this option with caution. Too many open windows will take hit on performance and resources

Coordinates format: 48.652, -122.5859 — Precision: 4

Magnetic variation format: -21.25 — Precision: 2

Info bubble:

Bubble info text

Border width: 2

Border color:

Background color:

Waypoints and route:

Starting airport settings:	En-route airport settings:	Destination airport settings:	Route settings:
Starting airport icon: Marker	En-route airport icon: Marker	Destination airport icon: Marker	Inner line color:
Starting airport icon color:	En-route airport icon color:	Destination airport icon color:	Inner line width (px): 2
Starting airport label: S	En-route airport label: E	Destination airport label: D	Outer line color:
			Outer line width (px): 4

Multi-Flight map viewer:

Selected route settings:

Inner line color: — Inner line width (px): 4

Outer line color: — Outer line width (px): 11

Cache:

☑ Enable caching — Cache folder size: 35.68 MB

THUMBS UP!

Clear cache

Ok

Picture 75: Map Viewer section of Settings

At the Map Viewer section of settings you can customize how the flight route will be drawn in the Map Viewer window. See ***Viewing flight on map*** chapter.

This section is split into 3 sections. The first one is, not very surprisingly though, the General section. The first line, labeled **Show these controls**, contains three switch buttons. The **+/- Zoom Buttons** will show/hide the zoom buttons at the left bottom edge of the map (The state of this button is reflected to the +/- button at Map Viewer window). The **Zoom Slider** will show/hide the zoom slider at the left bottom edge of the map above the zoom buttons (The state of this button is reflected to the **Zoom** button at Map Viewer window). And the **Mini Map** button shows/hides the mini map at right bottom edge of the map (The state of this button is reflected to the **Mini Map** button at Map Viewer window). All these three buttons will cause the respective controls to be shown when they are pressed in (they have gray color). When they are not pressed, the controls will be hidden. You can hover the mouse over the button and the tool tip text will tell you whether the controls are enabled or disabled. Read the ***Viewing flight on map*** chapter to see the buttons at Map Viewer window.

Below these there is check box named **Allow only one map viewer window** and it is checked by default. When it is checked, every time you press **View on map** button at Editor tab the flight route will be drawn into the same window (discarding the previous one if any). This means you can view one flight (one window) at a time. When this is unchecked, a new window will open every time you press the mentioned button. So you can view more than one flight route (more windows). But be advised against opening a lot of windows, as it can lead to great resource usage, like memory. Some bad things like crashing of FS Logbook Editor and loosing your unsaved data may occur when circumstances are good.

Last in the General section are **Coordinates format:** and **Magnetic variation format:** options. These set how the geographic coordinates and magnetic variation information will be formatted in Map Viewer window. For both of these you can specify Precision, which applies to decimal part of selected formatting, if it has any.

Next there are settings in the **Info bubble** section. Well, these control how the bubble that shows information about airports (route points) will look like. You can set **border width** and **color** and **background color**. Set the color by left clicking at the preview rectangle. If you right click on that rectangle., the color will reset to default value. You can also set **font** that will be used for airport information. **To set font, right click** in the info bubble preview on the left. If you want reset font to the default one, click in the info bubble preview with the middle mouse button (wheel).

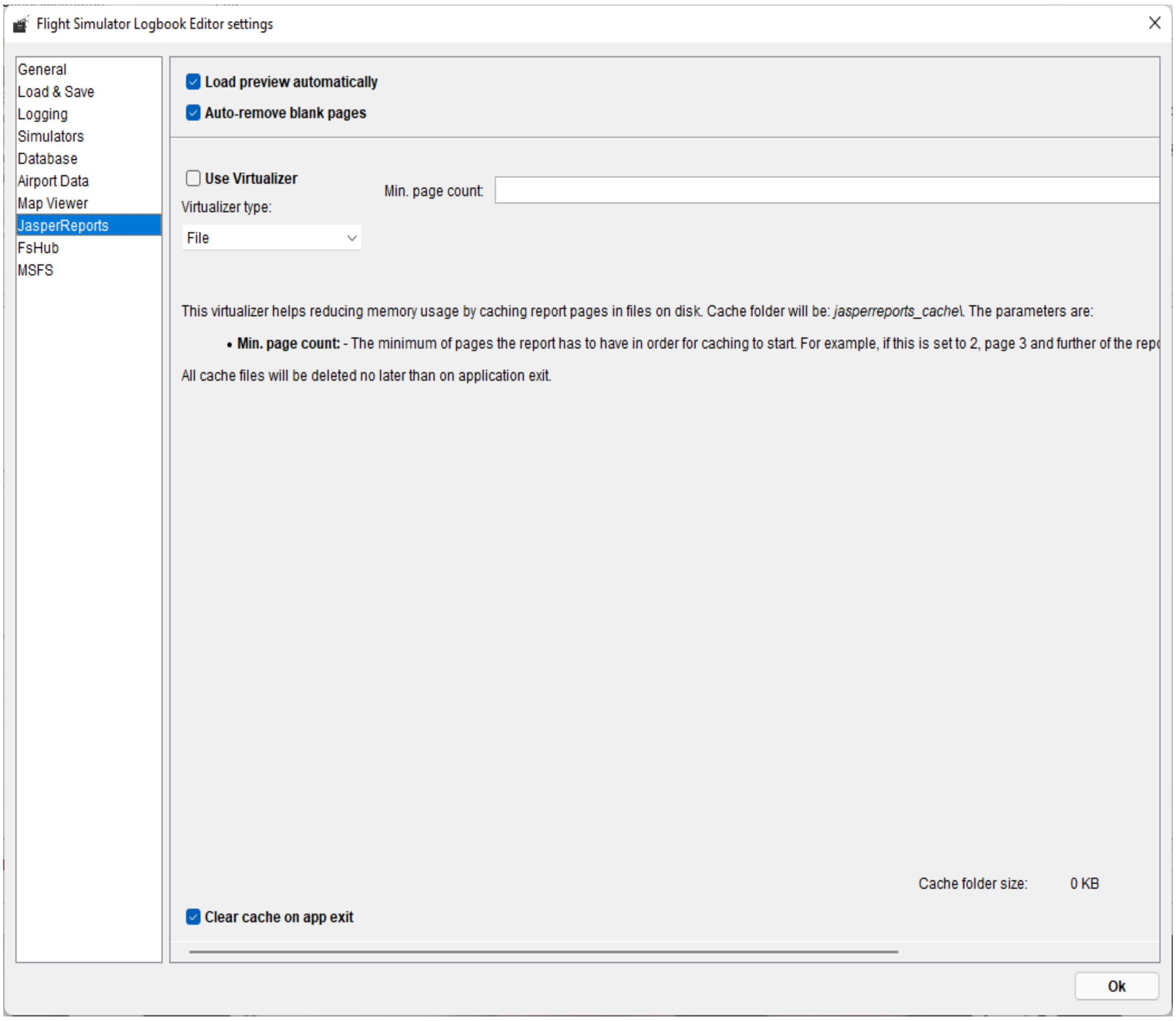
The **Waypoint and route:** section allows to customize the look of route points (airports) and the route itself. For each type of route point, ie. the starting airport, en-route airport and destination airport, you can choose the icon to be used, the icon color and the label. To select icon color, left click on the respective color preview rectangle. To remove the color, right click on the rectangle (The icon then will be drawn white as it is). The label is the text, that will be drawn over the respective icon. At ***Picture 28: Map Viewer window showing selected flight route*** you can see that, the icon of starting airport is orange marker with S printed over it and destination airport is red a marker with D over it, which corresponds to the default settings shown on ***Picture 75: Map Viewer section of Settings***. The label font can be set the same way as in the case of info bubble text. Right click with the mouse into the respective label text box and the font selection dialog will appear. Click the middle mouse button into the respective label text field to reset font to default value. For the flight route, you can set the color and width of inner and outer line. The color is set simply by left mouse click in the respective color preview box. These two does not react to right mouse button, as it makes no sense to remove color and make the path invisible. Note that for outer line to be drawn, it must be thicker than the inner line.

In FS Logbook Editor version 1.85 a support to display multiple flights was added. See the ***Viewing flight on map*** chapter. A new options group: Multi-flight map viewer was added to the settings. It allows you to define the look of the “route line” for selected flight, so that it can be easily identified within the displayed flights. The options work the same way as options for the “Route settings”.

The last section is **Cache**. There are only two controls under this. A check box named Enable caching, which is checked (enabled) by default and I advise to keep it that way. On the right side, you can see the size of the cache and there is also a **Clear cache** button, which let’s you delete all cached data. Map Viewer stores cached data in the **mapviewer_cache** folder which is stored in the FS Logbook Editor installation folder.

5.8. JasperReports

This section describes options available for JasperReports generation. This was added in FS Logbook Editor version 1.60. See the *JasperReports* chapter for more details about JasperReports and see below what you can configure:



☑ Load preview automatically

☑ Auto-remove blank pages

☐ Use Virtualizer

Min. page count:

Virtualizer type: File

This virtualizer helps reducing memory usage by caching report pages in files on disk. Cache folder will be: *jasperreports_cache*. The parameters are:

- **Min. page count:** - The minimum of pages the report has to have in order for caching to start. For example, if this is set to 2, page 3 and further of the repo

All cache files will be deleted no later than on application exit.

Cache folder size: 0 KB

☑ Clear cache on app exit

Picture 76: JasperReports options

There is not that much options for JasperReports, but they might be confusing for not so technical users (and maybe even for technical users). But they are included, because I like if the programs make decisions for you, but you still must be able to have influence on them. So the first one – **Load preview automatically** – tells FS Logbook Editor to generate preview of template filled with data whenever you select a JasperReports template. This same checkbox is at the *Picture 48: JasperReports dialog* dialog and it does not matter where you set it.

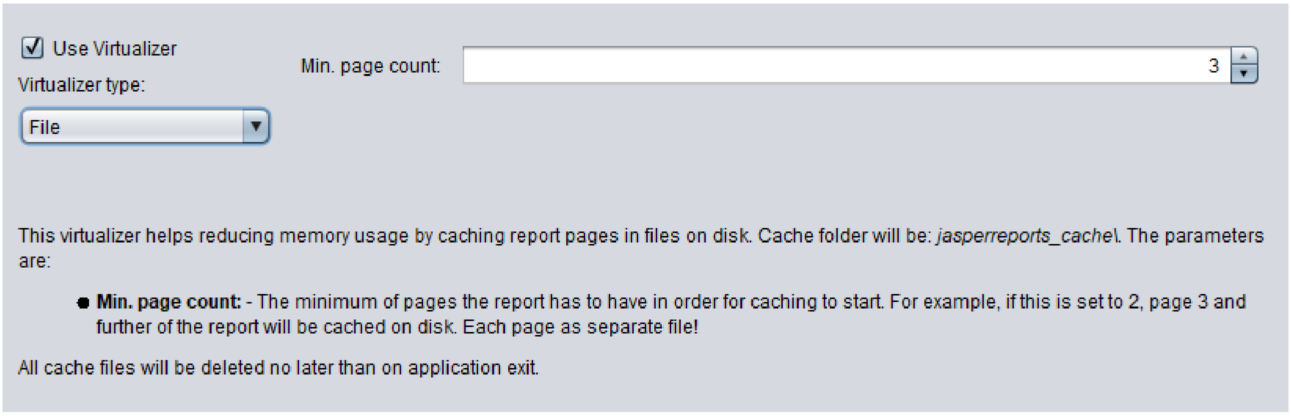
The **Auto-remove blank pages**, when set, will cause FS Logbook Editor to try to automatically remove blank pages from the generated report. It sometimes happens that the generated report contains blank pages, due to template design errors or something else. Enabling this option may help to mitigate it, but it may also cause errors in page numbering. You know, you may notice there is suddenly page missing in the middle of the document. But the blank pages, if any, often gets generated at the end of document, so you when this happens, most likely you will get the wrong total page number.

Below the line there is a section for Virtualizers usage. What are those? They might be a little tricky to understand, but … say you have a really big logbook and/or maybe a complex template … and maybe not so much memory in your computer (or maybe a lot of running programs at the time). Well, the filling of JasperReports with data is done in memory (for the preview, printing, exporting…). So chances are, FS Logbook Editor might run out of memory. (more specifically, Java virtual machine in which FS Logbook Editor runs, will run out of memory, but this is just technical…). To prevent this, JasperReports allows to store some data (pages of generated report) to somewhere else to alleviate the memory usage. That is what virtualizers are responsible for. Three types are implemented at this time. You switch between them in the **Virtualizer type** selection box. After changing Virtualizer type, the options on the right side will change and so will the help text below.

Below there is a label showing the actual size of the cache folder (**jasperreports_cache** within FS Logbook installation folder). The cache folder is the folder where virtualizers (described in further chapters) store their files. The files should be deleted automatically when no longer needed, but this may indeed fail in some circumstances. So, for the case the files get accumulated there more than you like, you can try to delete the contents of the folder using the **Clear cache** button. Note that this action may not delete all files. It will not delete the files that are currently in use by virtualizers (means you generated a JasperReport recently / a preview of report is still present at JasperReports dialog). The clean cache function may run automatically by FS Logbook Editor closure. Just check the Clear cache on app exit and FS Logbook Editor will try to clean the cache folder before closing itself. But the same applies as for the Clear cache button. It may not delete the very recent cache files. But this will at least prevent the accumulation of cache files from previous usage.

5.8.1. File virtualizer

File virtualizer helps reducing the memory usage by storing some of generated pages of the report in the files on disk. It creates one file per generated page. These files will be stored within the **jasperreports_cache** folder under FS Logbook Editor installation folder.



This virtualizer helps reducing memory usage by caching report pages in files on disk. Cache folder will be: *jasperreports_cache*. The parameters are:

- **Min. page count:** - The minimum of pages the report has to have in order for caching to start. For example, if this is set to 2, page 3 and further of the report will be cached on disk. Each page as separate file!

All cache files will be deleted no later than on application exit.

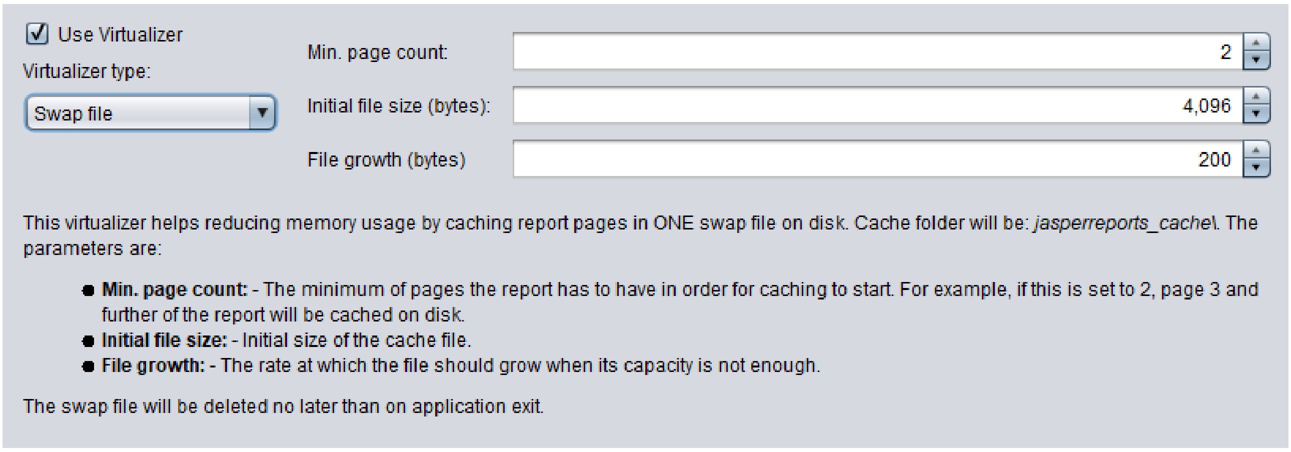
Picture 77: File virtualizer options

The file virtualizer has only one option – **Min. page count** – which determines a minimum of pages the report has to have in order for caching to start. In other words, it says from which page the caching will start. On the picture above, 3 is set. It means, page 4 and further pages of the report will each be stored in a separate file in the cache folder. The files should get deleted automatically when not needed.

The file virtualizer is probably the fastest one, but on the other side, can generate large amount of files.

5.8.2. Swap file virtualizer

The swap file virtualizer works essentially the same as ***File virtualizer***. But instead of creating new file for each page of the report, it stores all cached pages in one file. The swap file will be also stored in the **jasperreports_cache** folder under FS Logbook Editor installation folder and will be deleted when no longer needed.



Min. page count: 2
Initial file size (bytes): 4,096
File growth (bytes) 200

This virtualizer helps reducing memory usage by caching report pages in ONE swap file on disk. Cache folder will be: *jasperreports_cache*. The parameters are:

- **Min. page count:** - The minimum of pages the report has to have in order for caching to start. For example, if this is set to 2, page 3 and further of the report will be cached on disk.
- **Initial file size:** - Initial size of the cache file.
- **File growth:** - The rate at which the file should grow when its capacity is not enough.

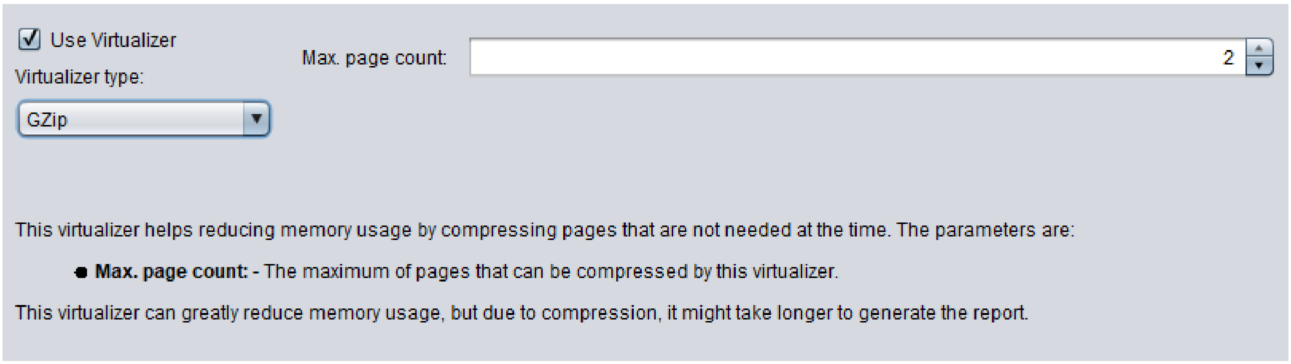
The swap file will be deleted no later than on application exit.

Picture 78: Swap file virtualizer options

The picture above shows options for swap file virtualizer. As in the case of file virtualizer, the **Min. page count** option specifies, from which page of the report the caching should start. The **Initial file size (bytes)** specifies, surprisingly, the initial size of the swap file in bytes. Because the swap file virtualizer uses just one file for all cached pages, it cannot know how "big" the file it will need. That is the reason why we specify here some initial size and also an amount of which the file should grow when its capacity is not enough for another pages to fit in. And the growth amount is the last parameter – **File growth (bytes)**. All of these options have some default values, but it is rather trial and error to find the sweet spot (So that the caching does not start too early or late and to fine tune the file size and growth).

5.8.3. GZip virtualizer

The GZip virtualizer works different than file and swap virtualizers. The GZip virtualizer does not create any files on disk, but it compresses pages that are not need at the time in memory – thus lowering memory usage.



Max. page count: 2

This virtualizer helps reducing memory usage by compressing pages that are not needed at the time. The parameters are:

- **Max. page count:** - The maximum of pages that can be compressed by this virtualizer.

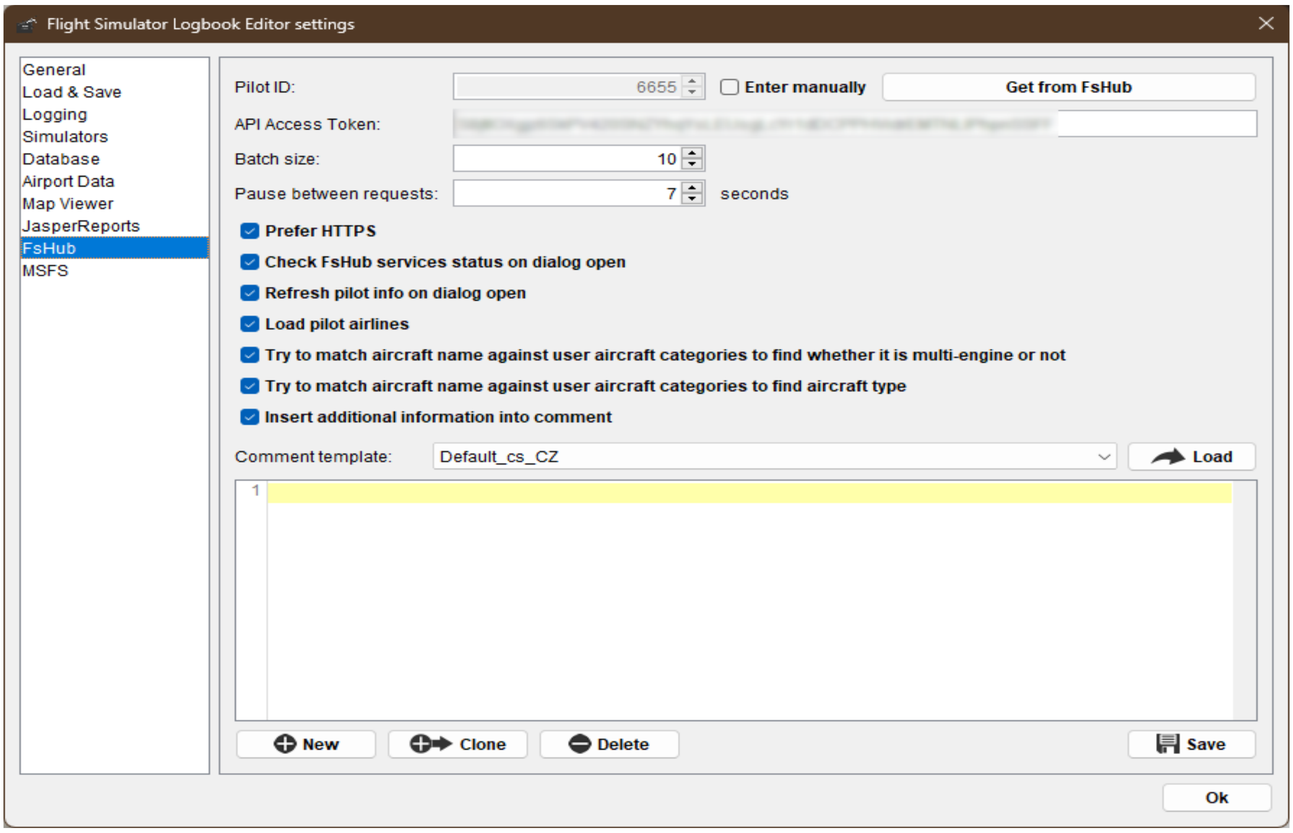
This virtualizer can greatly reduce memory usage, but due to compression, it might take longer to generate the report.

Picture 79: GZip virtualizer options

The only option here is **Max. page count** – limiting the maximum amount of pages that can be compressed. Although this virtualizer can greatly reduce memory usage and prevent out of memory problems, it can also greatly increase the time to generate report (due to the compression). It is also trial and error option. You will need to test what is too crazy for your system.

5.9. FsHub

This section contains settings related to import data from FsHub, two of them crucial for the import to actually work.


Flight Simulator Logbook Editor settings

- General
- Load & Save
- Logging
- Simulators
- Database
- Airport Data
- Map Viewer
- JasperReports
- FsHub
- MSFS

Pilot ID: 6655 ☐ Enter manually | Get from FsHub

API Access Token: [redacted]

Batch size: 10

Pause between requests: 7 seconds

- ✓ Prefer HTTPS
- ✓ Check FsHub services status on dialog open
- ✓ Refresh pilot info on dialog open
- ✓ Load pilot airlines
- ✓ Try to match aircraft name against user aircraft categories to find whether it is multi-engine or not
- ✓ Try to match aircraft name against user aircraft categories to find aircraft type
- ✓ Insert additional information into comment

Comment template: Default_cs_CZ | Load

New | Clone | Delete | Save

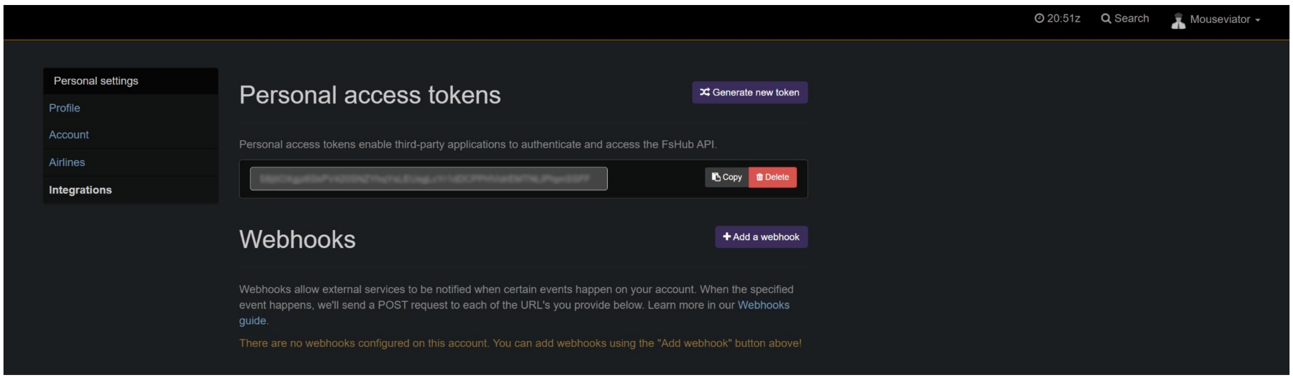
Ok

Picture 80: FsHub settings section

The crucial settings are the **API Access Token** and **Pilot ID.** API Access Token– that is a long string consisting of letters and numbers and is an "access key" for using FsHub API, or simply saying, it is what will allow you to download your flight data from FsHub. Where to get it? Well, to download your flight data from FsHub, you must have an account there and this is where you will get it. **Log in to your FsHub account**, than in the top right corner, where your login is, expand the menu and click on **Settings**. Than click on the **Integrations** (in the menu on the left side of the page now). On the page that will show, you should see **Personal access tokens** (See ***Picture 81: Finding out API (Personal) Access Token at FsHub account***). Either there will be a default one, or click on **Generate new token** to generate new one. Than, you can copy the token using the **Copy** button and insert into the **API Access Token** text field in FS Logbook Editor. **Pilot ID** is also needed, but FS Logbook Editor can determine your Pilot ID once you specify valid API Access Token. Click on **Get from FsHub** button to the right of Pilot ID to read your Pilot ID from FsHub. The **Enter manually** checkbox will allow you to override this and specify the Pilot ID manually if you want or need to.

Apart from that one/two crucial settings, there are bunch of others you can modify.

- **Batch size** – this determines the maximum amount of flights FS Logbook Editor will download in one request to FsHub server. The value can range between 1 and 100, default is 50.
- **Pause between requests** – this is the time, in seconds, that FS Logbook Editor will wait between subsequent requests to FsHub server. The minimum value is 7 seconds. The maximum … is not limited, wait as you wish :) If you have for example logged 100 flights and **Batch size** is 50, FS Logbook Editor will download the first 50, wait 7 seconds and them download the another 50.



Personal access tokens

Personal access tokens enable third-party applications to authenticate and access the FsHub API.

[redacted]

Webhooks

Webhooks allow external services to be notified when certain events happen on your account. When the specified event happens, we'll send a POST request to each of the URL's you provide below. Learn more in our Webhooks guide.

There are no webhooks configured on this account. You can add webhooks using the "Add webhook" button above!

Picture 81: Finding out API (Personal) Access Token at FsHub account

Both the **Batch size** and **Pause between requests** options **are there to protect the FsHub server from excessive load – and also you from locking yourself out of your account!!!** Because should you query the server „too much“, you can get banned. But don’t worry, the minimum values are set in way that this cannot happen. And of course, none of you want to do any bad to that great service, that is also free, right?

Below the described settings, there are several choices to check/uncheck:

- **Prefer HTTPS** – FS Logbook Editor will use HTTPS connection to FsHub server rather than plain HTTP.
- **Check FsHub services status on dialog open** – When checked, upon opening the FsHub Import dialog (see ***Importing flights from FSHub***), FS Logbook Editor will check the status of FsHub services. The result is textual information of the services status printed in the log window at FsHub Import Dialog. But even if the status is NOT OK (the service monitoring the FsHub services may be down too…, or there might be another unexpected issue), the GUI will be enabled allowing you to try to import flights anyway (assuming you have setup the **API Access Token** and **Pilot ID**).
- **Refresh pilot info on dialog open** – When checked, FS Logbook Editor will try to update pilot info upon opening the FsHub Import dialog. (see ***Importing flights from FSHub***).
- **Load pilot airlines** – This is related to the **Airlines…** button added to FSHub import dialog in FS Logbook Editor version 1.85. If enabled, list of pilot airlines will be downloaded when refreshing pilot info. (see ***Importing flights from FSHub***)
- **Try to match aircraft name against user aircraft categories o find whether it is multi-engine or not** – FsHub flight records does not contain information whether the used aircraft was multi-engine or not, so this information, that FS Logbook Editor supports, would always be false. When this option is enabled, FS Logbook Editor will try to find out aircraft name in user aircraft categories (See ***User aircraft categories***). User aircraft categories, since FS Logbook Editor version 1.8, have new property that specifies that the aircraft category contains multi-engine aircrafts. Since FS Logbook Editor version 1.85, this property can be set for each aircraft within the category individually. This way, if set

properly, this information can be determined automatically during import.

- **Try to match aircraft name against user aircraft categories to find aircraft type** - FsHub flight records does not contain information about aircraft type used in the flight, so this information that FS Logbook Editor supports, would always be the default – Fixed Wing. When this option is enabled, FS Logbook Editor will try to find out aircraft type in user aircraft categories (See ***User aircraft categories***). User aircraft categories, since FS Logbook Editor version 1.85, have new property that specifies what aircraft type the category contains. On category level, is is sort of default value for all aircrafts within the category, but each aircraft can override it. This way, if set properly, this information can be determined automatically during import.
- **Insert additional information into comment** – Flights logged at FsHub contains a lot of additional information for which there are no „columns" in FS Logbook Editor. I did not want to just discard these information and the solutions seemed to be, to allow you to insert them into the flight comment. As usually, I like the things to be configurable, so the solution is to use the „templates" – so everyone you can insert information that you are interested in in the format you want. The rest of the FsHub section is about that. It contains „simple" editor, that allows to **Load** existing templates, edit and **Save** them, create **New** ones, and **Clone** existing ones. I believe the GUI is quite self-explanatory, so try it out. The big white place is a text editor to write/edit the templates. It is a "smart" editor. If you **press Ctrl+Space**, it will show list of keywords supported (auto-completion window), along with short explanantory of their meaning. The keywords are enclosed in **{** and **}**. Keyword is special word that will be replaced by FS Logbook Editor by respective information from FsHub flight record. For example, the **{departure_icao}** keyword will be replaced by the ICAO code of departure airport of the respective flight. FS Logbook Editor contains two default templates – **Default_en** and **Default_cs_CZ** – one is in English, the other in Czech. Note that they are just named like this, **they will not switch automatically** if you change FS Logbook Editor language. The active template, that will be used during import (if enabled), is the one selected in the list. These default templates also contain all possible keywords! They are stored in the **res/fshub/templates** (as all other templates you here). If you delete the default templates, FS Logbook Editor will recreate them on next start, just like other crucial files.

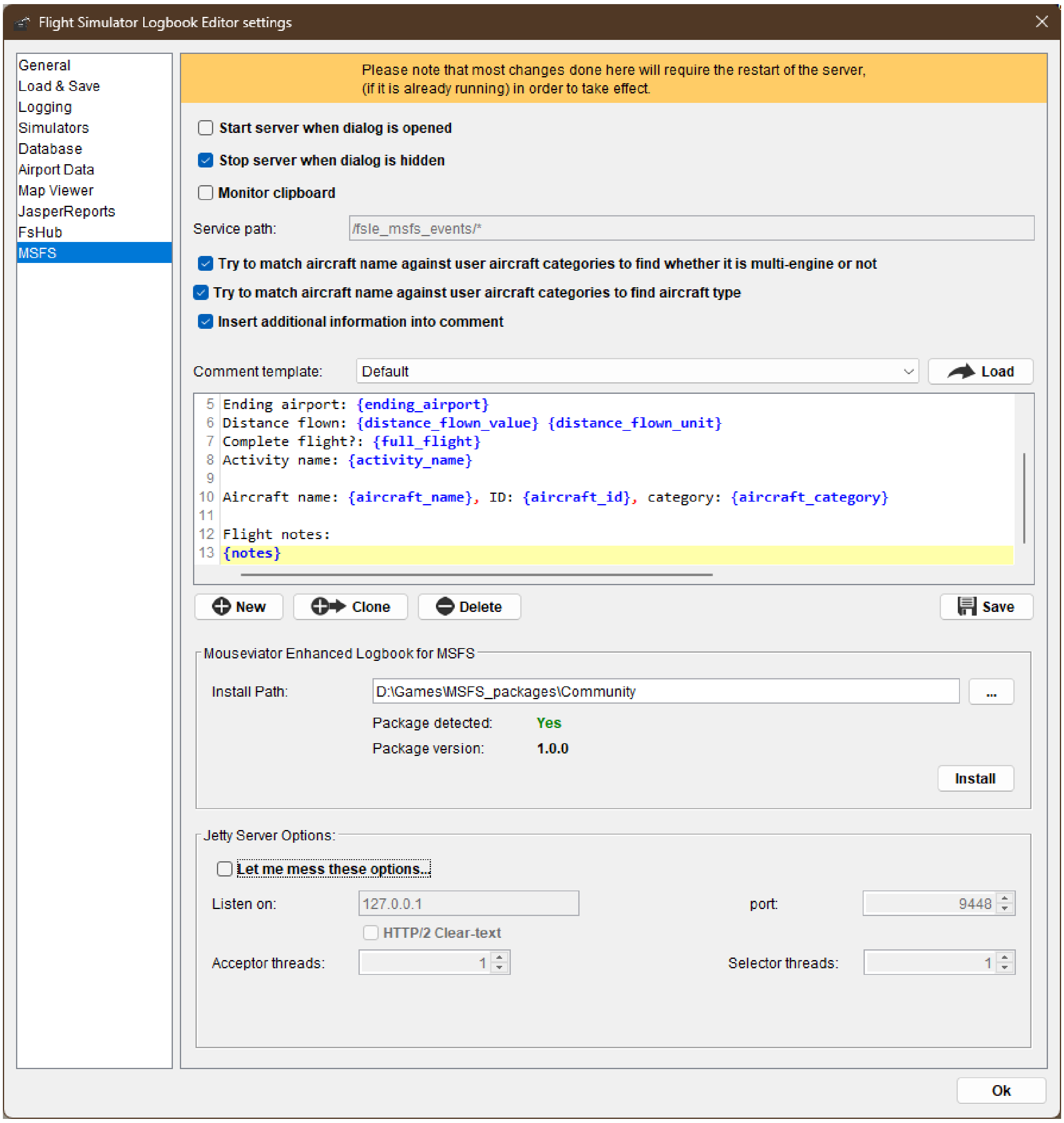
5.10. MSFS

The MSFS section of the Settings contains settings related to import of flights from Microsoft Flight Simulator. As described in Importing flights from Microsoft Flight Simulator 2020 chapter, the import depends on the presence of "**Mouseviator Enhanced Logbook**" add-on package within Microsoft Flight Simulator. The mentioned chapter also depicts that flights can be imported using clipboard or network. When using network import, FS Logbook Editor than becomes "server" and MSFS is client. The settings in this section relate to this server-client configuration.

So, server (refer to ***Picture 82: MSFS section of Settings, settings related to import flights from MSFS***), FS Logbook Editor contains embed HTTP server, in order to be a "server" for MSFS. The first two options you can find there are pretty easy:

- **Start server when dialog is opened** – When enabled, will automatically start the embed HTTP server when the MSFS Import Dialog is opened. (See ***Importing flights from Microsoft Flight Simulator 2020***). This is disabled by default. The server is sort of resource-heavy thing, is not supposed to be used very often by now, so better choice is probably to start it manually when needed.

- **Stop server when dialog is hidden** – When enabled, the embed HTTP server will be stopped once you close the MSFS Import Dialog is opened. (See ***Importing flights from Microsoft Flight Simulator 2020***). This option, on the other hand, is enabled by default. This is for the same reasons as the previous option – it is assumed the server will be needed only for the import, so it does not have to be running, consuming resources, when you are not importing anything. And you probably don't if you closed the dialog.
- **Monitor clipboard** – If this option is enabled, FS Logbook Editor will monitor clipboard for any text while the MSFS Import dialog is displayed. If it will find any MSFS flights within the clipboard, they will be automatically inserted into the logbook. This means, that you do not have to press the **Import from clipboard** nor **Import from clipboard – to new logbook** button at the import dialog. (See ***Importing flights from Microsoft Flight Simulator 2020***)
- **Try to match aircraft name against user aircraft categories o find whether it is multi-engine or not** – MSFS flight records does not contain information whether the used aircraft was multi-engine or not, so this information, that FS Logbook Editor supports, would not be present. When this option is enabled, FS Logbook Editor will try to find out aircraft name in user aircraft categories (See ***User aircraft categories***). User aircraft categories, since FS Logbook Editor version 1.8, have new property that specifies that the aircraft category contains multi-engine aircraft. Since FS Logbook Editor version 1.85, this property can be set for each aircraft within the category individually. This way, if set properly, this information can be determined automatically during import.
- **Try to match aircraft name against user aircraft categories to find aircraft type** - MSFS flight records does not contain information about aircraft type used in the flight, so this information that FS Logbook Editor supports, would always be the default – Fixed Wing. When this option is enabled, FS Logbook Editor will try to find out aircraft type in user aircraft categories (See ***User aircraft categories***). User aircraft categories, since FS Logbook Editor version 1.85, have new property that specifies what aircraft type the category contains. On category level, is is sort of default value for all aircrafts within the category, but each aircraft can override it. This way, if set properly, this information can be determined automatically during import.


Flight Simulator Logbook Editor settings

General
Load & Save
Logging
Simulators
Database
Airport Data
Map Viewer
JasperReports
FsHub
MSFS

Please note that most changes done here will require the restart of the server,
(if it is already running) in order to take effect.

- [ ] Start server when dialog is opened
- [x] Stop server when dialog is hidden
- [ ] Monitor clipboard

Service path: /fsle_msfs_events/*

- [x] Try to match aircraft name against user aircraft categories to find whether it is multi-engine or not
- [x] Try to match aircraft name against user aircraft categories to find aircraft type
- [x] Insert additional information into comment

Comment template: Default | Load

```
Ending airport: {ending_airport}
Distance flown: {distance_flown_value} {distance_flown_unit}
Complete flight?: {full_flight}
Activity name: {activity_name}

Aircraft name: {aircraft_name}, ID: {aircraft_id}, category: {aircraft_category}

Flight notes:
{notes}
```

New | Clone | Delete | Save

Mouseviator Enhanced Logbook for MSFS

Install Path: D:\Games\MSFS_packages\Community

Package detected: Yes

Package version: 1.0.0

Install

Jetty Server Options:

- [ ] Let me mess these options...

Listen on: 127.0.0.1 port: 9448

- [ ] HTTP/2 Clear-text

Acceptor threads: 1 Selector threads: 1

Ok

Picture 82: MSFS section of Settings, settings related to import flights from MSFS

- **Insert additional information into comment** – Flights logged in MSFS contains some additional information for which there are no „columns“ in FS Logbook Editor. I did not want to just discard these information and the solutions seemed to be, to allow you to insert them into the flight comment. As usually, I like the things to be configurable, so the solution is to use the „templates“ – so everyone you can insert information that you are interested in in the format you want. The rest of the FsHub section is about that. It contains „simple“ editor, that allows to **Load** existing templates, edit and **Save** them, create **New** ones, and **Clone** existing ones. I believe the GUI is quite self-explanatory, so try it out. The big white place is a text editor to write/edit the templates. It is a “smart“ editor. If you **press Ctrl+Space**, it will show list of keywords supported (auto-completion window), along with short explanantory of their meaning. The keywords are enclosed in { and }. Keyword is special word that will be replaced by FS Logbook Editor by respective information from FsHub flight record. For example, the **{ending_airport}** keyword will be replaced by the ending airport of the respective flight. FS Logbook Editor contains two default templates – **Default_en** and **Default_cs_CZ** – one is in English, the other in Czech. Note that they are just named like this, **they will not switch automatically** if you change FS Logbook Editor language. The active template, that will be used during import (if enabled), is the one selected in the list. They are stored in the **res/msfs/templates** (as all other templates you here). If you delete the default templates, FS Logbook Editor will recreate them on next start, just like other crucial files.

5.10.1. Embed server options

Now let's describe the grayed-out (by default) options. These are the **Service path** and all options within the **Jetty Server Options** panel. Jetty is the name of the server implementation from Eclipse, that is just for a side note.

The options are grayed-out by default, because they should not be changed – but I do not like to hard code things, so they can be changed, by clicking the **Let me mess these options…** checkbox.

- **Listen on** – is a host (IP address or host-name) that the server should listen on. By default, it is 127.0.0.1 – which means the server will listen on local interface only. If this is empty, the server will listen on all computer interfaces.
- **Port** – is a port the server should listen on.
- **Service path** – as an URL that the server will listen on for MSFS flights to come… This for the "*what if I will need the embed server for other services in the future too…*" scenario…

All the three mentioned options form an URL that the server listens on, in default configuration it is: "*http://127.0.0.1:9448/fsle_msfs_events/**". All of these are network related, techie stuff, so if you feel like I am writing in Clingon here, definitely, **do not change these options!** The "Mouseviator Enhanced Logbook" add-on package is pre-configured to use this address, so the two parties can connect without issues. **The only thing needed, is you may need to allow access to network for FS Logbook Editor when the system asks you so.**

The **Acceptor threads** and **Selector threads** are options that may be useful if the server is about to be heavily loaded. Again, not expected in the case of FS Logbook Editor, the default value of 1 for both should be more than enough. They are there just for the "*just in case*" … case.

The last "just in case" option is the **HTTP/2 Clear-text** check box, which, when enabled, will make the server to listen also on HTTP/2 Clear-text protocol. It is newer (so should be better?) protocol than standard HTTP, but in current implementation there is no need for it to be enabled.

5.10.2. Mouseviator Enhanced Logbook for MSFS

The "**Mouseviator Enhanced Logbook**" add-on package does the "client" side of transferring flights from Microsoft Flight Simulator to FS Logbook Editor. The **Mouseviator Enhanced Logbook for MSFS** settings section will help you determine whether the package is installed and in what version. It also allows you to install the version distributed along with FS Logbook Editor (which may not be current).

The **Install Path** shows where the package should be installed. You can select the path manually by using the **…** button. This should normally point to the **Community** folder of your Microsoft Flight Simulator installation, but if you are using some add-on management tool, it might be completely different location. If you defined some flight simulator of type "*Microsoft Flight Simulator (by*

ASOBO Studios)” within the ***Simulators***, either manually or by automatic Find feature, FS Logbook Editor will try to use them to find the path to **Community** folder automatically when **Install Path** is empty. For this to work, the **Logbook dir** of the respective MSFS type simulator must point to the simulators **LocalCache** folder. It will if discovered using **Find Simulators** button.

If the **Install Path** is known, no matter if discovered automatically or manually, FS Logbook Editor will than check for the “**Mouseviator Enhanced Logbook”** add-on package existence within the path and if it exists, will try to read the *manifest.json* file to determine package version. These information are then displayed in the panel.

The Install button will try to install/extract the “**Mouseviator Enhanced Logbook”** add-on package distributed with FS Logbook Editor into the install path and if succeeds, will perform the same existence and version check as if the install path was given manually. The panel than should look similar like on the ***Picture 82: MSFS section of Settings, settings related to import flights from MSFS.***

6. Customizing – a bit advanced topic

This chapter describes some customization's you can perform. These are an advanced topics, so don't be shamed if you don't understand what is written here and just skip this chapter.

6.1. Customizing date formats

At ***Settings***, in the ***General*** section you can choose format to display date of flight information at the table at ***Editor*** tab. There is just a couple of options which may not suit everybody's preference. These date formats are defined in the file called *date_formats.xml* that is stored at **res** folder in the program folder. While you do not need to know programming, it would help you much, since you would probably be familiar with date formatting and structure of XML files. But even if it never crossed your ways, you may be able to add your own date format.

The *date_formats.xml* may look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<date_formats>
  <date_format>dd.MM.YYYY HH:mm:ss</date_format>
  <date_format>dd.MM.YYYY</date_format>
  <date_format>dd.MM.YYYY HH:mm</date_format>
          <date_format>EEEE, MMMM d, YYYY</date_format>
  <date_format>EEEE, MMMM d, YYYY KK:mm a</date_format>
  <date_format>EEEE, MMMM d, YYYY KK:mm:ss a</date_format>
</date_formats>
```

As you can see, it is actually simple text file with some extra tags between < and > characters. These tags tells FS Logbook Editor what the information between the tag is and are program specific (some other program, even if working with same data, may use totally different tags).

It is important to preserve XML file structure. To add another date format, just add another pair of *<date_format></date_format>* tags and insert your date format in between them, just like in the sample above.

The harder part here will be to find out what are the characters meaning and which one you can use and how. This is why this goes under advanced topic and where the programming comes handy. FS Logbook Editor is written in Java and these date formats are used by SimpleDateFormat object to display nice human-readable flight date and time information in the Date column at ***Editor*** tab. You can ask at your favorite web search portal for the words: **Java SimpleDateFormat** and I am sure you will get a bunch of pages describing that. You do not need to read all the text about *SimpleDateFormat*, just find some page with the list of possible characters and their meanings with some examples, like at this official site:

http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html

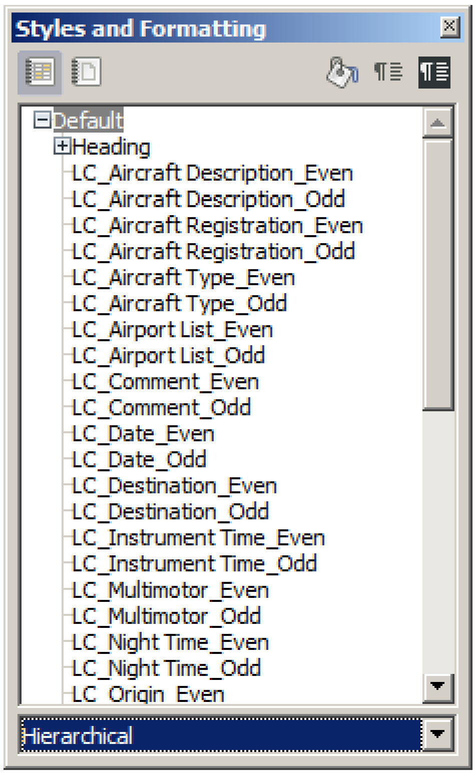
If you write some bad date format, it should not do more harm other than not being loaded in the list of available date formats. And one final note, you do not have to reload FS Logbook Editor after every change in this file, it is reloaded every time you open ***Settings*** window, so just close it and open it again if it was opened.

If you are fortunate enough to totally brake the default date and formats file – *date_formats.xml*, just close FS Logbook Editor, delete the *date_formats.xml* from the **res** folder and start FS Logbook Editor again – the default *date_formats.xml* will get recreated automatically.

6.2. Customizing ods template

At ***Settings***, in the ***Load & Save*** section you can choose which template to use when exporting logbook as Open Document Spreadsheet – .ods file. By default, there is only one template to choose from – *ods_export.ods* that is stored at **res** folder in the program folder. But if there is more, you can choose one to be used.

Guess what, you can add as many as you want. The simplest approach is to make a copy of *ods_export.ods* in the **res** folder with different name, such as *my_template.ods* or whatever you want. Than, open your template in Open Office, Libre Office or other spreadsheet program supporting open document spreadsheets (.ods files) and edit predefined styles. In Open Office and Libre Office, you can show Styles and Formatting window by F11 key, I believe. It looks like this:


Styles and Formatting

Default
Heading
LC_Aircraft Description_Even
LC_Aircraft Description_Odd
LC_Aircraft Registration_Even
LC_Aircraft Registration_Odd
LC_Aircraft Type_Even
LC_Aircraft Type_Odd
LC_Airport List_Even
LC_Airport List_Odd
LC_Comment_Even
LC_Comment_Odd
LC_Date_Even
LC_Date_Odd
LC_Destination_Even
LC_Destination_Odd
LC_Instrument Time_Even
LC_Instrument Time_Odd
LC_Multimotor_Even
LC_Multimotor_Odd
LC_Night Time_Even
LC_Night Time_Odd
LC_Origin_Even

Hierarchical

Picture 83: Styles and Formatting window at Libre Office

There is quite a lot of predefined styles in the default template for all logbook record attributes. I will not describe here how to edit those styles, since this is not purpose of this document. I will rather describe some of the styles and the idea behind their names. The names of the styles must stay the same, as FS Logbook Editor kind counts on with them. Changing their names actually defeats the purpose of the template.

All the custom styles used by FS Logbook Editor starts either with **LC**, **LH** or **LS**. **LC** at the beginning of the style name stands for **L**ogbook **C**ell. **LH** stands for **L**ogbook **H**eading and **LS** stands for **L**ogbook **S**tatistics. The LH styles are used in the first row of the resultant spreadsheet where the column labels (headings) are. LC are all other cells containing data from logbook records.

Even at the end of the style name indicates that this styles is used for spreadsheet cells at even spreadsheet row. Likewise, **Odd** identifies styles used at odd rows.

The text between LC, LH, LS and Odd, Even or to the end of style name identifies logbook record attribute this style will be applied to. To list them:

- **Date** – for flight date and time.
- **Origin** – for ICAO code of origin airport.
- **Origin_Name** – origin airport name. [5]
- **Origin_City** – origin airport city name. [5]
- **Origin_State** – origin airport state name. [5]
- **Origin_Country** – origin airport country name. [5]
- **Destination** – for ICAO code of destination airport (or the one last landed at during flight).
- **Destination_Name** – destination airport name. [5]
- **Destination_City** – destination airport city name. [5]
- **Destination_State** – destination airport state name. [5]
- **Destination_Country** – destination airport country name. [5]
- **Total Time** – for total flight time.
- **Instrument Time** – for time of flight flown in IMC conditions.
- **XCountry Time** – for cross country flight time.
- **Night Time** – for time of flight flown at night.
- **Aircraft Registration** – for aircraft registration.
- **Aircraft Description** – for aircraft description.
- **Aircraft Type** – for aircraft type.
- **Comment** – for your flight comment.
- **Airport List** – for cells where airport list will be (ICAO codes of airport landed at along with number of landings)
- **TT_Summary** – cell holding total flight time summary for whole logbook.
- **IT_Summary** – cell holding total instrument time summary for whole logbook.
- **NT_Summary** – cell holding total night time summary for whole logbook.

After you change your styles, save your template and you are ready to use it. You do not have to reload FS Logbook Editor to reload the list of available templates, it gets reloaded every time you open ***Settings*** window. If it was opened while you saved your template, just close it and open it again and you should find the template in the list.

If you are fortunate enough to totally brake the default template – *ods_export.ods*, just close FS Logbook Editor, delete the *ods_export.ods* from the **res** folder and start FS Logbook Editor again – the default *ods_export.ods* will get recreated automatically.

5 Since FS Logbook Editor v 1.40. Airport data must be available.

6.3. Translating FS Logbook Editor

FS Logbook Editor is multilingual. The GUI and most of the log messages are translated, but you may find that your natural language is not supported. You can do three things. First, you can wait until it is supported (but you will be questioning yourself if that is ever gonna happen), write your own translation (in that case, I will be happy if you want to share it with me. It may become official part of next release) or get over it.

In the case you want to translate the program, take a look inside the **lang** folder which is located at folder in which FS Logbook Editor is installed. You will see a couple of files here, with **.properties** extension. The names of the files also includes language codes that specifies to which language the given translation file belongs to. These codes starts with "**_**" character. They can be, for example:

- **_en** – For English language, country does not matter.
- **_cs_CZ** – for Czech language and Czech Republic.

These are the same codes supported by Java **Locale** object. The program supports usage of language, country and variant codes. To get the list of supported codes, use your favorite web search engine and search for: "java locale codes" or something similar. Or try this page:

http://www.oracle.com/technetwork/java/javase/javase7locales-334809.html

To translate all texts, you need to translate all these files:

- *Database<your language code>.properties* [6]
- *Dialogs<your language code>.properties*
- *Editors<your language code>.properties*
- *Filters<your language code>.properties*
- *FlightSimulatorLogbookEditor<your language code>.properties*
- *Logbook<your language code>.properties*
- *MapViewer<your language code>.properties*
- *PersonalMinimums<your language code>.properties*
- *PilotStatistics<your language code>.properties*
- *Readers<your language code>.properties*
- *Tasks<your language code>.properties*
- *Writers<your language code>.properties*
- *Utils<your language code>.properties* (Added in version 1.15. Some texts from *FlightSimulatorLogbookEditor<your language code>.properties* has been moved in here)

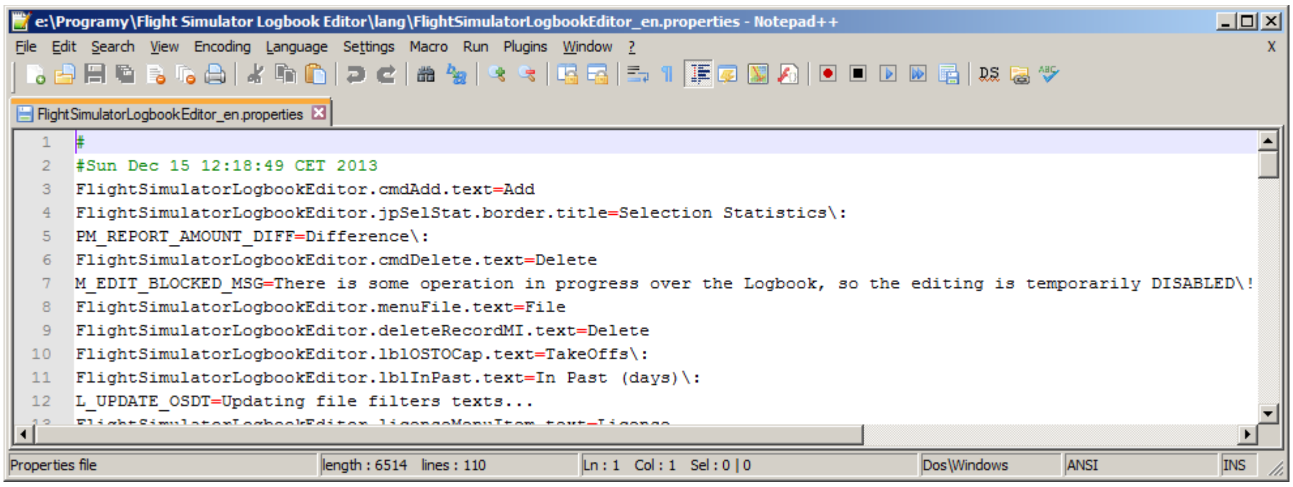
The best approach is to make a copy of existing files, change their language codes and than edit them so the texts within them match the language they are targeted to. For example, if you want to make German translation, you can copy all files with "_en" code and change their code to: "**_de**" or "**_de_DE**" (for German language variant for Germany). So you will have these files:

- *Database_de_DE.properties*
- *Dialogs_de_DE.properties*
- *Editors_de_DE.properties*
- *Filters_de_DE.properties*
- *FlightSimulatorLogbookEditor_de_DE.properties*

6 Since FS Logbook Editor v 1.40

- *Logbook_de_DE.properties*
- *MapViewer_de_DE.properties*
- *PersonalMinimums_de_DE.properties*
- *PilotStatistics_de_DE.properties*
- *Readers_de_DE.properties*
- *Tasks_de_DE.properties*
- *Writers_de_DE.properties*
- *Utils_de_DE.properties*

The **.properties** files are standard Java properties files, which are key-pair value files. The string (text) on the left side of "=" character is key that FS Logbook Editor uses to find appropriate translated text. You do not want to change this unless you want to break the translation. The text on the right side of "=" is actual translation. This is the text you need to translate. The files are simple text files, so you can use simple programs such as Notepad to edit them. The picture below shows an example of opened translation file:


e:\Programy\Flight Simulator Logbook Editor\lang\FlightSimulatorLogbookEditor_en.properties - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
FlightSimulatorLogbookEditor_en.properties
Properties file
length : 6514 lines : 110
Ln : 1 Col : 1 Sel : 0 | 0
Dos\Windows
ANSI
INS

```
#
#Sun Dec 15 12:18:49 CET 2013
FlightSimulatorLogbookEditor.cmdAdd.text=Add
FlightSimulatorLogbookEditor.jpSelStat.border.title=Selection Statistics\:
PM_REPORT_AMOUNT_DIFF=Difference\:
FlightSimulatorLogbookEditor.cmdDelete.text=Delete
M_EDIT_BLOCKED_MSG=There is some operation in progress over the Logbook, so the editing is temporarily DISABLED\!
FlightSimulatorLogbookEditor.menuFile.text=File
FlightSimulatorLogbookEditor.deleteRecordMI.text=Delete
FlightSimulatorLogbookEditor.lblOSTOCap.text=TakeOffs\:
FlightSimulatorLogbookEditor.lblInPast.text=In Past (days)\:
L_UPDATE_OSDT=Updating file filters texts...
```

Picture 84: Translation file - it is simple text file.

There are two rules you need to follow in order the translation to work. First one is the naming of the files, as described above. The second one is that FS Logbook Editor expects the files to be **UTF-8** encoded. There might be a problem, since basic Notepad that is part of Windows cannot change encoding. You will need some advanced editor, such as Notepad++, which is free. If the files are not UTF-8 encoded, they're still get loaded (if named properly), but you might see some weird characters in FS Logbook Editor. This will happen if the language you are writing translation for contains some special characters, such as Chinese. If the language uses only standard alphabet, there should be no issues if the file is not UTF-8 encoded.

6.4. Writing custom pilot statistics template

Rather than generating some solid predefined output for pilot statistics, which only few people would probably like, FS Logbook Editor generates pilot statistics output from template files (so the ones who don't like it can change it). These template files are simple text files. but they include some special tags, which will be replaced by actual values in the generated statistics output. FS Logbook Editor has an embed text editor, since version 1.8, that supports highlighting the pilots statistics template keywords, supports auto-complete and context help. Read more about it in the ***Little More Than Dumb Text Editor*** chapter.

Since you are reading this chapter, you probably want to modify some template for your own needs or create a new one. Continue reading to learn more about templates.

6.4.1. Where are the templates stored?

The templates are stored in **ps_templates** folder which is in the FS Logbook Editor installation folder. By default, you should find eight text files here. These are the default templates (four in English, four in Czech). You can either edit those or add new ones.

As in the case with translation files (Read ***Translating FS Logbook Editor*** chapter) or ODS templates (Read ***Customizing ods template*** chapter), if you damage the template to the extent that you want to return to the default one, just delete the damaged template form this folder and the default one will be recreated next time you start FS Logbook Editor.

Also notice that each template has language codes added as the last characters of the file name. (English ones have "**_en**", Czech ones have "**_cs_CZ**"). These are the same codes used to mark translation files. Please read the ***Translating FS Logbook Editor*** chapter for more information about language codes. They are here just for better distinguish of templates in the template selection box in FS Logbook Editor. FS Logbook Editor will let you select any template regardless of what translation you are using.

And as in the case of translation files, the templates are supposed to be **UTF-8** encoded.

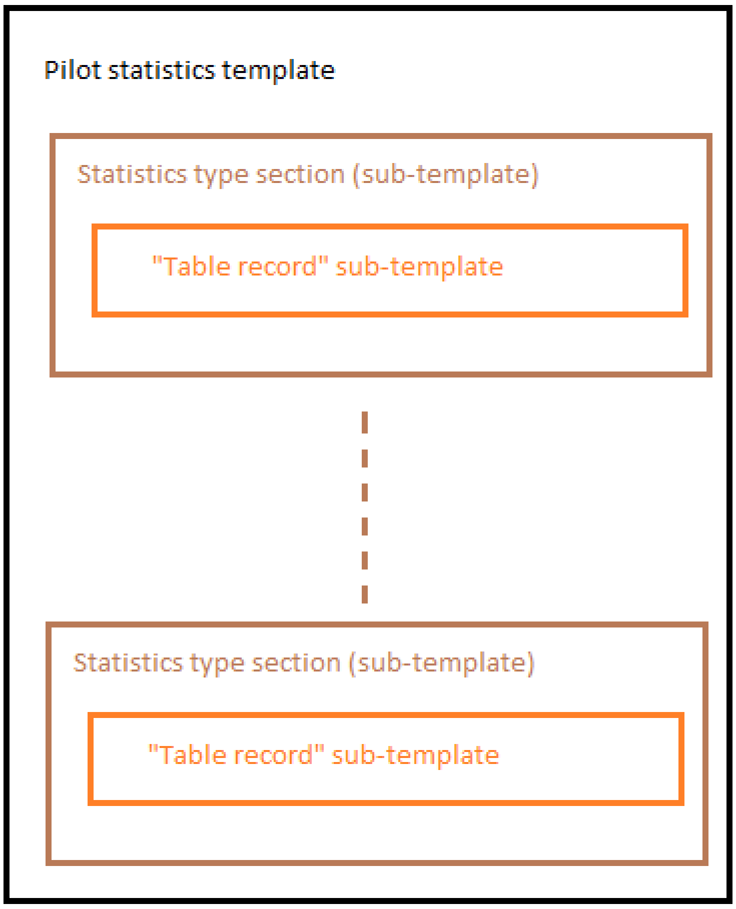
6.4.2. Template file structure

Pilot logbook in its nature is a table. Tables usually have records. In the case of pilot logbook, those records contains data about each flight. Thus, a pilot statistics generated from pilot logbook will most likely be some sort of table or tables. For example, one table can be a table of various flight times per aircraft, another one will be the table of flight times per aircraft type.

For both mentioned tables, we need to work with all records in the logbook, but compute the statistics differently (different criteria). So lets think about different statistics as different statistics types. And for each statistics type, FS Logbook Editor needs to know how to display it. This is done via the template, which contains "sections", or sub-templates, if you want, which tells FS Logbook Editor how to display every statistics type that FS Logbook Editor supports or which you want to compute. (Not all supported statistics types have to be computed via template).

Take a breath and keep reading :)

There is another sub-template in the statistics type sub-template – the one for resulting table row. We cannot know in the time of writing the template how many records (rows) will the resulting table have, thus we need a template for writing records. Like with the flight times per aircraft, every table row will contain various flight times for each aircraft in the logbook. Let's see it graphically below and/or take a look at example in the following ***Template tags*** chapter.


Pilot statistics template
Statistics type section (sub-template)
"Table record" sub-template
Statistics type section (sub-template)
"Table record" sub-template

Picture 85: Pilot statistics template structure

6.4.3. Supported statistics types

FS Logbook Editor pilot statistics supports the following statistics:

- Flight times per Aircraft
- Flight times per Aircraft type
- Flight times per User aircraft category
- Takeoffs and landings per Aircraft
- Takeoffs and landings per Aircraft type
- Takeoffs and landings per User aircraft category
- Takeoffs and landings per Airport
- Takeoffs and landings per City [7]
- Takeoffs and landings per Country [7]
- Takeoffs and landings per State [7]
- Distance flown per Aircraft [7]
- Distance flown per Aircraft type [7]
- Distance flown per User aircraft category [7]

7 Since FS Logbook Editor v 1.40

6.4.4. Template tags

AS mentioned in the first paragraph of this chapter, templates contains tags. The tags tells FS Logbook Editor what to place where. Rather than long writing, let's see an example right away:

```
Flight Times per Aircraft
==========================
<%ft_per_aircraft_start><%ft_per_aircraft_record_start>
--------------------------------------------------------
--------------------- <%key_name[1]> ---------------------
--------------------------------------------------------

*********************  Total values:  *********************
Total Time | Day Time | Night Time | Instrument Time | Multi-engine Time |
<%ft_total_tt> | <%ft_total_dt> | <%ft_total_nt> | <%ft_total_it> | <%ft_total_mt> |

*********************  Min values:  *********************
Total Time | Day Time |  Night Time | Instrument Time | Multi-engine Time |
<%ft_min_tt> | <%ft_min_dt> | <%ft_min_nt> | <%ft_min_it> | <%ft_min_mt> |

*********************  Max values:  *********************
Total Time | Day Time |  Night Time | Instrument Time | Multi-engine Time |
<%ft_max_tt> | <%ft_max_dt> | <%ft_max_nt> | <%ft_max_it> | <%ft_max_mt> |

*********************  Average values:  *********************
Total Time | Day Time |  Night Time | Instrument Time | Multi-engine Time |
<%ft_avg_tt> | <%ft_avg_dt> | <%ft_avg_nt> | <%ft_avg_it> | <%ft_avg_mt> |

Matching records: <%records_per_key>, Processed records: <%records_processed>, Total
records: <%records_per_logbook>

<%ft_per_aircraft_record_end>
\\\ Summary ///
Flight Time | Totals | Minimums | Maximums | Average
Total Time: <%ltft_total_tt> | <%ltft_min_tt> | <%ltft_max_tt> | <%ltft_avg_tt>
Day Time: <%ltft_total_dt> | <%ltft_min_dt> | <%ltft_max_dt> | <%ltft_avg_dt>
Night Time: <%ltft_total_nt> | <%ltft_min_nt> | <%ltft_max_nt> | <%ltft_avg_nt>
Instrument Time: <%ltft_total_it> | <%ltft_min_it> | <%ltft_max_it> | <%ltft_avg_it>
Multi-engine Time: <%ltft_total_mt> | <%ltft_min_mt> | <%ltft_max_mt> |
<%ltft_avg_mt>
Cross-country Time: <%ltft_total_xt> | <%ltft_min_xt> | <%ltft_max_xt> | <%ltft_avg_xt>
<%ft_per_aircraft_end>
```

Example 1: Part of statistics template

From the above example, you can guess that tags are those parts of text between **<%** and **>** characters. And you are right if you guessed so. There are multiple types of tags, which are described in following sub-chapters.

6.4.5. Paired tags

There are two types of paired tags. One that defines specific statistics type section (sub-template) in template file and one that defines "record" sub-template. In the example at ***Template tags*** chapter, the statistics type section paired tags are the red ones:

<%ft_per_aircraft_start>...<%ft_per_aircraft_end>

The statistics type section paired tags defines part of the template that is specific for the particular statistics type, like flight times per aircraft as in the example. All the supported tags within those paired tags will be replaced by respective values, the other text will be copied unchanged.

Second type of paired tags are the ones which defines "table record" sub-template. The orange ones in the example:

<%ft_per_aircraft_record_start>...<%ft_per_aircraft_record_end>

The text between those two tags will be used as template to generate each statistics row (in the example – a row with information about flight times for specific aircraft). The resulting output does not have to look like a table, it only depends on how the template is actually written. It is just simpler to think about it as of table row when explaining it.

Here is the list of supported statistics type section paired tags:

- *<%ft_per_aircraft_start><%ft_per_aircraft_end>* – for "Flight times per Aircraft" statistics.
- *<%ft_per_type_start><%ft_per_type_end>* – for "Flight times per Aircraft type" statistics.
- *<%ft_per_uac_start><%ft_per_uac_end>* – for "Flight times per User aircraft category" statistics.
- *<%tl_per_aircraft_start><%tl_per_aircraft_end>* – for "Takeoffs and landings per Aircraft" statistics.
- *<%tl_per_type_start><%tl_per_type_end>* – for "Takeoffs and landings per Aircraft type" statistics.
- *<%tl_per_uac_start><%tl_per_uac_end>* – for "Takeoffs and landings per User aircraft category" statistics.
- *<%tl_per_airport_start><%tl_per_airport_end>* – for "Takeoffs and landings per Airport" statistics.
- *<%tl_per_city_start><%tl_per_city_end>* – for "Takeoffs and landings per City" statistics.
- *<%tl_per_country_start><%tl_per_country_end>* – for "Takeoffs and landings per Country" statistics.
- *<%tl_per_state_start><%tl_per_state_end>* – for "Takeoffs and landings per State" statistics.
- *<%df_per_aircraft_start><%df_per_aircraft_end>* – for "Distance flown per Aircraft" statistics.
- *<%df_per_type_start><%df_per_type_end>* – for "Distance flown per Aircraft type" statistics.
- *<%df_per_uac_start><%df_per_uac_end>* – for "Distance flown per User aircraft category" statistics.

Here is the list of supported "table-record" sub-template paired tags:

- *<%ft_per_aircraft_record_start><%ft_per_aircraft_record_end>* – for a sub-template to generate "table record" for "Flight times per Aircraft" statistics.
- *<%ft_per_type_record_start><%ft_per_type_record_end>* – for a sub-template to

generate “table record” for “Flight times per Aircraft type” statistics.

- *<%ft_per_uac_record_start><%ft_per_uac_record_end>* – for a sub-template to generate “table record” for “Flight times per User aircraft category” statistics.
- *<%tl_per_aircraft_record_start><%tl_per_aircraft_record_end>* – for a sub-template to generate “table record” for “Takeoffs and landings per Aircraft” statistics.
- *<%tl_per_type_record_start><%tl_per_type_record_end>* – for a sub-template to generate “table record” for “Takeoffs and landings per Aircraft type” statistics.
- *<%tl_per_uac_record_start><%tl_per_uac_record_end>* – for a sub-template to generate “table record” for “Takeoffs and landings per User aircraft category” statistics.
- *<%tl_per_airport_record_start><%tl_per_airport_record_end>* – for a sub-template to generate “table record” for “Takeoffs and landings per Airport” statistics.
- *<%tl_per_city_record_start><%tl_per_city_record_end>* – for a sub-template to generate “table record” for “Takeoffs and landings per City” statistics.
- *<%tl_per_country_record_start><%tl_per_country_record_end>* – for a sub-template to generate “table record” for “Takeoffs and landings per Country” statistics.
- *<%tl_per_state_record_start><%tl_per_state_record_end>* – for a sub-template to generate “table record” for “Takeoffs and landings per State” statistics.
- *<%df_per_aircraft_record_start><%df_per_aircraft_record_end>* – for a sub-template to generate “table record” for “Distance flown per Aircraft” statistics.
- *<%df_per_type_record_start><%df_per_type_record_end>* – for a sub-template to generate “table record” for “Distance flown per Aircraft type” statistics.
- *<%df_per_uac_record_start><%df_per_uac_record_end>* – for a sub-template to generate “table record” for “Distance flown per User aircraft category” statistics.

6.4.6. Simple tags

All the other non-paired tags are referenced as simple tags here. All those tags will be replaced with respective computed values. If you have had read previous chapter, you know there is more types of statistics supported. Actually, generally, only two types at the time – the flight time statistics nad takeoffs and landings statistics. Both can be computed by different criteria, which at the end defines all statistics types mentioned in ***Template file structure*** chapter.

There are tags that has a meaning to use for flight times statistics, others for takeoffs and landings statistics and some common tags that can be used regardless of what statistics is being computed.

Tags for flight times statistics:

Also in here, we can split the tags in two groups. The ones that has a meaning to be used only within the respective "table record" paired tags (See: ***Here is the list of supported “table-record” sub-template paired tags***) and the others that may be used also within statistics paired tags (See: ***Here is the list of supported statistics type section paired tags***).

The following tags can be used only within „table record“ paired tags:

- *<%key_name>* – the computation criteria name (Either the aircraft name, aircraft type or user aircraft category name)
- *<%airport_icao>* – the same as *<%key_name>*. Use with statistics where key is airport.
- *<%airport_name>* - a name of the airport. Use with statistics where key is airport. Airport

data must be available.

- *<%ft_total_tt>* – total flight time
- *<%ft_total_dt>* – total day flight time
- *<%ft_total_nt>* – total night flight time
- *<%ft_total_it>* – total instrument flight time
- *<%ft_total_mt>* – total multi-engine flight time
- *<%ft_total_xt>* – total cross-country flight time
- *<%ft_min_tt>* – minimum flight time
- *<%ft_min_dt>* – minimum day flight time
- *<%ft_min_nt>* – minimum night flight time
- *<%ft_min_it>* – minimum instrument flight time
- *<%ft_min_mt>* – minimum multi-engine flight time
- *<%ft_min_xt>* – minimum cross-country flight time
- *<%ft_max_tt>* – maximum flight time
- *<%ft_max_dt>* – maximum day flight time
- *<%ft_max_nt>* – maximum night flight time
- *<%ft_max_it>* – maximum instrument flight time
- *<%ft_max_mt>* – maximum multi-engine flight time
- *<%ft_max_xt>* – maximum cross-country flight time
- *<%ft_avg_tt>* – average flight time
- *<%ft_avg_dt>* – average day flight time
- *<%ft_avg_nt>* – average night flight time
- *<%ft_avg_it>* – average instrument flight time
- *<%ft_avg_mt>* – average multi-engine flight time
- *<%ft_avg_xt>* – average cross-country flight time
- *<%records_per_key>* – number of logbook records matching given statistics type criteria (ie. how many records there were with given aircraft name, aircraft type or user aircraft category name)

<u>The following tags can be used within statistics type section paired tags and also within „table record“ paired tags:</u>

- *<%ltft_total_tt>* – total flight time per respective computed statistics.
- *<%ltft_total_dt>* – total day flight time per respective computed statistics.
- *<%ltft_total_nt>* – total night flight time per respective computed statistics.
- *<%ltft_total_it>* – total instrument flight time per respective computed statistics.
- *<%ltft_total_mt>* – total multi-engine flight time per respective computed statistics.
- *<%ltft_total_xt>* – total cross-country flight time per respective computed statistics.
- *<%ltft_min_tt>* – minimum flight time per respective computed statistics.
- *<%ltft_min_dt>* – minimum day flight time per respective computed statistics.
- *<%ltft_min_nt>* – minimum night flight time per respective computed statistics.
- *<%ltft_min_it>* – minimum instrument flight time per respective computed statistics.
- *<%ltft_min_mt>* – minimum multi-engine flight time per respective computed statistics.
- *<%ltft_min_xt>* – minimum cross-country flight time per respective computed statistics.

- *<%ltft_max_tt>* – maximum flight time per respective computed statistics.
- *<%ltft_max_dt>* – maximum day flight time per respective computed statistics.
- *<%ltft_max_nt>* – maximum night flight time per respective computed statistics.
- *<%ltft_max_it>* – maximum instrument flight time per respective computed statistics.
- *<%ltft_max_mt>* – maximum multi-engine flight time per respective computed statistics.
- *<%ltft_max_xt>* – maximum cross-country flight time per respective computed statistics.
- *<%ltft_avg_tt>* – average flight time per respective computed statistics.
- *<%ltft_avg_dt>* – average day flight time per respective computed statistics.
- *<%ltft_avg_nt>* – average night flight time per respective computed statistics.
- *<%ltft_avg_it>* – average instrument flight time per respective computed statistics.
- *<%ltft_avg_mt>* – average multi-engine flight time per respective computed statistics.
- *<%ltft_avg_xt>* – average cross-country flight time per respective computed statistics.
- *<%records_processed>* - number of records processed when computing this statistics. Not all records may be matching and/or may be matched multiple times (like in the case of user aircraft categories).

Tags for Takeoffs and landings statistics:

Also in ***Tags for flight times statistics***, we can split the tags in two groups. The ones that has a meaning to be used only within the respective "table record" paired tags (See: ***Here is the list of supported "table-record" sub-template paired tags***) and the others that may be used also within statistics paired tags (See: ***Here is the list of supported statistics type section paired tags***).

The following tags can be used only within „table record" paired tags:

- *<%key_name>* – the computation criteria name (Either the aircraft name, aircraft type, user aircraft category name or airport name)
- *<%to_total>* – total takeoffs
- *<%ld_total>* – total landings
- *<%to_min>* – minimum takeoffs
- *<%ld_min>* – minimum landings
- *<%to_max>* – maximum takeoffs
- *<%ld_max>* – maximum landings
- *<%to_avg>* – average takeoffs
- *<%ld_avg>* – average landings
- *<%records_per_key>* – number of logbook records matching given statistics type criteria (ie. how many records there were with given aircraft name, aircraft type, user aircraft category name or airport name)

The following tags can be used within statistics type section paired tags and also within „table record" paired tags:

- *<%ltto_total>* – total takeoffs per respective computed statistics.
- *<%ltto_min>* – minimum takeoffs per respective computed statistics.
- *<%ltto_max>* – maximum takeoffs per respective computed statistics.
- *<%ltto_avg>* – average takeoffs per respective computed statistics.
- *<%ltld_total>* – total landings per respective computed statistics.

- *<%ltld_min>* – minimum landings per respective computed statistics.
- *<%ltld_max>* – maximum landings per respective computed statistics.
- *<%ltld_avg>* – average landings per respective computed statistics.
- *<%records_processed>* - number of records processed when computing this statistics. Not all records may be matching and/or may be matched multiple times (like in the case of user aircraft categories).

Tags for distance flown statistics:

This statistics type is available since FS Logbook Editor v 1.4. Three keys are supported – a statistics per aircraft, aircraft type or user aircraft category. For all of them, you can use the following tags within the record paired tags:

<u>The following tags can be used only within „table record“ paired tags:</u>

- *<%key_name>* – the computation criteria name (Either the aircraft name, aircraft type or user aircraft category name)
- *<%df_total>* – total distance flown.
- *<%df_min>* – minimum distance flown.
- *<%df_max>* – maximum distance flown.
- *<%df_avg>* – average distance flown.

<u>The following tags can be used within statistics type section paired tags and also within „table record“ paired tags:</u>

- *<%ltdf_total>* – total distance flown per respective computed statistics.
- *<%ltdf_min>* – minimum distance flown per respective computed statistics.
- *<%ltdf_max>* – maximum distance flown per respective computed statistics.
- *<%ltdf_avg>* – average distance flown per respective computed statistics.

Common tags:

These tags can be used anywhere in the template, as they are not tight to any statistics type and their values are not record-specific.

- *<%records_per_logbook>* – amount of records in the logbook file the statistics was generated from.
- *<%logbook_name>* – Name of the logbook. The name of the logbook is supported only in FS2004 logbook file. Mostly, this will be the default value: STANDARD PILOT LOGBOOK.
- *<%logbook_file_name>* – Name of the logbook file.
- *<%logbook_file_path>* – Logbook file full path.

6.4.7. Value formatting

By default, values related to flight times statistics are formatted in "H:MM:SS" format (for example 1:02:03 or 12:13:14) and values related to takeoffs and landings are formatted as integers (except average values which are formatted as decimals with one decimal space). But you can change the value formatting. You can do that by inserting optional formatting info into tag as follows:

<%<tag_name>[<format><align><width>]>

for example:

<%ft_total_tt[2cwa]>

Let's explain what it all means.

<format> – a number that corresponds to one of supported value formatting. The values related to flight time supports the following formatting:

- **1** – *HH:MM:SS* – a value will be formatted like 01:02:03. Two digits are always used to express each part of flight time. Thus, leading zeros when necessary.
- **2** – *H:MM:SS* – this is a default format. Same as the previous one, except that for hours the leading zeros are not inserted. For example: 1:02:03 or 12:13:14.
- **3** – *HH:MM* – same as 1, except that seconds are discarded. For example: 01:02.
- **4** – *H:MM* – same as 2, except that seconds are discarded. For example: 1:02 or 12:13.
- **5** – *Decimal format.* Flight time is expressed as fraction of hours with two decimal places precision. For example a value of 1.50 means 1 hour and 30 minutes.

The values related to takeoffs and landings statistics supports the following formatting:

- **1** – *Integer value*. Simple, value is integer. For example 5 – for 5 takeoffs or landings. This is default formatting for all tags except the ones that computes average values.
- **2** – *Decimal value* with one decimal place precision. This is default formatting for tags that computes average values. For example 1.2.
- **3** – *Decimal value* with two decimal places precision. For example 1.25.

If you set any other than supported formatting value, the tag will use its default formatting.

<align> - The align tells how to align the value. Please note that this has a meaning only when used together with <width>. Supported values are:

- **l** – align value to the left.
- **c** – align value to the center.
- **r** – align value to the right.

<width> – Sets the width (length) of resulting value. In another words, how many characters will the resulting value occupy. The width must be specified as: **w** + one of supported values:

- **a** – the width will be computed automatically. The width (length) of the value will be the same as the width of longest value within specific statistics. The automatic width is only supported for tags within "table-record" paired tags (See: ***Tags for flight times statistics*** and ***Tags for Takeoffs and landings statistics***). For example, if we use this like <*%key_name[lwa]*> in flight times per aircraft statistics, all aircraft names will have the same length as the longest one. The shorter ones will have empty spaces inserted to the right (because the align is to the left).
- **Integer value** specifying the width (length) of the value. For example: <*%key_name[cw40]*> - in flight times per aircraft statistics, all aircraft names will be 40 character long. Empty spaces will inserted to the left and right of aircraft name in a way so the resulting value is 40 characters long and aligned to the center. The aircraft name can also be cropped from left and right if it is longer than 40 characters.

6.4.8. Final words for statistics templates

From reading the preceding chapters, it might seem hard to write pilot statistics template. But it is not that bad. It is much easier to learn by examples, so take a look at default templates in **ps_templates** folder and I am sure you will grasp the idea behind those templates.

6.5. *Creating JasperReports template for Logbook*

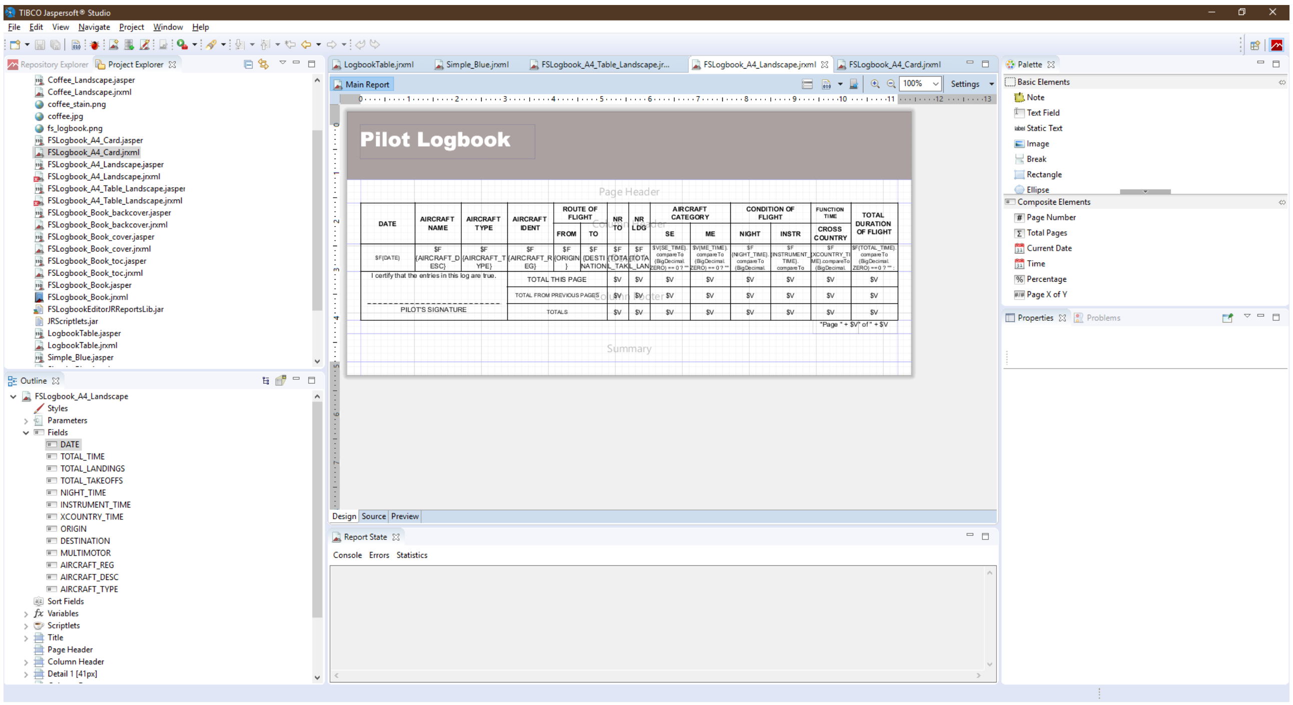
Well, I have to admit that the learning curve for JasperReports may be quite steep. And it requires some programming knowledge. And of course, it will require some time and patience to grasp it. Hope this great tutorial will still be online when you decide to try to make your own template:

https://www.tutorialspoint.com/jasper_reports/index.htm

The above tutorial is focused more on programmers. But at least the Getting Started and Life Cycle chapters are good reading to get at least the basic understanding of the concepts.

Those of us who does not need to write the report templates in notepad as some gurus, can use an editor to create a template. I used the official one - **TIBCO Jaspersoft® Studio**. Which is a free tool and works in the "you get what you click" concept :) (what you see is what you get). You can get it here:

https://community.jaspersoft.com/project/jaspersoft-studio



Picture 86: An example of template opened in Jaspersoft Studio

If you have read through some chapters of the tutorial mentioned above, you may have some idea about what are fields, parameters ... even a scriptlet maybe. These are the key concepts to grasp to actually design a template and possibly, fill it with data later on. If the following chapters, you can find out what fields, parameters and scriptlet FS Logbook Editor provides for you to use within the template.

6.5.1. *Examples*

FS Logbook Editor comes with couple of pre-made templates. These templates will get extracted into the **res/jasperreports** folder within the FS Logbook Editor installation folder. Anytime you delete a default template from this folder, it will be restored on next run of FS Logbook Editor. The templates are there with two extensions. One has the **.jasper** extension – this is compiled template and should be used preferably to generate reports. The other has **.jrxml** extension – this can be opened in the editor (or manually edited) and thus is easy to be used for learning purposes or editing. Actually, JasperSoft Studio can open even the **.jasper** template, but if you want to edit the template, use the **.jrxml** variant.

6.5.2. *Fields*

Below are fields and their data types that you can use when designing JasperReports report to use with FS Logbook Editor. Respectively, these are supported by main data source and each sub-data source which FS Logbook Editor passes to JasperReports engine when generating report from the logbook.

- **DATE** – a date of flight. Data type is: **java.util.Date**
- **TOTAL_TIME** – a total time of the flight. The value is fraction of hours (for example 1.5 means 1 and half hour). Data type is: **java.math.BigDecimal**
- **TOTAL_TAKEOFFS** – a total amount of takeoffs performed during the flight. Data type is: **int**
- **TOTAL_LANDINGS** – a total amount of landings performed during the flight. Data type is: **int**
- **NIGHT_TIME** – a time of flight flown during nigh. The value is fraction of hours (for example 1.5 means 1 and half hour). Data type is: **java.math.BigDecimal**
- **INSTRUMENT_TIME** – a time of flight flown during instrument conditions. The value is fraction of hours (for example 1.5 means 1 and half hour). Data type is: **java.math.BigDecimal**
- **XCOUNTRY_TIME** – a time of flight considered as cross-country flight time. The value is fraction of hours (for example 1.5 means 1 and half hour). Data type is: **java.math.BigDecimal**
- **ORIGIN** – an ICAO code of the origin airport. Data type is: **java.lang.String**
- **ORIGIN_NAME** – a name of the origin airport. Data type is: **java.lang.String**
- **ORIGIN_CITY** – a name of the city of the origin airport. Data type is: **java.lang.String**
- **ORIGIN_COUNTRY** – a name of the country of the origin airport. Data type is: **java.lang.String**
- **ORIGIN_STATE** – a name of the state of the origin airport. Data type is: **java.lang.String**
- **DESTINATION** – an ICAO code of the destination airport. Data type is: **java.lang.String**
- **DESTINATION_NAME** – a name of the destination airport. Data type is: **java.lang.String**
- **DESTINATION_CITY** – a name of the city of the destination airport. Data type is: **java.lang.String**
- **DESTINATION_COUNTRY** – a name of the country of the destination airport. Data type is: **java.lang.String**
- **DESTINATION_STATE** – a name of the state of the destination airport. Data type is: **java.lang.String**
- **DISTANCE_FLOWN** – a distance flown during the flight. The value is in meters. Data type is: **java.math.BigDecimal**
- **MULTIMOTOR** – whether a used aircraft had multiple engines or not. Data type is: **boolean**
- **AIRCRAFT_REG** – a registration number of the aircraft used. Data type is: **java.lang.String**
- **AIRCRAFT_DESC** – a description of the aircraft used. Data type is: **java.lang.String**

- **AIRCRAFT_TYPE** – a type of aircraft used. Will return type name in current FS Logbook Editor language. So if English, this will be *Glider/Fixed Wing/Amphibian/Rotor*. Data type is: **java.lang.String**
- **COMMENT** – a flight comment. Data type is: **java.lang.String**

You might be asking where is a field for an airport list. Well, the airport list, because it is a list, thus may contain more than one record, is implemented as another data source. You can access this data source in JasperReport report using the following expression:

> *(JRDataSource)(((com.mouseviator.jasperreports.datasource.JRLogbookDataSource) $P{REPORT_DATA_SOURCE}).getAirportList())*

And the airport list data source supports the following fields:

- **AIRPORT_CODE** – an ICAO code of airport landed at during the flight. Data type is: **java.lang.String**
- **NUM_LANDINGS** – an amount of landings performed at the respective airport. Data type is: **java.lang.Short**

6.5.3. *Parameters*

FS Logbook Editor also passes some values into each JasperReport report template as parameter for your usage. These parameters are:

- **LOGBOOK_NAME** – a name of the logbook. The logbook name is actually supported only by FS 2004 logbook file. By default (and for all other logbook files), this will be *STANDARD PILOT LOG.* Data type is: **java.lang.String**
- **LOGBOOK_FILE_NAME** – a name of the logbook file. Just the file name, without path. Data type is: **java.lang.String**
- **LOGBOOK_FILE_PATH** – a full path to logbook file. Data type is: **java.lang.String**

There is one more … type … of parameter being passed into each template. A **sub-data sources.** Each template works with main data source (a flight logbook in the case of FS Logbook Editor). If the template contains tables, charts or sub-templates – each of these elements requires their own data source. They cannot use the main one or even share the sub-data source. Each of these elements requires its own data source. These additional data-sources (sub-data sources) are passed as template parameters. The parameters are named: **subDataSource + number**. For example, **subDataSource1** – the first sub-data source, **subDataSource2** – the second sub-data source and so on. The data type of the these must be: *net.sf.jasperreports.engine.JRDataSource* (but an instance of logbook will be passed, so the same data as for main data source are available).

6.5.4. A scriptlet (with some useful functions)

What is scriptlet? Trying to stay simple, we can say it is a library with functions, which you can attach to JasperReports template and call the functions from within the template. These are usually some more complex functions, for example to manipulate dates, times etc. which cannot be easily done using variables or expressions within the template itself. For example, a logbook contains flight times and distance flown. The flight time is passed to report as fraction of hours and distance flown is in meters. If you want to display these values in different unit, like time in HH:MM:SS format, or distance in nautical miles, you would have to write pretty complex expression to convert these values. That is why, via scriptlet, FS Logbook editor offers the same functions that are used internally, for your usage in the template. These functions are all available in the **FSLogbookEditorJRReportsLib.jar** Java library. You will find that library in the **lib** folder in FS

Logbook Editor installation folder. Both, FS Logbook Editor and Jaspersoft Studio (when creating and compiling templates) requires this library to work with templates that uses the functions from within the library. FS Logbook Editor will always have it, as it is it‘s dependency library (that is why it is in the lib folder). For Jaspersoft Studio (or other editor), it will be best if you copy the library from the **lib** folder to the folder where you store the templates you are working on and link it to the Studio.

The following functions to format flight distance:

- String **formatDistanceAsKm**(BigDecimal distance, byte decimal_places, Locale locale)
- String **formatDistanceAsCm**(BigDecimal distance, byte decimal_places, Locale locale)
- String **formatDistanceAsM**(BigDecimal distance, byte decimal_places, Locale locale)
- String **formatDistanceAsNm**(BigDecimal distance, byte decimal_places, Locale locale)
- String **formatDistanceAsSm**(BigDecimal distance, byte decimal_places, Locale locale)
- String **formatDistanceAsInch**(BigDecimal distance, byte decimal_places, Locale locale)
- String **formatDistanceAsFeet**(BigDecimal distance, byte decimal_places, Locale locale)

I think the name of the functions are self explanatory. All have the same parameters:

- *BigDecimal distance* – a distance flown value. You will pass the value of the **DISTANCE_FLOWN** field as this parameter.
- *Byte decimal_places* – an amount of decimal places the resulting value should have. 2 for example.
- *Locale locale* – a locale in which to print the distance flown value. This influences some attributes, like what character is used as decimal separator - by nationality. For example, *Locale.getDefault()* for the default locale on your system, or *Locale.ENGLISH* to force English number formatting preferences.

An example of usage of these functions is below:

```
$F{DISTANCE_FLOWN}.compareTo(BigDecimal.ZERO) == 0 ? "" :
$P{LogbookScriptlet_SCRIPTLET}.formatDistanceAsNm($F{DISTANCE_FLOWN}, (byte)2, Locale.ENGLISH)
```

The above code is used in one of the pre-made templates to display distance flown. Well, it requires some Java programming knowledge, but it says, that if the value of distance flown is zero

```
$F{DISTANCE_FLOWN}.compareTo(BigDecimal.ZERO) == 0 ?
```

than display an empty string

```
""
```

and if not empty, display the distance flown formatted as nautical miles with 2 digit decimal places precision (the code that follows after the : does it):

```
$P{LogbookScriptlet_SCRIPTLET}.formatDistanceAsNm($F{DISTANCE_FLOWN}, (byte)2, Locale.ENGLISH)
```

There is also a second set of functions, but these take care of formatting flight time:

- String **formatTimeAsHHMMSS**(BigDecimal time) – formats time as HH:MM:SS. For example, if given time is 3 hours, 22 minutes and 25 seconds, the resulting value will be 03:22:25.
- String **formatTimeAsHHMM**(BigDecimal time) – formats time as HH:MM. For example,

if given time is 3 hours, 22 minutes and 25 seconds, the resulting value will be 03:22.

- String **formatTimeAsHMMSS**(BigDecimal time) – formats time as H:MM:SS. For example, if given time is 3 hours, 22 minutes and 25 seconds, the resulting value will be 3:22:25. The leading zero will not be printed.
- String **formatTimeAsHMM**(BigDecimal time) – formats time as H:MM. For example, if given time is 3 hours, 22 minutes and 25 seconds, the resulting value will be 3:22. The leading zero will not be printed.
- String **formatTimeAsDecimal1**(BigDecimal time) – formats time as decimal value (fraction of hours) with 1 decimal place precision. For example, if given time is 1 hour 30 minutes, the resulting value will be 1.5.
- String **formatTimeAsDecimal2**(BigDecimal time) – formats time as decimal value (fraction of hours) with 2 decimal places precision. For example, if given time is 1 hour 30 minutes, the resulting value will be 1.50.
- String **formatTimeAsReal**(BigDecimal time) – formats time as decimal value (fraction of hours) with default precision. You might end up will really long number (lots of decimal places). But it will be pretty precise.

Here is an example of usage of these functions:

```
$F{INSTRUMENT_TIME}.compareTo(BigDecimal.ZERO) == 0 ? "" :
$P{LogbookScriptlet_SCRIPTLET}.formatTimeAsHHMMSS($F{INSTRUMENT_TIME})
```

The above code is used in one of the pre-made templates to display instrument time. Well, it requires some Java programming knowledge, but it says, that if the value of instrument time is zero

```
$F{INSTRUMENT_TIME}.compareTo(BigDecimal.ZERO) == 0 ?
```

than display an empty string

```
""
```

and if not empty, display the instrument time formatted as HH:MM:SS (the code that follows after the : does it):

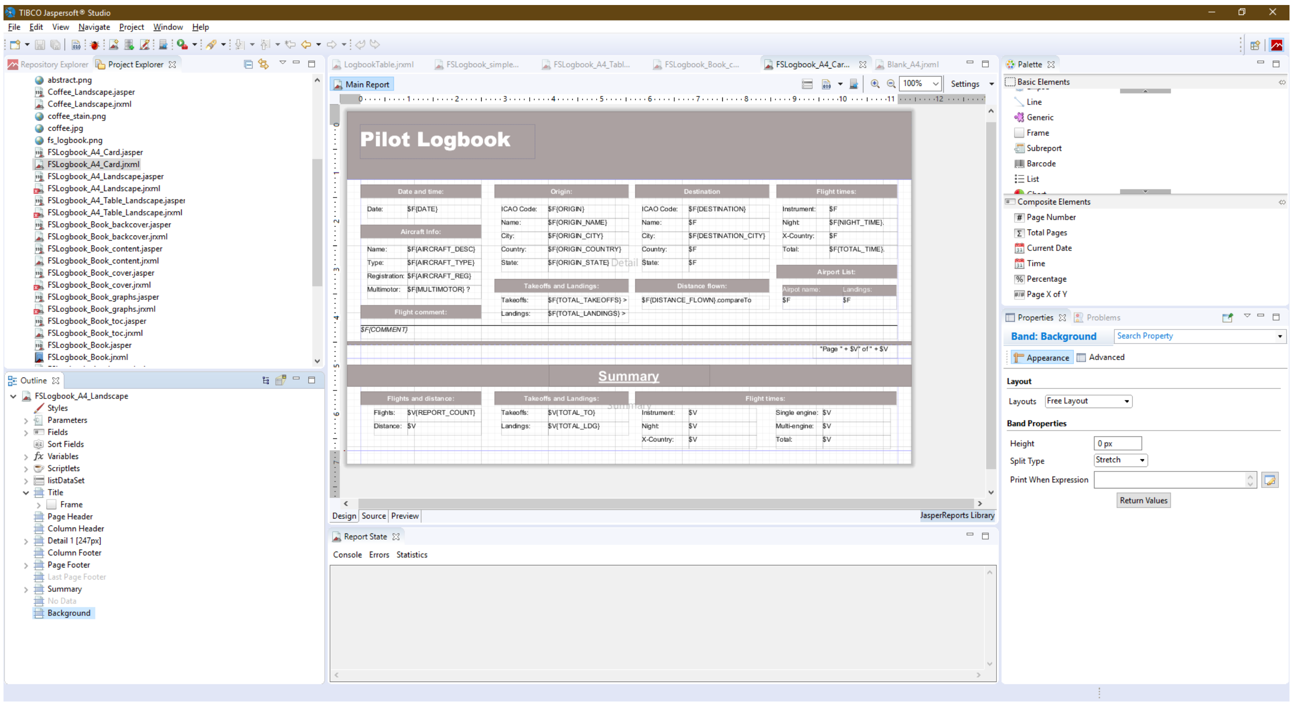
```
$P{LogbookScriptlet_SCRIPTLET}.formatTimeAsHHMMSS($F{INSTRUMENT_TIME})
```

6.5.5. Creating simple template using Jaspersoft Studio

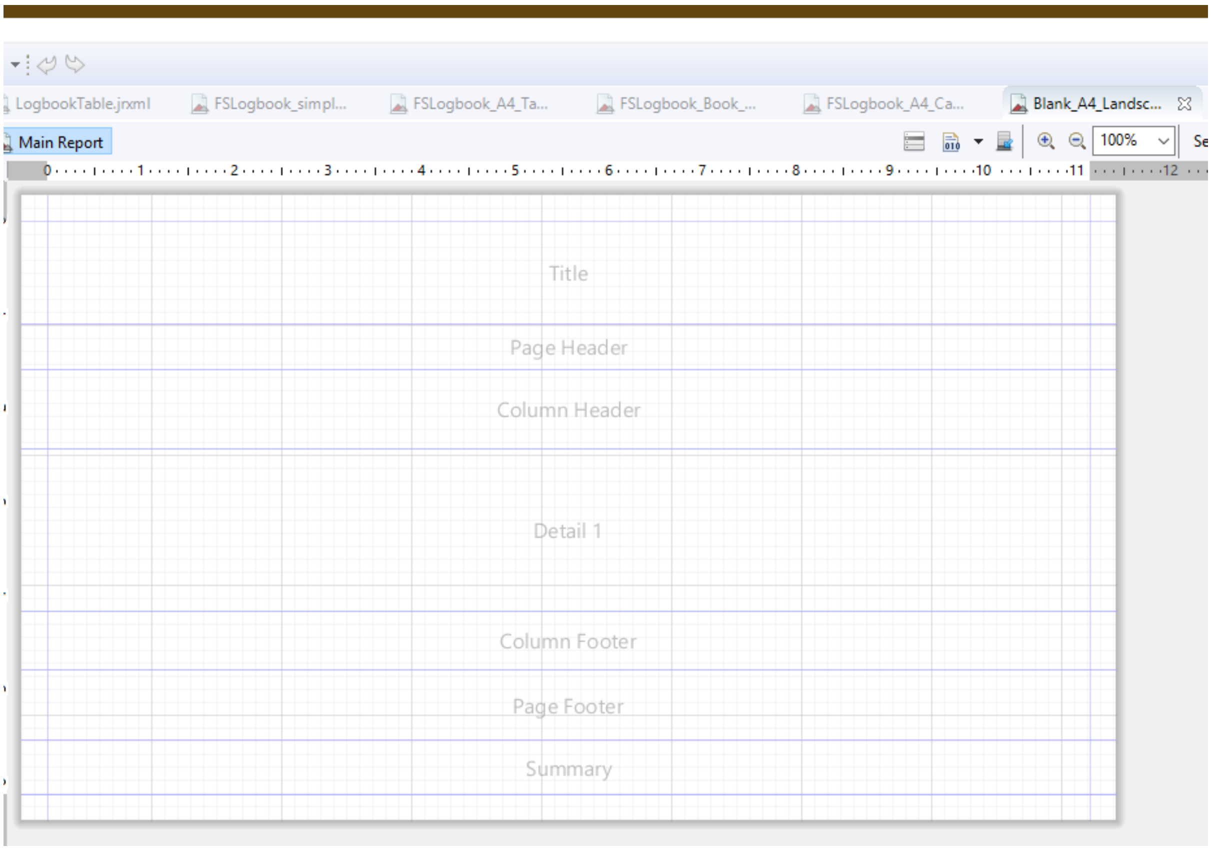
Ok, in this chapter I try to describe how to create a simple template using Jaspersoft Studio for use with FS Logbook Editor. The Jaspersoft Studio is the same software that I used to create all the templates shipped with FS Logbook Editor. At the time of writing this, the actual version was 6.8.0 and it was available here: *https://community.jaspersoft.com/project/jaspersoft-studio*.

Firstly, I am sorry that I would not write this "tutorial" as for the complete beginner. It will be very helpful, if you have some, at least the basic knowledge of programming. If you do not know what a data type is (like String) I guess not much of this chapter will make much sense to you, as some JasperReports principles might be confusing even for programmers. But hey, the "tutorial" chapter will not be that complex, so get into it :)

On the next picture you can see what we will try to achieve:


Picture 87: A JasperReports report template we will create

But you will start with the new, empty template. So in Jaspersoft Studio, choose **File** → **New** → **Jasper Report**. This will open dialog to create new report. Select "**Blank A4 Landscape**", give it some name (the name of the file – like *FSLogbook_A4_card_tutor*), and click **Ok**. Now you will have an empty template, looking like this:

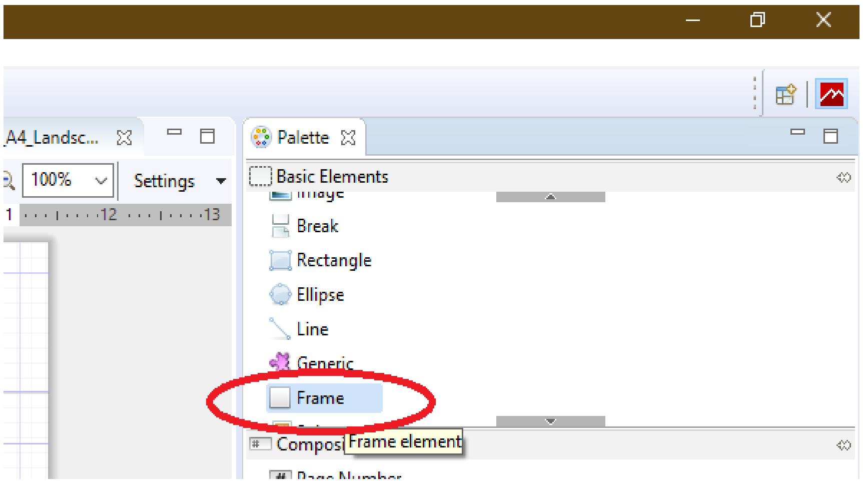


Picture 88: An empty JasperReports template

You can see there are several sections (called bands), separated by blue lines. Jaspersoft Studio is a WYSIWYG (What you see is what you get) editor, so you can grab those lines (or document borders), drag them and resize the respective band. You can read more about bands in Jaspersoft documentation. In this tutorial, we will use only the **Title**, **Detail**, **Page Footer** and **Summary** band.

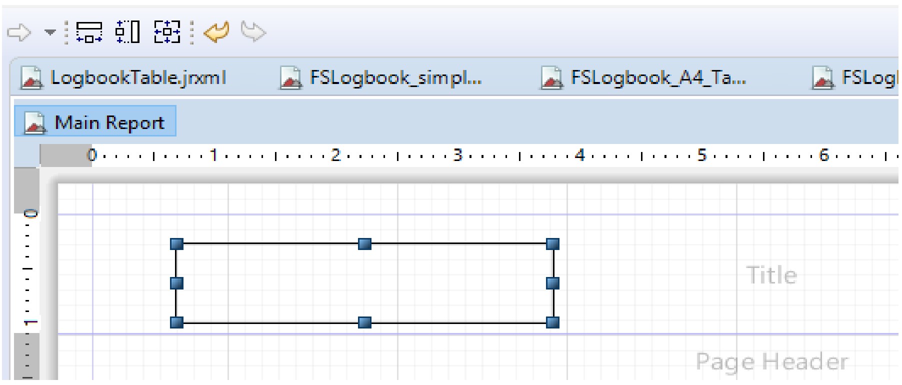
6.5.5.1. *Creating the title*

The title band in this example is made of one **Frame** element and one **Static text** element. So, in the Palette on the right side choose the Frame element (click on it).



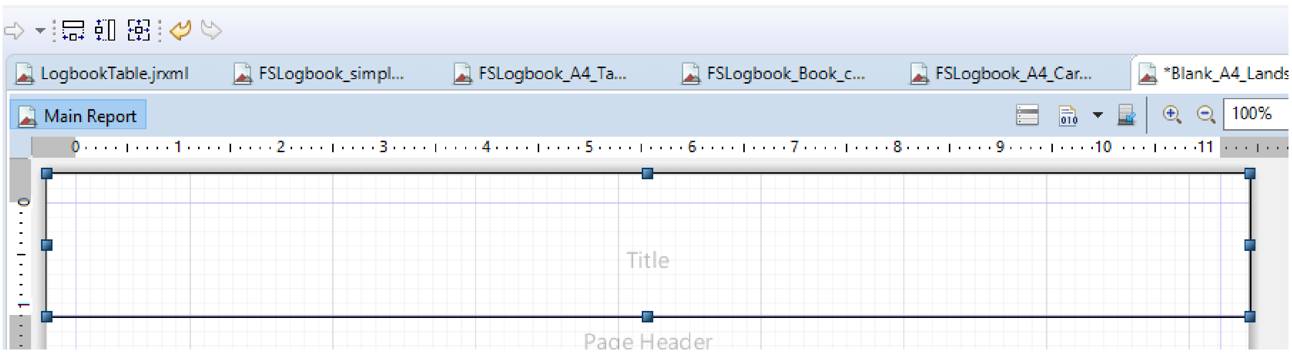
Picture 89: Selecting Frame element

Now click and drag somewhere in the Title band. You will see a dashed rectangle as you drag the mouse (with top left corner in the place where you clicked initially). Create a rectangle and release the mouse button. So your template should now look something like this:



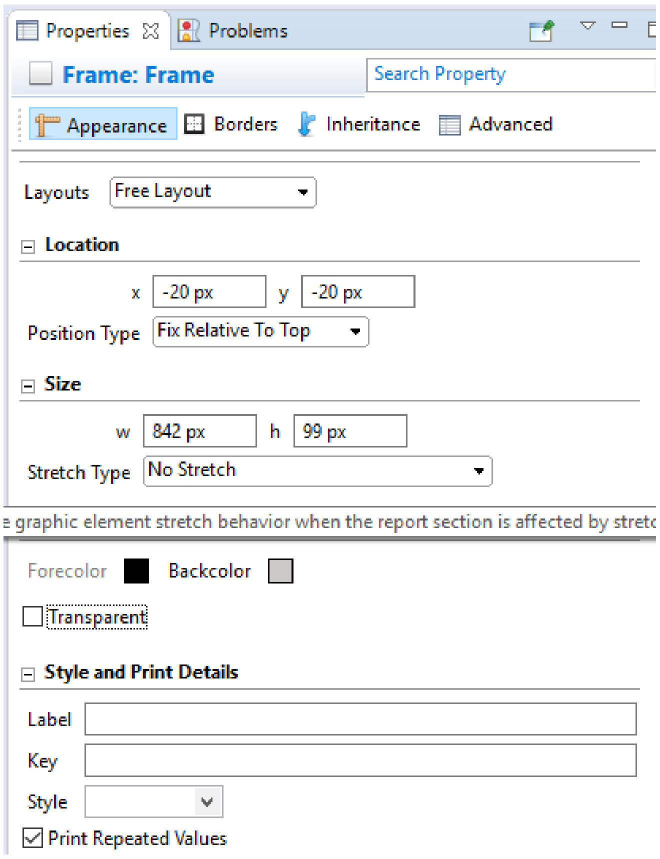
Picture 90: An initial Frame element creation

Now click and drag the blue boxes on the sides of the Frame you just created and “size” it so that the bottom line matches the blue line separating the **Title** and **Page Header** band. If you did not change any settings, the Frame should snap as you approach the blue line. For the left, right and top edge, drag them so that they snap to the document edges (over the first vertical and horizontal lines, which are borders of the content. The resulting band, should look like the one on the next picture:



Picture 91: Frame resized to match document borders

Now, with the Frame still selected, look at the bottom right side of your screen. There is a Properties palette, which, because the band is selected, allows you to edit properties of the Frame.



Layouts: Free Layout

Location

x: -20 px y: -20 px

Position Type: Fix Relative To Top

Size

w: 842 px h: 99 px

Stretch Type: No Stretch

graphic element stretch behavior when the report section is affected by stretc

Forecolor ■ Backcolor □

☐ Transparent

Style and Print Details

Label:

Key:

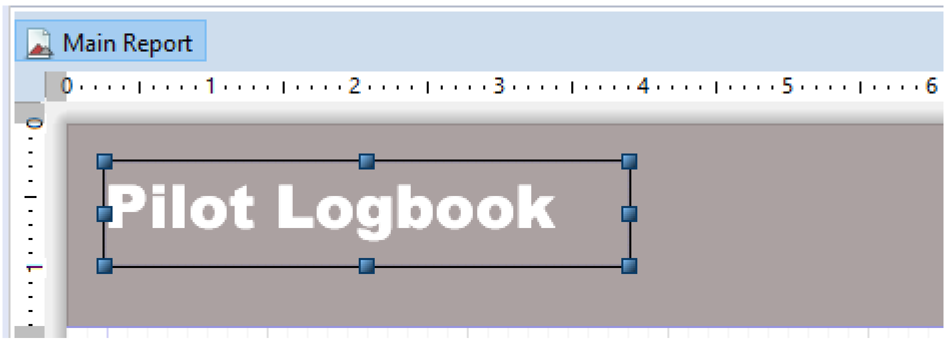
Style:

☑ Print Repeated Values

Picture 92: A Frame properties

For the Frame element, I just unchecked the **Transparent** property and choose a gray color for background. There are many other options you can change. Many of them will also give you a quick tip of what they do if you just hover with the mouse over them. Just don‘ t be shay and experiment with them. Look also into the **Borders** tab, if you want to customize borders.

Now we will insert a „Pilot Logbook“ text. Go back to the Palette where you chose Frame element a while ago, but this time, choose the **Static text** element. Create and size the static text element the same way you created the Frame element. Static text element, as the name suggests, may contain text. Double click it, type the „Pilot Logbook“ text and press Enter (to accept the value). Now focus back on the Properties palette. In the **Appearance** tab you can change Fore color (font color). I chose white. And in the **Static text** tab, you can change font (among other properties). I left the default font, but changed the size to something like 26 and made the text Bold. With a little luck, your static text now looks similar to the one below:

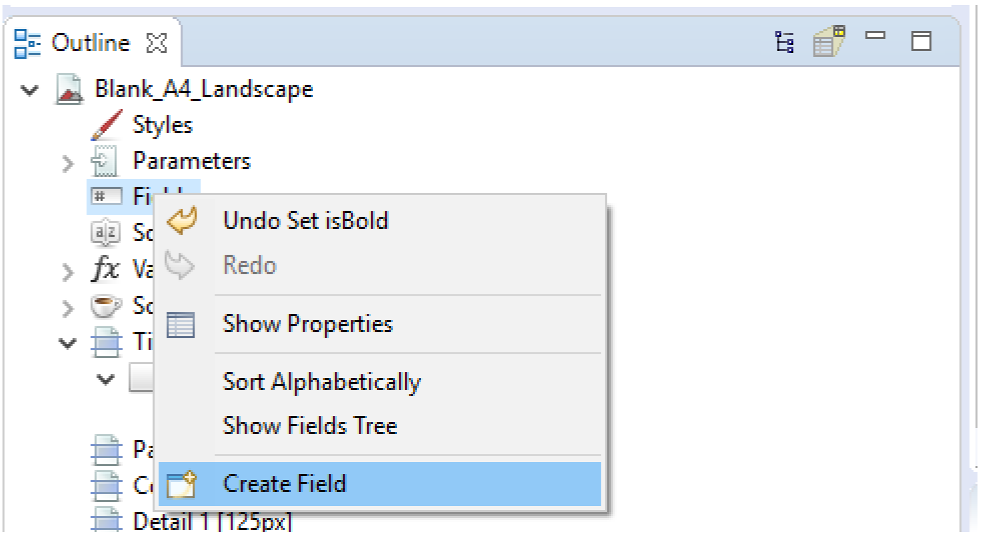


Picture 93: The "Pilot Logbook" static text

It is not that hard, is it? I believe that now you have the understanding on how to create elements in the template. And it is all about this, creating various elements and setting their properties. Ok, so let' s get into some more serious stuff.

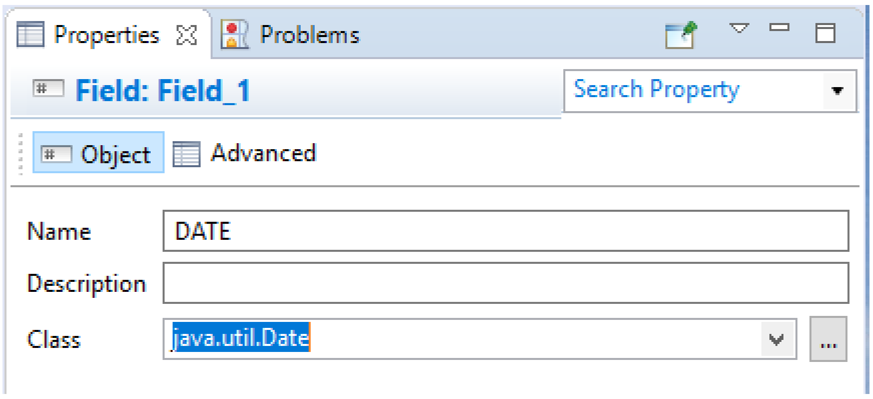
6.5.5.2. *Defining fields*

So far, we did not pay much of an attention to the left side of Jaspersoft Studio. Take a look now. At the left top part, you should see a **Repository Explorer** and **Project Explorer**. In Project Explorer, you will see all reports you added to Project, all you are currently working on. You open them, delete them, rename them and otherwise manage them here. Below those, there is an **Outline** view (palette) which displays the structure of your template in a tree view. It displays bands, but also components, that are not visible in the template – fields, variables etc. We need to create fields so that we can use them in the Detail band to print our flight information. Right click on the **Fields** section in the Outline view and choose **Create Field** from the menu:



Picture 94: Creating field

This will create new field, which will be named something like **Field_1** (A field word with some number). You can edit the field properties in the **Properties** window on the right side of the editor. Set it like on the picture below:



Field: Field_1

Name	DATE
Description	
Class	java.util.Date

Picture 95: Creating a DATE field

Set the **name** of the field to **DATE** and **class** to: **java.util.Date**. By this, you created a date of flight field. Later on, we will use it in the template and even later on, when you use FS Logbook Editor to generate report from this template, it will insert the date of flight of each respective logbook record in where the field is used. See chapter ***Fields*** for all the fields you can use. Note that all fields names and data types (class) must be exactly the same as specified in the ***Fields*** chapter, otherwise, the field will not work, will cause error or do some other nasty stuff.

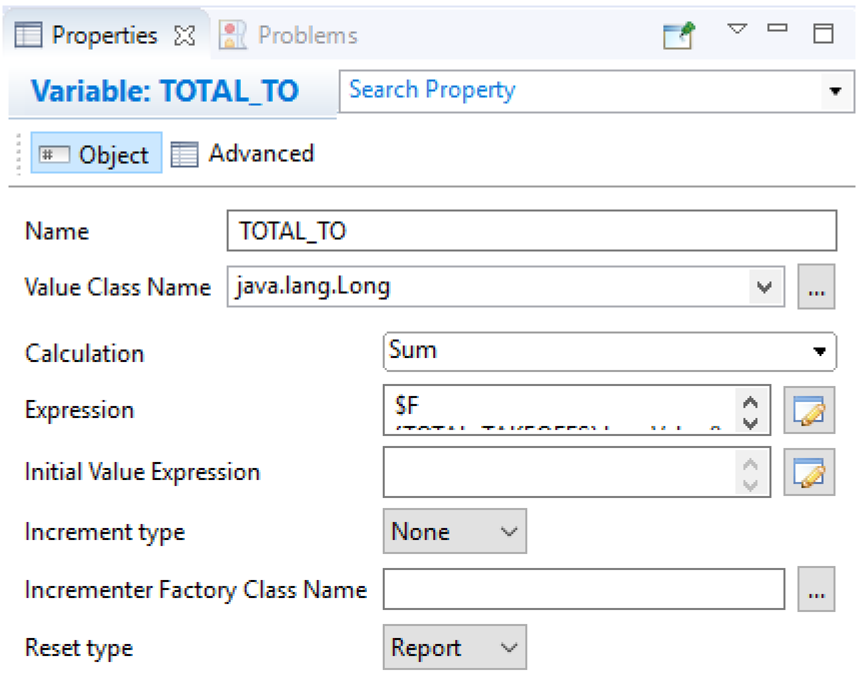
The example template uses all the fields available, so define them the same way as the DATE field. Just so you do not have to list back to the ***Fields*** chapter, we will use these fields:

- **TOTAL_TIME ,** Data type is: **java.math.BigDecimal**
- **TOTAL_TAKEOFFS ,** Data type is: **int**
- **TOTAL_LANDINGS ,** Data type is: **int**
- **NIGHT_TIME ,** Data type is: **java.math.BigDecimal**
- **INSTRUMENT_TIME ,** Data type is: **java.math.BigDecimal**
- **XCOUNTRY_TIME ,** Data type is: **java.math.BigDecimal**
- **ORIGIN ,** Data type is: **java.lang.String**
- **ORIGIN_NAME ,** Data type is: **java.lang.String**
- **ORIGIN_CITY ,** Data type is: **java.lang.String**
- **ORIGIN_COUNTRY ,** Data type is: **java.lang.String**
- **ORIGIN_STATE ,** Data type is: **java.lang.String**
- **DESTINATION ,** Data type is: **java.lang.String**
- **DESTINATION_NAME ,** Data type is: **java.lang.String**
- **DESTINATION_CITY ,** Data type is: **java.lang.String**
- **DESTINATION_COUNTRY ,** Data type is: **java.lang.String**
- **DESTINATION_STATE ,** Data type is: **java.lang.String**
- **DISTANCE_FLOWN ,** Data type is: **java.math.BigDecimal**
- **MULTIMOTOR ,** Data type is: **boolean**
- **AIRCRAFT_REG ,** Data type is: **java.lang.String**
- **AIRCRAFT_DESC ,** Data type is: **java.lang.String**
- **AIRCRAFT_TYPE ,** Data type is: **java.lang.String**
- **COMMENT ,** Data type is: **java.lang.String**

Staying in the Outline view, we will also define variables and the Scriptlet.

6.5.5.3. *Defining variables*

Variables are used to store partial results and do more complex calculations with data retrieved from data sources. You can than use these values in the report or in other variables. You create variables the same way you created fields. Find the **Variables** in the **Outline** window. Right click on the **Variables** and select **Create variable**. A new variable, like **Variable_1** (the word variable + number) will be created and you can edit variable properties on the right side of the editor in the Properties window. Let' s see how to create TOTAL_TO variable which counts the total amount of takeoffs per whole logbook. So, create new variable as described earlier. Now, set the variable properties as shown below:



Name	TOTAL_TO
Value Class Name	java.lang.Long
Calculation	Sum
Expression	$F
Initial Value Expression	
Increment type	None
Incrementer Factory Class Name	
Reset type	Report

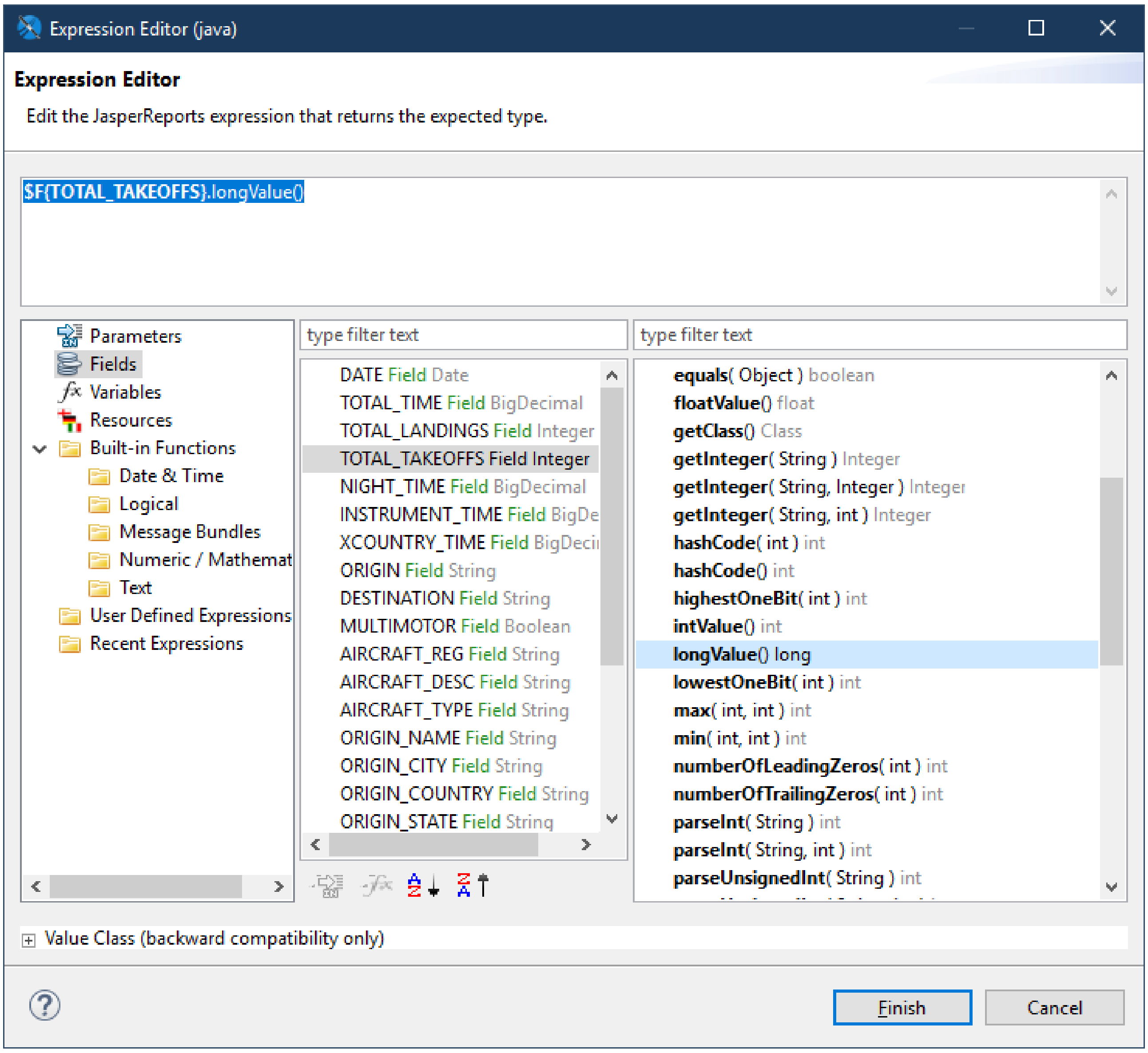
Picture 96: TOTAL_TO variable properties

Set the properties as follows:

- Name: TOTAL_TO
- Value Class Name: java.lang.Long
- Calculation: Sum
- Expression: $F{TOTAL_TAKEOFFS}.longValue()
- Increment type: None
- Reset type: Report

What does that all mean? Well, the **name**, is the name of the variable. This is how we will reference this variable further on. The **Value Class Name** defines the data type of the variable. Here, the programming knowledge comes into play. If you do not know, let's just say that **java.lang.Long** can store quite a big number. The **Calculation** property says, if any function should be performed on this variable. We selected **Sum**, which means Summary, so to add each value to previous one to calculate some total. Bu what we will summarize you ask? That is what the **Expression** property does. The $F{TOTAL_TAKEOFFS} says: take the value of the **TOTAL_TAKEOFFS** field we defined earlier as field. And the ".longValue()" converts the value to **java.lang.Long**. Because if you take a look at the definition of fields, you will see, that **TOTAL_TAKEOFFS** data type is **int**. For no-programmers, **int** represent also a number, but the maximum value is roughly twice smaller than for the **Long**. But since this variable summarizes all the values, **int** may not be enough, so that is why it is defined as **Long** and why we need to convert the value. The **Increment type** option defines, when the value of the variable should be changed. The **None** type means with every change of data source record (with each flight processed). And the **Reset type** property tells JasperReports engine when to reset the variable to zero (the **Initial Value Expression**). We set **Report**, which means at the end of the report, thus, after all flights are processed. This way, the variable computes the total amount of takeoffs from flights that the JasperReports engine will process when you use the template in FS Logbook Editor.

For the **Expression** and **Initial Value Expression** you can open the Expression editor by clicking at the button with the image of dialog on the right side of the input field.


Expression Editor (java)

Expression Editor

Edit the JasperReports expression that returns the expected type.

```
$F{TOTAL_TAKEOFFS}.longValue()
```

Parameters
Fields
Variables
Resources
Built-in Functions
Date & Time
Logical
Message Bundles
Numeric / Mathemat
Text
User Defined Expressions
Recent Expressions

type filter text

DATE Field Date
TOTAL_TIME Field BigDecimal
TOTAL_LANDINGS Field Integer
TOTAL_TAKEOFFS Field Integer
NIGHT_TIME Field BigDecimal
INSTRUMENT_TIME Field BigDe
XCOUNTRY_TIME Field BigDeci
ORIGIN Field String
DESTINATION Field String
MULTIMOTOR Field Boolean
AIRCRAFT_REG Field String
AIRCRAFT_DESC Field String
AIRCRAFT_TYPE Field String
ORIGIN_NAME Field String
ORIGIN_CITY Field String
ORIGIN_COUNTRY Field String
ORIGIN_STATE Field String

type filter text

equals(Object) boolean
floatValue() float
getClass() Class
getInteger(String) Integer
getInteger(String, Integer) Integer
getInteger(String, int) Integer
hashCode(int) int
hashCode() int
highestOneBit(int) int
intValue() int
longValue() long
lowestOneBit(int) int
max(int, int) int
min(int, int) int
numberOfLeadingZeros(int) int
numberOfTrailingZeros(int) int
parseInt(String) int
parseInt(String, int) int
parseUnsignedInt(String) int

Value Class (backward compatibility only)

Picture 97: The Expression editor

At the picture above, you can see how we can easily “click” the expression for the **TOTAL_TAKEOFFS** variable. In the left column, you see objects we can work with to create expression. There are fields, variables, resource. For this expression, we will use field, so **Fields** is selected in the left column. In the middle column, you can see all the fields defined. So select **TOTAL_TAKEOFFS**. And in the right column, you can see all the methods / functions the field /

variable / resource offers. Find the **.longValue()** method and double click it. Now the whole expression will be placed in the text window at the top of the dialog and we can close the dialog with the **Finish** button. The expression will than be placed in the Expression property.

The other needed variables are created the exact same way. I believe you can do it now. So only the list of variables and their respective properties follows.

- **TOTAL_LDG** (*total amount of landings per whole logbook*)
  - Name: TOTAL_LDG
  - Value Class Name: java.lang.Long
  - Calculation: Sum
  - Expression: **$F{TOTAL_LANDINGS}**.longValue()
  - Increment type: None
  - Reset type: Report
- **TOTAL_TOTAL_TIME** (*total flight time for the whole logbook*)
  - Name: TOTAL_TOTAL_TIME
  - Value Class Name: java.math.BigDecimal
  - Calculation: Sum
  - Expression: **$F{TOTAL_TIME}**
  - Increment type: None
  - Reset type: Report
- **TOTAL_NIGHT_TIME** (*total night time for the whole logbook*)
  - Name: TOTAL_NIGHT_TIME
  - Value Class Name: java.math.BigDecimal
  - Calculation: Sum
  - Expression: **$F{NIGHT_TIME}**
  - Increment type: None
  - Reset type: Report
- **TOTAL_INSTRUMENT_TIME** (*total instrument time for the whole logbook*)
  - Name: TOTAL_INSTRUMENT_TIME
  - Value Class Name: java.math.BigDecimal
  - Calculation: Sum
  - Expression: **$F{INSTRUMENT_TIME}**
  - Increment type: None
  - Reset type: Report
- **TOTAL_XCOUNTRY_TIME** (*total cross country time for the whole logbook*)
  - Name: TOTAL_XCOUNTRY_TIME
  - Value Class Name: java.math.BigDecimal
  - Calculation: Sum
  - Expression: **$F{XCOUNTRY_TIME}**
  - Increment type: None
  - Reset type: Report
- **TOTAL_DISTANCE_FLOWN** (*total distance flown for the whole logbook*)
  - Name: TOTAL_DISTANCE_FLOWN

  - Value Class Name: java.math.BigDecimal
  - Calculation: Sum
  - Expression: **$F{DISTANCE_FLOWN}**
  - Increment type: None
  - Reset type: Report
- **SE_TIME** (*single engine time for the flight. This will be computed for each record / flight. The expression tests if the aircraft in the flight is marked as multi-engine (motor). If yes, the single engine time will be zero, if not, it will be equal to the total flight time.*)
  - Name: SE_TIME
  - Value Class Name: java.math.BigDecimal
  - Calculation: No Calculation Function
  - Expression: **$F{MULTIMOTOR}** == Boolean.TRUE ? BigDecimal.ZERO : **$F{TOTAL_TIME}**
  - Increment type: None
  - Reset type: None
- **ME_TIME** (*multi engine time for the flight. This will be computed for each record / flight. The expression tests if the aircraft in the flight is marked as multi-engine (motor). If yes, the multi engine time will be equal to the total flight time, if not, it will be zero.*)
  - Name: SE_TIME
  - Value Class Name: java.math.BigDecimal
  - Calculation: No Calculation Function
  - Expression: **$F{MULTIMOTOR}** == Boolean.TRUE ? **$F{TOTAL_TIME}** : BigDecimal.ZERO
  - Increment type: None
  - Reset type: None
- **TOTAL_SE_TIME** (*total single engine time for the whole logbook. Note that this variable uses another variable – SE_TIME*)
  - Name: TOTAL_SE_TIME
  - Value Class Name: java.math.BigDecimal
  - Calculation: Sum
  - Expression: **$V{SE_TIME}**
  - Increment type: None
  - Reset type: Report
- **TOTAL_ME_TIME** (*total multi engine time for the whole logbook. Note that this variable uses another variable – ME_TIME*)
  - Name: TOTAL_SE_TIME
  - Value Class Name: java.math.BigDecimal
  - Calculation: Sum
  - Expression: **$V{ME_TIME}**
  - Increment type: None
  - Reset type: Report

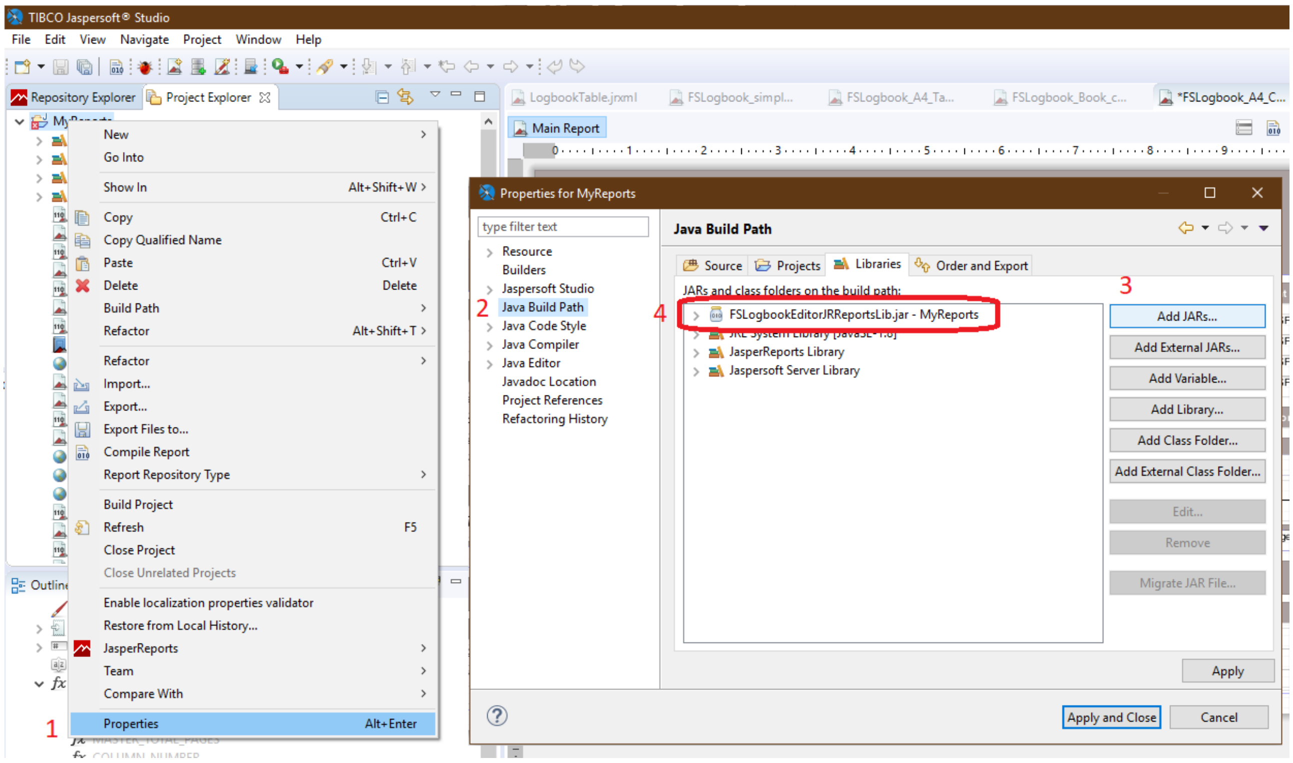
6.5.5.4. *Defining Scriptlet*

A scriptlet provides the functions described at the ***A scriptlet (with some useful functions)*** chapter.

Some of these functions we will use and thus, we need to define the scriptlet so that Jaspersoft Studio knows about the functions.

First of all, you need to copy the **FSLogbookEditorJRReportsLib.jar** from the **lib** folder inside the FS Logbook Editor installation folder, to, preferably, the folder in which you store your work-in -progress Jaspersoft templates. By default, this folder will be **MyReports** inside your user profile folder (On Windows, for example: *C:\Users\<your user name>\JaspersoftWorkspace\MyReports*).

Than, you need to link the library to your Jaspersoft project. In the Project Explorer window, right click on the project and select **Properties** (1).

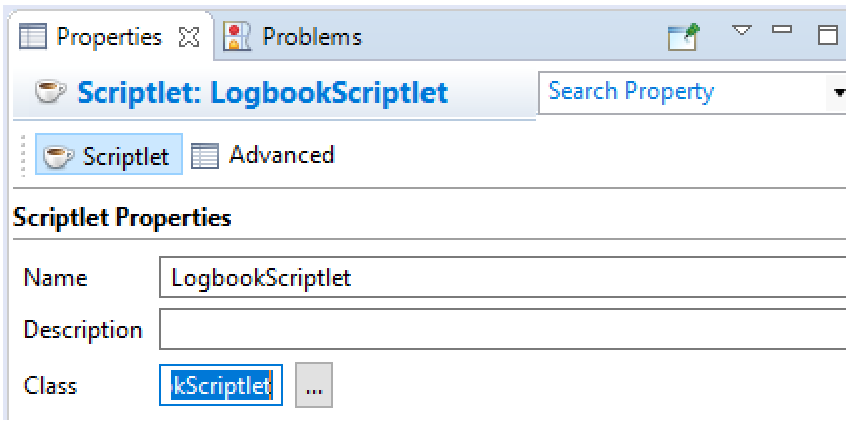


Picture 98: Adding scriptlet library to project

On the Properties dialog that appears select **Java Build Path** (2). Click on the **Add JARs…** (3) button. The standard file selection dialog will appear, so navigate to the folder where you copied the **FSLogbookEditorJRReportsLib.jar** file and select it. If you did that properly, the library will appear in the list of JARs and class folders (4) as in the picture above. Close the dialog with the **Apply and Close** button.

Now we can define the scriptlet itself. Back again in the Outline window (where you added fields and variables), find **Scriptlets**, right click and select **Create Scriptlet**. And set it’s properties:

- Name: LogbookScriptlet
- Class: com.mouseviator.jasperreports.scriptlets.LogbookScriptlet


Properties
Problems
Scriptlet: LogbookScriptlet
Search Property
Scriptlet
Advanced

Scriptlet Properties

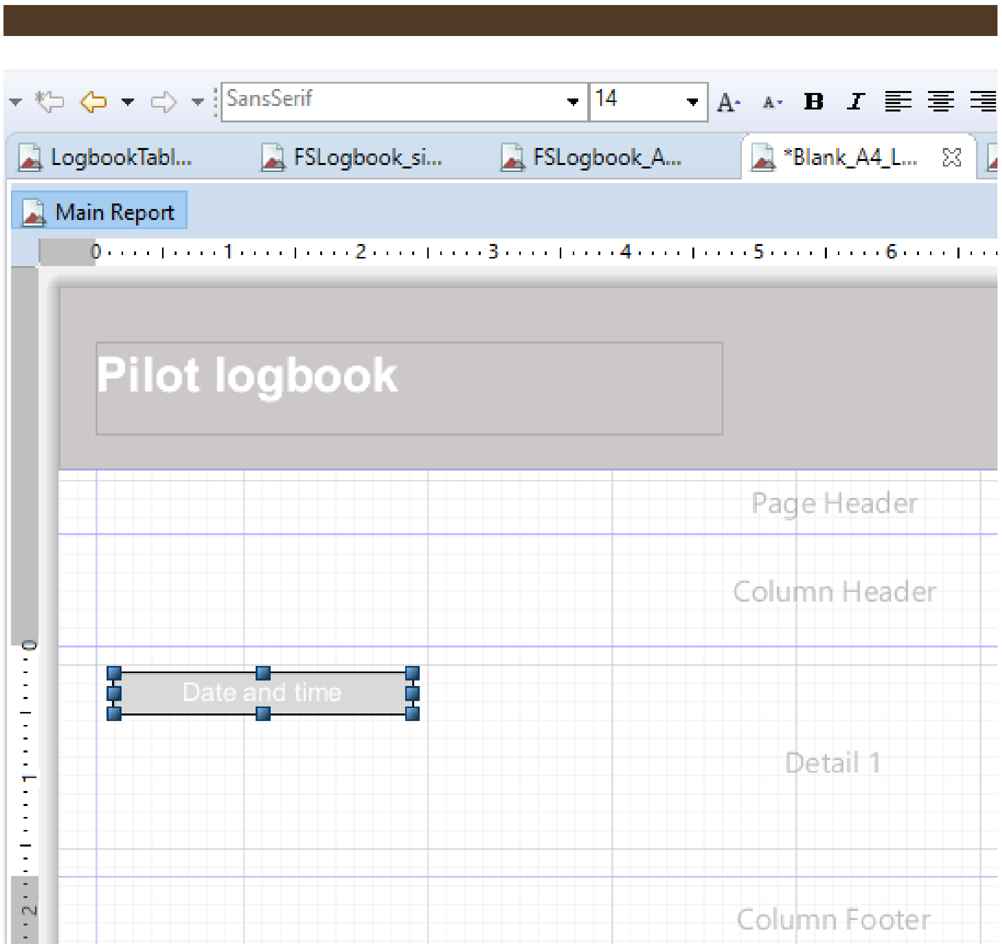
Name: LogbookScriptlet
Description:
Class: kScriptlet

Picture 99: Logbook scriptlet properties

That's it for the scriptlet.

6.5.5.5. Creating the body of the template, the Details band

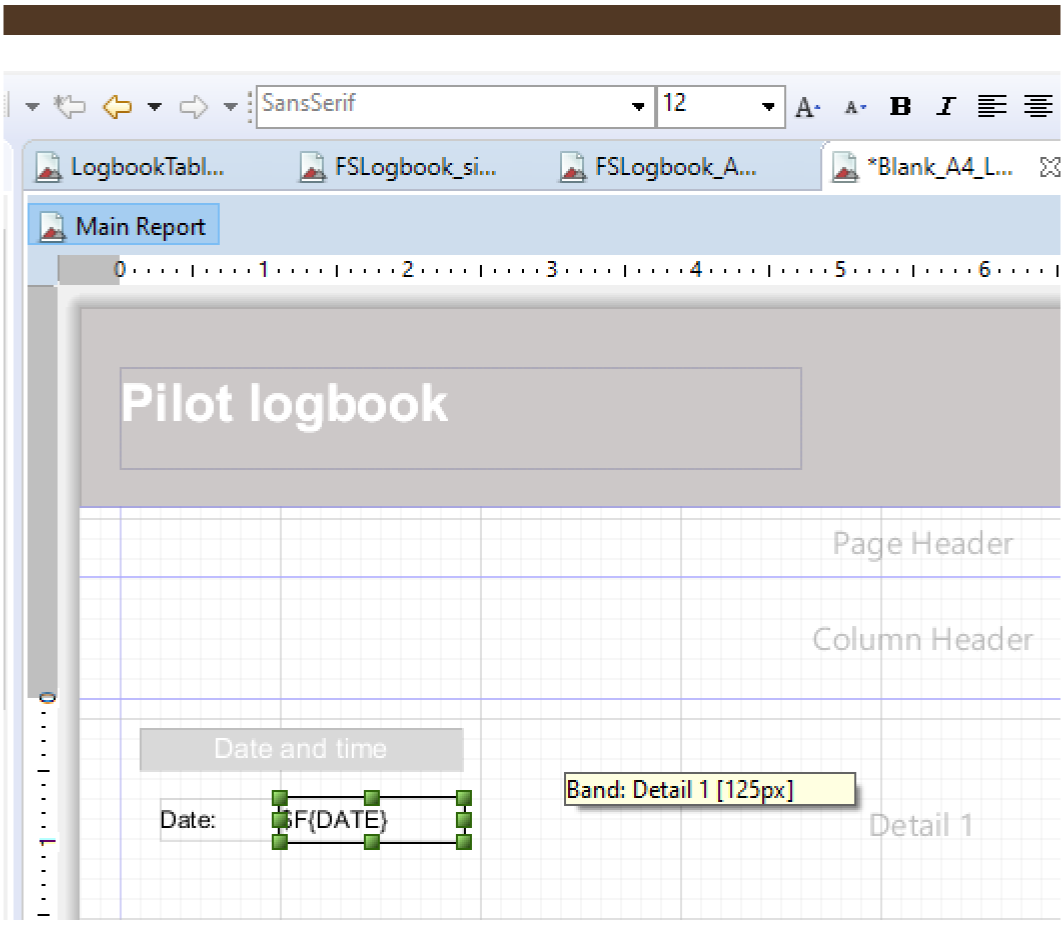
We will place the elements in the details band the very same way we placed elements (Static text and Frame) into the **Title** band. So, for start, select a **Static text** element and place it somewhere to the top left corner of the Details band. As a text, type "**Date and time**", make it **bold** and set its size to somewhat like **12 pixels**. Set the background color to the same gray as for the Frame in the Title band and fore color to white. Your template should look like on the picture below:



Picture 100: Creating static elements inside Deatils band

Below this “Date and time” static text, create another static text. Set its text to “**Date:**” and leave it transparent (no gray background like the previous one).

Now, this is when we are going to insert the actual magic, the Text element which will display the actual flight date. In the **Palette** select the **Text Field** element and place it to the right of the Date static text you just created. You create the Text field the same way you did create all the previous elements. What is going to be different, are the text field properties. So with the text field selected, look at the **Properties** window. Set the **Expression** property to: **$F{DATE}**. You can use the Expression Editor for this. Just click on the icon to the right of the Expression property. It will open the Expression Editor (you could see that on ***Picture 97: The Expression editor***). In the Expression Editor, select Fields and then double click the **DATE** field. Close the dialog with the Finish button. The Expression property should now contain the expression I mentioned a couple of words earlier. Set the **Pattern** property to: **dd.MM.yyyy hh:mm:ss**. This will format the date into more friendly format. (It is Java date format pattern. It would take too much place to explain if you have no idea, so search on the internet on how to format date in Java). That’s it for this text field. When FS Logbook Editor will fill this template, a date of the flight will be inserted here and formatted as per the pattern. You can set text font, alignment and other properties as for your liking. This is what your template should look like now:



Picture 101: A DATE text field added to template.

The rest of the details band is constructed in the same manner as this. So I will only list the properties of the rest of the Text fields you should create and let you do it on your own. Oh, OK, there is one exception, the airport list, but we will discuss it later in this chapter.

Properties for other Text fields:

- **Aircraft name:**
  - Expression: $F{AIRCRAFT_DESC}
- **Aircraft type:**
  - Expression: $F{AIRCRAFT_TYPE}
- **Aircraft registration:**
  - Expression: $F{AIRCRAFT_REG}
- **Multimotor:**
  - Expression: $F{MULTIMOTOR} ? "Yes" : "No"
  - *This one tests if MULTIMOTOR field (if aircraft used during the flight had more than one engine) is True, and displays "Yes" in the Text field if it is. If not, "No" id displayed.*
- **Origin:**
  - Expression: $F{ORIGIN}
- **Origin name:**
  - Expression: $F{ORIGIN_NAME}
- **Origin city:**
  - Expression: $F{ORIGIN_CITY}
- **Origin country:**
  - Expression: $F{ORIGIN_COUNTRY}
- **Origin state:**
  - Expression: $F{ORIGIN_STATE}
- **Destination:**
  - Expression: $F{DESTINATION}
- **Destination name:**
  - Expression: $F{DESTINATION_NAME}
- **Destination city:**
  - Expression: $F{DESTINATION_CITY}
- **Destination country:**
  - Expression: $F{DESTINATION_COUNTRY}
- **Destination state:**
  - Expression: $F{DESTINATION_STATE}
- **Takeoffs:**
  - Expression: $F{TOTAL_TAKEOFFS} > 0 ? $F{TOTAL_TAKEOFFS} : ""
  - *If TOTAL_TAKEOFFS is zero, an empty string "" is displayed. If not, the value of TOTAL_TAKEOFFS is displayed.*
- **Landings:**
  - Expression: $F{TOTAL_LANDINGS} > 0 ? $F{TOTAL_LANDINGS} : ""
  - *If TOTAL_LANDINGS is zero, an empty string "" is displayed. If not, the value of TOTAL_ LANDINGS is displayed.*
- **Instrument time:**
  - Expression: $F{INSTRUMENT_TIME}.compareTo(BigDecimal.ZERO) == 0 ? "" : $P{LogbookScriptlet_SCRIPTLET}.formatTimeAsHHMMSS($F{INSTRUMENT_TIME})

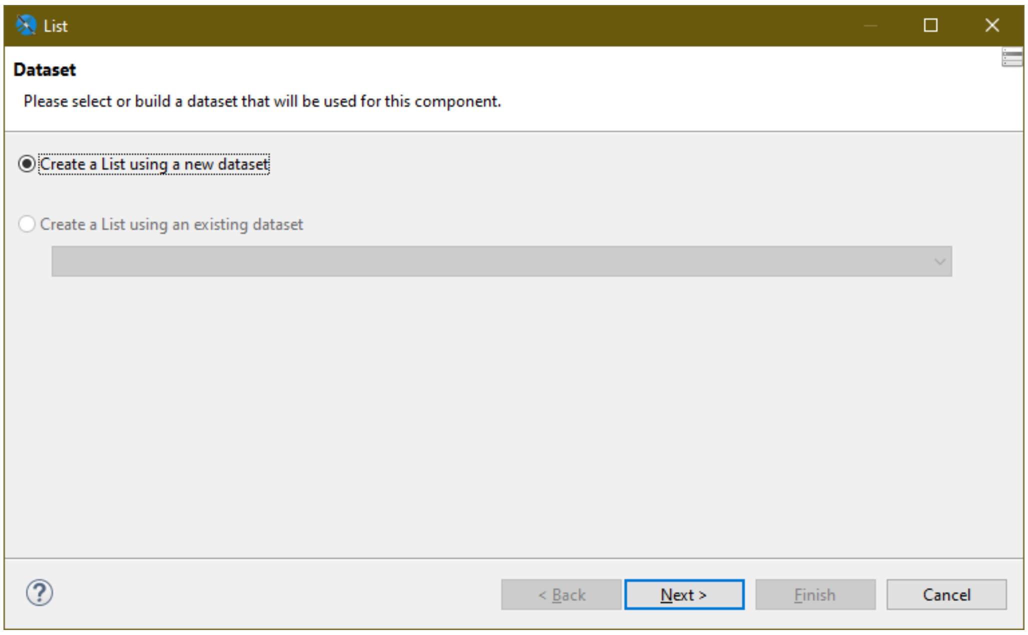
  - *If instrument time is zero, an empty string "" is displayed. If not, the value of instrument time formatted using function from the scriptlet is displayed.*
- **Night time:**
  - Expression: `$F{NIGHT_TIME}.compareTo(BigDecimal.ZERO) == 0 ? "" : $P{LogbookScriptlet_SCRIPTLET}.formatTimeAsHHMMSS($F{NIGHT_TIME})`
  - *If night time is zero, an empty string "" is displayed. If not, the value of night time formatted using function from the scriptlet is displayed.*
- **Cross country time:**
  - Expression: `$F{XCOUNTRY_TIME}.compareTo(BigDecimal.ZERO) == 0 ? "" : $P{LogbookScriptlet_SCRIPTLET}.formatTimeAsHHMMSS($F{XCOUNTRY_TIME})`
  - *If cross country time is zero, an empty string "" is displayed. If not, the value of cross country time formatted using function from the scriptlet is displayed.*
- **Total time:**
  - Expression: `$F{TOTAL_TIME}.compareTo(BigDecimal.ZERO) == 0 ? "" : $P{LogbookScriptlet_SCRIPTLET}.formatTimeAsHHMMSS($F{TOTAL_TIME})`
  - *If total time is zero, an empty string "" is displayed. If not, the value of total time formatted using function from the scriptlet is displayed.*
- **Distance flown:**
  - Expression: `$F{DISTANCE_FLOWN}.compareTo(BigDecimal.ZERO) == 0 ? "" : $P{LogbookScriptlet_SCRIPTLET}.formatDistanceAsNm($F{DISTANCE_FLOWN}, (byte)2, Locale.ENGLISH)`
  - *If distance flown is zero, an empty string "" is displayed. If not, the value of distance flown formatted using function from the scriptlet is displayed.*
- **Flight comment:**
  - Expression: `$F{COMMENT}`

Not so hard, is it?

6.5.5.6. *Creating the airport list*

The airport list for respective flight we create using the **List** element. But the List element is an element which requires it's own data source. But don't worry. As soon as you select the List element in the **Palette** and place it in the Details band, a Dataset wizard will open. This wizard will guide you through creation of the dataset (data source). Select **Create List using a new dataset** and click Next.

<content follows on next page>



List

Dataset

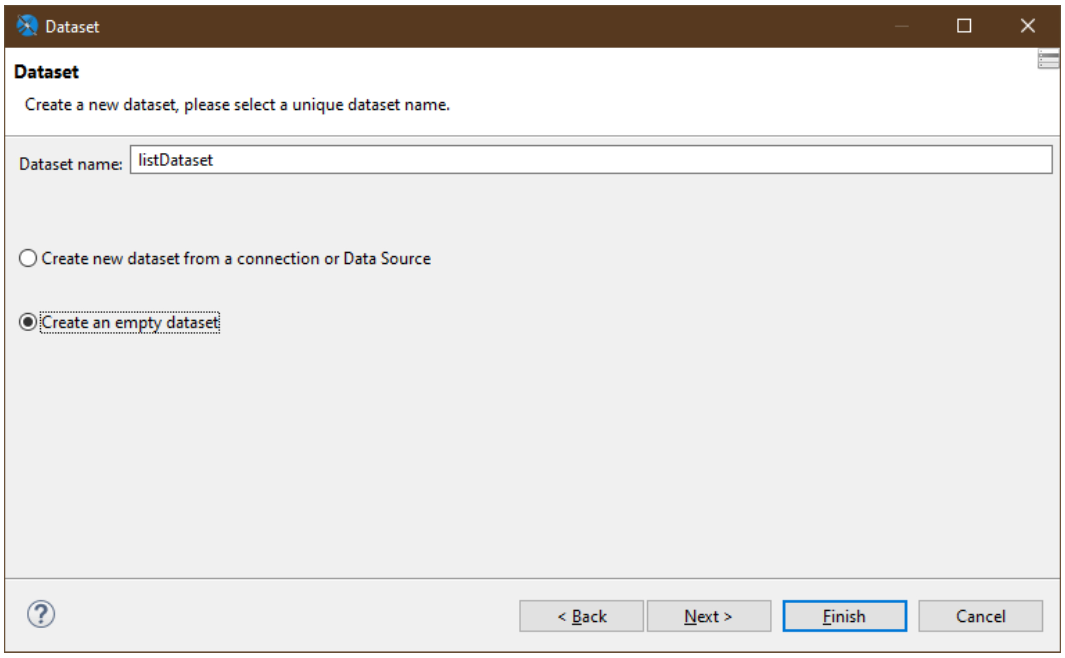
Please select or build a dataset that will be used for this component.

(●) Create a List using a new dataset

( ) Create a List using an existing dataset

Picture 102: Creating Dataset for the Airport list, step 1

On the next screen, give the dataset some name, like “**listDataset**”. Check **Create an empty dataset** and click the Finish button.



Dataset

Dataset

Create a new dataset, please select a unique dataset name.

Dataset name: listDataset

( ) Create new dataset from a connection or Data Source

(●) Create an empty dataset

Picture 103: Creating Dataset for the Airport list, step 2

Now you can find the **listDataset** in the **Outline** window on the left (where we defined Fields, Variables etc.). If you click on the arrow before the **listDataset**, it will expand it’s child items. You can see, that the **listDataset** has it’s own Parameters, Fields, Variables, Scriptlet and some other stuff. We will need two fields for the airport list – the airport code and an amount of landings. Let’s do it now. The process is the same as when defining the fields for the main data source, except that this time, they need to be under the **Fields** under the **listDataset**. Define these two fields:

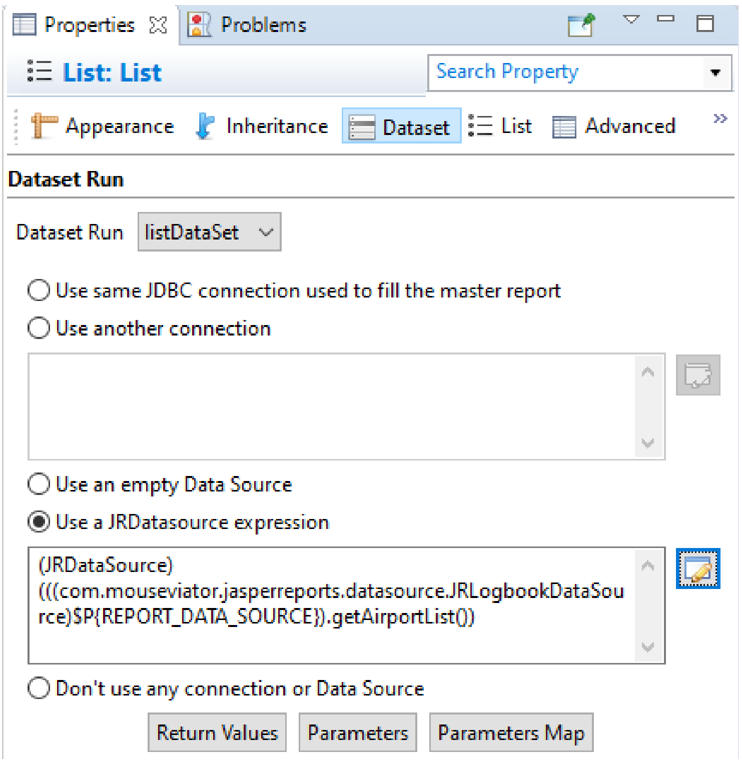
- **AIRPORT_CODE ,** Data type is: **java.lang.String**

- **NUM_LANDINGS ,** Data type is: **java.lang.Short**

Next, select the created List element in the Details band and look at it’s **Properties**. Click on the **Dataset** section. There, select the **Use JRDatasource expression** and type this code into the expression window below:

```
(JRDataSource)(((com.mouseviator.jasperreports.datasource.JRLogbookDataSource)
$P{REPORT_DATA_SOURCE}).getAirportList())
```

If you don’t understand the code above, don’t be sad about it. What it does is that it tells the JasperReports engine that it will get the data source for the airport list by calling the **getAirportList()** method of the main data source. Since the main data source is the flight logbook provided by FS Logbook Editor as data source, it will work. The stuff in the parenthesis is just retyping from the specific airport list data source class to the general JRDataSource one. To make sure you have it set right, compare with the following picture:



Dataset Run

Dataset Run: listDataSet

- ◯ Use same JDBC connection used to fill the master report
- ◯ Use another connection
- ◯ Use an empty Data Source
- ● Use a JRDatasource expression

```
(JRDataSource)
(((com.mouseviator.jasperreports.datasource.JRLogbookDataSou
rce)$P{REPORT_DATA_SOURCE}).getAirportList())
```

- ◯ Don't use any connection or Data Source

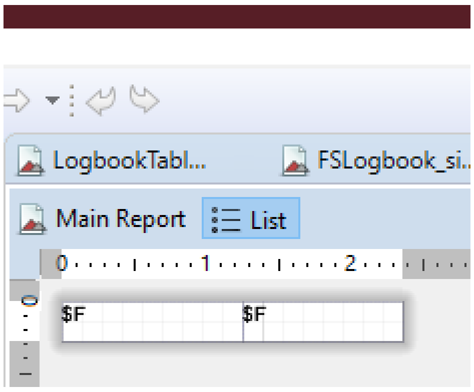
Picture 104: The Dataset settings for the airport list

To make this puzzle really do something, we need to place the two Text fields to display those airport codes with amount of landings performed on them. The workflow is the same as when creating all other text fields and content. Just, before you start clicking the Text fields, double click the List element you created for the airport list. It will open the design of the airport list itself. Don’t get scared that the whole template suddenly disappeared. It did not. You can switch back to it clicking at the Main Report tab on top of the design window. What the editor did, is that it just opened a detail view of the list so you can more easily place elements into it. So, let's do it. Place two text fields into the list right next to each other. One will display airport code and the other the amount of landings. Set them as follows:

- **Airport code:**
  - Expression: `$F{AIRPORT_CODE}`
- **Landings:**

- Expression: $F{NUM_LANDINGS}

Just to note, these text fields uses the Fields defined for the **listDataset** defined earlier. The Expression editor will offer them if you use it.



Picture 105: The list to display airport list

6.5.5.7. *Inserting page number, the Page Footer band*

Well, this one is petty easy. Just create two Text fields somewhere on the right side of the **Page Footer** band (is being displayed at the footer of each page) right to each other. Set the **expression** of the first text field to: "Page " + $V{PAGE_NUMBER}. It will display current page number. Set the **expression** of the second text field (which should be to the right of the first one) to: " of " + $V{PAGE_NUMBER}. This second one will display the total amount of pages. You can also achieve the same by placing the **Page X of Y** element from the **Composite Elements** section of the **Palette**.

6.5.5.8. *Creating the summary, the Summary band*

I believe you should be an expert in placing elements in the template by now. The summary band will be displayed after all records (flights), are processed. So, at the end of the template. And it is great place to put some summaries. Create the Static text elements and Text fields the same way as before. Below is the listing as how to set the respective expression for the respective text fields. Many of them uses variables we defined earlier which counts the total values as the template is filled.

- **Flights:**
  - Expression: $V{REPORT_COUNT}
- **Distance flown:**
  - Expression: $V{TOTAL_DISTANCE_FLOWN}.compareTo(BigDecimal.ZERO) == 0 ? "" : $P{LogbookScriptlet_SCRIPTLET}.formatDistanceAsNm($V{TOTAL_DISTANCE_FLOWN}, (byte)2, Locale.ENGLISH)
  - *If distance flown is zero, an empty string "" is displayed. If not, the value of distance flown formatted using function from the scriptlet is displayed.*
- **Takeoffs:**
  - Expression: $V{TOTAL_TO}
- **Landings:**
  - Expression: $V{TOTAL_LDG}
- **Instrument time:**
  - Expression: $V{TOTAL_INSTRUMENT_TIME}.compareTo(BigDecimal.ZERO) == 0 ? "" : $P{LogbookScriptlet_SCRIPTLET}.formatTimeAsHHMMSS($V{TOTAL_INSTRUMENT_TIME})

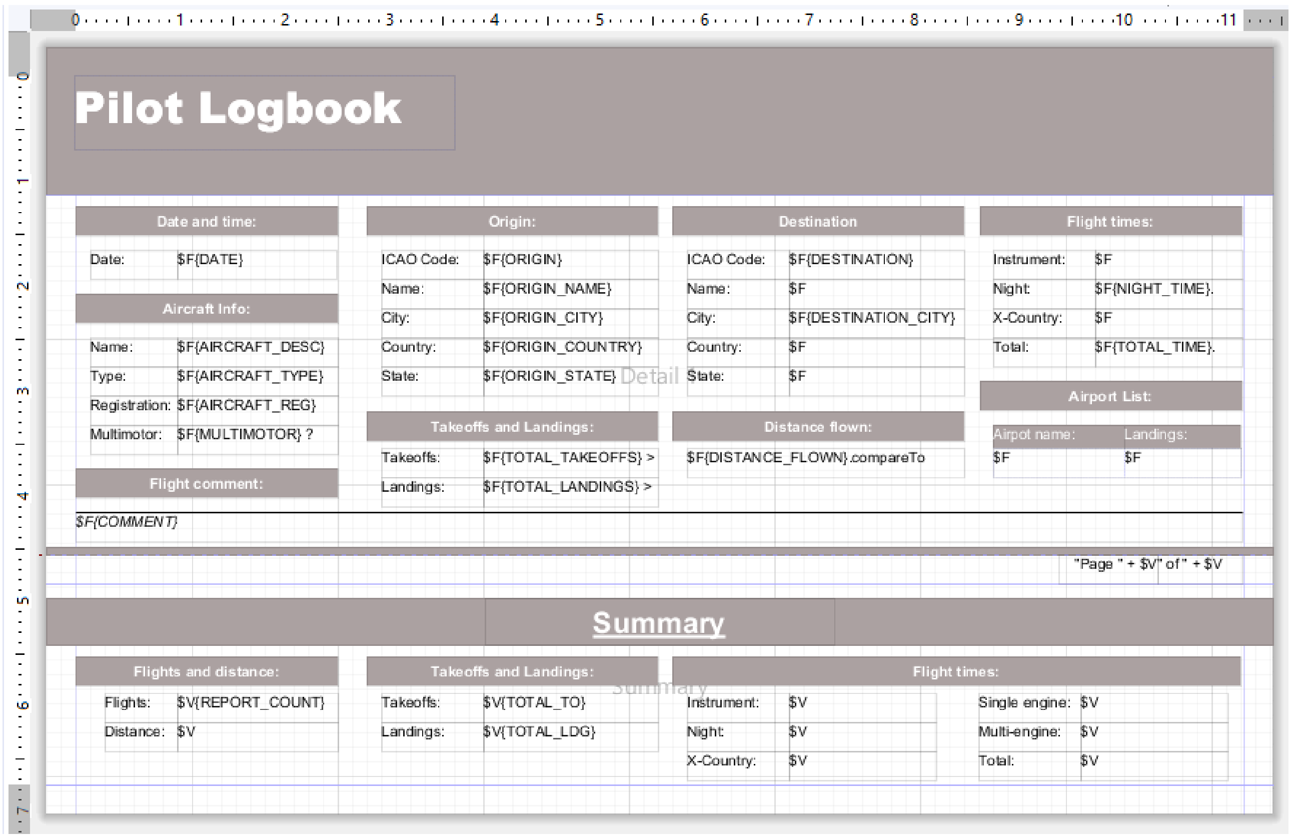
    - *If instrument time is zero, an empty string "" is displayed. If not, the value of instrument time formatted using function from the scriptlet is displayed.*
- **Night time:**
    - Expression: **$V{TOTAL_NIGHT_TIME}**.compareTo(BigDecimal.ZERO) == 0 ? "" : **$P{LogbookScriptlet_SCRIPTLET}**.formatTimeAsHHMMSS(**$V{TOTAL_NIGHT_TIME}**)
    - *If night time is zero, an empty string "" is displayed. If not, the value of night time formatted using function from the scriptlet is displayed.*
- **Cross country time:**
    - Expression: **$V{TOTAL_XCOUNTRY_TIME}**.compareTo(BigDecimal.ZERO) == 0 ? "" : **$P{LogbookScriptlet_SCRIPTLET}**.formatTimeAsHHMMSS(**$V{TOTAL_XCOUNTRY_TIME}**)
    - *If cross country time is zero, an empty string "" is displayed. If not, the value of cross country time formatted using function from the scriptlet is displayed.*
- **Single engine time:**
    - Expression: **$V{TOTAL_SE_TIME}**.compareTo(BigDecimal.ZERO) == 0 ? "" : **$P{LogbookScriptlet_SCRIPTLET}**.formatTimeAsHHMMSS(**$V{TOTAL_SE_TIME}**)
    - *If single engine time is zero, an empty string "" is displayed. If not, the value of single engine time formatted using function from the scriptlet is displayed.*
- **Multi engine time:**
    - Expression: **$V{TOTAL_ME_TIME}**.compareTo(BigDecimal.ZERO) == 0 ? "" : **$P{LogbookScriptlet_SCRIPTLET}**.formatTimeAsHHMMSS(**$V{TOTAL_ME_TIME}**)
    - *If multi engine time is zero, an empty string "" is displayed. If not, the value of multi engine time formatted using function from the scriptlet is displayed.*
- **Total time:**
    - Expression: **$V{TOTAL_TOTAL_TIME}**.compareTo(BigDecimal.ZERO) == 0 ? "" : **$P{LogbookScriptlet_SCRIPTLET}**.formatTimeAsHHMMSS(**$V{TOTAL_TOTAL_TIME}**)
    - *If total time is zero, an empty string "" is displayed. If not, the value of total time formatted using function from the scriptlet is displayed.*

Not so hard, is it?

6.5.5.9. The last steps, compiling the template

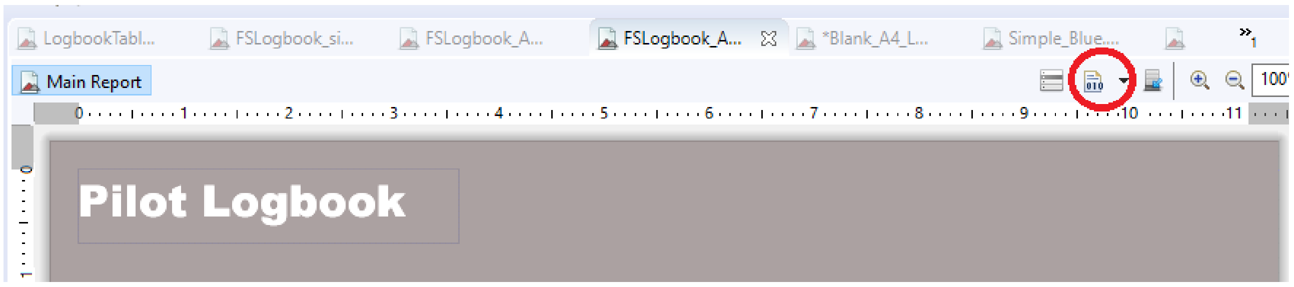
Now the template should be ready. For the last steps, you can do some adjustments like set the height of the bands we did not use (Page Header, Column Header, Column Footer) to 0 pixels, so they do not occupy any space. Chances are, your template now looks similar to the **FSLogbook_A4_Card.jrxml** included with FS Logbook Editor:

<content follows on next page>



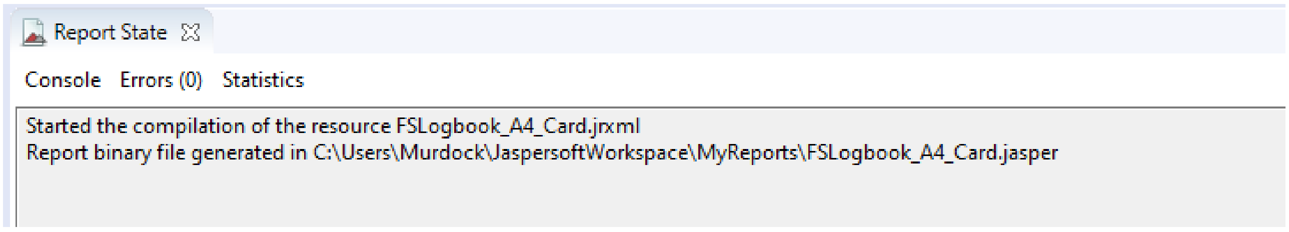
Picture 106: The final look of the template

If you did everything right, the template should contain no errors and you should be able to compile it. You can compile the template using the compile button above the template:



Picture 107: The compile button

You can also compile template using the **Compile Report** command, which you will find in the context menu after clicking at the template with right mouse button in the Project Exlorer window. And as mentioned before, if the template contains no error, it will get compiled. You will get the file with same name as the template have, but with the **.jasper** extension. You can see the result of the compile operation in the **Report State** window below the main design window:



```
Started the compilation of the resource FSLogbook_A4_Card.jrxml
Report binary file generated in C:\Users\Murdock\JaspersoftWorkspace\MyReports\FSLogbook_A4_Card.jasper
```

Picture 108: Report compilation result

And now, you can try to select this newly created template when creating logbook report with FS Logbook Editor. I guess, if you managed to read all the way down here, that you know how to do that. Wish you a good luck in designing your own templates, as it is sometimes frustrating.

Also, if you design some beautiful report for FS Logbook Editor and will think, that it should be part of FS Logbook Editor distribution, don't hesitate to contact me on *admin@mouseviator.com*. I think, I will not object about that.

7. Information supported by different logbook files

With version 1.3 FS Logbook Editor comes with support for X-Plane 10 logbook file. As a consequence, FSX binary logbook file (which FS Logbook Editor was primarily designed for), can no longer be used as „master logbook format“, because there is Cross-country flight time information in X-Plane 10 logbook file that is neither available in FSX, FS2004 or Prepar3D. For the possibility to store logbook with all information, there is a new file format called: **FS Logbook Editor XML file** with „**.fslex**“ extension. **Neither of the supported logbook files (ie. supported flight simulators) cannot save all information that can be edited by FS Logbook Editor.**

Please note: **FS Logbook Editor WILL NOT warn you while saving the logbook file that some information might not be saved (because the particular file format does NOT supporte them).**

To clear confusion a little bit, below is the list of information (logbook record attributes, such as date of flight, total flight time etc.) supported by each flight simulator logbook.

The respective logbook record attributes in the following lists are color coded, with the following meaning:

- **Green** – supported by respective file format (flight simulator)
- **Yellow** – supported only by FS Logbook Editor (respectively, flight simulator will NOT detect them)
- **Red** – NOT supported by respective file format (flight simulator)

7.1. Flight Simulator X binary logbook (.bin)

FSX binary logbook file supports most of the information that can be edited with FS Logbook Editor.

- Date of flight
- Total flight time
- Day flight time (Can be computed, no big deal)
- Night flight time
- Instrument flight time
- Cross-country fligh time
- Origin
- Destination
- Aircraft registration
- Aircraft description
- Aircraft type
- Whether the aircraft is multi-engine
- Flight comment
- Airport list (pairs or airport code and amount of landings on that airport)

7.2. Flight Simulator 2004 logbook (.log)

The Flight Simulator 2004 logbook by default supports about half of all logbook record attributes that you can edit with FS Logbook Editor, but with FS Logbook Editor you can get some more.

- Date of flight
- Total flight time
- Day flight time
- Night flight time
- Instrument flight time
- Cross-country fligh time
- Origin
- Destination
- Aircraft registration
- Aircraft description
- Aircraft type
- Whether the aircraft is multi-engine
- Flight comment
- Airport list (pairs of airport code and amount of landings on that airport)

Please see ***Add these data into remarks when saving FS2004 logbook*** to learn more about how to save and load the yellow fields.

7.3. X-Plane 10/11 logbook (.txt)

Logbook records at X-Plane logbook file contains Cross-country flight time field, that is supported neither by FSX, FS2004 nor Prepar3D logbook.

- Date of flight (Does NOT support time, only date)
- Total flight time
- Day flight time (Can be computed, no big deal)
- Night flight time
- Instrument flight time
- Cross-country flight time
- Origin
- Destination
- Aircraft registration
- Aircraft description
- Aircraft type
- Whether the aircraft is multi-engine
- Flight comment
- Airport list (pairs of airport code and amount of landings on that airport)

Logbook records at X-Plane 10 logbook also contains a field that stores amount of landings performed during particular flight. This field is not listed in the list above. The amount of landings is converted to airport list with records that contain the respective amount of landings at destination or origin airport (based on if they are present). If there is non-zero amount of landings in the record, but no origin nor destination airport, you might encounter UNMD airport code in the airport list. This is FS Logbook Editor constants signalizing that there was no airport to use, but to preserve the amount of landings, this constant is used. So for X-Plane 10 logbook, you can have airport list in FS Logbook editor, but if you use airport codes other than origin or destination there, they will be lost.

7.4. FS Logbook Editor XML file (.fslex)

Since this format was introduced to allow you to save the logbook with all information, it supports all of logbook record attributes (information).

8. Command line parameters

Since FS Logbook Editor version 1.86, it is finally possible to pass logbook to be opened from command line. It can be passed in three ways:

As an unnamed parameter (the path to the logbook must be the first parameter):

like this: *FSLogbookEditor.exe C:\Users\Bruce Almighty\Documents\logbook.fslex*

As named parameter using shortcut version of the parameter:

like this: *FSLogbookEditor.exe -lf C:\Users\Bruce Almighty\Documents\logbook.fslex*

As named parameter using full name of the parameter:

like this: *FSLogbookEditor.exe -logbookFile C:\Users\Bruce Almighty\Documents\logbook.fslex*

Note that those are example file paths though :), use some real ones.

9. 3rd party resources used

FS Logbook Editor includes 3rd party software ("software libraries"), that are subject to their own License Agreement. All copyrights regarding those software libraries belong to respective copyright holders. All concerned software libraries are placed within "lib" folder in the main program folder along with their respective license agreements.

As FS Logbook Editor evolved, the number of external libraries used increased. Now that is over 100 files and I am a little bit of lazy to manually rewrite the list that used to be down here to match. So since FS Logbook Editor version 1.85, the list is being automatically generated and has been moved to the "**doc/dependency-licenses**" folder in the main program folder.

Here I only keep those:

Java Runtime Environment – bundled with application to save you from the need to install one.
(https://adoptopenjdk.net)
Licensed under GPLv2 with Classpath Exception and OpenJDK Assembly Exception (often referred to as GPL2+CE)
(https://adoptopenjdk.net/faq.html#licensing)

MakeRwys.exe – an utility from Peter Dowson for extracting airport data.
http://www.schiratti.com/dowson.html

LorbySceneryExport.exe - an utility from Oliver Binder (Lorby-SI). It works with MakeRwys to make complete scan of sceneries.
https://www.lorby-si.com/downloads.html

9.1. Using another version of 3rd party library

Some of the libraries used requires that You are able to use different than included version of the library (like japura which is licensed under LGPL).

This is more likely for a programmers among you. Don't expect much details here, as I expect that if you want to use different versions of libraries required by FS Logbook Editor, you should have the knowledge to do it – you will find the way. So just a short mention about the easiest way you can take.

The easiest way to use another version of the library is to delete the included library from the lib folder within the FS Logbook Editor installation folder and place the new library here. But you must name it the same way the deleted library was named (otherwise, FS Logbook Editor will most likely crash because it does not have the library it needs).

Example: Say you want to use older version of japura library – **japura-1.18.2.jar**. So go to the **lib** folder and delete the **japura-gui-2.0.0.jar**, copy **japura-1.18.2.jar** into the **lib** folder and rename it to **japura-gui-2.0.0.jar**. As far as the interface of the library does not change significantly, FS Logbook Editor should work OK with the library.

10. Thanks

- To Kenneth Goodwin for suggestions regarding pilot statistics.
- Peter Dowson for permission to distribute MakeRwys with FS Logbook Editor (helps with airport data import for FSX/P3D)
- Oliver Binder (Lorby-SI) for permission to distribute LorbySceneryExport with FS Logbook Editor (helps MakeRwys to scan sceneries)
- All the users of my software, because without you, it would be all pointless.

"Happy flying!"

END OF DOCUMENT