FstarRC (Plan conversion module for Jeppesen FliteStar/FliteMap)

Freeware by Pete Dowson, 11th February 2012

Support Forum: support-pete-dowson-modules

Note: All my Windows based software is always available in the latest versions from http://www.schiratti.com—just follow my name when you get there**.**

Version 2.30 of FstarRC

This package contains the following parts:

FstarRC.exe	FliteStar loader and dll injector, version 2.30
FstarRC.dll	The module itself, which runs inside FliteStar (version 2.30)
FstarRC.rws	Runways database for FS2004*
FstarRC.pdf	This document

NOTE: If you cannot *see* the DLL, please go to the Explorer's View menu. Select "Folder Options" and then the "View" Tab. Then choose either the "Show all files" button or the "Do not show hidden files"—anything *but* the "Do not show hidden or system files" button! (Windows now seems to regard all DLLs as system files and not the "application extensions" they usually are!).

* This file should be replaced by new ones generated from your own FS9, FSX or Prepar3D installation, by running my MakeRunways program. Optionally you can also place the generated F5.CSV file into the samefolder. This will then be used to supplement missing radio frequencies in FliteStar with ones from FS.

THIS VERSION IS COMPATIBLE WITH WINDOWS 2000, XP, Vista and Win7 (32- and 64-bit)

Introduction: What is FstarRC?

This package mainly provides Jeppesen FliteStar and FliteMap users with a direct link to JDT LLC's Radar Contact. As an aside it can also produce .sbp plans for SquawkBox and Enrico Schiratti's Project Magenta CDU, .efpu plans for Chris Brett's EFIS98, and .pln plans for FS2000's GPS, FS2002's GPS, FS2004's GPS, Wilco's 767PIC, and AETI's ProFlight2000.

The main requirement is that you have version 8 or 9 of Jeppesen FliteMap or FliteStar. All the testing has been done with versions 8.04, 8.1, 8.2 and 8.5 of FliteMap and 8.1, 8.2, 8.3, 8.31, 8.4 and 8.5, 9.04 and 9.521 of FliteStar.

Installation

Follow these steps:

1. Copy all the files into your FliteStar/FliteMap folder (the one in which Flitestar.exe resides).
2. Drag a shortcut to FstarRC.exe to your desktop, or wherever you want it.

Then, whenever you are running FliteMap or FliteStar to produce a plan for flying with FS, double click the FstarRC icon *instead of* the original Jeppesen icon. FstarRC will load FliteStar.exe for you, and when the main Window is ready it will insert the FstarRC.dll into the process (it allows a minute or two for you to respond to the usual preliminaries from FliteStar).

From then on, in that FliteStar/Map session, whenever you print a plan you will also get additional output—the Radar Contact .apl file and optional additional files for use with SquawkBox, Project Magenta and EFIS98.

Note that on Windows 98 or Me (but NOT 2000 or later) you can operate this a different way. You can load and run FliteStar or FliteMap first, then, before printing the Navigation Log for a plan, run FstarRC. The latter will find the running FliteStar program and not try to load it again.

Finally, in case the title bar for FliteStar or FliteMap, does not actually start with "FliteStar …" or "FliteMap …" you can specify the word it should look for at the start of that title bar, on the command line in the shortcut for FstarRC. For example, add a space then FliteMap after "FstarRC.exe".

Using FstarRC

The prime output is an .apl file which can be compiled in Radar Contact. In Radar Contact versions before 3 this is by selecting the "FliteStar" tab on the RC front end. In versions 3 and 4 just select the .apl input option and find the file. You need to get RC to look in the directory into which you direct the .apl files produced by FstarRC—I use a folder called "FliteStarPlans inside the RC folder. For Radar Contact versions before 3 you also need any old EXE file in there to allow you to select it in RC as the "source".

FstarRC processes the FliteStar plan only when you **print** the Navigation Log. For it to work properly you *must* select FliteStar's "Expanded Log" format (select the 'Report' tab and 'Nav Log', right click on the Log itself and make sure "Expanded Log" is checked). You don't have to have all the "Optional Blocks" selected, but I recommend at least the 'Plain Language Route' block. The size of the Log on screen isn't relevant.

When you first run FstarRC it will generate a file called FstarRC.ini, in the same folder. You will want to edit this, at least to set the Paths to which FstarRC will send the resulting plans. For more details about this, see the section about setting options, next.

Setting FstarRC options

Options are set in the FstarRC.ini file Here's my own .ini file, as an example:

```
[Paths]
RC="\\centre\d\fs2k\modules\fsnavigator\flpexport"
EFIS98="\\right\e\EFIS98\FlightPlans"
SquawkBox="\\left\f\Squawkbox"
SchirattiCDU="\\right\e\EnricosCDU\Routes"
FS2000="\\centre\d\fs2k\pilots"
FS2002="\\centre\g\fs2002\flights\myflts"
FS2004="\\centre\My Documents\Flight Simulator Files"

[Settings]
IncludeAlternate=Yes
DuplicateFreqsOkay=Yes
RoundCheckpointAltsUp=Yes
HardSurfacesOnly=Yes
MinRunwayLength=3000
MinDepartureDistance=12
MaxCheckpoints=200
MaxLastCheckpointDistance=2
InsertCrossingPoint=Yes
MaxDescription=160
IncludeAlternate=Yes
SetExtrasForRCV4=No
Log=No
```

The section you'll mainly want to change is the one called [Paths]. As you see above, I use RC, EFIS98, SquawkBox, Enrico's CDU and FS's GPS, so I've filled in all of the paths. If none of them are filled in you will only get the RC .apl file, and that will be inserted into the FliteStar folder. The RC path will then be filled in accordingly, as you will see if you let FstarRC generate the .ini file to start with.

Also you'll notice that all my paths are on other PCs on my Network. In fact I run FliteMap on my Notebook, which only runs FliteMap (in moving map mode) when I'm flying. (If you are using FliteMap rather than FliteStar and you would like to use it in moving map mode with FS, let me know. I have a module called GPSout.DLL to do just that).

The filenames used in each instance are as follows. In each case FFFF is the departure airport's ICAO code and TTTT is the destination:

RC: FFFFTTTT.apl, or FFFFTTTTn.apl where 'n' is the next free decimal number, used when the original file exists and you elect NOT to overwrite it in the prompt which then appears.

EFIS98: FFFFTTTT.efpu, or FFFFTTTTn.efpu—depending solely on the RC name, *not* on whether there's a previous .efpu file with the same name. ***No such check is made!***

SquawkBox/CDU: Both these are identical formats, and the filename is the same for both: FFFF-TTTT.sbp or FFFF-TTTTn.sbp—depending solely on the RC name, *not* on whether there's a previous .sbp file with the same name. ***No such check is made!***

FS2000–FSX: FFFFTTTT.pln or FFFFTTTTn.pln—depending solely on the RC name, *not* on whether there's a previous .pln file with the same name. ***No such check is made***

If there are any errors in producing the plan, this will be indicated in the Message Box telling you that the RC file has been produced. The details of the errors are provided in text format as comments at the start of the RC .apl file. Note that there are no checks made on the validity of the paths. If they are wrong you will not get the files.

Here's an explanation of each of the [Settings] parameters:

IncludeAlternate=No

If you are using Radar Contact Version 3 or later, then you can set this parameter to 'Yes', and FstarRC will include the Jeppesen planned alternate destination details in the resulting .apl file for Radar Contact to use. It will also include "Checkpoint_N_id" entries for RCV4 to show the waypoint names.

DuplicateFreqsOkay=No

RCV2 and RCV3 do not like any COM frequencies at the arrival airport to be the same as any of those at the departure airport. You can set this to 'Yes' for RCV4.

RoundCheckpointAltsUp=Yes

FliteStar altitudes at each checkpoint are often very specific calculated values, and cannot be set on many AutoPilot, MCP or CDU systems. This option ensures that those altitudes are rounded up to the next multiple of 100 feet in the output plans. The printed plan is not affected.

HardSurfacesOnly=Yes

The runways database provided includes *all* runways available in the default FS2000 Pro installation, excepting only helipads. If you are okay landing on grass and other such surfaces set this option to 'No'. if you leave it at 'yes' only concrete and asphalt surfaces will be considered.

MinRunwayLength=5000

This ignores runways shorter than the specified value, given in feet. If you really want to takeoff from or land at Meigs you'll need to change this from its default of 5000 feet to 3500 or so, as I have.

MinDepartureDistance=4

RC cannot deal with checkpoints too close to takeoff, whilst it is still operating your ATC from Tower. To allow it time to switch to Departure, checkpoints which are near have to be discarded. This parameter allows you to set the distance *in nautical miles along the route* within which checkpoints should be ignored in producing the RC file. Note that no checkpoints are omitted from the EFIS98, SquawkBox or CDU files, only from the RC input. This allows you to make use of all of the checkpoints in a detailed SID, via your FMS, without upsetting the RC ATC adventure.

If there are any checkpoints omitted you will be warned of 'changes' in the completion Message Box, and there will be comments for each missing one in the .apl file.

MaxCheckpoints=29

RC version 2 and earlier only allowed 29 checkpoints at maximum. This parameter is here to allow this to be changed. RC versions 3 and later have no known restriction here, and FstarRC version 1.85 has been tested with over 100 checkpoints with no problems.

Note that this does *not* include any checkpoints omitted due to either of the last two parameters, so there may be more than the stated number of checkpoints in the other files.

FstarRC still produces all the output files even if there are too many checkpoints, but you will be told it is a fatal error so you can do something before compiling it.

MaxLastCheckpointDistance=10

With RC it is important to have a final checkpoint quite near, if not actually at, the destination airport. This is because RC imposes a crossing restriction (250 knots at 10000 feet, normally) at a point 40nm directly away from the final checkpoint. If that checkpoint is a long way from the airport, you will find you have to descend too early, wasting fuel.

This parameter gives a maximum distance, in nautical miles, for the final checkpoint in your RC plan. If the last checkpoint derived from the FliteStar plan is further away than this, FstarRC will automatically insert a final checkpoint *at* the airport.

To stop FstarRC inserting this waypoint at all, set this parameter to 0. Note that the files provided for SquawkBox, CDU and EFIS98 do not contain this inserted checkpoint in any case.

InsertCrossingPoint=Yes

The RC crossing restriction, where you should be at 10000 feet (normally) and 250 knots, is at 40nm from the last checkpoint. This is 40nm direct, which is not necessarily the same as route distance. Therefore it isn't always easy to work out where the crossing restriction is going to occur, especially if the final checkpoint is not a VOR/DME.

One way around this is to place your own user checkpoint at the correct position. FstarRC will do this for you, by default, if possible. If such a point falls very close to an existing checkpoint, another is *not* inserted—a comment is placed against the checkpoint entry in the .apl file to show which checkpoint is the relevant one.

FstarRC names the inserted checkpoint "Xrstr" (for "Crossing restriction") by default. Lower case is used deliberately to distinguish it from real intersections. If you wish to use a different name, provide a 5-character value in place of the "Yes" in this parameter. Note that this inserted checkpoint appears in *all* output plans, excepting of course FliteStar's print-out.

MaxDescription=160

This sets the maximum description length for the resulting RC plan. There's an overall limit of 255 characters, but Radar Contact uses some of these. It used to only use about 32 characters, but it is getting larger, with Web Site references as well, so the amount available for FliteStar information is more limited. The default, 160 characters, allows for the current maximum usage by RC version 2.1.

FstarRC tries to put the maximum of useful information in here, but it has now had to remove the Date, Aircraft type and Tail number. If space is limited the other details may go as well. If the FAA/JAA plan is truncated an asterisk (*) will be appended to indicate this.

Navigation Log changes, and Airport COM frequencies

FstarRC makes some subtle changes to the printed Log:

1. The title is changed to "RC Navigation Log"
2. The name "FliteStar" or "FliteMap" has the FstarRC name and version appended
3. All airport COM frequency details are replaced by only those chosen for output to RC, and a list of Runways and ILS frequencies is also appended.

This last point needs some further explanation:

FliteStar provides a large variety of COM frequencies for each airport. I don't pretend to understand what they are used for, but I've tried to map them as well as possible onto those frequencies that can be used in RC. These are the equivalents I am currently assuming (corrections or suggestions welcome!):

RC need:	Met from (in priority order):
ATIS	ATIS, INFO, MULTICOM
Delivery (CLNC) `	CLNC-DEL, PRE-TAXI-CLNC, CTRL, DEL
Ground	GROUND, GND-CTRL, RAMP/TAXI-CTRL, GATE-CTRL, CTRL
Tower	TOWER
Departure	DEP-CTRL, CTRL, TCA, TMA, DEP, TOWER, GROUND
Approach	APPR, APP-CTRL, ARR-CTRL, DIR-APP, TCA, TMA, RADAR, RADIO, UNICOM, ARR
FSS	FSS, CLNC-DEL, REMOTE-FSS

FstarRC only selects frequencies within the FlightSim tunable COM range, and simply omits those it cannot assign using the above rules. [**NOTE** that RC version 1 ignored frequency assignments from the input .apl file in any case, as does the Beta version 2 up till the RCV2.EXE dated 21st April 2000. Then it certainly accepts them okay when the FS Navigator tab is used, but it always overrides the Departure Ground frequency no matter what the input screen says. RCV2.1 has a FliteStar tab and fixes this last problem]

Runway Amendments

The runway database supplied with FStarRC is generated from default FS2002 scenery files (actually the AFD files– Airport and Facilities Data). This should be fine for most purposes, as then the runway lists provided to Radar Contact will abide by FS2002's knowledge rather than Jeppesen's data in FliteStar.

Unfortunately there are cases where this won't be correct. One is where your scenery in FS has been enhanced or modified by additional scenery packages, for instance adding a newly opened runway and maybe closing another. The other is where you are using Jeppesen data which is not of the same vintage as FS's data and something more drastic has happened, such as the assigned ICAO codes have changed.

An older example of the latter is the change at Austin, Texas, where the FS2000 "KAUS" airport is still Mueller Municipal in the city, with the International airport outside the city being KBSM (Bergstrom Intl). Recently the city airport was either closed or downgraded and KBSM became KAUS. This made FstarRC choose the old runways (in the city) rather than those for KBSM, since it is, naturally enough, going by the ICAO identity. (Hopefully this is okay in FS2002).

FStarRC has provisions for changing its runway database. These work without having to regenerate the whole binary, sorted and optimised RWS file. You provide details of the changes in a separate, simple text file, named FStarRCR.txt, containing one line for each runway. For any airport with a runway change *all* runways for the airport must be given, even if they are unchanged, as the presence of the ICAO code in a line in this text file will stop FstarRC looking for that airport in the RWS file.

The format of the entries is very strict, so I'll use an example (KAUS) to explain what is needed:

KAUS,17R,N30 12.8,W97 40.8,541,10951,2,172,110.30
KAUS,35L,N30 10.8,W97 40.7,487,11156,2,352,110.30

The generic format is:

ICAO, Runway, Latitude, Longitude, Elevation(ft), Length(ft), Surface, Heading, ILS frequency

Each line starts with the 4-character ICAO code, in capitals. Each item of data is then separated from the previous one with a comma (,), with spaces being ignored between parts.

The runway is identified as usual, by its number, optionally followed by L, C, R, or H (for Helipad).

Latitudes and Longitudes are given starting with N, S, E, or W as appropriate, then either of the forms:

n.nnnn … (i.e. degrees with decimal degrees

n n.nnnn … (i.e. degrees space minutes with decimal minutes

In general the latter is more convenient as it matches Jeppesen's own usual method.

The runway threshold elevation is in feet.

The runway length is its usable length and is in feet.

The surface type code is a decimal number chosen from the following (based on FS2000's own coding):

0	unknown
1	dirt
2	concrete
3	asphalt
4	grass
5	gravel
6	oil-treated
7	mats
8	snow
9	coral
10	water

Note that only runways with surface types 2 and 3 are listed in the results if the parameter **HardSurfacesOnly=Yes** is used or defaulted.

The heading is given in degrees magnetic.

If there's no associated ILS, omit the final parameter (and the comma). Otherwise add the frequency, always in the 6 character format nnn.n.

Published by Peter L. Dowson, 11th February 2012