

AI FLIGHT PLANNER

... a comprehensive, fully integrated AI flight planning tool for both FS9 and FSX

INDEX

- 1.** GENERAL
 - 1.1** Airports
 - 1.2** Aircraft
 - 1.3** Flight Plans
 - 1.4** Traffic File Issues
 - 1.5** Conversion - FS9->FSX
- 2.** INSTALLING/UNINSTALLING AI FLIGHT PLANNER
- 3.** THE AI FLIGHT PLANNER MAIN WINDOW
- 4.** MAIN MENU FUNCTIONS
 - 4.1** Creating a New Flight Plan Files
 - 4.2** Loading Flight Plans and Associated Files
 - 4.3** Flight Plan Validation
 - 4.4** Flight Plan Subsets
 - 4.5** Saving/Compiling Flight Plans and Associated Files
 - 4.6** Conversion of FS9 Flight Plans for use with FSX
 - 4.7** Managing the Aircraft List
 - 4.8** Managing Airports
- 5.** BUTTONS AND INDICATORS
 - 5.1** Flight Plan List
 - 5.2** Moving Items to and From the Flight Plan List
 - 5.3** Leg List
 - 5.4** Airport List
 - 5.5** Aircraft List
- 6.** USING THE FLIGHT PLAN EDITOR
 - 6.1** Arrival and Departure Times
 - 6.2** Departure Airport
 - 6.3** Loading Flight Plans
 - 6.4** Built-In text Editor
 - 6.5** Flight Plan Editor
 - 6.6** New Flight Plans from "Scratch"
- 7.** MANAGING AIRCRAFT DATA
 - 7.1** Loading and Saving Aircraft Data
 - 7.2** Adding an Aircraft to the Aircraft List
 - 7.3** Editing an Aircraft Already in the Aircraft List
 - 7.4** Deleting Aircraft
- 8.** COLLECTING AND EDITING AIRPORT DATA
 - 8.1** Collecting Airport Data
 - 8.2** Customizing the Airport List
 - 8.3** Editing and Manually Adding New Airports
 - 8.4** Airport Data Bulk Update

<u>9.</u>	SUPPORT
<u>10.</u>	ACKNOWLEDGEMENTS
<u>11.</u>	END USER LICENSE AGREEMENT

1. GENERAL

AI Flight Planner provides all the facilities necessary to create, develop, edit and maintain AI traffic flight plans for both the FS9 and FSX versions of Microsoft Flight Simulator (MSFS). It includes an embedded compiler/decompiler. Hence, AI Flight Planner can use either TTools-format *.txt* files or compiled MSFS traffic *.bgl* files as “source code” for flight plans and associated aircraft data – or you may create one or both “from scratch”. AI Flight Planner provides its own airport data.

In addition to the features expected in any flight planning tool, AI Flight Planner:

- appends data from additional flight plan and aircraft files to that already loaded; data from TTools-format files and traffic files may be mixed;
- has a leg-oriented flight plan editor that can handle daily and weekly traffic in an identical manner;
- permits entry of arrival and departure times based on any time zone or using local times for both;
- displays at any time the arrivals and departures at any airport;
- generates traffic for a user-specified subset of the airports included in a flight plan;
- removes specific airports from default traffic files; and
- includes in the compiled traffic file only the airport and aircraft information used by the included flight plans, reducing system overhead.

Thus, it is possible with AI Flight Planner to generate efficient AI traffic files pertaining to only a region or single airport from multiple, airline-system-wide flight plan files.

AI Flight Planner manages the three main components of MSFS AI traffic as follows.

1.1 Airports – All the airport information required for MSFS traffic files is derived from an internal airport list named *AirportList.dat*. *AirportList.dat* includes every stock airport in FS9 and FSX and recognises when an airport exists in only one or the other. To address new or modified airports included in scenery add-ons, AI Flight Planner includes:

- an editor to add/update individual airports,
- a bulk update facility which adds/updates airports based on a TTools-format airport file, and
- an airport collection facility that regenerates *AirportList.dat* with or without add-on airports.

For those with state-of-the-art computer systems, the size of *AirportList.dat* should not present a problem. However, users with older computers or with limited RAM may experience unacceptable delays when accessing airport data.

So, AI Flight Planner allows you to specify a subset of the available airports for inclusion in a custom airport list and use it instead.

As well, using a “tree-view” style window, you can easily find the ICAO code and other details of any airport “known” to the system.

- 1.2 Aircraft – AI Flight Planner accepts aircraft information from TTools-format aircraft *.txt* files and compiled traffic *.bgl* files. Data from both types of files may be used simultaneously. That information can be supplemented by “pointing” to a MSFS aircraft folder and then selecting from the available AI aircraft variants stored in that folder. AI Flight Planner will extract from the associated *aircraft.cfg* file all the information required to use the selected aircraft. Indeed, all required aircraft information may be obtained by the latter method and saved to a TTools-format file for later use.

Aircraft data may be edited using either AI Flight Planner’s custom editor or in TTools format using the default text editor on the host system.

- 1.3 Flight Plans – As with aircraft data, AI Flight Planner will use data from multiple flight plan files. As well:
- the airports involved in the loaded flight plans are shown in a separate list;
 - flight plans may be edited either singly, with:
 - the leg-oriented editor, or
 - the built-in plain-text editor, orsingly or collectively using the Find/Replace feature;
 - the length of the flight plan leg currently in the editor is shown, both in distance and predicted duration, along with the predicted arrival time;
 - “reasonableness” error checking is performed at several stages
 - when a flight plan file is saved, “companion” aircraft data and airport files are also saved, maintaining full compatibility with other TTools-format-based tools;
 - flight plans for only a selected subset of the included airports may be saved in a separate file, in which case the companion aircraft file will include only the aircraft used by those flight plans.

- 1.4 Traffic File Issues – There are three different AI flight plan compilers in widespread use:
- TTools, a compiler/decompiler for FS9 and earlier versions of MSFS,
 - AIFPC, a TTools-compatible compiler/decompiler for FS9 and FSX (AI Flight Planner includes the AIFPC compiler/decompiler), and
 - Traffic DataBase Builder (TDBB), Microsoft’s AI traffic generation utility for FS9 and FSX.

The source data formats used by TDBB are very different from those used by TTools and AIFPC.

There is a common misconception that flight plans for FS9 must be in TTools format and those for FSX must be developed using TDBB. And, often, you will hear “FS9 flight plans can’t be used with FSX”. But, the reality is, TTools flight

plans are compatible with FSX. And TDBB, to the surprise of many, was available in FS9 as well as FSX.

But, there are several significant issues to be aware of when using TTools-format flight plan files for FSX:

- FSX accepts traffic files compiled for FS9. However, if you attempt to mix TTools-format traffic files (compiled for FS9) with traffic files compiled for FSX, the compiled-for-FSX traffic may be suppressed. This has nothing to do with the day-of-week encoding scheme discussed below. It pertains solely to the respective formats of the traffic files. So long as the original FS9 flight plan file is (re)compiled for FSX (using AIFPC, since TTools is not FSX-compatible and the TDBB source format is different) there's no such problem. (AI Flight Planner requires you specify the target system (FS9/FSX) at compile time and, if you wish, has the compiler add a "_FS9" or "_FSX" suffix to the traffic file name to help avoid confusion.)
- In FS9, the day-of-week encoding scheme for weekly flight plans uses 0 for Sunday, 1 for Monday and so on to 6 for Saturday. In FSX, Microsoft changed the day-encoding to 0 for Monday and 6 for Sunday. Thus, if a flight plan file developed initially for FS9 is compiled and used in FSX without taking this difference into account, the AI appear a day late (AI Flight Planner handles the day-of-week encoding automatically and hides most of the related detail.)
- TTools flight plan source format permits user-specification of arrival time for a flight plan leg by prefixing the time with the "@" symbol. This is a feature for which there is no parallel in TDBB. With TDBB, arrival time is always based on the user-specified departure time, the cruising speed of the aircraft used to fly the flight plan and the distance between the departure and destination airports. But, because the "@" symbol is not "understood" by the simulator's "AI engine", its use in TTools involves some "trickery". And, it almost works! Unfortunately, if an arrival time specified with "@" is more than about 37 minutes later than the time calculated by the "AI engine", the AI for that leg of the flight time never materializes for landing. Instead, it spawns in a parking spot at the destination airport some time later in preparation for the next leg. This is most likely to occur when simulating scheduled airline long-haul operations where the scheduled arrival time is often substantially later than the simply-calculated (distance/speed) arrival time.

To address the latter issue, AI Flight Planner has two compile modes:

- normal, where measures are taken to avoid this "37-minute issue"; and
- "raw", where AI Flight Planner passes the user input directly to the compiler.

The two modes are necessary because there is one situation that cannot be handled in the normal mode. That is, an AI departing from some distance away (> 75-100 miles) that is specified to do touch 'n gos upon arrival prior to landing. When the normal compile mode is used for such a flight plan, the AI will arrive late, how late depending on the distance between the originating and destination airports. Should the flight plan file being compiler be susceptible to either of these situation, the user will be warned. Of course, if the potential for both situations exist, the user must either change the offending flight plan(s) or choose between the two situations.

Users who decompile AI Flight Planner generated traffic files compiled in the normal mode will find an unusually-named airport as the first entry in the airports.txt file. This is a flag used by AI Flight Planner to indicate that it has taken special measures to address the “37-minute problem.”

1.5 Conversion - FS9->FSX - AI Flight Planner will convert your FS9 flight plan and traffic files for use with FSX. This conversion addresses two main attributes:

- Airports – The ICAO code of many FS9 airports changed between FS9 and FSX. AI Flight Planner will update those airports for both flight plan and traffic file conversion. As well, there is no equivalent in FSX for some FS9 airports.
- Day-Encoding – please refer to the discussion in the previous section.

Flight plan files may be converted for use with FSX in either of two ways. If a FS9 flight plan file is to continue to be used for FS9 and well as FSX, it's probably easiest to leave the source (flightplans...txt) file in FS9 format and have AI Flight Planner to do the conversion each the file is compiled for use with FSX. Simply select the target MSFS version in the compile dialog. However, if the flight plan file is to be used only with FSX, you may wish (but do not need) to convert the source file. This is accomplished with the *Convert FS9 FP to FSX* menu item.

For FS9 traffic file conversion, AI Flight Planner provides a bulk-conversion function, allowing you to select any combination of folders and files for conversion using a “treeview” structure. For simplicity, only compiled-for-FS9 traffic files are shown in the structure. Folders known not to contain FS9 traffic files, while still shown, are dimmed. The selected folders may contain a mix of FSX and FS9 traffic files; only the FS9 files will be affected. Following conversion, any missing airports or other problems encountered will be noted in a conversion report.

Two “convert” buttons are provided in the conversion dialog; one to convert airports only, the other to convert both airports and day-encoding. If some traffic files have already had their day-encoding adjusted and others not, they should be separated and each set converted separately.

But, in many cases, you will not know the extent to which a traffic file has already been converted for use with FSX, and you may be concerned about “re-conversion”. You need not be. AIFP will scan each file looking for FS9-only airports. If the airports have already been updated for FSX, AIFP will find none that need updating. Day-encoding is more problematic. If the traffic file was originally compiled with AIFPC and AIFPC did the day-encoding conversion, this will have been noted in the traffic file. Unfortunately, if the day-encoding conversion was done manually or not at all, AIFP has no automatic way of knowing how to handle day-encoding during conversion. However, should a file for which the day-encoding has already been converted for use with FSX be re-converted, it's “not the end of the world”. The only effect is that the AI will operate on incorrect days. Should that occur, simply decompile the errant traffic file (AI Flight Planner's *Open Traffic .bgl* function), step the days forward or back

as necessary using the day-encoding radio buttons and the Change Day Code button and re-compile the file.

2. INSTALLING / UNINSTALLING AI FLIGHT PLANNER

AI Flight Planner is installed simply by copying the files from the downloaded archive to the desired destination folder. The complete set of files includes:

- *AI Flight Planner.exe*
- *AIFP.ini* (created on first execution of AIFP.exe)
- *AirportList.dat*, *AirportList_Base_FS9.dat* and *AirportList_Base_FSX.dat*
- *bglAnalyze9.exe* and *bglAnalyzeX.exe*
- *FS9Chg.txt*
- *Timezones.txt* and *Timezone.dat*
- *AI Flight Planner.htm* (this user manual)
- a sub-folder named *Work* for AI Flight Planner's internal use

and several information files.

However, if you are upgrading AI Flight Planner from an earlier release and have added or changed any airport or updated the time zone information, you should not load *AirportList.dat* or *Timezone.dat* included in the archive since they would overwrite your changes.

AI Flight Planner does not affect the system registry.

AI Flight Planner is a VB NET.Framework 2.0 application. If NET.Framework 2.0 is not already installed on your computer, the "redistributable" can be downloaded from the Microsoft website at no charge.

Vista users will need full administrative privileges to use some of the features of AI Flight Planner, in particular Collect Airports and Convert FS9 Traffic Files for FSX.

To uninstall AI Flight Planner, simply delete the folder in which it is installed.

3. THE AI FLIGHT PLANNER MAIN WINDOW

AI Flight Planner's main window is partitioned as follows.

- The main menu bar is in its traditional position across the top of the window. Used to initiate functions not directly associated with the editing of flight plans, it comprises four sections: *Flight Plans*, *Aircraft*, *Airports* and *Help*; in this manual, references to menu selections will be in the form "Menu Name / Item name".
- The left-hand side of the main window holds the *Flight Plan List* - a list box containing all loaded flight plans (in TTools format) - and related buttons and search functions. Each item in the *Flight Plan List* is identified with a line number.
- To the right of the *Flight Plan List*, towards the bottom of the window, is the *Airport List*, a list of all the airports referenced by the flight plans in the *Flight Plan List* and the editor.

- The top-center portion of the window is occupied by the *Time Zone Selector*. Flight plan times may be displayed/edited in either local times or a selected time zone. However, for local times to be used, geographic information (city, country, position) on all the airports involved in the flight plan must be included in AI Flight Planner's internal airport list.
- The upper right-hand corner is devoted to the *Aircraft List*, which includes the available aircraft and permits selection of the aircraft to be used in the currently-selected flight plan. The *Aircraft List* includes a field for the entry of the aircraft's tail (registration) number.
- The remainder of the screen is consumed by the *Flight Plan Editor*. Across the top of the area is the *Flight Plan Base Data*, which displays and allows editing of the flight plan data common to all the legs of a flight plan. Below that is the leg editing area and below that again, the *Leg List*

Whenever a change is made to the *Flight Plan List*, the *Aircraft List* or the *Leg List*, a red ### indicator will appear to the right of the list name to remind the user to save the list.

As a general rule, only those menu items, buttons and data entry fields valid for use in the current context of AI Flight Planner are enabled. In particular, all the fields of the editor are disabled unless a flight plan is being edited. All the fields of the *Flight Plan Base Data* must be completed before the fields and buttons applicable to the *Leg List Editor* are enabled. All leg fields except *Override ETA* (which may be left blank) must be completed before the buttons involved in placing the edited leg data (back) into the *Leg List* are enabled. Buttons pertaining to the insertion of data into or deletion of data from any list are enabled only when an item in the relevant list is selected. And, finally, the buttons for moving a completed flight plan from the editor (back) into the *Flight Plan List* are enabled only when there has been a change to the leg data and there are at least two legs in the *Leg List* – the minimum for a valid flight plan – unless touch-'n-go operation is specified.

So, at any time, if a button, menu item or field data-entry field disabled, it is because that button, menu item or field is not useable in the current context of AI Flight Planner – probably because some required data is missing.

Many buttons and menu items have keyboard shortcuts. If you don't see the shortcuts, it is because Windows is preventing them from being displayed until you press the <Alt> key. You may override this feature of Windows using Control Panel. In Windows XP, the control to do so is found at Display – Appearance – Effects. In other versions of Windows it may be elsewhere.

4 MAIN MENU FUNCTIONS

- 4.1 Creating a New Flight Plan File – The main menu item *Flight Plans/Start New File Set* will clear the *Flight Plans List*, the *Airport List* and the *Aircraft List* in preparation for the creation of new flight plans and the specification of new aircraft as described below.
- 4.2 Loading Flight Plans and Associated Files – The loading of flight plan and aircraft data is initiated from the main menu. The available functions are:

- *Flight Plans / Open File Set* clears the *Flight Plan List* and *Aircraft List* and displays a dialog box to specify the name of a TTools flight plan file.
- *Flight Plans / Append File Set* is identical to *Open File Set* except that the *Flight Plan List* and *Aircraft List* are not cleared.
- *Flight Plans / Open Traffic .bgl* operates like *Open File Set* except that the flight plan and aircraft data will be derived from the designated MSFS traffic file.
- *Flight Plans / Append Traffic .bgl* is identical to *Open Traffic .bgl* except that the *Flight Plan List* and *Aircraft List* are not cleared.
- *Flight Plans / Reload Last File (Set)* reloads the last flight plan file opened (before any appends), whether a TTools file or a traffic file. The *Aircraft List* is also reloaded if a companion aircraft file existed.
- *Aircraft / Load New Aircraft File* clears the *Aircraft List*, displays a dialog box to specify the name of a TTools-format aircraft file to be loaded.
- *Aircraft / Append Aircraft File* is identical to *Load New Aircraft File* except that the *Aircraft List* is not cleared.

Please be aware that the times in the flight plan information returned by the *Flight Plans / Open Traffic .bgl* function may not match exactly the times originally specified by the user. This is because of the way times are stored in the traffic file. The “consolidated” mode of operation of the *Leg List* may be affected (even though AI Flight Planner allows a 2-minute “window”). As well, the returned aircraft speed may be in error by 1 knot.

- 4.3 Flight Plan Validation – The flight plans in the *Flight Plan List* may be validated using the *Flight Plans / Validate Flight Plan List* menu item. If there is more than a single error, the error list is presented in Notepad format and may be printed.

Please note that the leg reference numbers in the error report refer to the leg order based on UTC/GMT time. These reference numbers may not be applicable to the legs when displayed in the *Leg List*. Should this be the case and it is not readily apparent from the *Leg List* display which leg is at fault, you can identify the leg by reference number using the flight plan plain text editor.

- 4.4 Flight Plan Subsets – AI Flight Planner allows the creation of subsets of the current flight plans based on airports selected in the *Airport List*. Subsets may be either inclusive or exclusive. An inclusive subset includes each flight plan that references any airport selected in the *Airport List*. An exclusive subset includes every flight plan that does not reference those airports. Among other things, this feature allows the generation of regional flight plan lists and the exclusion of designated airports from the default traffic files.

There are two associated menu items:

- *Flight Plans / Create Subset (Inclusive)*
- *Flight Plans / Create Subset Exclusive)*

For both items, the user will be given an opportunity to specify whether or not a subset of the *Aircraft List* also is to be generated.

- 4.5 Saving/Compiling Flight Plans and Associated Files – The following main menu items control the saving and compiling of flight plan and aircraft data.
- *Flight Plans / Save File Set* saves the *Flight Plans List* and the *Aircraft List* in TTools format under the name of the last flight plan file loaded. This menu item remains disabled until changes to either the *Flight Plan List* to the *Aircraft List* are made. This menu item is also disabled when a flight plan file is appended.
 - *Flight Plans / Save As . . .* displays a dialog box to specify a file name under which the *Flight Plan List* and *Aircraft List* will be saved. The data will be saved in TTools format.
 - *Flight Plans / Compile Traffic .bgl* operates similar to *Flight Plans / Save As . . .* except that after the files are saved temporarily, a dialog box is opened and the data is compiled into a MSFS traffic file (.bgl) under user control. In this mode, the compiled file is pre-processed by AI Flight Planner to address certain idiosyncrasies of the simulator's AI engine (discussed in Section 1.4).
 - *Flight Plans / Compile Raw* also compiles the user files but without the pre-processing of the normal compile mode.
 - *Flight Plans / Compile Again* recompiles without user intervention the current file set using the same mode, target file name, MSFS version, etc. as the last compile.
 - *Aircraft / Save Aircraft File* operates similar to the *Flight Plans / Save File Set* item except that only the *Aircraft List* is saved.
 - *Aircraft / Save As* operates similar to the *Flight Plans / Save As . . .* except that only the *Aircraft List* is saved.

There are no restrictions on the naming of traffic files to be decompiled by AI Flight Planner. However, each traffic file compiled by AIFP will be named "Trafficbgl".

- 4.6 Conversion of FS9 Flight Plans for use with FSX – The *Flight Plans / Convert FS9 FP to FSX* menu item examines the destination airports of each flight plan in the *Flight Plan List* and, where the corresponding airport in FSX has a different ICAO code, updates the destination airports accordingly. This function also flags FS9 destination airports for which there is no FSX equivalent. The conversion report is presented in Notepad format and may be printed.

Like flight plan validation, the leg reference numbers in the conversion report refer to the leg order based on UTC/GMT time. These reference numbers may not be applicable to the legs when displayed in the *Leg List*. Should this be the case and it is not readily apparent from the *Leg List* display which leg is at fault, you can identify the leg by reference number using the flight plan plain text editor.

A second function, *Convert FS9 traffic Files to FSX*, applies similar conversion to one or more traffic (.bgl) files designated by the user as noted in Section 1.5.

- 4.7 Managing the Aircraft List – Changes to the *Aircraft List* are initiated from the *Aircraft* menu.
- *Add New Aircraft* opens a dialog box that allows aircraft to be added to the *Aircraft List*.

- *Modify Selected Aircraft* opens a dialog that facilitates modification of the parameters of the aircraft selected in the *Aircraft List*.
- *Send Aircraft List to Text Editor* opens the entire *Aircraft List* in the user's default text editor in TTools format. The edited file will be reloaded automatically into the *Aircraft List* when the text editor is closed. If the reference number of the aircraft selected when the editor was opened still exists upon closing, that aircraft will be re-selected.
- *Delete Selected Aircraft* deletes the selected aircraft from the *Aircraft List*
- *Delete Unused Aircraft* deletes from the *Aircraft List* any aircraft not used by the flight plans in the *Flight Plan List*

4.8 Managing Airports – All airport-related functions are initiated from *Airports* menu. Most of these functions open a context specific dialog box.

- *Get Airport Information* displays a “tree-view” from which you can find the ICAO identifier of any airport known to the system and see other information about the airports.
- *Arrivals & Departures* allows you to view all the arrivals and departures scheduled in the flight plans then in the *Flight Plan List* for any airport. The arrivals and departures for any airport may be “dumped” to Notepad.
- *Change Local Time Offset* allows AI Flight Planner's time zone database to be updated.
- *Edit/Add Airport* permits editing of the parameters of an existing airport or adding new airports to *AirportList.dat*.
- *Bulk Airport Update* reads data from a TTools- format aircraft file and uses this data to update the internal airport list.
- *Collect Airports* controls the collection of airport data for both stock and add-on airports.
- *Customize Airport List* displays a “tree-view” display similar to that for *Get Airport Information* but which includes checkboxes to allow you to specify the airports to be included in a custom airport list. This specification may be on a country, region, city or individual airport basis or any combination. Following selection, on user command, a new airport list named *AirportList-Custom.dat* will be created that will be used in place of *AirportList.dat* until you direct otherwise. For more information, please refer to *MANAGING AIRPORT DATA – Customizing the Airport List*.
- *Use Master Airport List* forces AI Flight Planner to use *AirportList.dat* instead of *AirportList-Custom.dat* for airport data purposes.
- *Save Selected Airport List* saves a list of the airports selected in the *Airport List* to a designated file.
- *Select Airports from List* selects the airports in the *Airport List* that are contained in the designated file.

5 BUTTONS AND INDICATORS

While the use of AI Flight Planner should be largely intuitive, the following descriptions may be help you appreciate its full capabilities.

5.1 Flight Plan List – The following buttons initiate *Flight Plan List* functions:

- *Start New File* clears the *Flight Plan List*. This is similar to the main menu item *Flight Plans / New* except that the *Aircraft List* is preserved.
- *Delete Item* deletes the selected item from the *Flight Plan List*.
- *Open Text Editor* opens a dialog box containing a multi-line text box together with Replace, Insert, Add and Exit buttons (If an item in the *Flight Plan List* is selected, that item will be copied to the text box. Both flight plans and comments may be created/edited).
- *Change Day Code* changes day encoding of the flight plans in the *Flight Plan List* from FS9 to FSX or vice versa depending on the state of the day-encoding radio buttons.

Immediately below the *Flight Plan List* are two radio buttons which indicate the day-encoding scheme used in the flight plans (see *USING THE FLIGHT PLAN EDITOR, Loading Flight Plans*). These radio buttons also allow the user to indicate the correct scheme should AI Flight Planner not be able to determine it automatically. However, the *Change Day Code* button must be used to actually change the day-encoding in the flight plans.

To the upper right of the *Flight Plan List* are several controls which allow you to search the flight plans in the *Flight Plan File*. Select the field to be searched in the *Field* combo box, specify whether the match parameter is to be less than (<), equal to (=) or greater than (>), enter the value you wish to find in the *Find* text box and click one of the “find” buttons. The *Find* button selects (highlights) the first flight plan in the *Flight Plan List* matching the criteria. Use the *Next* button to search for the second and subsequent occurrences of the desired field. The *All* button selects all the flight plans in which the criteria are matched.

Immediately below the “find” buttons is the Replace text box and the Replace and Replace All buttons. The former replaces all occurrences of the match criteria in a single flight plan with the contents of the *Replace* text box (The *Replace* button is disabled if there is no flight plan selected or if more than one is selected.) As the name suggests, the *Replace All* button replaces every occurrence of the “find” criteria in the *Flight Plan List*.

5.2 Moving Items To and From the Flight Plan List – Items are copied from the *Flight Plan List* into the *Leg List* and editor by double-clicking on the item. The movements of items from the editor (back) into the *Flight Plan List* are controlled by the following buttons.

- *Update FP in List* copies the edited flight plan to the *Flight Plan List*, replacing the item that was selected when the flight plan was loaded into the editor
- *Add FP to List* adds the edited flight plan at the bottom of the *Flight Plan List*
- *Insert FP into List* inserts the edited flight plan into the *Flight Plan List* immediately above the selected item.

These buttons are not enabled unless there are a sufficient number of legs in the *Leg List* for the flight plan to be valid and further, in the case of *Insert FP*, an item in the *Flight Plan List* is selected.

5.3 Leg List – Items are copied from the *Leg List* into the leg editor by double-clicking on the item. The movement of items from the leg editor into the *Leg List* and other *Leg List* manipulations are controlled by the following buttons:

- *Save Edits* copies the leg data from the editor back to the *Leg List*, replacing the item in the list that was selected when the leg was loaded into the editor.
- *Add Leg* clears the leg editor and creates a new blank leg at the bottom of the *Leg List* (where the new leg data will be saved when *Save Edited Leg Data* is later clicked).
- *Insert Leg* clears the leg editor and creates a new blank leg immediately above the selected item in the *Leg List* (where the new leg data will be saved when *Save Edited Leg Data* is later clicked).
- *Delete Leg* deletes the selected leg from the *Leg List* after user confirmation.
- *Start New FP* clears the *Leg List*, creates a new blank entry in it and initializes other editor data.

The *Delete Leg* and *Insert Leg* buttons are not enabled unless a leg is selected in the *Leg List*. The *Add Leg* and *Insert Leg* buttons are both disabled after either is used and remain disabled until the associated blank line in the *Leg List* is either updated with the edited data or deleted.

Finally, the *Leg List* has two modes of operation when the selected repeat period is *Week*:

- “Individual”, which displays each leg in the flight plan individually, and
- “Consolidated”, which consolidates legs that operate on more than one day of the week into a single list item followed by a day-of-week indicator.

You can select the desired mode using the *Consolidate Weekly FPs* checkbox located in the bottom right-hand corner of the window. Daily flight plans are always displayed in the “individual” mode.

5.4 Airport List – As the name implies, the *Select All* button selects all the items in the *Airport List*. Items may be selected individually in the traditional manner using the keyboard <Ctrl> key.

5.5 Aircraft List – The items in the *Aircraft List* may be sequenced numerically by aircraft reference number or alphabetically by aircraft title using the applicable radio button.

6. USING THE FLIGHT PLAN EDITOR

6.1 Arrival and Departure Times – MSFS requires flight plan data in a traffic file to be based on UTC/GMT time and the *Flight Plan List* reflects this requirement. However, AI Flight Planner allows arrival and departure times of flight plans in the editor and in arrival/departure summaries to be entered and displayed either in either local time (at the originating and destination airports) or based on a specific time zone. Subsequent conversion to UTC/GMT time for use by MSFS is automatically.

Local time may reflect either winter or summer time, as specified by the user. However, flight plans are not date-specific, so arrivals and departures in the

transition periods to and from daylight savings time (where applicable), may be in error by an hour.

When using local time, the user should appreciate that AI Flight Planner's time zone database reflects real-world time zones – which may not match in all cases the time zone calculated by MSFS. MSFS uses a geographic approximation technique to determine the time zone in the area where the user aircraft is located. Hence, the two may disagree at airports close to the edge of time zones. Since MSFS' calculation of time zones can be affected by add-ons, such discrepancies cannot always be resolved by AI Flight Planner. As well, while significant efforts were expended in making AI Flight Planner's time zone data base complete and accurate, time zone data for smaller airports, especially in developing countries, sometimes is not readily available and, hence, may be in error.

When local time is in use, AI Flight Planner displays the UTC/GMT offset it uses for each airport in the *Airport List*. If an offset is not correct, it is easily changed (permanently) using AI Flight Planner's time zone database editor.

Unlike TTools, AI Flight Planner does not accept "@" and "TNG" as prefixes for arrival times. Touch 'n go operation is specified using a checkbox. The function performed by the "@" symbol is automatic with AI Flight Planner. (Any arrival time that differs by more than two minutes from the system-calculated arrival time is assumed to have been specified by the user. This buffer is necessary to accommodate "jitter" introduced by de-compilation.)

AI Flight Planner determines whether the specified arrival time is for the day (repeat period) of departure or the following day (repeat period) and, when not the day (repeat period) of departure, applies an appropriate suffix. If arrival time in 24 hour format is earlier than departure time, arrival is assumed to be the following day (repeat period) and the arrival time field will be suffixed with "+1" – a notation used in many airline schedules. Where the flight crosses the International Date Line, the suffix may also be "-1" (eastbound flights leaving Asia just after midnight) or "+2" (westbound flight leaving North America just before midnight). Like arrival time prefixes, you need not enter these suffixes; AI Flight Planner will apply them where appropriate. But, you may override the system-generated suffix and it will be used instead where possible.

When a sub-daily repeat period is selected, the hour value of the arrival and departure time entries must be less than the repeat period. For example, for a repeat period of 4 hours, the maximum acceptable arrival or departure time is 03:59. When the repeat period is changed to a smaller value such that previously-entered arrival and departure times in the editor are invalid, the arrival time and predicted ETA suffixes may become what appears to be a nonsensical number.

In weekly consolidate mode, the departure times displayed in the *Leg List* are in "day-time" configuration, i.e., d/HH:MM. This is to maintain proper sequencing of the entries. Departure time entries in the editor are always in HH:MM with day-of-week information being derived from the day-of-week checkboxes in the *Base Data* area.

- 6.2. Departure Airport - A flight plan leg specifies only the destination airport. The departure airport for any leg is the destination of the previous leg in time. For the first leg of a flight plan, the originating airport is the destination of the final leg – with one exception. If, when starting a new flight plan, you enter a departure airport for the first leg, AI Flight Planner will continue to report that airport as the departure airport for the first leg irrespective of the following legs. When it comes time to save the flight plan, if the destination of the final leg does not match this airport, AI Flight Planner will alert you. You will then have the option to either save the flight plan as it exists, with the destination airport of the last leg becoming the departure airport for the first leg, or to further edit the flight plan.

Otherwise, AI Flight Planner uses the departure airport (solely) to calculate the distance and duration of the flight specified in the leg and to predict the arrival time. When a flight plan leg is moved into the editor, AI Flight Planner attempts to determine the airport from which that leg departs. For a daily (or more frequent) flight plan, this is a simple task – it is the destination of the leg immediately previous in the *Leg List* to the leg of interest. However, when in the consolidated display mode, the previous leg (in a weekly flight plan) is not necessarily the leg immediately previous in the *Leg List*.)

Consider, for example, the simple case of a flight plan that operates between airports A and B from Monday to Friday but makes an intermediate stop at airport C on Wednesday. Such a situation requires three legs to be specified; A to B on M/T/T/F and A to C and then C to B on Wednesday. In the *Leg List*, the sequence of the legs will be A to B, A to C and then C to B. So, there will be two intervening legs between the A to B leg and the ongoing leg from B. To accommodate such situations, AI Flight Planner looks back up the *Leg List* to find an earlier flight plan scheduled for the same day of the week and assumes that to be the departure airport.

However, depending on the complexity of the flight plan, AI Flight Planner may not correctly identify the departure airport. (This is a trade-off for being able to specify a leg only once irrespective of how many days of the week it operates, and is of little consequence, since the departure airport is only used to calculate the distance and duration and to predict the ETA of the flight.) Should AI Flight Planner not determine the correct departure airport, you may override its selection to allow the calculation of distance, duration and ETA for the leg. This action will have no effect on the saved/compiled flight plan, which will always reflect the leg sequence.

- 6.3 Loading Flight Plans – Flight plans may be loaded from two sources:
- TTools-format *.txt* files - click *Flight Plans / Open File Set* and specify the desired flight plan file; if there is a companion aircraft file, it also will be loaded; and
 - compiled MSFS traffic *.bgl* files - click *Flight Plans / Open Traffic .bgl* and specify the traffic file of interest; the traffic file will be decompiled and both flight plans and aircraft data will be loaded.

The flight plans are saved to the *Flight Plan List* after the associated aircraft data (if any) has been loaded into the *Aircraft List*. Error/warning messages will be generated if:

- either file is not in the expected format; or
- a flight plan refers either to:
 - an airport not included in *AircraftList.dat* (or the custom version), or
 - an aircraft not in the *Aircraft List* (these error messages are suppressed if the *Aircraft List* is empty).

At the completion of the file loading activity, summaries of any missing airports and/or aircraft are provided. In the case of airports, at the users direction, the Edit Airports dialog will be opened automatically and *AirportList.dat* may be updated under user control based on the companion airports file if available (generated automatically when a traffic file is decompiled). (See also relevant sections under *MANAGING AIRPORT DATA*.)

FS9 and FSX flight plans in TTools format are identical save for day-of week encoding. In FS9, Sunday is encoded as 0, while FSX uses 0 for Monday. So, AI Flight Planner needs to know which day-encoding scheme is in use so that, for weekly flight plans, the editor reflects the proper day of the week and the AI operates on the correct days. Where the flight plan data comes from compiled traffic files, the decompiler tells AI Flight Planner which encoding scheme is used. However, there is no way for AI Flight Planner to determine automatically the day-encoding scheme used in TTools-format flight plan files. AI Flight Planner defaults to FS9 day encoding after opening a TTools-format flight plan file. (Day-encoding is left unchanged when appending files.) If the FSX day-encoding scheme is used in the loaded flight plans, it is left to the user to so indicate using the radio-buttons located below the Flight Plan List.

Previously-loaded flight plan and aircraft data may be supplemented at any time by appending data from additional files. Appended data may be derived from either TTools-format files or compiled traffic files, irrespective of the source of the previously loaded data. Before appending TTools-format flight plan files, AI Flight Planner will ask for confirmation that the current day-encoding is appropriate for the file to be appended. Before appending flight plans from a decompiled traffic file, AI Flight Planner will check that the day encoding is the same and, if not, announce the issue. Where necessary, day-encoding in the already-loaded data may be changed using the *Change Day Code* button.

- 6.4 Built-In Text Editor – TTools formatting supports embedded comments. AI Flight Planner provides a simple text editor to allow insertion of comments into, and editing of comments already in, the *Flight Plan List*.

To open the text editor, double click on a comment line in the *Flight Plan List*. Alternately, select an item in the *Flight Plan List* and click the *Open Text Editor* button. The text editor supports both single-line and multi-line comments. Each line in a multi-line comment should be terminated using the keyboard <Enter> key and the succeeding line commence with either “,” or “//”.

For simple changes to flight plans, it may be more convenient to use the text editor rather than the flight plan editor. To edit a flight plan in the text editor,

select it in the *Flight Plan List* and click the *Open Text Editor* button. The data is formatted to make each leg readily identifiable. Should edits destroy this clarity, it may be restored with the *Format* button.

In addition to replacing the *Flight Plan List* item selected when the text editor was opened, the contents of the text editor may be added at the end of the *Flight Plan List* or inserted immediately above the selected item.

- 6.5 Flight Plan Editor – Flight plans are moved into the flight plan editor by double clicking on the flight plan in the *Flight Plan File*. The *Aircraft List* combo box will highlight the aircraft used by the flight plan (if it is in the *Aircraft List*), the other base data of the flight plan will be displayed in the base data area across the top of the main window, the flight plan legs will be inserted into the *Leg List* and, lastly, the top item in the *Leg List* is selected, parsed and displayed in the editor.

Each leg of the flight plan is displayed in the *Leg List*, time sequenced. As noted earlier, for weekly flight plans, a leg which is flown on two or more days may be shown in either of two ways:

- “individual”, i.e., each leg in the flight plan shown individually, or
- “consolidated”, i.e., legs that operate on more than one day of the week but are otherwise identical consolidated into a single list item.

The *Consolidate Weekly FPs* checkbox located in the bottom right-and corner of the window control this mode.

In “individual” mode, the legs must be entered in operational sequence. This is not a requirement in the consolidated mode, however. In the latter mode, the position in the *Leg List* at which a new leg is entered determines the departure airport for that leg - for the purpose of calculating distance, duration and ETA. When the data for a new leg is entered into the *Leg List* or when edited data is saved, AI Flight Planner will automatically place the edited leg based on departure time.

The times displayed in the *Leg List* and in the editor reflect the time zone selected by the user with the *Time Zone Selector*. (The *Flight Plan List* always displays UTC/GMT time.) When the time zone is changed, the flight plan currently in the editor is reloaded from the *Flight Plan List*. So, time zone changes should be made before any edits are applied to the flight plan in the editor. However, if the time zone is changed after edits have been made, the user is given an opportunity to save the edits before the new time zone takes effect.

Flight plan editing is straightforward. (Nonetheless, if you are not familiar with the individual flight plan fields, it is strongly recommended that you download Lee Swordy's TTools and refer to its user documentation.) The base data for the flight plan (i.e., the data that applies to all legs, such as selected aircraft and activity level) is entered into the editor at the time the flight plan is moved from the *Flight Plan List*.

An individual leg is moved into the editor by double-clicking on it in the *Leg List*. Leg data in the editor is displayed in a combination of text boxes, check boxes

and radio buttons. In general, error checking, where applicable, is performed when you move the cursor away from an edited text box. Error checking of the whole leg is performed when the edited data is saved.

When a leg is loaded into the editor, AI Flight Planner will (attempt to) identify the departure airport and calculate the distance between it and the destination airports as well as the expected duration of the flight (based on the cruising speed of the selected aircraft) and anticipated arrival time. Of course, if AI Flight Planner picks the wrong departure airport in a complex weekly flight plan, these calculations will be of little use. Should that be the case, however, you may enter the proper destination airport.

When you enter or change an airport ICAO code, the newly-entered code is validated as soon as you move the cursor to another field. If the code is valid, the airport entered into the *Airport List* if it is not already there and the name of the city will be shown. If invalid, an error message will be issued and the user given an opportunity to enter the airport into the system. If unsure of the ICAO code for the intended airport you may enter “?” optionally preceded by a character string. If:

- no character string is entered, i.e., just “?”, the Get Airport Information dialog box will be opened; locate the ICAO code of interest, select (click on) it and close the dialog box;
- the character string contains “>”, a list of ICAO codes where the city name or airport name contains the entered string will pop up; or
- otherwise, a list of ICAO codes which start with the entered string will pop-up.

In all cases, the selected ICAO code will be placed in the relevant field and validated. When editing an existing flight plan file, should the ICAO code of an airport that is not known to the system be entered but the airport is contained in the companion TTools-format airports file, you may direct AI Flight Planner to use the data from the airports file. You will also be given the opportunity to add the airport to *AirportList.dat* using the Airport Editor which will be initialized for the airport of interest.

An AI aircraft is known to MSFS Air Traffic Control (ATC) by either its flight number or its registration number. If the *ATC Callsign FN* button is selected, a flight number must be entered for the leg. If the *Reg* button is selected, the aircraft associated with the flight plan must have a tail (registration) number.

When all edits have been applied to the leg, use the *Save Edited Leg Data* button to update the item in the Leg List to which the edits apply. The original copy of the leg in the *Leg List* is updated, irrespective of which leg is then selected in the list when the *Save Edited Leg Data* button is clicked.

New legs may be inserted into or added to the *Leg List* as necessary. To insert a new leg, select the leg in the *Leg List* above which the new leg is to be inserted and click on the *Insert Leg* button. A blank leg will be inserted into the *Leg List* at that point as a “place-holder”. To add a new leg at the bottom of the *Leg List*, click on the *Add Leg* button. Similar to *Insert Leg*, a blank leg will be appended to the leg list and the departure airport set. Please note that the *Save Edited Leg*

Data button will not be enabled until all fields except arrival time have been entered.

When a new leg is added to/inserted in a weekly flight plan, there is no day of the week information. So initially, AI Flight Planner will assume that the leg originates at the destination of the leg immediately above it in the *Leg List*. However, once the days of operation are specified, AI Flight Planner attempts to refine its earlier choice by selecting a departure airport from an earlier leg that operates on the same day(s). But, as noted earlier, if you enter a departure airport for the initial leg, AI Flight Planner will continue to use it instead.

When all edits have been made to a flight plan, the edited flight plan may update the original copy of the flight plan in the *Flight Plan List* or be added to the bottom of, or inserted immediately above the selected flight plan in, the list using the *Update FP in List*, *Insert FP in List* or *Add FP to List* button, as applicable.

- 6.6 New Flight Plans from "Scratch" – To start a new flight plan "from scratch", click on the *Start New FP* button. This will clear the *Leg List* if there is any data in it, place a blank entry in the *Leg List* and clear the editor fields. Initially, only the fields for the base data will be enabled. Once all the base data for the flight plan has been entered and the aircraft selected, the *Leg List Editor* fields/controls will be enabled. This sequence ensures that all required data is available when needed. In the *Leg List* editor, all fields must be completed except for arrival time, which may be left blank. (When the flight plan is saved, AI Flight Planner will calculate arrival time if not specified based on the selected aircraft cruising speed.) When the required data for the first leg has been entered, save it to the *Leg List* using the *Save Edited Leg Data* button. Create additional legs as necessary using either the *Add Leg* or *Insert Leg* buttons. (In all cases, please note that the *Save Edited Leg Data* button will not be enabled until all fields except arrival time have been entered.)

Before attempting to enter flight plan leg data, it is recommended that the associated aircraft exist in the Aircraft List. (The leg editor will not be enabled until an aircraft has been selected.) This data may be derived from one or more TTools-format aircraft files and/or created or supplemented using the *Aircraft / Add New Aircraft* capabilities of the system initiated from the main menu.

Once all the legs of the new flight plan have been entered, save the flight plan to the *Flight Plan List* using the *Add to FP List* or *Insert in FP List* buttons as appropriate. These buttons are not enabled unless there are sufficient legs in the *Leg List*, i.e., two for all but TNG operation.

7 MANAGING AIRCRAFT DATA

AI Flight Planner does not restrict you to using aircraft data associated-by-name with the flight plan data ("companion" file). When a TTools-format flight plan file or a compiled traffic file is loaded into AI Flight Planner, "companion" aircraft data (if it exists) will also be loaded. However, the user is free to replace this data or supplement it with data from other sources.

- 7.1 Loading and Saving Aircraft Data - The items in the *Aircraft* main menu item:

- *Load New Aircraft File*
- *Append Aircraft File*
- *Save Aircraft File*
- *Save As ...*

perform the functions noted in MAIN MENU FUNCTIONS section.

7.2 Adding New Aircraft to the Aircraft List – Aircraft may be added to the *Aircraft List* by clicking on the *Aircraft / Add Aircraft* menu item. This will open the Aircraft Editor dialog box, which allows specification of new aircraft in either of two modes.

- Specify a FS9 or FSX aircraft folder using *Aircraft Folder Select*, which will list in the *Aircraft Title* combo box all the AI aircraft available in that folder. Select the desired aircraft in the combo box. The *Cruise Speed* text box will be filled in automatically with the cruising speed specified in the relevant *aircraft.cfg* file, but it may be edited. If *Cruise Speed* is edited at this time, the cruising speed in the *aircraft.cfg* file will be updated and, hence, affect the performance of the aircraft in the simulator; or
- Enter the title for of the new aircraft (which must be unique in the *Aircraft List*) into the *Aircraft Title* text/combo box and a cruising speed into the *Cruise Speed* text box.

AI Flight Planner automatically displays the next highest aircraft reference number in the *AC#* text box. However, the user is free to assign any other unused reference number. Once all three fields have been entered, the *Add* button will be enabled. Click it to add the new aircraft to the Aircraft List. The dialog box remains open until closed by the user.

7.3 Editing an Aircraft Already in the Aircraft List – To edit an aircraft in the *Aircraft List*, select the aircraft and click on the *Aircraft / Modify Selected Aircraft* menu item. This action also opens the Aircraft Editor dialog box.

Edit the *Aircraft Name*, *AC#* and *Cruise Speed* fields as necessary, bearing in mind that if either the title or reference number is changed, the new value must be unique in the *Aircraft List*. When ready, click on the *Replace* button. Please note that, in this case, a change to *Cruise Speed* will not be reflected in the *aircraft.cfg* file (since all association with the aircraft folder from which the aircraft may have been added has been lost). The dialog box is automatically closed when the replacement is accepted. If you change the *AC#*, AI Flight Planner will offer to update all references to the old number.

7.4 Deleting Aircraft – To delete an aircraft from the *Aircraft List*, simply select the aircraft and click on the *Aircraft / Delete Selected Aircraft* menu item. All aircraft not used by the flight plans in the Flight Plan List may be deleted using the *Aircraft / Delete Unused Aircraft* menu item.

8 COLLECTING AND EDITING AIRPORT DATA

AI Flight Planner uses its own internal airport information – even if there is companion airport data for the flight plan file(s) loaded. Generally, if the user has created a custom internal airport list, AI Flight Planner will use this custom list as its source of airport

information. But, the user may revert to the master *AircraftList.dat* at any time using the *Airports / Use Master Airport List* menu item.

- 8.1 Collecting Airport Data - AI Flight Planner includes an airport data collector which collects data for FS9 and or FSX stock airports as well as for add-on airports for both. For add-on airports, the user is given the opportunity to select the add-on folders of interest. The collected airport data is displayed to the user in a standard list box. The data is saved to *AirportList.dat* under user control.

The airport data collection function first attempts to locate the FS9 and FSX top folders using registry entries. If it is successful, the path(s) will be displayed at the top of the form and the associated stock and add-on airport checkboxes enabled. If either top-level folder is not found, the associated checkboxes will remain disabled. If FS9 or FSX exists on the user's computer and is not located automatically, the user may indicate the relevant top folder using the *Select* buttons.

As noted below, collection of airport data takes a while. Consequently, AI Flight Planner "ships" with the data for both FS9 and FSX stock airports already collected. Data for the individual versions are in *AirportList_Base_FS9.dat* and *AirportList_Base_FSX.dat*. *AirportList.dat* holds the consolidated data.

The scope of the collection process is controlled by the four checkboxes located beneath the top-level folder path displays. Data may be collected for FS9 stock airports, FSX stock airports, add-on airports for either version or any combination of the four. Should you wish to re-build *AirportList.dat*, for example to include both stock data as well as your own collection of add-on airports, AI Flight Planner gives you the opportunity to re-use the existing stock data rather than re-collect it.

Each time airport data is collected, any updates previously applied by the user and saved in the file *AirportList - Updates.dat* are re-applied. Also, if you want the data (latitude, longitude and elevation) in any add-on files found to replace that of the corresponding stock airports, check the *Update from Add-on* checkbox.

Once collected, the selected data is displayed in the list box which occupies most of the area of the form.

Because of the extent of control over the collection process, this feature may find other uses. For example, if you wish to know which airports are installed in a given add-on folder, the airport collector will answer the question. Hence, airport data for purposes other than updating *AirportList.dat* may be explored. Data is not copied to *AirportList.dat* until the user clicks on the *Save "AirportList.dat"* button, so you may safely experiment.

Airport data collection requires several minutes, much longer than the airport scanning mechanism in most airport editors. This is because those editors have a specialized, built-in decompiler while, for the moment, AI Flight Planner uses a generic decompiler, *bglAnalyze9* and its FSX counterpart *bglAnalyzeX*. AI Flight Planner must wait for *bglAnalyze* to fully decompile a file before being able to determine whether or not the file is of interest. This is particularly problematic

when collecting data for add-on airports, since every *.bgl* file in the *lscenery* folders must be examined even though only one may contain airport data. Future versions of AI Flight Planner may offer a significant improvement in this regard. Since AI Flight Planner ships with the data for the FS9 and FSX stock airports already collected and will use this data unless you direct otherwise, you shouldn't have to endure these delays very often.

Should you wish to terminate the airport data collection process, click on the *Abort* button.

In addition to a progress bar, for amusement while scanning, the number of airports for which data has been collected to that point and the country for which airports are currently being examined are shown.

- 8.2 Customizing the Airport List – The airport data included with AI Flight Planner includes every stock airport FSX – which number about 25,000. Thus, the file *AirportList.dat* is huge – nearly 2mb. For those with state-of-the-art computer systems, this will not present a problem. However, if you have an older computers or limited RAM, you may wish to work with only a subset of the available airports.

To generate a custom airport list - which will be selected automatically on subsequent start-ups of AI Flight Planner:

- click on the *Airport / Customize Airport List* menu item,
- select the airports/cities/regions/countries of interest in the “tree-view” display, and
- click on the *Generate Subset* button.

The newly generated airport data file will be named *AirportList-Custom.dat*. If custom airport data already exists, the user will be warned before overwriting it.

- 8.3 Editing and Manually Adding New Airports – Errors have been noted in MSFS airport data. As well, the original airport data may not match that contained in add-on scenery loaded into the user's system. While small changes are unlikely to have any noticeable effect on AI operation, the user may wish to have *AirportList.dat* reflect the corrected data. As well, since AI Flight Planner will not compile a flight plan file containing an airport for which it does have positional information, any new airport created in add-on scenery must be added to *AirportList.dat* (and its custom counterpart, if used) before AI can be programmed for that airport.

To modify the data for any airport or add a new airport, click on the *Airports / Edit/Add Airport* menu item, which will open the Airport Editor dialog box with the controls pertaining to bulk-update (see next section) disabled. Then, proceed as follows:

- if custom airport list exists, designate whether the update is to be applied to the custom or the original data;
- enter the airport ICAO code and click the *Open* button to edit an airport or click the *New* button and then enter the new ICAO code for a new airport
- enter/update the remaining data as necessary, and

- click the *Apply Update* button.

To avoid potentially lengthy delays between successive edits due to the size of *AirportList.dat*, the airport data file is read into memory when the editor is opened and any changes you make are entered only into the memory-resident version of the file. The file on disk is not updated until the *Save File and Exit* button is clicked. If you wish to exit without saving, click the *Exit* button.

Where both custom and original data is to be updated, the change must be entered twice or the custom file re-created.

Whenever an airport is entered or updated manually, a record is saved in a file named *AirportList – Updates.dat*. If *AirportList.dat* ever needs to be regenerated, these updates will be applied automatically

- 8.4 Airport Data Bulk Update – Despite the large number of stock airports included in FSX, some small local airports, grass strips and water airports are missing. Scenery developers often model these missing airports. Where flight plans for such airports have previously been prepared, AI Flight Planner is able to update *AirportList.dat* based on the data in the corresponding TTools-format airport file.

To update AI Flight Planners airport data from a TTools-format airport file:

- click on the *Airports / Bulk Update* menu item
- check the *Update Existing* checkbox if you wish airports already in the *AirportList.dat* to be updated based on the data in the TTools file; and
- if custom airport data exists, designate whether the update is to be applied to the custom or the original data; and
- specify the TTools-format airport file using *TTools Airport File for Update Select*.

The first airport in the file, or the first one not already in *AirportList.dat*, as applicable, will be loaded into the editor. If the airport was previously in *AirportList.dat*, the full record as updated by the TTools-format file data will be shown. Make whatever further changes are necessary and then click on the *Apply Update* button. Scroll forwards or backwards through the airports in the file meeting the *Update Existing* criteria using the *Next Airport* and *Previous Airport* buttons. If you wish all airports to be processed a single step without any manual updates, simply click on *All* after checking one or both of FS9 and/or FSX (the same selections will be used for all airports updated using *All*). Please note that once you click *All* and the related processing is complete, all airports in the file are then known to the system; hence, the *Update Existing* checkbox must be checked in order for any airports to be displayed.

As with *Edit/Add Airport*, when finished click on the *Save File and Exit* button. To exit without saving, click the *Exit* button.

Whenever an airport is entered or updated using the bulk update feature, a record of the change is saved in a file named *AirportList – Updates.dat*. If *AirportList.dat* ever needs to be regenerated, these updates will be applied automatically

9 SUPPORT

AI Flight Planner's support forum is located in the "Tools support" area at <http://fsdeveloper.com>. Please direct your problem reports, suggestions for improvement and other comments there. When you report problems, please include relevant details. In particular, the version number, the exact error message and a summary of what you were doing at the time are likely to be particularly helpful

I have also creating a support website at <http://members.shaw.ca/aifp>. The most recent release of AI Flight Planner will be available from that site.

While I can't promise to resolve every issue you report or include every feature addition you propose, I will undertake to support and enhance AI Flight Planner in a manner consistent with it becoming and remaining the AI Flight Planning tool of choice for Microsoft Flight Simulator.

10 ACKNOWLEDGEMENTS

I wish to thank the following people whose prior efforts or contributions made AI Flight Planner possible:

- Lee Swordy for TTools
- Winfried Orthmann for bglAnalyze (which is used in the generation of the airport lists),
- Peter van der Veen for AIFPC and for permission to include its traffic file compiler/decompiler in AI Flight Planner,
- Karl Swartz for his Great Circle Mapper website (<http://gc.kls2.com>) which seems to be the most comprehensive source of airport time zone data available at no cost
- my beta team, especially Martin, Colin, John and Jim (FS Developer user names "gsnde", "colin_d", "JohnG" and "jvile" respectively) who expended a lot of effort in testing AI Flight Planner and/or advising me, and
- Arno Gerretson for hosting the AI Flight Planner support forum on FS Developer.com.

11 END USER LICENSE AGREEMENT

You are granted a free, non-exclusive right solely to install and use AI Flight Planner on your computer system(s) for your own personal enjoyment and, subject to what follows and the rights of others, to use and distribute flight plan and aircraft data in TTools text or MSFS compiled format files created or modified with AI Flight Planner ("derivative files").

You may not:

- upload AI Flight Planner to any file distribution system,
- reverse engineer, disassemble or decompile any part of AI Flight Planner,

- incorporate AI Flight Planner in whole or in part into any commercial product or facility, "shareware" or "freeware", or
- use AI Flight Planner or any derivative files in the development, marketing or support of, or incorporate derivative files in, any commercial product

without the express written permission of me, the author. Use for commercial purposes may be subject to a license fee.

In this end user license agreement, any reference to AI Flight Planner includes the files *AirportList.dat*, *Timezone.dat*, all other files included in the downloaded archive (.zip) file and any derivatives thereof.

Your use of AI Flight Planner is entirely at your own risk. I accept no liability whatsoever for any damage arising from its use no matter how caused.

By installing this software, you are deemed to have agreed to the foregoing.

AI Flight Planner - © 2008 - Don Grovestine