

AIRFIELD LIGHTS TOOLBOX – Version 3

Airfield Lights Toolbox (or AFLT for short) is a comprehensive utility for the creation of custom airfield lights and light arrays for Microsoft Flight Simulator and Lockheed Martin Prepare3D (collectively referred to as "FlightSim") airports. AFLT includes 3D models for the usual airfield light fixtures and approach lighting structure configurations from which you should be able to create just about any airfield light arrangement. Where the stock models are not sufficient for your purposes, you may modify them or add your own.

Lights for use with FS9 or FSX can be either BGL_LIGHTs, effects or a mix. AFLT creates appropriate light models on-the-fly and attaches those light models to the applicable 3D model from its stock models or your replacements.

Unfortunately, with P3Dv2, Lockheed Martin discontinued support for certain technologies used in AFLT Version 1. As well, P3Dv2's handling of effects left a lot to be desired; P3Dv3 is much better in this regard. So, for the most part AFLT does not support P3Dv2.

For P3Dv3 and later, AFLT Version 2 generates simobjects on-the-fly to which are attached appropriately-coloured effects. These simobjects are controlled using SimConnect. Nonetheless, the generation of lights is handled in the same manner as for FS9 and FSX. The use of simobjects has a substantial adverse impact on FPS. But, since the recent versions of P3D have eliminated the legacy technology on which scenery-based AFLT lights are based, there is no alternative. P3Dv3 may be problematic, depending on the number of lights; P3Dv4, being 64-bit based is much better.

AFLT Version 3 introduces a new concept. It permits the use of simple 3-plane lights (pioneered by Christian Bahr) that grow in size as the user aircraft gets further from the airfield. (Effects are still available.) Users can specify an approximate visible range for these new lights. While still FPS-intensive, the new lights give a substantial improvement over effects. Instructions on adjusting light characteristics are given in Appendix "D".

For simplicity, references in this manual to FSX apply equally to FSX, FSX-SE and P3Dv1. The term "P3D" includes P3Dv3 and later versions. As noted above, P3Dv2 has "difficulties" that preclude custom lighting with AFLT.

INSTALLING, EXECUTING and UNINSTALLING AFLT

Compatibility – AFLT Version 3 may be used with AFLT Version 2 data. The only significant difference is that, in Version 3, every 3-plane light style has a user-specified visible range. If no range is specified, as would be the case with Version 2 data, a default range is used.

Installation (All Versions) - To install AFLT, simply copy the files and folders from the downloaded archive into a folder of your choice (referred to as "the AFLT folder"). If you plan to use AFLT with FS9 or FSX, copy the contents of *AFLT\Effects\FS9* or *AFLT\Effects\FSX_P3D*, as applicable, from the archive into the applicable FlightSim Effects folder.

AFLT does not affect the system registry. It is a Microsoft .NET Framework 4.5 application. If .NET Framework 4.5 or later is not already installed on your computer, the "redistributable" can be downloaded from the Microsoft website at no charge.

Execution - To execute AFLT, double-click on AFLT3.exe.

Depending on your computer configuration and your Windows UAC setting, users may require administrator privileges when running AFLT. This is almost a certainty if either FlightSim or AFLT is installed on your C: drive. If you need but do not have administrator privileges, Windows will "complain" and you will not be able to access the necessary files. To run AFLT with administrator privileges, right-click *AFLT.exe*, select "Run As ..." and then "administrator". You can also specify Run as Administrator in a shortcut.

Windows 7 users may wish to run AFLT in the XP compatibility mode. Running it otherwise may result in a "this program may not have installed correctly" message when AFLT is shut-down. Despite the error message, there is no known problem - other than the annoyance factor.

Initialization - When you shut-down AFLT for the first time, an additional file, *AFLT.ini*, is created and saved to the AFLT folder. AFLT "remembers" key settings from one session to the next in this file. The next time you run AFLT, those settings are restored.

FSX DX10 Preview Mode – AFLT is generally compatible with FSX DX10 Preview Mode – with one exception. When you pan around the airfield or approach from different directions, whole blocks of custom lights disappear and, eventually, re-appear. This is a known deficiency of FSX DX10 Preview Mode and affects several types of lights. To correct, this you must have the applicable ..._LightField.bgl from Steve Parson's DX10 Fixer in the \scenery folder to which you save the AFLT lights and his fx_dx10field.fx effect in your Effects folder. Note that when you uninstall the DX10 Fixer libraries, that effect is deleted from the Effects folder, so you may have to replace it.

Alternately, use custom effects with FS9 models for DX10 Preview Mode

Automatic Updates - Whenever AFLT is started, it checks the support server to determine if a more recent release is available. If so, it will download that release for you. The updated release must be manually installed in the normal manner.

If you decline an update, you will be asked if you wish to be advised of future updates. If you decline, the "No Automatic Updates" item in the *AFLT.ini* file will be set to "True". To reinstate automatic update checking, manually edit *AFLT.ini* to set this item to "False".

Un-Installation - To uninstall AFLT, delete:

- the AFLT folder and all its contents, and
- any effects whose name starts with "fx_AFLT_" from your Effects folder(s), and
- for P3D, remove the *AFLT for P3D* folder from *Documents\Prepar3d v_ Add-ons*.

Airfield Light Size – The stock texture files that control the size of AFLT's BGL_LIGHTs results in lights that some, including me, think are too big and bright.

The AFLT archive includes replacement textures that will reduce the size of BGL_LIGHTs generally. Please note, however, using these files will affect the lights at all airports, not just those where AFLT-generated lights are used. If you want to try them, copy the file:

- for FS9, *halo.bmp* into your *FS9\Texture* folder, or
- for FSX, *fx_2.bmp* into the *Effects\Texture* folder,

overwriting the stock files of the same name. Be sure to backup the stock files first in case you don't like what happens with these replacement files.

(I obtained *halo.bmp* from an attachment to a fsDeveloper.com post. Unfortunately, I have been unable to find that post again in order to give proper credit. So, thank you, whoever you are. The *fx-2.bmp* file in the archive is my own work product.) Feel free to re-tweak these files as you think necessary.

For simobjects or other use of custom effects as the light source, you may edit the AFLT stock effects to achieve whatever light characteristics (including size) you wish.

AFLT STOCK FILES

Airfields Lights Toolbox comes with everything you'll need to start creating all the usual airfield lighting elements.

- 3D base models (FS9 and FSX), effects and textures for the following fixtures are included:
 - bi-directional lights – medium and high intensity - runway lights
 - omni-directional taxiway edge lighting
 - dual-headed obstruction light
 - split light for runway thresholds
 - uni-directional approach lights in both standalone and 1-, 2-, 3-, 4- and 5-light configurations on typical supports
 - strobe lights (uni-directional and omni-directional) both standalone and mounted on a tower up to 30'/10m high
 - wigwags
 - PAPI/VASI in 2- and 4-light PAPI configurations (both right and left), 2- and 3-ganged VASI configurations and a standalone unit that may be used to create an array of any VASI configuration, and
 - aeronautical beacons - flashes red continually and, optionally, strobes during the daytime while flashing red at night.

They, together with their textures, are saved in AFLT's *BaseModels* folder. Each model is more fully described in Appendix "A"

- *.ini* Files – The INI folder contains an *.ini* file for each of the elements. These *.ini* files define the guid to be used when calling the element as well as “instructions” to AFLT for creating the element
- *Colors.txt* - a simple text file specifying the RGB value for each color available for use. Both the BGL_LIGHT and effect color and the color of the model's light lens are controlled by this file. The format is simple and self-evident from an

- *Types.txt* - a tabular text file specifying for each light type the base model file to be used, the position(s) of the attached lights relative to the attachpoint location and other "default" data. Its format is described in Appendix "B". If you add any base models, you'll have to make corresponding entries in *Types.txt* before you will be able to use them. Entries in this file MUST use a period (.) as a decimal Separator.
- Sample approach array definition files for the most common types of approach lighting. To use any of these files, you must first substitute appropriate numeric values for latitude, longitude, elevation, heading and, if used, an alphanumeric tag for the corresponding strings in the first non-comment line of the sample file.

LIBRARY FOLDER

Since the work products are likely to relate to an individual airport, as a first step you should specify a folder where AFLT is to save its work for each airport, called the "Library Folder". A good location is your development folder for that airport. But, it may be anywhere. Also essential to the Library Folder is an airport identifier and a path to the scenery folder where object libraries and textures are to be saved.

The Library Folder must be specified as the first step in the use of AFLT.

The Library Folder is specified as the top item on the Main Panel. You may leave this entry blank, in which case AFLT will create a Library folder in your AFLT folder; but doing so means the folder may contain remnants of work products from all your projects.

Normally, the project identifier would be the airport's ICAO. But, where you need two or more AFLT "projects" for a single airport, a single ICAO is not sufficient. Therefore, the identifier – entered into the Project ID textbox on the Main Panel - may be any unique character string. Files/folders associated with simobjects will be prefaced with this identifier.

Object libraries are always saved to your Library Folder. AFLT will also copy the object libraries and the associated textures to another folder of your choice – entered into the Scenery Folder Path textbox in the Library Folder section of the Main Panel. Normally, this would be a subfolder in FlightSim's AddOn Scenery folder. If it is, you need only specify the path relative to the Addon Scenery folder. But, you can have the object libraries and textures saved to any location, in which case you need specify a fully-qualified path. If the specified path ends with "\scenery", AFLT assumes a companion \Texture subfolder. If the path does not end with "\scenery", AFLT assumes a folder containing \Scenery and Texture subfolder. If either anticipated subfolder does not exist, AFLT will create it/them on the first save.

QUICK START

It is strongly recommended you read this manual from cover to cover, especially the section on SIMOBJECTS if you are creating lighting for use in P3D.

If you have completed the Library Folder specification and all you wish to do is place some runway and taxiway lights using stock objects, you are now ready to do so. Unfortunately, if you wish to place PAPI/VASIs or wigwags (see CREATING/EDITING LIGHT ELEMENTS) or approach or threshold arrays (see CREATING LIGHT ARRAYS), you've got some more reading and work to do.

To use an airport editor (such as ADE) to position your light elements, follow the instructions in the first few paragraphs of CREATING AN LIBRARY OF ELEMENTS. Alternately, create a .xml file containing "Scenery Objects" using the "call" guid(s) set out in the respective .ini file(s) for the elements of interest and copy that file into your airport's scenery folder.

You'll not only need to position your airport lights; you'll also have to disable the corresponding Flightsim stock lights.

Once that's done and you've recompiled your airport, click Make Library and select the Flightsim version. If you are using P3D, enter the path to the .xml file containing the element positioning information and airport definition. (Both data must be in the same .xml file.) Select the elements of interest from the FSX and/or FS9 model lists and click Create/Save library.

If you've done all this correctly, your AFLT light elements should be displayed the next time you "fly" to the airport of concern.

OPERATING OVERVIEW

The starting point for every light to be created with AFLT (generally referred to as an "element"), is a base model, i.e., a 3D model of the light fixture, or support or whatever. AFLT's stock base models include both FS9 and FSX .mdl files for all usual light types. These stock models should satisfy most of your requirements. Each of these models has a single "empty" attachpoint to which AFLT will attach one or more BGL_LIGHTS, effects or 3-plane lights. The positions of the lights relative to the location of the attachpoint in each base model are defined in *Types.txt*.

BGL_LIGHTs are very processing-efficient - more so than effects. BGL_LIGHTs together with simple 3D models have a minimal affect on FPS. However, the only control you have over the display of BGL_LIGHTs is their brightness. And, their maximum range is sometimes a problem – particularly when used in VASI/PAPI applications. Effects are much more flexible in that you can adjust their display in a variety of ways. But, of course, a bigger FPS "hit" is the price you pay for this added flexibility.

By default, when FSX encounters a call to display a BGL_LIGHT, it substitutes a similarly-coloured stock effect. These special-purpose effects are non-adjustable and have similar shortcomings as the BGL_LIGHTs they replace. Unfortunately, FSX does not permit nesting of attachpoints, i.e., one attachpoint being attached to another. Use of effects requires this capability. Consequently, for FSX scenery-based lights, effects may only be used with FS9 models. As described later, AFLT creates simobjects for use with P3D.

The size and colour of AFLT-stock effects is controlled by the .fx file for the effect. You may adjust these attributes and several others to suit your own requirements. The stock runway edge light effect must be oriented to the runway heading even though the element is classified as omni-directional. If you create your own 3D models that use other than stock effects, you must add that effect to the appropriate AFLT Effects subfolder(s).

Light elements are specified using the Main Panel (dialog) as described later in this manual. The data entered on the Main Panel is captured in an .ini file for each element. The .ini files, which are in plain text, are saved in the Library Folder's INI sub-folder and are named as the element to which they apply, i.e., *element.ini*. These .ini files are the starting point for the Make Library function.

AFLT supports a full range of pilot-controlled lighting, PCL, through the specification of on/off criteria using a system global variable (e.g., a COMS frequency or a transponder code) and, where appropriate, airport operating hours. With FS9 base models, if PCL is used during the daytime (for example, in low visibility conditions), in addition to the lights being on, the light lens "glow". (That's what the "day-glo" texture is used for.) Unfortunately, for technical reasons, this same capability is not available with FSX models.

Like any other scenery, in order to be rendered by Flightsim, the elements (other than simobjects) must be collected in an object library file. Those object libraries must be saved to the *lscenery* subfolder of the airport with which they are to be used and the associated textures in the *texture* subfolder. With P3D, a collection of simobjects is created and automatically saved to the *AFLT for P3D\Simobjects* sub-folder of the Library Folder – later transferred to the "use\Documents\Prepar3D v*n* Addon Files" folder. In either case, the Make Library function saves the object libraries, simobjects and textures for you.

An essential step in creating a custom lighting arrangement with AFLT is the generation of scenery object placement data in .bgl format. While such a file could be created manually as a collection of "<SceneryObject>" items in a .xml file that is later compiled, a more typical approach is to use an airport editing utility such as ADE. To do so requires that the utility first "know about" your custom light objects. With ADE, this is accomplished by loading the library .bgl generated by AFLT's Make Library facility into ADE's Library Object Manager. (If you are creating custom lights for more than one Flightsim version of an airport, you only need to load the library .bgl into ADE for one of them since AFLT will use the same guid for all airports.)

To create more complex light arrays like, for example, a runway end configuration or an approach lighting system, AFLT allows you to specify the contents of the array as a series of X/Y/Z positions relative to a base point specified by latitude, longitude, elevation (Z-axis) and orientation (Y-axis). These coordinates are included in text-formatted "def" file. A ".def" file may hold the specifications for one or more arrays. AFLT's Make Array function compiles a .def file into a .bgl file (the "language" understood by FlightSim and P3D) and generates a series of .ini files for use by the Make Library function. This .bgl file includes only the placement data for the array elements. The elements themselves must be included in your object libraries by running the Make Library function afterwards.

Orientation is an integral part of the specification of certain light element. So, even though two such elements are otherwise identical, if their orientation differs, they must be specified individually – e.g., wigwags of different runways.

On dialogs having Flightsim version selection radio-buttons or checkboxes, only those buttons or checkboxes corresponding to versions actually installed on the user's system are displayed. Consequently, what is displayed on your system may differ slightly from the illustrations.

SIMOBJECTS

While the specification procedure for the airfield light elements and arrays, and library generation is (virtually) identical for all versions of FlightSim, that's where the similarity ends. As noted previously, lights for P3D are simobjects controlled by Simconnect. This necessitates very different internal handling of the related files by AFLT.

AFLT saves simobjects and related files to the *AIFP for P3D\SimObjects* sub-folder of the Library Folder. This folder and all its contents are created automatically for you. Once you have created a project using simobjects, that folder will contain two subfolders named *Simobjects* and *XML*. The *AIFP for P3D\XML* subfolder of the library Folder contains the files that control the simobjects.

As well, AFLT creates (or adds to) a P3D Add-on Package (folder) and places that Add-on Package, named *AFLT for P3D*, in the appropriate *Documents\Prepar3D v n Add-on* folder. For details of a P3D Add-on Package, please refer to the Add-ons section of the SDK.

When you start Flightsim after installing simobjects, a small dialog containing two buttons: Connect and Disconnect is created and minimized. This dialog, which has no use normally (but cannot be suppressed) is primarily to assist you in testing and allows you to make changes to the lights without having to restart P3D each time. Once P3D's startup is complete, the airfield lights should appear and the Disconnect button will be highlighted. To "tweak" your lights, open the dialog using the icon in the taskbar, click Disconnect, make whatever changes are necessary and regenerate the simobjects library and reinstall. Then click Connect. The revised lights should appear. This dialog will disappear when you shut down P3D, or you may close it – which will have the same effect as Disconnecting.

To uninstall all simobjects, simply delete the Add-on package. If you have AFLT-generated simobjects for multiple airports and only wish to delete those pertaining to a single airport, you should delete from *Documents\Prepar3D v n Add-on\AFLT for P3D\XML* the .xml file whose name is prefaced by the airport ICAO (or other airport ID you may have chosen), and from *AFLT for P3D\SimObjects* the subfolder similarly named.

SimObjects called by any file in *AFLT for P3D\XML* **MUST** be present in a corresponding *AFLT for P3D\SimObjects* sub-folder. Otherwise, the speed of execution of P3D may be affected. The user will be notified of any missing simobjects. AFLT will not create control information for simobjects not created at the same time.

Be aware that the 3D light bases are handled as scenery objects even when the lights are simobjects. Unfortunately, the two are not always displayed by P3D in the exact same place even though the same geographic location is used for both. The difference may only be a few (simulated) millimeters, but that's enough for the light to be positioned within the 3D models of approach lights, PAPI/VASIs and wigwags - and, hence, less visible - instead of in front of the model. Should this happen to you, the situation can be corrected by applying a Y-Bias, i.e., moving the light further in front of the 3D model. The bias is entered in centimeters in the middle section of the Main Panel. 1 cm. is often enough.

EFFECT CONSIDERATIONS

All the effects used by AFLT reside either in AFLT's *Effects* folder (FS9, FSX and P3Dv1) or in P3D's add-ons folder (P3Dv3 and later). Their names are prefaced with "fx_AFLT_" and reflect the light type and color to which they apply. The attributes of these effects may be changed to suit your specific requirements, but it is recommended such editing be undertaken only by experienced users.

AFLT uses Type 25 effects for FS9, FSX and P3Dv1 (notwithstanding the suggestion in the SDKs that Type 19 should be used for lights) since the latter seem to have very limited range. Later versions of P3D use Type 19 effects with a special texture to extend their visible range.

CREATING/EDITING A LIGHT ELEMENT

Light elements are created using the Main Panel, shown below.

The steps are as follows. They can be performed in any order after the first three.

- Specify the Library Folder where the elements are to be saved, by one of:
 - using the Select button (only the top folder need be specified; AFLT will look after the sub-folder organization),
 - selecting one of the previously-accessed libraries from the Library Folder combobox items, or
 - typing into the combobox's text area

When you restart AFLT, this field will be initialized to the last-used Library Folder.

- If simobjects are to be created, enter the airport ICAO (or other unique identifier) in the designated field. It will be used to identify the components associated with the specific project. You will be unable to create the simobject library unless the ID has been specified.
- Scenery object libraries should be saved in the same `\scenery` folder as the airport `.bgl` with which they are to be used, and the required textures in the companion `\texture` folder. AFLT assumes that `\scenery` subfolder will be in Flightsim's *Addon Scenery* folder. Enter the path (relative to *Addon Scenery*) to the folder containing the `\scenery` and `\texture` sub-folders where the scenery-based light libraries and arrays are to be saved.
- Click the New button associated with the Element combo-box and enter a name for the new element, or select an existing element using the combo-box. If an existing element, the panel will be initialized with the last entered value for each parameter. If you need to change the name of an element after initial entry, do so

Airfield Lights Toolkit - Version 3.1.01

Library Folder
D:\Scenery Development\CYYJ\AFLT Select

Project ID
CYYJ

Scenery Folder Path (may be relative to Addon Scenery)
CYYJ (Dev)\CYYJ\Scenery

Pilot Control
Control Control Freq./Code Auto-Off (min.) 0

Airport Operates
☒ 24 hrs. --OR-- From 0000 Until 2359 UTC Offset (hr.) 0

SimObject Y-Bias (cm.)
Approach 0 PAPI/VASI 0 Wigwag 0

Make Library Make Array

Element
New PAPI_27 Light Only
Change Name Effect BGL_LIGHT # of Lights 3 Kill Shadow
Guid e914ee844c042684eb11d09732f18c90 Format FS9 Std. Att. to Scenery

Light Type PAPI-Left Color Front Twy-Green Back

Strobes Sequence of 8 16 24 Seed Limit Visibility

Orientation Runway/User-Facing Heading (deg.) 286.15 Slope from Horizontal (deg.) 3

Visibility Light Spread - Horizontal (deg) 60 Vertical 0

LOD-Variable Light Size (m.) - Min. Default Max. Default Range (nm.) Default

Create/Save Element

Main Panel

- Select the model type from the Light Type combo-box.
- If the Color combo-boxes are enabled, select from the available colors. (The content of these combo-boxes is determined by the *Colors.txt* file. If neither combo-box is enabled, a standard color will be used (e.g., amber for wig-wags, red and white for VASI/PAPI, etc.)
- If the light type is a Strobe, specify:
 - the flash sequence as 1-8 of 8 or 1-16 of 16 or 1-32 of 32, and
 - whether the visibility is to be limited (default is omni-directional) and

- a "seed" value (1-8) which notionally controls the firing of the first flash in the sequence ("running-rabbit"). Each strobe in a running-rabbit must be given the same seed value. A different value should be used for each running rabbits at an airport so they do not have identical sequence start times.
- If the model type is Beacon, the Strobe fields will be replaced with the following:
 - the period of the blinking light in seconds, i.e., the time between the start of two successive "blinks",
 - the on-time of the red light, in seconds (0 for strobes only),
 - the number of strobe flashes/minute, and
 - an arbitrary start seed value (1-8) to avoid all beacons flashing/strobing in unison.

Strobe-only and start-seed parameters are effective only for P3Dv3 or later versions.

- If you wish the visible range of your light to be other than the default value, specify the desired approximate range in nautical miles.
- Specify whether the light source is to be a BGL_LIGHT or an effect by checking the applicable radio-button and, if the former, also specify the number of BGL_LIGHTs to be used at each location. Normally, this will be 1, or 2 for strobes. But use of more may be necessary in special applications where extra brightness is required. (Single BGL_LIGHT brightness is set by the RGB values in *Colors.txt*.) Appropriately-colored effects are always used for simobjects, irrespective of this specification.)
- Enter the Orientation and Visibility Limit parameters when those text boxes are enabled. (For new elements, the Visibility Limits, where applicable, will be initialized to default values from *Types.txt*.) If these text boxes are not enabled, the selected light type has no use for such data.
- Check Kill Shadow if you do not need a shadow. Otherwise:
 - if a custom shadow .mdl file exists in the folder along with the selected base model (its name will be suffixed with "_Shad"), it will be used (custom shadows may be used to improve processing efficiency), or
 - if no custom shadow file exists, the shadow incorporated in the base model file (if any) will appear.

(Shadows consume processing resources. If a shadow isn't necessary - like, perhaps, for an obstruction light which is generally well away from the user aircraft - "kill" it. Shadows should always be suppressed for lights on towers/supports where part of the tower/support is buried. Otherwise, FlightSim will display a shadow for the below-ground portion as well.)

- Finally, click the Create Element button. AFLT will place a file in the library's INI folder specifying the elements characteristics. AFLT will also create a test model to ensure all required resources are available. If all is well, a confirmation message will be displayed for a short time.

In certain situations, it may be desirable to suppress the display of the 3D model. (This is automatic for elements designated as "Surface"). This would be the case for

approach lights that extend onto the runway or, perhaps, the last in the sequence of ODALS for a displaced threshold. To accomplish this, check Light Only.

As mentioned previously, FSX does not permit nesting of attachpoints. But, an effect-based light source must be “attached” to a FSX base model.. Consequently, if an FSX base model is called as an attached object (like an obstruction light object), the light would not display – since such an arrangement would constitute nesting. To avoid this, elements may be specified as “Attached. To Scenery” (by checking the Att. To Scenery checkbox), in which case a FS9 base model with an integrated the light source is used.

You may also edit any existing light – whether created from the Main Panel or as an array element (see Creating Light Arrays below) by selecting its name from the Element combo-box. The name of array elements is prefaced with “%”. Note, however, changes to array elements are only effective until the next time you Make (that) Array at which time the specification of that element will revert to that in the applicable .def file. So, any changes to array elements that are to be permanent must be made to the .def file.

To delete an element you no longer need, locate that element’s .ini file in the AFLT folder “INI” and delete it.

PILOT-CONTROLLED LIGHTING and AIRPORT OPERATING HOURS

While specified on the main dialog along with the light elements, pilot-controlled lighting (PCL) is an airport attribute and applied to all elements in the library folder. If PCL is required at this airport, select a pilot-controlled lighting control from the Control/Variable combo box (radio/navigational systems). If COMM is selected, please note PCL is only triggered if the COMM unit then-currently selected for transmit is tuned to the specified frequency.

Then, specify the “lights-on” value for the control/variable in the Value for Lights On textbox. Enter the value exactly as the value will appear on the control (e.g., 1234 for the transponder or 123.70 otherwise).

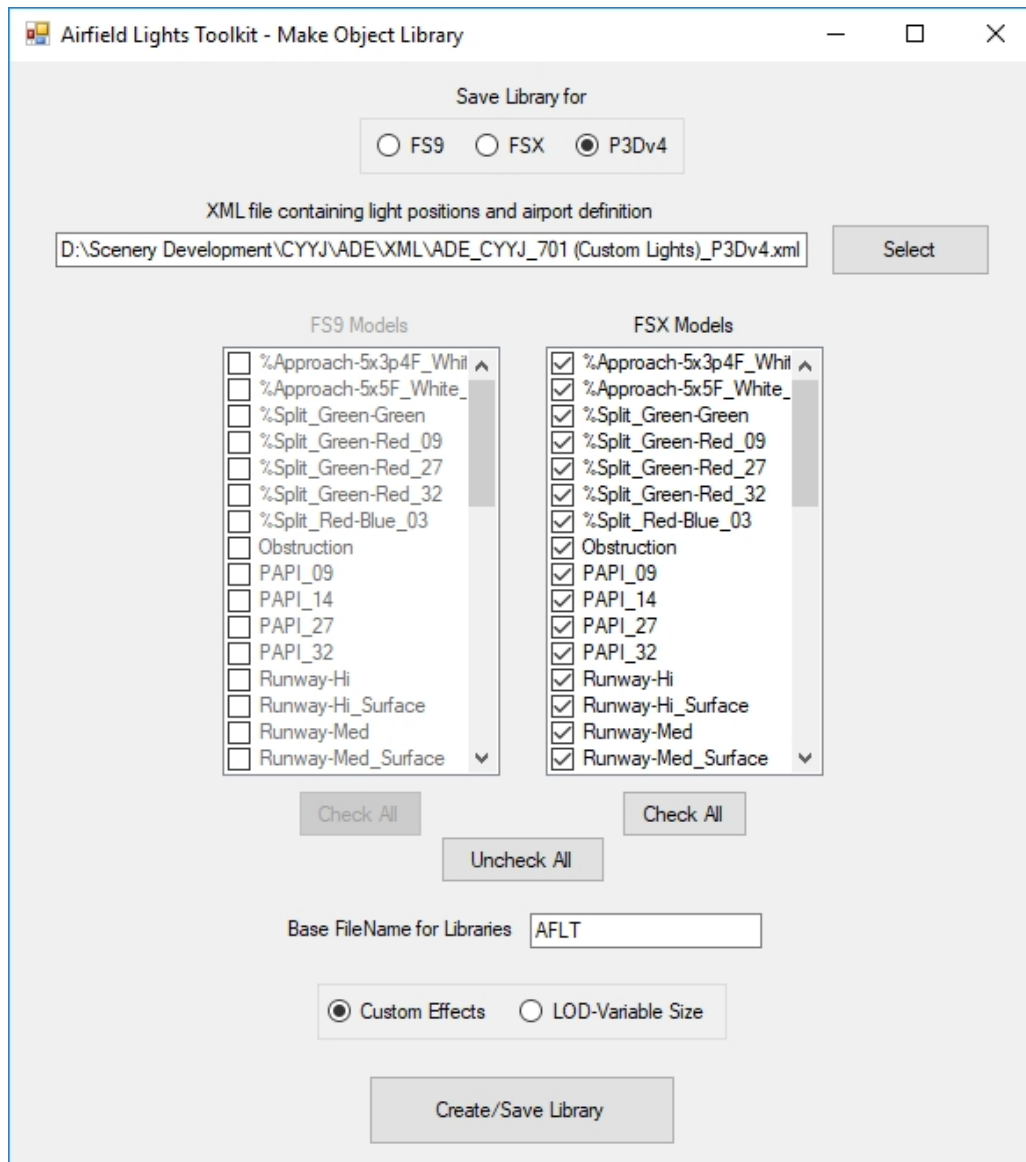
If you want to be able to deselect the PCL lights-on criteria but have the lights remain on, enter the number of minutes after which the lights will automatically be turned off in the Auto-off (min.) text box.

For airports having 24 hr operation, check the 24 hrs. checkbox. Otherwise, enter the operating hours (i.e., the hours outside of which the lights will normally be off) in local time and the airport’s UTC offset in hours (decimal hours if necessary). **Please be aware, pilot-controlled lighting based on airport operating hours is “expensive” in processing overhead and should be used only if required.**

There is no save control for PCL parameters or airport operating hours. They are saved automatically to the Library folder. When object libraries are created, the then-current (displayed) PCL parameters are applied. So, to revise your pilot-controlled lighting scheme, simply specify the new PCL parameters and regenerate the applicable object libraries using the Make Library function.

CREATING A LIBRARY OF ELEMENTS

Click the Make Library button on the main panel and you'll be presented with the dialog below. The FS9 Models and FSX Models list boxes contain all the available elements for that version. Elements prefaced with "%" are automatically generated array elements.



Make Library Dialog

If you plan to use an airport editor to place your light elements, you must first create an object library to inform that editor of the element guids. After the involved light elements, are available (having created new ones or edited stock elements as may be necessary) click Make Library, select the Flightsim version of interest and then click Create/Save Library. Ignore any warnings about no airport data or .xml file. The object library, named “_Bases.bgl” or “_Lights.bgl”, prefixed by “AFLT” (unless you changed the Base Filename for Libraries), will be saved to your Library Folder as well as the location you

specified as the Scenery Path Folder on the Main Panel. (That latter file will be overwritten later.) Load that library into your airport editor. Unfortunately, ADE only recognizes FSX or P3D object libraries. So, if you only have FS9, you'll have to enter the guids manually.

If you are compiling for P3D and your library is to include lights the positions of which are specified in an airport (or other) .bgl file of your making, you must first specify the path to the .xml file from which the .bgl was created, i.e., containing the "SceneryObject" code for the custom lighting elements AND the airport definition, from which AFLT will extract the data to locate and control the simobjects. If you are in the habit of having ADE (for example) separate the airport from the scenery objects and make two .bgl files, you'll have to recompile the airport to create the required single .xml file. (If you have programmed arrays, the related positioning information will have been generated when the arrays were compiled.)

Irrespective of the version of Flightsim for which you are creating an object library, if you wish to place the elements using an airport editor such as ADE, you must first "tell" that editor about the objects you wish to place. You may be able to do this manually, one-by-one, entering the respective "call" guid(s) found in the respective .ini file(s) for the element(s) of interest. Alternately, you may create an initial object library with ADE and load that library into your airport editor. In the case of ADE, you would use the Object Library Manager to do so.

Click the radio-button for the FlightSim version with which the library is to be used. If you click a P3D radio-button, the FS9 model listbox and associated controls are disabled, since FS9 models may not be rendered properly by P3D. Similarly, if you click the FS9 button, the FSX model controls are disabled. All models in one of the lists will be checked. Uncheck any models you do not want.

For FSX, both model lists are enabled and all the models in the FSX list are checked. But due to the "no nesting of attachpoints" issue previously discussed, the stock Obstruction Light and any other All in One elements will be checked in the FS9 models list. These elements will not even appear in the FSX models list. If you wish to use the FS9 base model for other elements, check the item in the FS9 list. The corresponding item in the FSX list will be automatically unchecked. (Similarly, items in the FS9 list are unchecked when you check their FSX counterparts.) To make this task easier, the lists are sequenced identically with blank spaces for base models that don't exist or can't be used for that version. As well, both list-boxes scroll together.

By default, object library files are named starting with "AFLT_". If you wish to name your files otherwise, enter another filename prefix into the Base Filename for Libraries textbox.

Individual light elements will have been defined to use BGL_LIGHTs (FS9/X only) or effects. The group-box near the bottom of the Make Library dialog allows you to override this initial specification in the object library about to be generated. For FS9 and FSX, you may use custom effects in place of BGL_LIGHTs (and their FSX equivalent). For P3D, which uses custom effects by default, you may instead use LOD-Variable-size lights (i.e. 3-plane objects that increase in size as the LOD decreases).

When all desired items have been selected, (don't forget to include array elements) click the Create/Save Library button. When you do so, AFLT will confirm the path to the folder where the files will be saved. That's all there is to it!

While an object library including FS9 objects is being generated, you may notice a small dialog rapidly flashing on and off. This is the FS9 compiler being called repeatedly for each element. Unfortunately, there's no ability to suppress this.

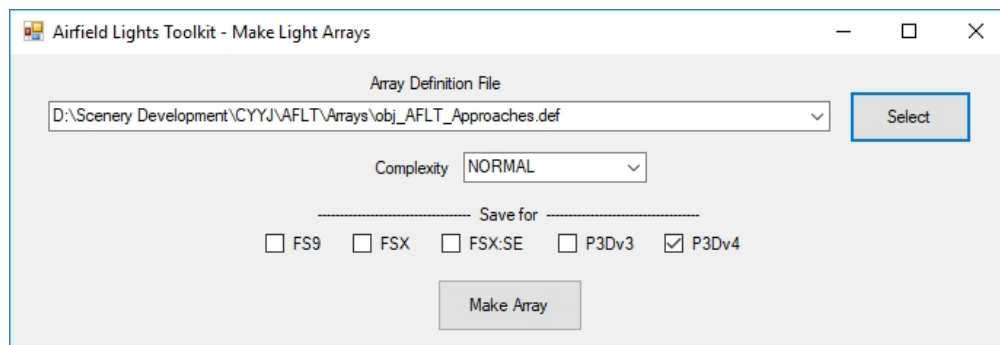
AFLT will copy the object libraries or simobjects, as applicable, and associated textures to the proper folders. Please note when creating custom models, AFLT requires textures for the base models to be in the *Base Models\Textures* folder. If the required textures are not found there, an error message will be issued you'll have to transfer the textures manually. The color parameters of any effects utilized by the library are automatically updated to reflect the RGB component of the specified color in the Colors.txt file.

If no errors, a confirmation message is issued. Any required textures not found in the designated texture folder (if specified) will be listed in place of the confirmation message. The object library ".bgl" files (one for lights and depending on the selected override, if any, also one for each version of the bases) are also saved in the Library Folder. So it's easy to transfer them later if, for instance, the intended add-on scenery folder doesn't exist when you create the library.

Please note. Neither FSX nor P3D supports nesting of attachpoints involving effects and will not render an attached effect if the FSX or P3D object to which the effect is attached is itself an attached object. An example is the Obstruction Light, which is usually attached to a scenery object. With FSX, only the FS9 version of the obstruction light can be selected. (FSX supports attachment of FS9 models). Unfortunately, this alternative is not available with P3D since FS9 scenery models have not been supported since v3.

CREATING LIGHT ARRAYS

Creation of light arrays requires an array definition (".def") file.



Make Array Dialog

A ".def" file is a text file setting out a reference point for one or more arrays and, within each array, the location of each light element relative to that *reference point*. The format

of array definition files is described in Appendix "C". These files are saved in the *Arrays* subfolder of your Library Folder.

Whether you create your own array definition or modify one of the samples, you must determine the geographic location to be used as the *reference point*. Usually, the reference point will be the center of the threshold of the runway to which the array elements pertain. You can determine this location using an airport editor, such as ADE. If you are enhancing a stock airport, position the user aircraft over the center of the threshold and record the latitude, longitude and elevation using a utility such as TCalc or SlewMaster, the latter by the author.

To add one or more light arrays to your airport, once the necessary .def file(s) exist, click on the Make Array button on the main panel. The Make Array dialog will open.

To generate an array, select:

- the name of the ".def" file to be used,
- scenery complexity, and
- the Flightsim versions for which the array applied, and

click Make Array.

For scenery-based lights, AFLT will save the resulting .bgl file to the *Addon Scenery* subfolder specified on the Main Panel. For simobjects, AFLT will create a custom .xml file and save it to the *AFLT for P3D/XML* sub-folder in your Library folder.

As during library creation, while the array is being generated you will notice a small dialog rapidly flashing on and off. This is the compiler. (Each element in the array is test-compiled to ensure all required resources are available.)

The supplied sample ".def" files assume nominal, uniform spacing between the various light elements with all elements placed at or just above runway elevation. Certainly for test purposes, and probably also for production, this spacing will be adequate. However, in the real world, there are numerous reasons why nominal spacing isn't used. As well, depending on how far your flatten extends beyond the ends of the runways, the elevation of the approach lights may need adjustment. It's up to you to decide whether the default positioning is adequate and, if not so, to edit the .def file.

Once you have generated the array, create a library (i.e. Make Library) including the array elements and "fire-up" FlightSim. If things aren't exactly as you intended, don't panic. Review your array definition (all the answers will be there), correct it and regenerate the array and library file.

SPECIAL CIRCUMSTANCES

Simobject-Based Lights Displayed Inside 3D Models – As noted in the SIMOBJECTS section above, you may find with some of your approach lights, PAPI/VASIs and wigwags the simobject-based light is displayed within the 3D model instead of in front of it. To correct this situation, add a Y-bias for the category of elements affected using the controls in the middle section of the Main Panel and re-make your object Library.

VASI - Siting of VASI elements will require a little extra work on your part. VASI configurations are airport dependent and comprise two independent (for the purposes at hand) sets of elements, a downwind bar(s) (closest to the runway threshold) and an upwind bar(s). You'll have to make and place them individually.

The downwind bar is normally aimed 0.5 degrees below the intended glideslope. The upwind bar is aimed along the glideslope. The distance between the upwind and downwind bars, which may vary between 500 and 1000' (150-300m), is airport dependent. The nominal separation is 700' (215m). In these circumstances, you'll need to use two VASI2 or VASI3 models.

Alternately, you could create an array in which all 4, 6 or whatever number of elements are used, the elements being placed automatically.

Obstruction Lighting - Please note, for light type Obstruction, two base models are provided – an FS9 model, named “Obstruction” and an FSX model solely for use with P3D, named “Obstruction_P3D”. For both FS9 and FSX, if you wish to implement obstruction lights, you MUST use the FS9 model “Obstruction”. For P3D, you MUST use the FSX model “Obstruction_P3D”. Although you may select Obstruction_P3D for use with FSX, it will not illuminate.

Lights in Close Proximity - It appears that FS9 has difficulty initializing the rendering of certain lighted models in very close proximity when approaching from a distance, resulting in a “crash”. Given the complete absence of diagnostic information at the time of the “crash”, the precise cause of the problem is not known. FSX does not appear to be similarly affected.

In the case where the problem was discovered, when the light source was BGL_LIGHTs, two strobes 2.5'/0.7m apart triggered the crash, but two others 5'/1.5m apart were OK, suggesting a minimum separation of 3.5'/1m. When effects were substituted for the BGL_LIGHTs, the latter two strobes also resulted in a crash, indicating FS9 is less tolerant of effects. On the other hand, steady lights as close at 1'/0.3m were rendered without difficulty, suggesting that blinking lights, e.g., strobes and, beacons. are most susceptible.

So, should you experience a unexplainable FS9 crash when approaching an airport with AFLT custom lighting, review the spacing of the lights - especially strobes and beacons.

SUBSTITUTION/ADDITION OF BASE MODELS

You may replace the provided base models and add new light types to AFLT. But, before doing so, you should examine the stock textures *obj_AF_Lights*, *obj_AF_Lights_Dayglo* and *obj_AF_Lights_LM* (found in the *Base Models\Textures* folder) to appreciate how they are used and their inter-relationship. Unfortunately, “dayglo” textures cannot be used with FSX models.

Other than for wigwags, the only firm requirement is that a custom base model contain a single, empty attachpoint. Both the attachpoint and the model can be named as you wish. You may use any texture for the model and lens – or different textures for each. For FS9 models, if you use the stock texture *obj_AF_Lights.bmp* for the light lenses, AFLT will automatically apply *obj_AF_Light_Dayglo.bmp* whenever the lights are on. If

you use your own texture, AFLT will look for a file (.bmp or .dds) named "*texture_Dayglo*" and use it when the lights are on. If it doesn't find a "dayglo" texture it will complain, but the model will still be useable (however, of course, the lens will not "illuminate" in the daytime). If you provide it, a "_LM" texture will be applied as usual.

For custom wigwag models, in addition to the attachpoint, the two light lens must be:

- a separate element of the model,
- displaced negatively in the Y-dimension sufficiently far from other portions of the model to avoid flickering,
- displayed unconditionally (no LOD assignment), and
- textured with a file whose name ends with "_WigwagLens".

For FS9 models only, AFLT will alternately apply the noted texture and one whose name ends with "WigwagLens_On" – which should be more brightly colored than *texture_WigwagLens* but otherwise identical. Please refer to the stock "WigWagLens" textures.

It is recommended the attachpoint be placed as follows:

- X-axis:
 - if a single light is to be shown (e.g., a runway edge light), at the location of the light; or
 - if multiple lights (e.g., an approach lighting tower), at the "centre" of the lights.
- Y-axis:
 - for omni-directional lights, at the center of the modeled lens;
 - for uni-directional lights, just in front of the lens.
- Z-axis:
 - the elevation at which the light is to be displayed, for lights placed on the airport proper, (runway/taxiway edge lights, runway end split lights and approach lights using in threshold bars, PAPIs, VASIs and wigwags), and
 - at the model reference point (0/0/0) for lights to be placed remotely (approach light standards, strobes, beacons).

For runway/taxiway lights especially, which will be numerous, should you need to raise or lower them you can do so by modifying the model rather than the individual placement data.

Place each version (FS9/FSX) of your model in the applicable Base Models folder and, if you use non-stock textures, place a copy of your new texture (32 bit, DXT and/or DDS formats as necessary) in the *Base Models\Textures* folders.

If you are simply replacing a stock base model (using the same name) and your attachpoint is in the same location as in the stock model, you need do nothing further. However, if you have moved the attachpoint, have renamed the file or are adding a new base model, you will have to make a new, or update an existing, entry in Types.txt. (Please see Appendix "B").

DISTRIBUTING YOUR AIRPORT

You are permitted to distribute AFLT-generate custom lighting with your airport so long as your airport is made freely available and its distribution and use does not involve an end-user paying a fee of any kind.

Distribution of your airport with scenery-based lights does not require any special action other than ensuring for AFLT's effects are available. With simobjects-based lighting, you need only copy the Add-on package, but your users may need instructions for setting up the "user\Documents\Prepared v\Add-on Files" folder.

SUPPORT

Airfield Lights Toolbox's support forum is located in the [Scenery Design - Airport Design support area at http://www.fsdeveloper.com](http://www.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 AFLT version number, the exact error message and a summary of what you were doing at the time are likely to be particularly helpful. If you are asking for support with a particular lighting issue, please attach the complete Library Folder. Without it, it's unlikely I can be of much help.

I also have a support website at <http://stuff4fs.com> for all my airports and development utilities. (Navigate to the User Applications / AFLT page.) Among other things, the site lists all known problems with the latest release. The most recent release of Airfield Lights Toolbox is available from that site as are occasional development releases of new features.

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 Airfield Lights Toolbox in a manner consistent with it becoming and remaining the tool of choice airfield lighting for Microsoft Flight Simulator.

Enjoy,
Don Grovestine
don@stuff4fs.com

END USER LICENSE AGREEMENT

As used in this end user license agreement, the terms "Airfield Lights Toolbox" and "AFLT" shall be construed as encompassing the full contents of the downloadable archive (.zip) file created and/or posted for distribution on "download sites" by the author, including without limitation: the executables file AFLT.exe, AFLT(exe).exe and AFLT (Install).exe, the associated user manual, the files Colors.txt, Types.txt and the sample array definition files, any derivatives thereof and certain files proprietary to third parties including but not limited to the files BGLC_9.exe, bglcomp.exe, bglcomp.xsd and Ionic.Zip.Reduced.dll.

You are granted a free, non-exclusive right solely to install and use Airfield Lights Toolbox on your computer system(s) for your own personal enjoyment and, subject to what follows and the rights of others, to use and distribute work products created or

modified with Airfield Lights Toolbox (“derivative files”) as described in the section of this manual entitled DISTRIBUTING YOUR AIRPORT WITH SIMOBJECTS.

You may not:

- upload Airfield Lights Toolbox in whole or in part to any file distribution system,
- reverse engineer, disassemble or decompile any part of Airfield Lights Toolbox,
- incorporate Airfield Lights Toolbox in whole or in part into any commercial product or facility, “shareware” or “freeware”, or
- use Airfield Lights Toolbox 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 the author. Use as may be permitted for commercial purposes may be subject to a license fee.

Your use of Airfield Lights Toolbox is entirely at your own risk. You assume and are responsible for any and all liabilities and damages arising therefrom no matter how caused.

By installing or otherwise using any part of Airfield Lights Toolbox, you are deemed to have agreed to the foregoing.

Airfield Lights Toolbox - Copyright © D. Grovestine 2013 - 2018

Appendix "A" - Base Models Supplied

The following base models - both FS9 and FSX versions - are supplied and their use is defined in *Types.txt*:

Approach.mdl - uni-directional light for approach lighting systems threshold bars
Approach-n x spacing.mdl - approach lights mounted on 20' supports (which may be buried to the extent necessary) "n" is the number of lights mounted on the support and spacing is the nominal spacing in feet except for 4x2p5 and 5x3p4 where the spacing is 2.5' and 3.4' (40") respectively. The range of lights should be adequate to configure most approach lighting systems specified by the FAA.
Obstruction.mdl – (FS9 model only) dual obstruction light assembly on a 5' post
Obstruction_P3D.mdl – (FSX model only) dual obstruction light assembly on a 5' post
Omni.mdl - omni-directional light for taxiway and runway edge lighting application
PAPI.mdl - 4 light PAPI system. Same model used for both *PAPI_Right* and *PAPI_Left*
Split.mdl - bi-directional light for runway end applications
Strobe.mdl - strobe light mounted on a 5' pole
Strobe-T.mdl - strobe light mounted on a 30' tower
Tower_Beacon.mdl - a 50m' high tower. You may choose to have a blinking light atop it or a blinking light at night and a strobe during the day
VASI.mdl - single VASI
VASI-2.mdl - 2-unit VASI/PAPI with 16' separation
VASI-3.mdl - 3-unit VASI with 16' separation
WigWag.mdl - typical wigwag with two alternately-blinking amber lights

Elements on towers or supports may be set to any height. The unneeded portion of the tower/support will be "buried".

None of these models will cast shadows normally. Custom shadows (provided - model file name suffixed with "_Shad") may be applied to all models except *Approach-n x spacing.mdl*, *Tower_Beacon.mdl*, *Obstruction.mdl*, *Strobe.mdl* and, *Strobe-T.mdl*. *Obstruction.mdl* is generally used in locations where a shadow would add little realism. In the other cases, a portion of the model usually will be buried but would cast a full shadow anyway.

You may add to this set of base models or replace individual models. It is recommended you not change the names of stock elements. If you do, or if you add new elements, you will have to add or edit the corresponding line in *Types.txt*. Since models used in scenery are regenerated during library creation, to "retrofit" a model, simply edit *Types.txt* as necessary and rebuild the libraries that use it.

Appendix "B" - Types.txt Format

Each line in Types.txt describes a model type, as follows:

- Column 1 - Model Name - These names are loaded into the Light Type combobox at startup.
- Column 2 - Name of base model used for this light
- Column 3 - Light type. These names are predefined in the system:
- APPROACH - uni-directional light. The supported versions may be placed up to 10m. (30') above the ground.
 - BEACON - a simple blinking red light (approx. 30 flashes/minute) up to 50m (160') above ground level. Optionally, you may specify a strobe during the day
 - OMNI - omni-directional light in a simple housing
 - PAPI - a sequenced four unit PAPI in a typical housing
 - SPLIT - bi-directional light in a simple housing
 - STROBE - sequentially-flashing bluish lights up to 10m. (30') above the ground in a simple housing
 - VASI - white above/red below specified glide-path angle in a typical housing
 - WIGWAG - alternate blinking lights in a typical housing
- Column 4 - The number of BGL_LIGHTs to be used to determine brightness.
- Column 5 - Default horizontal spread of light(s) in degrees. A blank entry implies full (360 degree) visibility.
- Column 6 - Default vertical spread of lights, centered about light tilt if any. A blank entry implies full (90 degree) visibility.
- Column 7 - Minimum size of LOD-variable light
- Column 8 - Maximum size of LOD-variable light
- Column 9 - Visible range (in nautical miles) of LOD-variable light.
- Column 10 - "True" or "Yes" if light is subject to pilot control; otherwise "False" or "No"
- Column 11 - M (metres) / F(feet) indication pertaining to following column
- Column 12 - X/Y/Z offset(s) from attachpoint (in feet or meters). Values for each axis are separated by "/". If more than one light to be displayed, subsequent positions are separated by ";".

Appendix "C" - Array Definition Format

It is suggested you refer to one of the sample array definition files when reading this appendix.

Each array definition begins with "<" and must be terminated with ">", much like a SceneryObject in a scenery ".xml" file. The start and termination characters may be on a line by themselves or included (in the appropriate location) in one of the other lines. Comments (lines beginning with ";" or "//") may be interspersed between array specifications.

The first non-comment line (i.e., the "header line") of the array definition specifies the reference point for the array. It must include:

- latitude,
- longitude,
- elevation,
- heading, and
- an optional "tag" (identification string) which, if provided, must be unique for each array

all separated by "|".

Latitude and longitude can be in any of the standard formats (*DD.ddddddd*, *DD MM.mmmm* or *DD MM SS.ss*). Elevation may be in feet or meters but, if in feet, must be suffixed with "F". Heading (true) must be a number between 0 and 359.99. For approach lighting systems, use the runway heading. Be as precise as possible in all cases. The "tag", if entered, will usually be the applicable runway number - but it may be anything so long as it is unique. (The system-generated filename for an oriented array element includes a "tag", i.e., a unique identifier, to differentiate it from other similar, but differently-oriented, elements at the same airport.) If AFLT requires a "tag" for any element in an array and none is provided, the array reference heading will be used.

The definition of each element in the array occupies a single line. It comprises:

- Light Type, which may be suffixed with "_LO" if you wish the base to be omitted for, for example, runway centerline lights,
- the X/Y/Z position in meters or feet relative to the reference point except for type "Beacon" where the elevation is AGL, and
- supplementary data (where necessary), such as color, strobe sequence, etc.

Dimensions in feet must be suffixed with "F". Dimensions in meters may be suffixed with "M" for clarity. Y-values are the distances from the reference point when travelling along the specified heading. Thus, for an approach lighting system, if the associated runway heading is used, the Y value for all points prior to the threshold (when on approach) will be (-)ve whereas those on the runway or beyond the threshold will be (+)ve. X-values are (-)ve when left of the (extended) runway centerline and (+)ve if to the right when facing in the direction of the specified heading. Except where noted otherwise, Z-values are relative to the elevation at the reference point.

A ".def" file format would, then, be similar to the following:

```
; Title of Array 1
< latitude | longitude | elevation | heading | tag
element name | X-value | Y-value | z-Value | supplementary data
element name | X-value | Y-value | z-Value | supplementary data
element name | X-value | Y-value | z-Value | supplementary data | alternate tag />
```

Supplementary data need only include color data - and then only for elements for which color is specified as the first item below. (Colors must be one of the names specified in the Colors.txt file.) The remainder is optional and, if not provided, default data (as specified in Types.txt) will be used. Supplementary data comprising two or more items must be delineated by a forward slash ("/"). Individual items may be omitted but, unless all following items are also omitted, an empty field (i.e., "/") place-holder must be included for each omitted item.

The possible contents of the supplementary data field depends on the type of element:

<u>Light Type</u>	<u>Modifier</u>
Approach	Color / No. of BGL_LIGHTs (bulbs) / horizontal spread / vertical spread / tilt
Beacon	Beacon (red) period (sec.) / on for (sec.) – 0 for strobes only / strobes per minute / no. of bulbs
Omni	Color / no. of bulbs
PAPI/VASI	No. of bulbs / glide slope (deg) / horizontal spread / vertical spread / range (nm)
Split	Front color / Back color / no. of bulbs
Strobe	Sequence number / no. of flashes (8 or 16) / seed value / no. of bulbs / horizontal spread / vertical spread
all other types	Color / no. of bulbs

A final supplementary item, Range (in NM) may be added for any light type. If used, a blank field must be included for each default item noted in the previous table.

For strobes, the *seed value* is an arbitrary number (1-8) which determines the timing of the first flash of each set of sequenced strobes ("running rabbit") so that there will be slight variations in the timing from one running rabbit to the another.

For all light types, if you wish to kill the shadow, preface any other supplementary data with "K/", or just "K" if that is the only data in the field.

AFLT creates a name for each element in the array by concatenating the *light-type* and *color* or other distinguishing data and suffixing that with the *tag* or *heading* from the header line. This should be adequate in most cases. But, if you wish, you may specify an alternate tag for any item. It may be any character string.

AFLT includes sample ".def" files for approach lighting systems commonly encountered, including ALSF-2, SSALR, MALSR, MALSF and ODALS in its *Sample Defs* folder. Use these as your starting point. Unless you need to alter the Z-values because of the surrounding terrain, all you need do is specify the relevant values in header line.

The FAA recommends that the light plane of an approach lighting system be horizontal whenever possible. The sample ".def" files reflect this. However, we don't live in a perfect world. If you want your approach system model to closely reflect that portion of the real world it simulates, it may be necessary for you to establish the elevation of each light bar and enter it into the ".def" file. But, unless the slopes are drastic, the "default" plane should suffice.

Appendix "D" – Adjusting Light Characteristics

AFLT supports Three types of lights:

- BGL_LIGHTs
- Custom Effects
- 3-plane brightly-colored objects.

BGL_LIGHTs are rendered in FS9 only. The only adjustment available to users is to vary the number of “bulbs” in the element definition. In FSX, a light element specified to use BGL_LIGHTs is rendered as an automatically-created effect the same color as specified for the BGL_LIGHT. BGL_LIGHTs do not apply to P3D v3+, which default to custom effects.

The custom effects shipped with AFLT use a special texture with inverse-sized mipmaps, i.e., the relative size of the image in the mipmaps *increases* as the distance from the light increases, increasing the visible range of the effect. Some of these effects, in particular, the runway edge lights are bidirectional, i.e., limited visibility to the side. These characteristics may not be to your liking. Users familiar with editing effects are free to adjust any aspect of these effects, including choosing a different texture.

The characteristics of the 3-plane lights (other than color) are defined in Types.txt. There are three entities: minimum size, maximum size and range. As would be expected, the lights are rendered at minimum size at LOD_100 (LOD_600 in FS9) and at maximum size at LOD_1. Light size increases or decreases in increment of 10% as the user aircraft moves towards or away from the light. Types.txt may be edited with Notepad or any other text editor.

Unfortunately, LOD is calculated differently in P3D v3+ than in the earlier versions of Flightsim. In P3D, LOD is calculated as a multiple of the radius (RADI block) in the .mdl file., whereas in the earlier versions of Flightsim, LOD is a function of the rendered object size relative to the vertical pixel count of the display window. AFLT addresses this difference in technique as best it can. But, you may need to use different size and range values to get consistent results across all Flightsim versions. It can do no harm to experiment.