

# Scenery Workshop

## Kingman Example Airport

To get a better idea on how the whole process of creating an airport for Aerofly FS 2 works, we have provided the airport 'Kingman' with all required source files. Before creating your airport, the process of exporting via export plugins and then converting using the Aerofly FS 2 Content Converter should be tested. But before we export our Kingman airport we will briefly describe what the scenery definition file does, and we will summarize the different scenery mesh classes that our Aerofly FS 2 engine supports.

## Scenery Definition File (.tsc)

The .tsc is a simple text file that defines the type of scenery object, its 3d model origin on the globe, elevation and use of auto height, its name as short and long version, description text and specific to airports: ICAO code identifier, parking positions, runway endpoints and threshold points, runway width and its approach light system, tower view position and more. We will address some of these features later on in this workshop.

Changing the .tsc file can be done in two different ways:

- Manually editing the .tsc file with a UNICODE capable text editor.
- Using the convenient export options of the exporter plugins in both 3D Studio Max and Cinema 4D to generate the file.

## Scenery Object Classes Overview

The Aerofly FS 2 engine supports the following scenery classes. A more detailed description will be given further down.

- ground (runways, taxiways, aprons)
- decals (runway and taxiway lines)
- objects (buildings and static vehicles)
- alpha to coverage (billboard trees, vegetation)

## Exporting an Airport

Go ahead and open the airport 3D model file in 3D Studio Max or Cinema 4D, whichever is preferred. Then the runway, decals, objects and alpha-to-coverage (atc) object have to be exported one after the other. When one of these is changed in the future, only the affected objects will need to be exported again. Running the Content Converter will update the results afterward. To export the ground mesh follow these steps:

1. Select the object 'kigm\_kingman\_rwy'.
2. In 3D Studio Max select 'Export selected' and choose 'IPACS TGI/TSC Exporter' as a file type. In Cinema 4D use the menu option 'IPACS Scenery TGI Export' to export the selected object as '.tgi'.
3. In the save-file-dialog browse to the folder 'kigm\_kingman' where the 'kigm\_kingman.max / .c4d' project file currently opened is stored, if not already in the right directory.
4. Edit the filename to 'kigm\_kingman\_rwy.tgi'.
5. Finally, click save to start the export process.
6. In 3D Studio Max pick 'Scenery' in the pop-up dialog and set the scale to 10000.

To export the runway markings and taxiway lines (decals 3D objects) repeat the same process but this time select the object 'kigm\_kingman\_decal' and all subobjects and export them as 'kigm\_kingman\_decal.tgi'.

Next, the remaining objects like houses, static vehicles and so on are exported by selecting the 'kigm\_kingman\_obj' and all its subobjects. Save the .tgi as 'kigm\_kingman\_obj.tgi'.

Follow the same pattern to export the alpha to coverage '\_atc' objects of the scene.

Finally the airport description file (.tsc) has to be created. Select all helper objects in 3D Studio Max or null-objects in Cinema 4D and export them into the same project folder as before by using the file extension .tsc or the export option 'IPACS Scenery TSC Export'. Name the .tsc file identically to the project folder name, for example 'kigm\_kingman.tsc' for the Kingman example airport.

You now have all required intermediate files to start the conversion to Aerofly FS 2 using the Aerofly FS 2 Content Converter.

## Notes using the Content Converter

Before the Aerofly FS 2 Content Converter can be started, several conditions have to be met. Otherwise the Content Converter throws an error and does not proceed.

- The input and output folders have to be set to valid file paths that fit the specific folder structure of your machine. To configure these paths edit the 'content\_converter\_config.tmc' text file in the project folder with a UNICODE capable text editor.
- All files referenced by the 3D model file have to be in the input folder.
- They have to be 'flat' in the input folder, meaning that they can't be in a sub-folder. If a texture is not found in the same folder as the exported .tgi files, the Content Converter will interrupt the conversion process.
- Files that don't belong to the project, especially unused images can cause the Content Converter to fail. If an areal image was used as a positioning guidance it must not be contained in the same folder as the 3D model. Such files can be put into a sub-directory or somewhere else.
- All 3D models referenced in the airport .tsc file must have been exported as .tgi files beforehand. For the Kingman airport those are 'kigm\_kingman\_rwy.tgi', 'kigm\_kingman\_obj.tgi', 'kigm\_kingman\_decal.tgi' and 'kigm\_kingman\_atc.tgi'.
- The .tsc file has to be contained within the project folder.
- The Content Converter only updates the output files if the intermediate files are newer than the already converted output files.

## Starting the converter

When the above conditions are met the Aerofly FS 2 Content Converter can be executed.

Within a project folder right click on the file 'content\_converter\_config.tmc' and select the context menu entry 'Run with Aerofly FS 2 Content Converter'. The Content Converter will then try to convert all intermediate 3D model files (.tgi) and textures in .bmp, .png and .tif format within the current (input) folder.

The converted files will be saved in the output directory defined in the .tmc configuration file. Usually, this is the folder 'Aerofly FS 2/scenery/places' in the current Windows user Documents folder. Writing the files into these directories automatically installs the scenery or aircraft for the Aerofly FS 2 Flight Simulator.

The converter can also be started by a batch file with the scenery configuration file as a parameter for the Aerofly FS 2 Content Converter executable. An example batch file 'convert\_kingman.bat' is provided for the sample airport Kingman, but the file should be edited (right click, edit) before the first execution to confirm the file paths within the batch file are set correctly.

## Coordinate System

The entire airport is modeled in a local coordinate system whose origin is moved to the longitude and latitude or web-mercator coordinates defined in the .tsc file when the airport is loaded in the simulator.

The x-axis of the modeling coordinate system points towards east; the y-axis is geographic north and the z-direction is defined up from the earth's geographic center towards the sky. Z is zero at airport elevation if 'auto height' is selected and it is zero at sea level if 'auto height' is not selected.

## Elevation and Auto-Height

There are two different approaches to defining the airport's elevation: 'auto height' and manual (auto height off).

- Auto height 'on' is used for more than 99% of the airports and is the preferred option for several reasons. It uses Aerofly's elevation data to transform the runway mesh and building positions to their final elevation. This has the major advantage that the entire airport ground can be modeled flat in 2 dimensions. This is much simpler and faster to create and it still features sloped runways in the engine. Objects like buildings will automatically be set to the correct ground level.
- When using the manual mode the entire ground mesh, runway and objects have to be modeled at the correct elevation above sea level.

## Model Scaling

Scenery objects have to be scaled to fit web-mercator units. Close to the location defined in the .tsc

file one web-mercator unit still equals one modeling unit but further away from the equator objects theoretically need to be bigger. For most airports sizes the effect is minimal and they can be modelled at a 1:1 scale with very little length error at the furthest point away from the .tsc reference point.

## Scenery Object Classes in Detail

We already briefly introduced the different object classes available in the Aerofly FS 2 engine. Now we will look closer into the individual properties of the ground mesh, decals, static objects and alpha-to-coverage objects.

### Ground

#### Runways, taxiways, apron

Usually, the ground mesh is one large joined mesh object that includes the runways, taxiways, aprons and surrounding area and is identified in Cinema 4D with the suffix '\_\_runway' (note the double underscore) and exported as '\_rwy.tgi'. Having multiple ground mesh objects is possible but can create visual issues when they are close together and is therefore not recommended.

A ground mesh can have different materials assigned that divide the mesh into several material selections. All material selections for the ground mesh must end with '\_\_ground' in their name e.g. 'taxiways\_\_ground'. Polygons in the ground mesh with no materials assigned to them will be treated as the surrounding area and the Aerofly engine will assign the default satellite textures.

Seen from above the ground mesh object can have any shape. Experience shows that a rounded island shape creates the best looking results. Using the vertex-color alpha channel in 3D Studio Max or the vertex-weight in Cinema 4D ground textures can be blended into the areal images linearly. Assigning vertex-weight (vertex-color alpha component) of 100% will use only the assigned material, smooth out the terrain elevation data and remove all auto-places trees. 0% will result in a usage of only areal images. This means a vertex-color or vertex weight has to be assigned in order to see any custom textures.

As the auto height feature changes the height per vertex, a smoother surface can be achieved by creating a denser mesh.

#### Ground materials

The materials for the ground mesh must have at least one UV-color texture assigned. Multiple UV-color maps are possible to reduce visual patterns created by the texture tiling. The default Aerofly FS 2 airports use two different UV-color textures with a 40m tiling color texture and 1200m variation 'dirt' texture. Other texture assignments by the modeling software have no effect on the exported '\_rwy.tgi' file. Instead, the shader will search for any files that have the same name but end with '\_bump' or '\_specular' and use them for the ground material in the rendering engine.

Summary:

- `_color` texture is a mandatory RGB, 8bit per color channel color map for the diffuse shader color.
- `_bump` texture is an optional gray 8bit RGB height map for optical surface height variations which will be converted to a `.tsn` normal map by the Content Converter.
- `_specular` is optionally used by the shader to create a soft gloss effect. It can have an alpha channel in addition to the three 8bit RGB color channels but doesn't have to.
- `_specular_alpha` is an optional 8bit RGB texture that overwrites the specular map's alpha channel with its calculated RGB brightness and controls the width of the specular reflection spot. The Aerofly FS 2 Content Converter will merge the `_specular_alpha` texture into a single `_specular` texture file.

## Decals

### Runway markings, taxiway lines

As decals we understand lines and markings on the ground that are usually painted onto the surface in the real world and require a higher level of detail than the ground itself. In the Aerofly FS 2 engine decals are rendered with a dynamic height offset over ground mesh to prevent z-fighting of the decals and the surface blow. The decal objects themselves must have a priority suffix at the end of their name, e.g. `'__priority0'`. The priority ranges from 0 to 9 with `'__priority9'` being the highest priority rendered on top of all other. Usually only `'__priority0'` and `'__priority1'` are needed because rarely more than two lines intersect at the same point. All decals are exported as one intermediate file with the ending `'_decal.tgi'`.

A decal object consists of several flat polygons and can have multiple materials assigned to it. Like the ground mesh, the elevation of decals is adjusted per vertex by the engine if the auto-height feature is selected. For very high curvature in taxiways or runways due to large changes in elevation, the decals should be cut along the edges of the ground mesh polygons below to follow the curved terrain better.

### Decal materials

Materials for the decals and markings must have a UV color map. The texture must have the extension `_color` and use three 8bit RGB color channels. Optionally an 8bit alpha channel can be used.

## Objects

### Buildings, vehicles

Houses, static vehicles and so on are created as individual polygon objects that also can have multiple materials assigned to them. When auto height is enabled, the Aerofly engine will offset the origin of each of these objects by the elevation information at that their origin. Because the terrain could be sloped it is recommended to have the base of the houses at an elevation slightly below zero in the modeling software to prevent floating above the ground. All objects are exported together as one `'_obj.tgi'` file.

## Object materials

One of two combinations are possible for materials used for objects:

1. `_color` (mandatory), `_reflection` (opt.) and `_light` (opt.)
2. `_color` (mandatory), `_specular` (opt.), `_specular_alpha` (opt.) and `_light` (opt.)

The mandatory diffuse color texture must end with `_color` and have the three 8bit RGB colors and an optional 8bit alpha channel for transparent windows as an example.

Reflection textures are 8bit RGB-textures and must end with `_reflection`. Bright colors create higher reflectivity.

Specular maps are used for non-reflective but shiny objects like metal surfaces. They are identified by the suffix `_specular`. The three 8bit RGB components describe the reflectivity of each of the colors and an optional 8bit alpha channel changes the specular spot size. Optionally an 8bit RGB `_specular_alpha` texture can be used to overwrite the alpha channel of the output `_specular` texture.

Light maps with the suffix `_light` can be used for light emitting things like windows at night. The three 8bit RGB colors are available.

## Alpha-to-coverage

### Vegetation and others

Billboard trees or flat grass planes that consist of only very few intersecting polygons fall into this category. The Aerofly FS 2 engine uses different techniques to render their transparency quicker which makes this category the best choice for large sets of partly transparent objects that are not very important.

## Object materials

The mandatory diffuse color texture must end with `_color` and usually is a single texture with three 8bit color and an 8bit alpha component.

## Guidelines for creating or porting own scenery objects

1. Create a new project folder where all textures and the 3D model will be stored.  
The Aerofly convention for naming airports is `icao_airport_name` e.g. `'ksfo_san_francisco_int'` and for cities its `'san_francisco_city'` as an example.  
The names must be in English letters, all lowercase and underscores instead of spaces. Spaces or umlauts mustn't be used to assure cross-platform compatibility.
2. Start 3D Studio Max or Cinema 4D and insert a background reference image that covers scenery region sufficiently. Save the project with the same name as the newly created project folder.
3. Paste all textures into the project folder and rename them following the same naming rules as

above. Refresh the texture assignments in the modeling software if necessary.

4. Scale the entire scene to 1:1 by using ground reference points on the background image that are easy to identify. Vast scenery projects should probably be split into multiple regions if they are not using web-mercator coordinates to avoid wrong scaling at great distances.
5. Copy the dummy- or null-objects from our example airport and adjust the names, ICAO identifier, etc. or copy the .tsc of the example airport to the project folder and set the fields manually.
6. Decide on an airport reference point on the background image and write its coordinates (lon. lat. or web-mercator) into the dummy- or null-object for the airport location or into the .tsc file manually.
7. Create one ground object that is one joined mesh.
  - 3D Studio Max specific: The materials for ground objects should follow the following naming conventions: `__airport_runway`, `__airport_apron`, `__airport_outside`
  - Cinema 4D specific: All material selections assigned to the ground objects must have the `'__ground'` suffix in their name. The name of the ground object must end with `'__runway'`.
1. Continue with decals, add the suffix `'decal__priority0'` to their names for now, adjust the priority by renaming individual lines later. Group all decals together so that they can easily be selected for exporting them all at once.
2. Use helper objects to set the coordinates of all runway endpoints and their threshold points. The threshold points will affect the positioning of the selected approach lighting system.

From:

<https://www.aerofly.com/dokuwiki/> - **Aerofly FS Wiki**

Permanent link:

<https://www.aerofly.com/dokuwiki/doku.php/sdk:scenery>

Last update: **2017/01/16 11:52**

