# AutoAsm V1.0

22nd October 2005

by Chris Wright

chris@kline.demon.co.uk

## INTRODUCTION

AutoAsm is a program that creates LWM polygons (water or land) and VTP lines (roads, streams etc), as well as other scenery objects for use in FS2002 and FS2004. The input data is always in the form of a colour bitmap, though several data formats can be imported and converted to suitable bitmaps.

The following scenery objects can be created:

1. LWM polygons for water or land.
2. VTP lines for roads, streams, railways etc.
3. VTP areas.
4. Terrain mesh.
5. Landclass.
6. Static objects such as trees or buildings linked to VTP lines (similar to vector autogen).
7. Dynamic objects such as road traffic linked to VTP lines.

The design process for LWM polygons and VTP lines is as follows:

1. Create or import a colour bitmap. LWM and VTP polygons are represented by line polygons. The lines must be of single pixel width and coloured i.e. non-grey. AutoAsm will not read in grey lines, so the bitmap can be annotated with black, grey or white text.
2. On the Main panel, set the project parameters such as project name, geographical location and size, and save.
3. Read the image into AutoAsm. The lines and polygons will be displayed in the map area with the same colours.
4. In the Line Types panel, set the individual line parameters. For each line a set of parameters have to be defined, for example water or land, LWM, VTP or both, VTP texture number, line width. Save the project again.
5. Create the VTP and LWM asm files. Optionally AutoAsm will compile the files and copy the bgl scenery files to Flight Simulator.

Roads that cross bodies of water can be automatically converted to bridges. If an elevation bitmap is available, individual lakes can automatically be set to the correct elevation so that they match the surrounding terrain.

There are a number of possible sources for the LWM / VTP input bitmaps:

1. Polygons and lines created out of your imagination, for example when creating an imaginary island.
2. Bitmaps exported from TerraScene.
3. Bitmaps exported from LWMViewer or GIS programs such as fGIS.
4. Any suitable maps or aerial/satellite images you can use to draw on coastlines and roads etc.
5. A Flight Simulator lookdown map screenshot, useful for integrating new polygons and lines with existing default or add-on scenery and buildings.

AutoAsm can also import .dat files , .mp files (polish format) as well as .bln and .ung files. Dat files are available from http://www.ngdc.noaa.gov/mgg/shorelines/shorelines.html

You can create terrain mesh and landclass, again by reading in suitable bitmaps. Terrain mesh bitmaps are 8 bit monochrome .bmp types (the pixel brightness defines the mesh elevation), while landclass bitmaps are 8 bit colour .bmp types (the colour index directly defines the landclass type).

# CONTENTS



# TUTORIALS

# ACKNOWLEDGEMENTS

Many thanks to Jim Keir who gave me permission to include LWMViewer in the package. LWMViewer is a perfect companion to LWM/VTP creation programs. AutoAsm includes an LWMViewer button. After creating your new LWM/VTP scenery bgl's, click on the button to open LWMViewer and view your scenery. If there are any problems with your input data it will be immediately apparent!

Many thanks to the experts who have contributed so much to our knowledge of this amazing flight simulator, in particular Dick Ludowise and Christian Stock. Thanks also to Dick for the use of his include files.

Many thanks also to Edgar Knobloch for the use of his dyn_header.inc file.

Many thanks to Microsoft for the various tools such as bglc and resample - and thanks for a rather good flight simulator as well.

# INSTALLATION

There is no reason why the use of this program should harm your computer or your files. But I take no responsibility should your computer emit clouds of smoke or print unintelligible garbage on the screen.

To make a fresh installation:

1. If required, create a new directory where you want to locate the AutoAsm directory.
2. Double-click on the autoasm.zip file to run WinZip. Check that Use Folders is ticked.
3. Select all the files *except* AA_dyn_lib.bgl and the bmp files with the 'dyn' and '0dyn' names.
4. Browse for the desired location and extract the selected files. A directory named AutoAsm will be created inside the directory. If desired you can move or rename the AutoAsm directory, but be sure to correctly update the WORKING DIRECTORY setting.

When you first run AutoAsm you should first enter the AutoAsm directory location into the WORKING DIRECTORY text box (at the top of the screen and near the LH edge). For example, if the directory is located at c:\scenery_tools\AutoAsm then you should enter 'c:\scenery_tools\autoasm'.

Note: one of the Microsoft tools does not seem to work with file paths that have spaces, so installing into the Program Files directory may cause problems.

If you want to use the AutoAsm library of dynamic objects:

1. Select the file AA_dyn_lib.bgl and extract it into the Flight Simulator scenery folder e.g. FS2004\addon scenery\scenery.
2. Select all the .bmp files beginning with 'dyn_' or '0dyn_' and extract into the Flight Simulator textures directory e.g. FS2004\addon scenery\texture or FS2004\texture.

If you already have the previous version of AutoAsm installed (V 0.8, 28th November 2004), then all you need to do is extract autoasm.exe and the manual, overwriting the original files. If you want to use dynamic objects you should create a new AutoAsm sub-directory named 'Dynamic_objects' and extract these three files into it: dyn_header.inc, Fs6def.inc and model_id.inc.

If you have an existing AutoAsm installation then you could simply extract into the existing directory, over-writing any files with the same name. A safer method would be to install into a new directory and then copy all needed project files into the new directory (for example bitmaps in the bitmaps sub-directory and the .ini and .dat files in the settings sub-directory).

# RUNNING THE PROGRAM

This version of AutoAsm is optimised to run at a Windows resolution of 1280*1024. With standard font size it will fill approximately two-thirds of the display window. With font size increased by 8% it will fill the display window.

The program can be used at 1024*768, but part of the windows may be cut off at the right and there will be other minor display defects. The AutoAsm windows can be moved in the normal way to improve visibility.

If you *really* want to run it at 1024*768 let me know and I'll see if I can make a special version for this resolution.

Run the program by double-clicking the AutoAsm exe file, or by any other standard Windows method.

When AutoAsm opens you will see a collection of buttons and readouts, with the map display taking up the lower half of the screen. The following is a very brief tour of the program.

Note: a detailed description of all the panel functions is given in Section 12.

Before proceeding, let's fill the map area in the bottom half with something more interesting than a grid.

1. In the WORKING DIRECTORY slot, type in the full path to your main AutoAsm directory (e.g. c:\autoasm). While you're about it, you could fill in the path to Flight Simulator under FLIGHT SIMULATOR PATH (e.g. c:\FS2004). Click on FINISH to exit the program and then run AutoAsm again. This demonstrates that all settings are saved on exit and restored the next time you run the program.
2. In PROJECT NAME, type in 'demo' and then click on SETTINGS/LOAD. This will load in the settings for the demo project. Note that, after loading the new settings, the settings you entered in Step 1 do not change.
3. Click on READ DATA. The bitmap demo.bmp will be read in. After a few seconds you will see the lines and polygons in the map display.

You can use the zoom and pan buttons at the right to take a closer look. The smaller grid squares are the LOD13 areas, while the larger grid squares are the LOD8 squares. If you left-click on a line the selected point is highlighted in green. You can see information on the point in the text slots just above the map area. The settings for the line are shown in the STATUS slot.

In the top LH of the screen you will see a small group of buttons named 'SELECT FUNCTION'. The buttons are labelled 'MAIN', 'LINE TYPES', 'Objects', 'Terrain', 'LOD8 Squares', 'TOOLS' and 'EDITOR'. Click on these and you will see that the panel to the right changes as appropriate. Select 'Line Types' to set parameters for the lines. Select 'TERRAIN' if you want to create terrain mesh and landclass, and so on. When you've finished looking at the various panels, select the MAIN panel again.

You can change the background colour according to your preference. Select the Tools panel and try adjusting the sliders. The first three sliders adjust the R,G,B colour while the other two adjust the brightness of the large and small grids.

On the MAIN panel you enter settings for the project such as the project name, the working directory and settings that determine how the input bitmap is to be read in. Here you can also load and save settings for the project.

The fixed panel just above the map display has the buttons for reading in bitmaps and creating the LWM and VTP asm files. If COMPILE and TO FS are checked AutoAsm will call the Microsoft compiler (bglc.exe) to compile the asm files and copy the resulting bgl files to Flight Simulator. The STATUS slot displays status and error messages. If you click on a line in the map display the STATUS slot will display the line's parameters. The readouts below it and just above the map display give information on the selected point (to select a point, simply click on it). POLYGON indicates the polygon number and its colour index. X13 and Y13 are the point co-ordinates inside the current LOD13 square. LONG and LAT are the geographical co-ordinates of the selected point.

If you click on the + button at the left of the display the selected point will move to the next point in the list. The INDEX readout indicates the index number of the point (actually the index position inside the program data arrays). The selected point is highlighted in green. The button group to the right of the display allows you to pan and zoom the map display. The pan and zoom buttons with two pluses or minuses have a greater effect. Click on CENTRE to centre the selected point. Clicking on a point while pressing the RETURN key will also centre the selected point.

After creating your scenery bgl files, you can click on LWMViewer to load LWMViewer and display the scenery you just created. If LWMViewer indicates your scenery is okay you can call Flight Simulator to inspect your work. Click a point on the map to select a location and then click on RUN FS. It will open Flight Simulator at the exact location you selected.

Note: if you call LWMViewer after creating the LWM or VTP scenery, but not both, LWMViewer will give an error message when it opens. If this happens, just click ok and continue.

After a brief look around we're ready to create some scenery. You should still have the demo bitmap loaded. If not, repeat steps 2 and 3 above. Just above the map and at the left you'll see the WRITE LWM button. This function converts the LWM data (e.g. the water polygons) to an assembly file that can then be compiled into a .bgl scenery file. If the COMPILE check box is checked then the Microsoft compiler (bglc.exe) will compile the assembly file into a .bgl file. If the COPY TO FS check box is checked then the .bgl file will be copied to Flight Simulator (the SCENERY BGL PATH text box must be correctly set up).

The resulting assembly and .bgl files are placed in the main AutoAsm directory. The files bglc.exe, TDFHeaders.inc and TDFMacros.inc must also be located in this directory.

Immediately below the WRITE LWM button you'll see the WRITE VTP button. This has the same function except that it creates the VTP assembly and .bgl files.

The OUTPUT FILE text box defines the file names. If OUTPUT FILE is 'test' then the files will be: test_LWM.asm, test_LWM.bgl, test_VTP.asm and test_VTP.bgl.

Now we're ready to create the scenery. But first a quick note.

All the demo projects except the island project are located at N10 E62, which is off the east coast of Africa. This means you don't need to worry about the default scenery as it's only water. You'll find the file background.bgl in the main AutoAsm directory. Simply copy it to Addon Scenery\scenery in the FS directory. This provides some background landclass for the demo projects.

4. Check that these check boxes are enabled (checked): ISLANDS, AUTO FILL, AUTO CULL, COMPILE and TO FS. These should always normally be enabled. Similarly, FIX PIXELS and LWM ONLY should normally be disabled (unchecked). Note: with AutoAsm version 0.8 ISLANDS, AUTO FILL and AUTO CULL will always default to on when the program is run or a project loaded. The only reason any of them might be set to off would be for fault finding.

5. Click on WRITE LWM. This creates the LWM assembly file and scenery. As the assembly file is created progress is indicated by black dots crawling over the polygons and filling in the water areas. This should take less than half a minute. At the end you will briefly see a DOS window open when bglc.exe is called to compile the scenery.

6. When WRITE LWM has completed, click on WRITE VTP. This creates the VTP assembly file and scenery file.

When WRITE VTP has completed, the LWM and VTP scenery files have been created and copied to Flight Simulator.

7. Click on LWMViewer. With FULL SCREEN enabled LWMViewer will open in full screen. Examine the polygons and check that the scenery is correct. Close LWMViewer. Note that LWMViewer displays the bgl files created in the main AutoAsm directory and not the files copied to FS.

8. Click on one of the polygons in the map view. The point will be highlighted in green. Click on RUN FS. AutoAsm calls Flight Simulator and then closes. FS will open in slew mode with your aircraft directly above the point you clicked.

That's it. You have created your first scenery in AutoAsm. But it's pretty boring scenery, consisting of a few circles and simple polygons. If you want to be more adventurous, try this. Repeat the steps 4 to 8 above, but with one important difference: type the name 'ts_demo' into PROJECT NAME.

TS_demo is a cut-down TerraScene bitmap, but it's still far more complex than the 'demo' bitmap. Read it in, take a good look at the map view, create the scenery, check it with LWMViewer and finally run it in FS.

So far we've loaded in pre-packaged projects so there was no need to worry about some of the more detailed aspects. Now we'll look at one of the most important aspects of creating new projects: line settings.

# LINE SETTINGS

In AutoAsm scenery is always created from data contained in colour or grey-scale bitmaps. The bitmaps contain no information on the nature of the lines. A set of parameters for each line is stored in a .dat file. If the project name is 'test' then the line parameters for the project are stored in test.dat, in the settings sub-directory.

The parameters for each line is stored in a text string. Here's an example:

1050, water, both, 50, 0, 3, 8, 2, 30, coastline

Each parameter is separated by a comma and an optional number of spaces. At the top of the Line Types panel you'll see a line of labels for the parameters. They are as follows:

type: this is the texture type, e.g. 1050 for a shoreline or 1138 for a road.

water/land: defines whether the polygon is water or land (only applicable for LWM polygons).

lwm/vtp: defines whether the line is LWM, VTP, BOTH or VTP AREA. If set to 'both' then the polygon will create water and also a VTP line, usually a shoreline.

width: sets the texture width for VTP lines.

height: sets the height in meters for water polygons. -9999 can be entered for mesh-clinging water.

bridge: sets the bridge type for road sections that cross water.

layer: sets the layer number for the line. A higher number gives higher priority.

smooth: sets the amount of smoothing (zero for no smoothing, typically 1 or 2 for roads).

random: sets the amount of added randomness (zero for roads, typically 20 for coastlines).

comments: add any comments here e.g. 'coastline'.

The basic procedure for assigning parameters to a line is as follows:

1. Read in the bitmap and go to the Line Types panel (you can load in the bitmap to the Editor or read in the bitmap by clicking on READ DATA).
2. Click on a line in the map display to select it.
3. Enter the parameters into the small boxes running along the top of the Line Types panel.
4. Click on UPDATE POLYGON.

The parameters are copied to the selected line. If you then click on the line the parameters will be displayed in the STATUS slot.

You can create a collection of pre-made settings in the large box on the right. To do this, simply click on ADD and the settings in the editing boxes at the top are copied into the large box. To remove settings in the box, click on a line to select it and click on REMOVE.

To apply settings from the large box, select the polygon in the map display and double-click on the settings in the large box.

To copy the settings from a text line in the large box to the editing boxes at the top, simply double-click on the line.

To copy settings from a polygon on the map display to the small editing boxes, double-click on the polygon.

To copy settings from the editing boxes to a selected polygon on the map display, click on UPDATE POLYGON.

To check that the correct settings have 'taken' to a polygon, click on the polygon and the settings will be displayed in the STATUS slot.

Settings are actually applied to a given colour index rather than to a specific line. Therefore all lines of the same colour will have the same settings. This is useful if you want to make the same changes to a large number of lakes. But it's not so useful if you want to give different settings to the lakes e.g. different elevations. In this case you can use the polygon tool in the Editor. This tool changes the colour index of a polygon to the next unused value. Optionally it can also change the polygon's elevation to a specific value.

If your source of bitmaps generates consistent colours (e.g. TerraScene) then you don't need to laboriously add the settings for each new project. Simply load a previous project, rename it, enter the new geographical bounds, make any other changes as required and save the project with the new name. When you load the new bitmap you will find that all the lines have the correct settings because they were 'inherited' from the old project.

# CREATING NEW SCENERY

This section summarises the complete process of creating new scenery. Let's suppose we want to create a few roads and lakes by drawing them on a bitmap. The procedure would be as follows:

1. In your paint program, open a 512*512, 8 bit colour bitmap and draw a number of line polygons to represent the lakes. The colours should be in the range 1 to 249, while the background should be colour zero. If you want to apply different settings (e.g. different shoreline textures) to the lakes then use different colours for the polygons (the same applies to roads). Draw a number of lines for the roads. Ideally the lines should be a single pixel in width with no redundant pixels. In practice there will be quite a few redundant pixels and maybe some breaks, but they'll be cleared up with the Editor tools.

To add some islands, draw one or two polygons inside the existing ones.

2. We'll call our new project 'lakes', so save the bitmap to the AutoAsm bitmaps sub-directory as 'lakes.bmp'.
3. Load the 'demo' project and change the project name to 'lakes'.
4. In the Editor panel, change INPUT BITMAP and OUTPUT BITMAP to 'lakes'. Save the project.
5. Click on LOAD. You will see your polygons and lines in the map display.
6. Go to the Line Types panel and assign settings to the lines and polygons (for more details see the 'Assigning Line Types' section). After the line types have been assigned, save the project.

We now need to fix any bitmap errors. I'll summarise this process here, but it is described in more detail in the 'Using the Editor' section. Provided your main polygons are lakes and not islands, and that the polygons do not touch the bitmap edge, then we don't need to use the BITMAP EDGES tool. The main polygons will be automatically assumed to be water.

Of course, if you drew the lines very carefully without any breaks or redundant pixels then you could skip steps 7 and 8.

7. Run the REPAIR BITMAP tool.
8. Run the SEARCH tool and fix any errors. Repeat until you get the green banner (zero errors). Save the bitmap.
9. From the Main panel, click on READ DATA to read in the bitmap. As all the LWM pixel errors were fixed in steps 7 and 8, the scenery polygons should display correctly, with any islands shown as red and the lakes filled with water.
10. Check that COMPILE and TO FS are checked and then click on WRITE LWM to create the LWM scenery.
11. Click on WRITE VTP to create the VTP scenery.
12. Check the created bgl scenery files with LWMViewer.
13. Click on a selected point in the scenery and then click on RUN FS. Flight Simulator will open in slew mode at the point you selected.

# USING THE EDITING TOOLS

AutoAsm always uses colour bitmaps for data entry. The bitmaps consist of single pixel-width lines and polygons. Line settings (e.g. width and texture type) are applied to each colour index that is used. Therefore, all lines of the same colour index will have the same settings (an Editor tool allows you to create unique settings for individual lines). You can edit the bitmap palette to any colour you want with a paint program - as the line's colour index does not change, changing the colour does not change the settings. There is one restriction: all lines and polygons must be non-grey (i.e. coloured). More specifically, at least two of the R, G and B components must be different. AutoAsm ignores all grey pixels. This means that you can annotate the bitmap with text etc providing you use a grey colour (including pure white or black).

If you import a conventional map bitmap you should first convert it to 8 bit grey scale and then draw colour lines over it as required.

The background colour must be grey. The background should also preferably be zero index colour, as this will speed up some operations, including map drawing.

When AutoAsm scans a bitmap it looks for valid pixels. When it finds one it then traces the entire line or polygon. When there are no more pixels of the same colour it assumes it has reached the end of a line. If it returns to the same pixel position then it has traced a complete polygon. However, if there are defects (e.g. breaks or redundant pixels) then the read-in process may fail.

## The Perfect Polygon

Sounds a bit like a Hollywood block-buster, doesn't it? But perfect polygons are a lot more useful (and considerably less dangerous) than perfect storms.

The definition of a perfect polygon is easily stated: every pixel of a polygon must touch precisely two other pixels of the same colour.

If there is a Golden Rule for bitmaps, this is it. It means that there must be no breaks, because the pixels at the edge of the break will touch only one other pixel. If there are extra (redundant) pixels, then the Golden rule is also broken, because at least one pixel will touch more than two other pixels.

Lines should also conform to the rule, with one exception: the pixels at the start and end of the line will touch only one other pixel.

Imperfections in water polygons are far more serious than those in lines such as roads. They can cause spurious squares or spikes of water or land to appear, and auto detection of islands and bridges may fail.

In previous versions of AutoAsm, these errors had to be fixed manually with a paint program. It required skill, experience and patience. And, with large and complex bitmaps, a lot of time. Fortunately the editor tools introduced with this version of AutoAsm have changed all that. They allow errors to be fixed in a fairly consistent manner that requires less user skill. A degree of automation also makes the task a lot easier. With a little experience the errors in a complex bitmap with hundreds of thousands of points can be corrected in fifteen minutes or less.

At the end of the process you will see a green banner that informs you there are no errors. After saving the bitmap you should then be able to read in the bitmap and create the scenery without problems.

The Editor and associated tools can be found on the Editor panel. Tools that require user mouse input (e.g. clicking on the bitmap to add a point) are located in the Editor box. Tools that scan the entire bitmap to perform their function are located outside the box. For a concise description of each text box, button etc, see the Panels section.

Note that currently the tools mostly correct errors in LWM polygons only (more support for VTP lines may be added later). This is because LWM errors are *far* more serious than VTP errors. For example, LWM errors can create spurious and very obvious squares or triangles of water.

## Correcting Bitmap Errors

The normal sequence for correcting a bitmap would be:

1. Load the bitmap into the Editor.

2. IMPORTANT: If not already done, use the Line Types panel to define all the lines and polygons as VTP or LWM types. You can refine the settings later, but all lines MUST be defined as LWM or VTP.
3. Run the REPAIR BITMAP tool.
4. Define whether the top left pixel is land or water and then run the BITMAP EDGES tool.
5. Run the SEARCH tool repeatedly until the green banner (no errors) is achieved.
6. Optionally run the MAKE BRIDGES tool to convert any road sections that run across water to VTP-style bridges (strictly speaking, this does not correct bitmap errors).

Note: just to add emphasis, step 2 is essential. The detailed settings don't matter at this stage, but all the lines MUST be correctly defined as VTP or LWM type. That's because lines are processed differently depending on whether they're VTP or LWM. Currently most repair functions are for LWM only as LWM errors are far more obvious.

If all lines are completely undefined then they will default to VTP type and SEARCH will find no errors (but you will get an error message if no lines are defined as LWM).

If your source of bitmaps generates consistent colour indexes (e.g. TerraScene) then bitmaps can 'inherit' settings from previous projects, making it unnecessary to apply new settings. To do this, load an existing project, then change the project name to match the new bitmap, change any other settings as required (e.g. the geographical position and size settings) and save the new project. The lines and polygons will then have the same settings as in the previous project.

## REPAIR BITMAP Tool

Click on REPAIR BITMAP to run this tool. It scans the entire bitmap and corrects a number of problems. It will delete redundant or isolated pixels. It also deletes corner pixels. Corner pixels are also redundant pixels, but they have a specific form. For example, if a pixel has pixels immediately above it and to the right, then it's a corner pixel. The tool will also correct line breaks up to the size set by SEARCH RADIUS (LWM) on the Main panel.

This tool will not fix all pixel errors. That job will be left to SEARCH.

## BITMAP EDGES Tool

Very often part of a water polygon will be cut off by the edge of the bitmap. This appears as a large polygon break along the bitmap edge. This tool automatically scans the bitmap edge and adds lines to correct the breaks. The line has a special attribute (NO VTP): the lines drawn along the bitmap edge will form complete LWM water polygons as required, but they will not form a VTP shoreline, which is also required (otherwise you would see shorelines in the middle of lakes).

This tool raises an important point. A bitmap filled with line polygons is fundamentally …. The polygons could be lakes. They could just as easily be an archipelago of islands. Things get even more difficult if there are polygons inside polygons.

Suppose a bitmap has one large polygon with some polygons inside it. AutoAsm will always work consistently: the outer polygon is assumed to be a water polygon. Therefore the inner polygons are islands. This is fine if the outer polygon really is a lake. But what if it's actually an island and the inner polygons are lakes? The solution would be to draw a rectangular outer polygon along the edge of the bitmap. Then the entire bitmap would be filled with water and the outer polygon would be an island as required. The BITMAP EDGES tool will do this for you, as well as completing lakes cut off by the edge.

For this to work correctly you need to inform the program whether a single specified pixel is land or water: I chose the top left pixel. You do this by setting the PIXEL 0,0 text box to land or water as required. Simply click in the text box to toggle the values. You can choose the colour for the NO VTP lines that will be added along the edges: to do this, click in the text box to the right of the NO VTP button.

Click on the BITMAP EDGES to run the tool. After a few seconds you will see that lines with the colour that you chose have been added along the edge. In many cases this process will work correctly: you will see that where a water polygon was cut off by the bitmap edge, now a NO VTP line has correctly completed the polygon. But it can fail - if so you may find that lines have been drawn across land areas.

You should scan around the bitmap edge to make sure that all is well. The BITMAP EDGES process starts at the top left pixel and works around in the clockwise direction. Therefore you should start at the top left corner and check the edge going clockwise. If you find an error you will probably find that the

NO VTP lines remain out of sync i.e. placing lines where there is land and not where there is water. Ironically, if there is a second error it will probably put the process back into sync!

There are several polygon problems that could cause BITMAP EDGES errors:

1. A polygon is cut off by the edge but it does not have a pixel on the edge. Solution: add the missing pixel.
2. A polygon is cut off by the edge but it has two or more pixels actually on the edge. Solution: remove the extra pixels. Where a polygon meets the edge there should be just one pixel.
3. A valid polygon happens to run along the edge for some distance. This is quite rare but it will happen from time to time. Solution: edit the polygon so that it does not run along the edge. Second solution: delete the pixels running along the edge, but leave a pixel on the edge where the polygon first touches the edge and also where the polygon leaves the edge - this simulates a perfect polygon cut off by the bitmap edge that will be fixed correctly by the tool.

If you find an error while scanning around the edge, first click on UNDO to remove the added lines. Look at the pixels at the point where the error first occurred and decide what the problem is (e.g. a missing pixel on the edge). Then use the Editor tools to edit the polygon. When you've finished editing, run BITMAP EDGES again and check that the error is fixed, then continue scanning around the bitmap in search of further errors. When you've corrected all the line errors, save the bitmap.

If for any reason there are still errors in the added lines, then the SEARCH tool will find them!

## SEARCH Tool

This tool scans the entire bitmap. It looks for pixels that break the Golden Rule i.e. any LWM pixel that does not have exactly two neighbours of the same colour.

When the tool finds an error it immediately stops and displays the error with a fairly high zoom value so you can see the pixels clearly. A small cross-hair indicates the problem pixel. A message in the STATUS slot suggests what the problem is. The Editor tools are enabled and the small floating tool window appears. You can choose several editing tools from this window and you can move it close to the editing point for convenience. You can also choose tools from the main Editor box.

Suppose there is a single pixel break in the polygon. Simply choose the Points tool and left-click to place a pixel in the gap, thereby fixing the break. The colour is automatically set to the colour of the polygon. If there is a larger break, choose the LINES tool. Click on a pixel at one end of the break and then click on the pixel at the other end of the break. The tool will draw a line between the two, thus fixing the break.

If you have enabled the AUTO JUMP feature (tick box just below the SEARCH button), as soon as you place the pixel the search resumes, starting at the place where it stopped. It will continue searching until it finds the next error. If AUTO JUMP is disabled, the search will not resume until you click on NEXT on the floating window. This allows you to see the effect of your edits before resuming the search.

Using AUTO JUMP is quicker and more convenient but, particularly when you're starting on the learning curve, you may prefer to see the effect of your edits before resuming the search (it seems more 'intuitive'). However, if you use AUTO JUMP you can set a time delay, e.g. one second, before the search resumes. This allows you to see the effect of your edits before the screen clears. As you gain experience you'll probably want to set a smaller delay - I usually use a zero delay for maximum speed.

On the first run you'll probably fix plenty of errors, but not all. When the search is complete you will be informed of the number of errors that were fixed. Run the tool again. On the second run you'll probably find a much smaller number of errors. Keep repeating the search runs until you get a clear run with zero errors. You will be rewarded with a green banner that informs you there were no errors. The green banner indicates that, as far as LWM polygons are concerned, you have a perfect bitmap. Two features (automatic island detection and VTP bridges) require a green banner bitmap to work reliably.

The rule is simple: keep running the SEARCH tool until you get the green banner. Particularly if it's a large and complex bitmap, you should give yourself a pat on the back when you get your first green banner. But don't forget to save the bitmap or you'll have to do it all over again!

## VTP Bridges

Now that you have a perfect bitmap you can optionally use the MAKE BRIDGES tool. It scans the bitmap and wherever it detects a road section crossing water it converts the section to a VTP bridge.

This tool should only be used on a 'green banner' bitmap, as it requires perfect water polygons to correctly determine whether a particular point is in the water or not.

Before running the tool, you should decide what bridge types you want (the numbers are given in the text file terrain.cfg in the main Flight Simulator folder). For example, if you enter 1220 and 1240, the tool will randomly assign bridge types in the range 1220 and 1240.

Click on MAKE BRIDGES to run the tool. It scans the entire bitmap. When it finds the first pixel of a road line, it scans horizontally from the pixel to the left hand bitmap edge, counting the number of LWM lines that it passes. Note that the PIXEL 0,0 text box must be correctly set. It then checks the road line for any sections that pass across water.

Each road section in water is changed to one of ten unique colour indexes that have bridge settings automatically assigned. A node point is also added at the two ends of the bridge. This ensures that the end of the road and the beginning of the bridge are at the same point.

As each bridge is created an entry is added to the DIAGNOSTICS WINDOW list. When the tool has finished, click on the DIAGNOSTICS WINDOW and you will see a list of the new bridges at the bottom. If you double-click on one of the bridge text lines the map view will jump to that location, thus allowing you to easily inspect the new bridges.

You can click on the NEXT button (just below the MAKE BRIDGES button) to make the map display jump to the next bridge.

The list is not permanent - it will be lost when you exit AutoAsm. In this case, click on the SEARCH button (just below the MAKE BRIDGES button) to create a new list. You can then use NEXT to inspect the bridges on the bitmap.

## Editor Tools

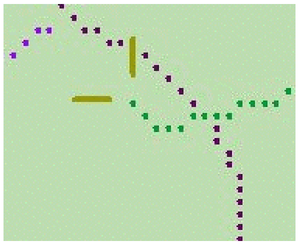
To enable editing, click on the EDIT button. A small floating window appears. To select an Editor tool, click on one of the buttons in the Editor box (for convenience a subset of these is also included on the floating window). Where appropriate, when a tool has been selected, left-clicking adds points or lines while right-clicking deletes points or lines. The tools are as follows:

1. Points: add or delete pixels.
2. Lines: add or delete line sections.
3. Pick Colour: click on a pixel to choose the current colour.
4. Set Colour: click on a line or polygon to set its colour to the current colour.
5. Poly: left-click on a polygon to change its colour index to a new and unique value - or right-click to delete the entire polygon. This tool can also be used to set a unique elevation for a lake polygon.
6. No VTP: left-click on a line section to change it to the NO VTP colour index (251).

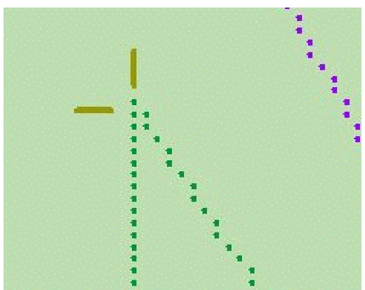
Refer to the Panels section for a detailed description of these Editor tool functions. To use them you must first click on EDIT to enable them. If they are not enabled then clicking on the bitmap will not modify it. To disable the Editor tools, click on EXIT.

# BITMAP ERRORS

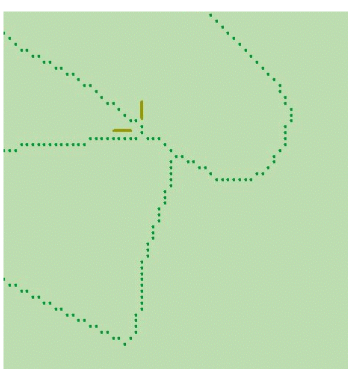
The following examples show typical pixel errors. The LWM polygons are green.



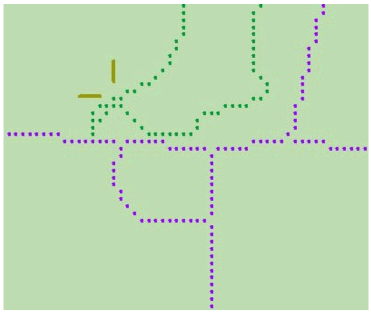
A line break. Zoom out to see the entire polygon and then left-click to draw a line that repairs the break.



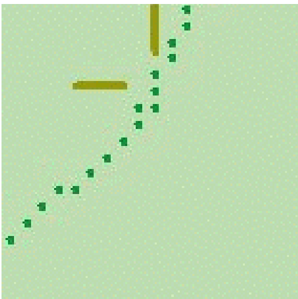
Right-click to delete the three redundant pixels at the top. The pixel below them is a corner pixel, so delete it as well (the third pixel down on the left).



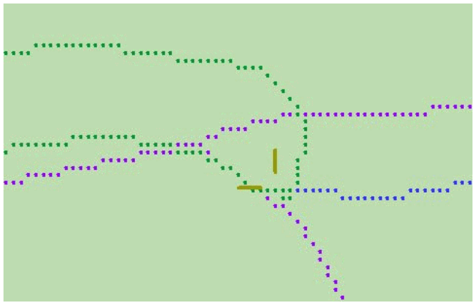
It looks like two polygons are touching. You'll need to delete several pixels to separate the two polygons and then add pixels to repair the breaks. Alternatively you could delete the five common pixels to convert it into a single polygon.



The ‘toe’ on the ‘foot’ is probably a small lake. Delete the polygons below the cross hair to detach the small polygon. You’ll need to edit the small polygon to remove the redundant and corner pixels. Or you could delete the small polygon entirely.



A classic corner pixel. Delete the pixel on the left on the seventh line down. Or you could delete the pixel to the right of it. After deleting either pixel the corner pixel is gone.



Although the mauve VTP line crosses the green LWM polygon, that’s not a problem because it does not create a break. But the two lowest green pixels are a problem. Delete them.

# RESERVED COLOUR INDEXES

For all LWM/VTP bitmaps, zero colour index should be used for the background whenever possible. Colour indexes in the range 1 to 249 are available for polygons. The colour indexes 250 to 255 are reserved for special purposes as follows:

250: VTP MARKED

When a bitmap is being read in by AutoAsm, every VTP pixel that has been found is marked with colour index 250. This ensures that any VTP point will only be found once.

251: NO VTP

Water polygons that have been defined as 'both' will create water and also a VTP line, usually a shoreline. However, there are occasions when part of a water polygon should not have a shoreline, for example a dam. When a water polygon is cut off by the edge of the bitmap, the polygon must be completed by a line running along the edge. This line should not have a shoreline.

You can ensure that a selected section of a polygon does not have a shoreline by changing its colour to 251. The editor has the NO VTP tool for doing precisely this. Go to the Editor panel and select the colour you want by clicking in the box just to the right of the NO VTP button. Click on EDIT to enable the editor tools and click on the NO VTP button. Click on two points of the polygon. The selected section is then changed to a NO VTP line with the colour you selected.

The BITMAP EDGES tool also creates NO VTP lines. These lines run along the edge of the bitmap to complete any polygons that happen to be cut off by the edge. If the lines were the same colour as the polygons then you would see shorelines running across lakes!

252: NO OBJECTS

This is similar to NO VTP, except that it inhibits the creation of objects along any line sections converted to NO OBJECTS (see Section…. for more information about creating objects). Currently there is no NO OBJECTS tool, so you would need to add this in your paint program.

253: NODE POINTS (yellow)

The primary purpose of node points is to force VTP lines (usually roads) to join together.

Suppose two different-coloured road lines form a T-junction on the bitmap. Because there is a finite distance (the pixel spacing) between the two roads, there will be a small gap. If a common pixel is changed to a node point, it will force the two roads to join, forming a natural T-junction. In the text diagram below the A and B points represent two roads forming a T-junction. The node point is placed at position N. This forces the A and B lines to join together, thus correctly forming a T-junction.

```
A              A
A              A
A              A
ABBBBBB   →    NBBBBBB
A              A
A              A
A              A
```

Auto VTP bridges also use node points. On the bitmap a bridge consists of a road with a gap, the gap being filled by the bridge line. A node point is added at each end of the bridge line, thus ensuring that the bridge joins with the roads.

A node point could be placed at each road junction. When adding objects linked to VTP lines objects will not be placed on a node point, which greatly reduces the probability of a building being placed at the centre of a cross-roads!

The editor has a tool (ADD NODES) aimed at automating the process of adding node points, but it is still under development and its use is not encouraged.

254: ISLANDS (red)

When an LWM bitmap has polygons inside other polygons (e.g. islands inside lakes) AutoAsm uses the following convention: as it moves along the data stream the water is always to the right. If you click on a point to select it and then click on + (just above the map display on the left), you will see the selected point moving along the data stream (more specifically, each successive point occurs in the next location in the arrays - the INDEX value is the pointer into the arrays). The water is always to the right. Therefore, if it is a lake polygon the data stream moves in the clockwise direction (think about it!)

If a polygon is an island, this is automatically detected by the program, which then causes the polygon to be traced in in the counter-clockwise direction. Remembering that water is always to the right, if the polygon goes in the counter-clockwise direction then water is 'projected' outwards. This makes sense, as generally islands, by definition, are not filled with water!

When AutoAsm detects an island polygon, it forces the colour to 254 so that it can be handled correctly.

For illustration, load the TS_demo project and read in the bitmap. You will see a number of islands marked in red. If you click on one and press the + button you will see that it does indeed go counter-clockwise.

Islands are automatically detected by scanning from the top of the islands polygon to the LH bitmap edge and counting the polygon lines that it passes through. Therefore it is essential that all LWM polygons are perfect (i.e. 'green banner') for island detection to work reliably.

255: LWM MARKED

When a bitmap is being read in by AutoAsm, every LWM pixel that has been found is marked with colour index 255. This ensures that any LWM point will only be found once.

# WORKING WITH OBJECTS

The Objects panel is used for creating static and dynamic objects that are linked to VTP lines. Refer to the Panels section for a concise description of the functions on this panel. To create static objects an installation of Airport for Windows is required.

## STATIC OBJECTS

AutoAsm can add static objects linked to VTP lines, for example buildings or light poles running along the edges of roads, or buildings and trees running along the sea shore. Gerrish Gray's excellent tree objects are perfect for this. This feature is similar to the default FS vector autogen, but it gives a lot more flexibility. Note that the library objects used for vector autogen can easily be converted into api files for use with this feature; several are included in the main AutoAsm directory.

To add objects, you apply additional settings to lines in the Objects panel. You can apply multiple object types to a line, for example buildings, trees and telegraph poles. When you've finished applying object settings the program then generates a scenery file (or files) that can be loaded into Airport ready for compiling into scenery.

There seems to be a limit to the number of objects that Airport can handle. For this reason AutoAsm will automatically split the Airport files into two or more files as required, so there is no major limit to the number of objects you can add. The CALL AIRPORT button automatically calls Airport and loads the files that were created. Simply compile each file in Airport and then delete it until you run out of files.

If you reduce some object settings then the number of files may be smaller and therefore the number of Airport bgl files would also be smaller. If this occurs be very sure to delete the excess bgl files previously created.

Objects don't have to be closely linked to roads. You could, for example, add a large circular, zero-width VTP line and apply trees with large offsets, thus creating a forest with thousands of trees (frame rates permitting!) By editing the line settings you could radically change the forest with a few key strokes.

Where appropriate, settings have two numbers separated by a comma. The two numbers specify a minimum and a maximum value. The program automatically chooses random values between the two limits, thus creating a more natural random effect.

On the left of the Objects panel you will see two large text squares and a single text slot at the top. The upper large square is the Object Palette. Here you can store ready-to-use object definitions. Each object definition is a text string with a series of settings separated by one or more spaces. You can assign multiple object definitions to a given VTP line.

The lower box shows the object definitions assigned to the selected line or polygon. If you double-click on a line in the map display the box will display all the objects assigned to the line. If necessary you can scroll down to see more objects.

The single-line text box at the top is where you enter parameters for a new object or edit parameters for an existing object. If you double-click on an object line in either of the large text boxes the object parameters will be copied to the top line for editing. If you click on OK then the edited settings will be copied back to the line you originally double-clicked.

A line of labels above the editing slot at the top of the screen indicates the parameter names (e.g. 'density group …..' ). The labels are different according to whether the object is static or dynamic. Click the Toggle Labels button to choose the appropriate labels. The first label at the left indicates the type ( 'static object type' or 'dynamic object type' ).

Parameters in the top line are separated by one or more spaces. Each parameter (e.g. offset) is split into two numbers, separated by a comma. The two numbers are a minimum and a maximum value. AutoAsm randomly chooses a value between the two, so that objects have a natural variation. If you set the two values equal then there will be no variation. Look at some of the object strings that come with the demo projects for examples.

Here's an example:

*buildings|cottage.api 5 3 2, 5 left .5, .9 2, 3 3, 8 20, 60 23, 66 10, 38 17, 20 cottage*

You can add any number of spaces so that the parameters form neat columns under each parameter label. The parameters are as follows:

*object type:* this is the path and file name of the API relative to the API directory. The API directory is set by the ‘object path’ slot at the top right of the panel. An example should clarify this.

Suppose you want to store your Airport API’s at c:\airport\api and you also want to store buildings in a sub-directory named 'buildings'. Enter this path into the ‘object path’ slot. Then, if you enter 'buildings\house.api' under ‘object type’, the full definition for the API would be c:\airport\api\buildings\house.api

If, instead of entering a complete path to an API, you enter the reserved word ‘building’, a FS2000-style building will be created.

*density:* this sets the frequency at which the object appears along the VTP line. A suitable density for a telephone pole might be 50. Increasing the value increases the frequency (smaller spacing between the objects).

*group:* not currently implemented. Just place a zero for this parameter.

*offset:* the distance of the object from the VTP line. A setting of 5 to 10 would be suitable for a building that is to be close to a road.

*side:* determines which side of the road the object will be placed. Values are *left, right* or *both.*

*size:* sets the size of the object. The effect will depend on the actual object and is the same value that appears when you open an object in Airport.

*parameters:* general parameters used for expansion. There are six pairs of parameters. When using Gerrish Grey’s trees, some of these parameters are used for setting tree types.

*comments:* enter any suitable comments here.

Example procedure: to add some telegraph poles along a road and some trees along a shoreline.

1. Read in a project that has some roads and shorelines.
2. In the Objects panel, set the object path text box to the api path in Airport e.g. c:\airport\api
3. Place the telegraph pole api into the Airport api sub-directory that you set in step 2.
4. In the Objects panel, create settings for telegraph poles as follows:
5. In the entry slot at the top of the panel, enter this text string:

TelegraphPole.api 50 0 0.7,0.7 left 0.5,0.5 0,0 0,0 0,0 0,0 0,0 0,0 telegraph pole

Add sufficient spaces so that the items line up with the labels at the top - this is not essential, but it makes it easier to read the settings.

The settings are:

TelegraphPole.api - the name of the api file.

50 - the density setting. Increase this to place the poles closer together.

0 - the group setting. This is not normally used, but a zero setting causes the objects to be placed at equal intervals along the line, which is appropriate for telegraph poles.

0.7,0.7 - the offset from the line. The two values are equal so there will be no random variation i.e. all poles will be the same distance from the road.

left - places the poles to the left of the road.

0.5,0.5 - the object size. As both values are equal all the poles will be the same size.

0,0 0,0 0,0 0,0 0,0 0,0 - the six additional parameters. As this object does not use them we may as well set them all to zero. Note: if these parameters are not used they must still be placed in the text string.

telegraph pole - the comments line.

6. Click on the ADD button just to the right of the OK button. This adds the object string you just created to the object palette. You can then add this to any objects as required.
7. Enter a new string into the top slot to define some trees:

trst20_3.api 8 2 3, 5 left 3,8 0, 0 2, 2 304, 304 3, 10 1, 1 1, 1 trees

This calls up one of the API's from Gerrish Gray's excellent trees add-on. Of course, the tree API's must be placed in the same sub-directory as TelegraphPole.api. Here some of the extra parameters are used, for example 304, 304 defines a tree type. Note that the trees will be placed to the left of the line. If applied to a shoreline they must be on the left, otherwise they will be in the water!

8. Click on ADD to add this entry to the object palette. Optionally save the project.

We now have our telegraph pole and tree definitions stored in the object palette. Now we have to apply them to the roads and shorelines. We'll apply the telegraph pole first.

9. Click on a road line to select it.

10. Click on the telegraph pole entry in the object to select it and click on the ADD button that's between the two DELETE buttons. This copies the telegraph pole settings to the lower POLYGON OBJECTS box. As you probably guessed, this box contains the settings that are applied to a specific line. Now it contains just the telegraph pole settings.
11. With the road line still selected, click on UPDATE POLYGON. The telegraph pole settings are copied to the selected road. All roads with the same colour will have telegraph poles.
12. Repeat the above procedure, but this time select a shoreline and copy the tree settings to the POLYGON OBJECTS box. If you don't want telegraph poles running along the shorelines, select the telegraph pole settings in the POLYGON OBJECTS box and click on the DELETE button immediately to the right of the box. Finally, with the shorelines still selected, click on UPDATE POLYGON.

Now your telegraph pole and tree settings have been applied to the appropriate lines. We're ready to create the Airport scenery files.

13. Click on WRITE AIRPORT FILE. The Airport file (or files) is created. Each object is marked as a red dot on the map display, so you can get a good idea of where the objects have ended up.
14. Click on CALL AIRPORT. Airport is opened with the new scenery file or files already loaded. Note: the 'airport path' text box must be correctly set e.g. c:\airport
15. Ensure that the scenery path in Airport is correctly set and then click on the compile icon. Airport then compiles the scenery and places the scenery bgl file in Flight Simulator.
16. Delete the scenery window in Airport. If more than one scenery files were created by AutoAsm then you will see the next scenery window. Compile and delete each succeeding scenery window until all of them have been processed. You can now run FS to check the scenery. You should see the telegraph poles running along the selected roads and trees along the shoreline.

We'll now change a setting to illustrate how you can edit the object types.

17. With the map loaded, open the Objects panel and double-click on a shoreline that has trees assigned to it. The tree settings appear in the POLYGON OBJECTS text box.
18. Double-click on the tree settings. They now appear in the editing slot at the top of the panel. Edit the string to make any changes you want e.g. increase the density setting.
19. When you're satisfied, click on the OK button. The edited settings are copied back to the POLYGON OBJECTS box.
20. With the shoreline still selected, click on UPDATE POLY. Your new, edited settings are copied to the shoreline.

## Additional parameters

For most objects the six additional pairs of parameters are not used, but there are two special cases: FS2000-style building objects and Gerrish Gray's trees.

If, instead of specifying an api file, you use the reserved word 'building', then FS2000-style buildings will be created. An example:

building 1.7 1 3, 5 both .7, 1.5 1,4 5,8 9,12 13,16 1,1 1,1 building

For the building object, the first four of the six extra parameter pairs are used to specify the building textures in this order: Ground level 1, Level 2, Level 3 and Roof level. In this example the ground levels would randomly have textures in the range 1 to 4, while the roof levels would have textures in the range 13 to 16.

This example calls up some Gerrish Gray tree api's:

trees_1\trst20_3.api 5 2 3,5 both 5, 10 0,0 2,2 304,304 3,10 305,310 8,12 trees

In this example the api is located in the trees_1 sub directory. If the object path is set as 'c:\airport\api' then the full api path would be c:\airport\api\trees_1\trst20_3.api

The parameters in this example are as follows:

3,5: the Scale setting (it defines the area covered by the trees and not the actual tree size).
0,0: the hemisphere (0=N, 1=S). Both values should be the same.
2,2: the season (range 1 to 4). Both values should be the same.
304,304: the tree type for tree 1. In this case tree 1 will always be type 304.
3,10: scale value for tree 1 (the actual tree size). This will give a random range between 3 and 10.
305,310: the tree type for trees 2 and 3. This will give a random range between 305 and 310.
8,12: scale value for trees 2 and 3. This will give a random range between 8 and 12.

# DYNAMIC OBJECTS

In addition to static objects, you can assign dynamic objects to VTP lines, the most common application being moving road traffic as well as ships and boats. You assign dynamic object settings in exactly the same way as for static object settings though of course many of the parameters are different. You can select appropriate parameter labels at the top of the OBJECTS panel by clicking the *Toggle labels* button.

IMPORTANT: The object type must be preceded by 'dyn_', for example 'dyn_LIB_CESSNA_ID' or 'dyn_00000122'.
If it is not preceded by 'dyn_' AutoAsm will not recognise it as a dynamic object.

You can select several different types of dynamic object as follows:
1. Objects stored in a library with a short GUID number e.g. dyn_00000125 In this example the short 8-digit number specifies the GUID number for the object in the library. You will find instructions on how to build your own dynamic object libraries below.
2. Library objects as above but with a long (full) GUID number.
3. Default Microsoft SDK objects e.g. dyn_LIB_NEW_FUEL_TRUCK_ID These objects include a Cessna which can fly at a fixed altitude (dyn_LIB_CESSNA_ID).
4. Train objects e.g. dyn_train_00000126 You can assign how many sections the train has. The GUID number specifies the library object for the train carriages.

Here's an example parameter string for a library object with a short GUID number:

| dynamic object type | density | speed | offset | side | direction | spacing | altitude | comments |
|---|---|---|---|---|---|---|---|---|
| *dyn_00000122* | *10* | *5* | *3* | *left* | *1* | *3* | *0* | *dynamic red car* |

with a long (full) GUID number:

| dynamic object type | density | speed | offset | side | direction | spacing | altitude | comments |
|---|---|---|---|---|---|---|---|---|
| *dyn_12345678,23456789,3456789A,00000122* | *10* | *5* | *3* | *left* | *1* | *3* | *0* | *red car* |

and for an SDK object:

| dynamic object type | density | speed | offset | side | direction | spacing | altitude | comments |
|---|---|---|---|---|---|---|---|---|
| *dyn_LIB_CESSNA_ID* | *2* | *2* | *0* | *left* | *1* | *3* | *2000* | *dynamic Cessna* |

and for a train object with six carriages:

| dynamic object type | density | speed | offset | side | direction | spacing | altitude | comments |
|---|---|---|---|---|---|---|---|---|
| *dyn_train_00000126* | *6* | *4* | *5* | *left* | *1* | *3* | *0* | *dynamic 6 carriage train* |

Note that for trains the density sets the number of carriages and not the number of trains. Each section of rail (i.e. the VTP line) will have two trains, one starting at each end of the line.

The parameters are as follows:
1. The object type. Any short GUID numbers should be eight digits without any leading zeroes. Long GUID numbers should have four groups of eight digits with each group separated by a comma.
2. Density - the number of instances of this object assigned to the VTP line. The instances of this object will be randomly placed along the VTP line. If it is a train object it defines how many sections (carriages) it will have, so the larger the density the longer the trains.
3. Speed, in arbitrary units. Increase the value to increase the object speed. Typical values for road traffic would be 3 or 4.

4. Offset - the offset between the object's path and the centre of the VTP line (a typical value might be 5).
5. Side - determines whether the object offset is to the right or to the left of the road centre.
6. Direction. A value of 1 (one) will cause the objects to move from the beginning of the road to the end. A value of 0 (zero) will cause the objects to move in the opposite direction.
Parameters 4, 5 and 6 can be used together to create traffic moving in both directions and on separate lanes.
7. Spacing. A larger value will cause objects to be more widely separated. The larger the value, the greater the time period between individual objects being spawned. Occasionally two objects will be placed on top of each other. Increasing this value should reduce the probability of this happening, though objects will take a bit longer to spawn when the scenery area is activated. (Note: objects are now randomly spawned along the length of the line instead of just at the beginning, so this parameter is less important).
8. Altitude. If this value is set to zero objects will move along the terrain surface. FS does not adjust object elevations very frequently (typically every two seconds) so if a car is climbing a hill it will repeatedly vanish and re-appear as FS adjusts the elevation. By using suitable altitude settings you can create aircraft flying defined paths. However, with this version of AutoAsm the aircraft cannot land or bank when making turns.
9. Comments e.g. 'dynamic train'.

Here's an example of a two-lane road. The side and direction settings are different and the offset ensures the cars in the two lanes do not collide!

| dynamic object type | density | speed | offset | side | direction | spacing | altitude | comments |
|---|---|---|---|---|---|---|---|---|
| *dyn_00000122* | *10* | *5* | *3* | *left* | *1* | *3* | *0* | *dynamic red car* |
| *dyn_00000122* | *10* | *5* | *3* | *right* | *0* | *3* | *0* | *dynamic red car* |

You can add many object types to a road to increase the variety and realism, but don't forget that any scenery area has a maximum 'allowance' of 220 objects due to a limit in Flight Simulator. Also, remember that if you increase the objects on a given road, all other roads (i.e. VTP lines) that have the same colour index will also have their number of objects increased.

## DYNAMIC LIBRARY OBJECTS

Apart from a small selection of SDK default objects (e.g. the Cessna), most objects will be called from one or more dynamic object libraries. The specific object is defined by a unique GUID number. GUID numbers consist of four words of eight hex characters.
You can enter the full GUID number e.g. 12345678,23456789,3456789A,00000106
For convenience you can also use a shortened GUID number. As well as requiring less typing the other parameters don't get pushed over to the right by a long GUID number!

Short GUID numbers must follow this convention:
1. The first three GUID words must be 012345678 023456789 03456789A (note: these include a leading zero as required by fsregen). I chose these so that they are easily remembered!
2. The fourth GUID word can have any value, although this AutoAsm release uses values in the range 100 to 1FF, so it would be best to avoid that range. To avoid clashes, any users are welcome to contact me by email and I will assign a suitable range.
When assigning an object type in AutoAsm, do not include the leading zero (e.g. dyn_10002789). However, when entering the values in fsregen you should add a leading zero (e.g. 010002789).

## THE AUTOASM DYNAMIC OBJECTS LIBRARY

I’ve included a library named AA_dyn_lib.bgl. To use it just copy the file into a suitable scenery directory e.g. c:\FS2004\addon scenery\scenery. The vehicles are fairly crude but they will get you going. You can of course build your own vehicle objects. The objects included in the library are as follows:

dyn_00000120 orange car (the full GUID is 12345678, 23456789, 3456789A, 00000120)
dyn_00000121 small blue truck
dyn_00000122 red car
dyn_00000123 white van
dyn_00000124 bus
dyn_00000125 van
dyn_train_00000126 train (the density setting sets the number of carriages, not the number of trains).

The distance between train carriages is set by timing. Unfortunately the time period can only be set in integer seconds, so the timing is limited. As a result there may be gaps between individual carriages of a train. You can reduce or eliminate the gaps be setting a slower speed for the train - or by making longer carriages in gmax!

## CREATING DYNAMIC LIBRARY OBJECTS

To create a new dynamic object, follow this procedure. You need three tools: gmax, bglc and fsregen. Be sure to use the same version of bglc.exe as the one included with AutoAsm. Middleman is also highly recommended (this procedure uses Middleman). This procedure uses short GUID numbers.

1. Design the object in gmax. For obvious frame rate considerations the object should be fairly simple. It must be textured (probably due to a bug in FS, untextured objects may or may not be visible, depending on the angle of view and possibly the phase of the moon).
2. Export the object to a suitable location (from the gmax menus choose File / Export or Export Selected).
3. Locate the xxxx_0.asm file in the export location. If you named the object 'test' the file will be 'test_0.asm'
4. Before compiling the library this file must be modified to make it compatible with dynamic objects. Open the file (e.g. test_0.asm) in a text editor such as Wordpad. Locate the line similar to 'car3_topG0 label BGLCODE'. It is the first line of code after the comments. Insert the following text immediately below this line and save the edited file (use CTRL-C to copy and CTRL-V to paste):

```
; Dynamic Code
SHADOW_VPOSITION 0
SUPER_SCALE A1, 10000, 105, 16
SHADOW_VICALL A2, 18h
A1 label word
VPOSITION A3 , 10000, 105, 0
SUPER_SCALE A3, 10000, 105, 16
VINSTANCE_CALL A2, 18h
SETWRD 2Ch, 25
A3  label word
BGL_RETURN
A2  label word
; End Dynamic Code
```

Note: many thanks to Dick Ludowise for making this code available on the Avsim Scenery forum.
If you want the object to be lit at night with a headlight plane, you'll need to add three extra lines of code - see the section below (DYNAMIC OBJECTS AT NIGHT).

5. Run fsregen and click on Library Compiler. This, not surprisingly, opens the library compiler.
6. In the top slot of the compiler window, browse for the asm file created by gmax when you exported the object, e.g. test.asm. When you've selected the file, click on Open. You'll see the path to the file displayed in the top slot.
7. Edit the first three GUID words to: 012345678, 023456789, 03456789A,
8. Enter the fourth word e.g. 000000555

In this example the complete entry would be: 012345678, 023456789, 03456789A, 000000555 and you would enter its description in AutoAsm as ‘dyn_00000555’.

9. Click on the Add to List button at the right. The entry appears in the Library Object Files box.

10. If you want to place more than one object into the library, repeat steps 6 to 9. Be VERY sure you update the GUID number each time before adding to the list!

11. In the Output File slot at the bottom enter the path to where you want the library bgl file to be placed e.g. c:\fs2004\addon scenery\scenery\test.bgl

12. Click on the Create Library Source File button and then click on the Compile to BGL button.

The dynamic object library has now been created and placed into the FS scenery folder.

If you want to place multiple objects into the library file there is a problem. When compiling the asm files bglc will fail because all the asm files use the same labels in the code that was added in step 4 above. The labels in the above text are A1, A2 and A3. You will need to modify the labels. For example, you could change the labels in the first file to A11, A21 and A31, and in the second file to A12, A22 and A32, and so on. The same applies to the text below, where the variable is A4.

## DYNAMIC OBJECTS AT NIGHT

You can add a xxxx_LM.bmp texture to light up your vehicle at night e.g. red brake lights. To simulate the effects of headlights that light up the road you can add a flat, horizontal plane in front of the vehicle and map on to it a suitable texture in the xxxx_LM.bmp file. This works fine at night but in day time the flat plane will create a shadow, even if the day texture makes it invisible (i.e. transparent). To fix this problem, follow this procedure that adds three extra lines to the xxxx_0.asm file. Many thanks to Lou (Firestriker) for producing this solution on the Avsim scenery forum. It adds a time of day condition so that the flat plane does not exist in day time.

1. After exporting the object from gmax, open the xxxx_0.asm file described in Steps 3 and 4 above. In addition to adding the code as described above, you need to add three extra lines of code to suppress the shadows.

2. Identify the Node section that belongs to the flat headlight plane. The name refers to the name of the flat plane in gmax, in this case Plane02. You could check the name in gmax and search for it in the asm file. It will be toward the end of the file and comes after the vertex lists.

3. Add the three lines of text below which are marked by a line of asterixes and then continue at Step 4 above.

```
DRAW_TRI_END
BGL_TRANSFORM_END ; Transform Box02

; Node 5 - Plane02 transform:
BGL_TRANSFORM_MAT     -0.181430,27.807377,0.731340,     1.000000,0.000000,0.000000,
0.000000,3.060000,0.000000, 0.000000,0.000000,1.000000
MATERIAL 0,2 ; <255,255,255,255> DYN_TEXF.BMP;DYN_TEXF_LM.BMP;;

VAR_BASE_OVERRIDE VAR_BASE_GLOBAL ;******************
IFIN1 A4, tod, 2, 4  ;*********************

DRAW_TRI_BEGIN 90, 4
DRAW_TRI    2,   3,   1 ; poly=98 part=5
DRAW_TRI    0,   1,   3 ; poly=97 part=5
DRAW_TRI_END

A4 label word  ;*****************************

BGL_TRANSFORM_END ; Transform Plane02
BGL_TRANSFORM_END ; Transform RotateAroundX
```

## ASSIGNING DYNAMIC OBJECTS TO VTP LINES

The process is virtually identical to that for assigning static objects, but there are several global settings that are specific to dynamic objects. We'll take a look at these and then follow it up with a tutorial.

Run AutoAsm and select the Objects panel. Near the right hand edge of the panel there are two boxes, one for static objects and the other for dynamic objects. Each has a tick box at the top. If you tick both boxes, when you click on the WRITE FILES button the files for both static and dynamic boxes will be created. If only the box for dynamic objects is ticked then only dynamic object files will be generated. In general I would recommend that you generate static and dynamic object files in separate passes.

The next setting in the dynamic objects box is DENSITY. This controls the density of the path and not the objects. If you set it to 0.5 then every second point will be used to generate the path. Therefore if you change the setting from 1 to 0.25 the number of path points will be reduced to a quarter.

There appears to be a limit in Flight Simulator of about 1,600 path points. If this is exceeded then FS may lock up - generally the only way out is a Windows re-boot, which is not at all pleasant. When the files have been created the STATUS slot displays the number of path points. If there are more than 1,600 you can reduce the DENSITY setting as appropriate until there are less than 1,600 path points. Note that with low DENSITY settings the objects will not follow the roads so accurately, particularly at bends.

The next setting is LENGTH. If it is set to, say, 100 then roads with less than 100 points will not have dynamic objects assigned (the actual number of path points is fairly approximate). It's quite unrealistic for short roads to have dynamic objects, as they would be constantly appearing and vanishing. Increasing the value of LENGTH is also a good way to reduce the total number of path points.

The final setting is FILE SPLIT. In any scenery area there appears to be a limit of 1,600 path points and 220 dynamic objects. If the scenery area is large then the objects will be very sparse indeed. This setting allows the dynamic object scenery to be split into sub-areas, each with a separate bgl file. If you split the area into four then you will have approximately four times as many objects in the complete scenery area.

If FILE SPLIT is left blank then no file splitting will be performed and there will be only one bgl file. If a value of 2 is entered, then the area will be split into two, both horizontally and vertically, giving four sub-areas. Similarly a value of 3 will split the area into nine sub-areas.

## THE ISLAND DEMO

The Island demo now has dynamic objects assigned, so you can easily try out this feature. If you already have AutoAsm installed, replace the three files island.bmp (in the bitmaps directory), island.ini and island.dat (in the settings directory) with the new ones in the zip. The yellow roads on the southern part of the island have traffic assigned, and there are two trains on the railway that crosses the lower of the two bridges.

Select the island project in the usual way and read in the bitmap. Double-click on a yellow road and then select the Objects panel. You will see the dynamic objects assigned to the road in the lower large text box. Double-click on the railway and you will see that, as well as a train, several Cessnas are also assigned.

Ensure that DYNAMIC OBJECTS is check ed and then click on WRITE FILES. This will create the dynamic scenery bgl file. Click on a yellow road and then click on RUN FS. After Flight Simulator has loaded you should see that the roads in the southern part of the island are filled with busy traffic as well as a few trains and plenty of Cessnas flying past. Of course, to create the rest of the island you'll need to follow the steps in the Island tutorial. If you haven't created the island you'll still see the dynamic objects, but they'll be moving over empty sea!

## TUTORIAL: A SIMPLE DYNAMIC OBJECT SCENERY

The basic procedure for creating a dynamic object scenery is as follows:

1. In your paint program create a 512*512 8 bit colour bitmap and draw a single vertical line about half the height of the bitmap. Save it to the AutoAsm bitmaps directory with the name test.bmp
2. Run AutoAsm. Enter 'test' into PROJECT NAME and check that all other settings are suitable, particularly the settings for geographical location and size. Load the bitmap (click on READ DATA). You don't need to change the line settings as it will default to a VTP road.
3. Select the Objects panel. Check DYNAMIC OBJECTS and uncheck STATIC OBJECTS.

4. Set the dynamic settings as follows:
   DENSITY: 1
   LENGTH: 1
   FILE SPLIT: blank
5. In the editing slot at the top, enter the following: (enter the parameters, not the labels!)

| dynamic object type | density | speed | offset | side | direction | spacing | altitude | comments |
|---|---|---|---|---|---|---|---|---|
| *dyn_00000122* | *50* | *5* | *3* | *left* | *1* | *30* | *0* | *dynamic red car* |

This will generate 50 instances of the red car, so the road is going to be busy!

6. Click on the ADD button so that the settings are copied to the large box at the bottom.
7. Edit the settings to:

| dynamic object type | density | speed | offset | side | direction | spacing | altitude | comments |
|---|---|---|---|---|---|---|---|---|
| *dyn_00000121* | *50* | *5* | *3* | *right* | *0* | *30* | *0* | *dynamic truck* |

This object is a truck. Side and direction are changed, so the second set of objects will travel in the opposite direction and on the opposite side of the road.

8. Click on the ADD button. Now both parameter strings appear in the large box at the bottom. These must now be applied to the VTP line.
9. Left-click on the line in the map display to select it. The point you clicked on turns green.
10. Click on the UPDATE POLYGON button. This applies the dynamic object settings to the road.
11. Click on the WRITE FILES button. After a second the STATUS slot displays the total number of objects and path points. The bglc DOS window opens as the asm file is compiled to a bgl scenery file. When the window closes, the bgl file has been compiled and copied to Flight Simulator.
12. Click on WRITE VTP to create the road.
13. Click on RUN FS. After a few seconds Flight Simulator will open with you hovering in slew mode above the centre of the road. After a few seconds the objects start to appear randomly along the road. You should see red cars going north and trucks going south with (hopefully!) a gap between them so they do not crash. Set the FS time to dusk and you should see fairly realistic headlight effects.
14. Exit FS and run AutoAsm. It will have the same settings as when you exited. Load the bitmap and go to the Objects panel. Now we'll add a Cessna (actually, a lot of Cessnas!).
15. Edit the settings in the editing slot to:

| dynamic object type | density | speed | offset | side | direction | spacing | altitude | comments |
|---|---|---|---|---|---|---|---|---|
| *dyn_CESSNA_ID* | *50* | *3* | *0* | *right* | *1* | *30* | *1000* | *dynamic Cessna* |

This will create 50 Cessnas flying south along the road. The altitude is in meters. Of course, if the terrain height exceeds this you'll need a larger value.

16. Double-click on the line in the map display. This copies the existing object settings to the large box at the bottom.
17. Click on ADD to copy the Cessna settings into the large box and click on UPDATE POLY to copy the settings to the VTP line (as always, the line must be selected by clicking on it).
18. Create the scenery (click on WRITE FILES) and return to Flight Simulator. Now you should see that the air above the road is filled with an armada of Cessnas. The total object count is now 150, so we're still comfortably below the maximum limit of 220. Feel free to add more objects!

## LIMITATIONS

Sadly the number of traffic objects you can expect to see is limited. The limitations are in bglc and Flight Simulator, I hasten to add, and not in AutoAsm! The limitations are:

1. When compiling the dynamic scenery, bglc will issue error messages if the distance between a label and a reference to it exceeds 65000 characters. I optimised the AutoAsm code to reduce the likelihood of this occurring, but it can still happen in dense sceneries. If this occurs you can reduce the DENSITY setting or increase the LENGTH setting.
2. If the number of path points exceeds about 1,600 in the active scenery area then Flight Simulator may lock up. Usually the only way out is to reboot Windows. After creating the dynamic scenery (WRITE FILES button) the number of path points is displayed in the Status slot.
3. The total number of objects available in a scenery area appears to be limited to about 220. If this number is exceeded by, say, one hundred, then roads that come later in the list will not have any dynamic objects. If the number is exceeded by a large amount then FS may lock up, though it may not be necessary to reboot Windows. The number of objects is displayed in the Status slot.

The SPLIT FILES feature is designed to increase the number of available objects and path points, but the effectiveness of this is somewhat limited by yet another limitation - or error - in Flight Simulator as follows:

4. If the SPLIT FILES feature is used, the load and unload bounds for each sub-square are set to be equal. This means that as you fly from one sub-square to the next, the objects in the previous square should vanish just as the objects in the next square appear. When flying westwards or southwards this is exactly what happens. But when flying northwards or eastwards you have to fly about three minutes of arc (about 3 miles) before the objects in the new square appear. If the squares are, say, 5 minutes across, then for a lot of the time no objects will be visible. This may well be caused by a bug in Flight Simulator.

# SETTING LAKE ELEVATIONS

AutoAsm provides two distinct methods for setting lake elevations so that they sit naturally in mountainous terrain. The first is the polygon tool in the Editor. It allows water polygons to be set to specific elevations. If there are not too many lakes and their elevations are known, then this is the preferred method.

The second method relies on the use of an elevation bitmap to set the lake elevations. Of course, if the terrain mesh was created with AutoAsm then the elevation bitmap already exists. If not, then it would have to be created, which might not be easy. AutoAsm reads in the elevation bitmap and, when creating the LWM asm file, sets the elevation of LWM water polygons as defined by the elevation bitmap. As this process is automatic it is particularly useful for scenery areas that have many lakes in mountainous terrain.

Method 1: the Polygon tool

Of course, individual water polygons can be set to specific elevations in the normal way. However, if the polygons are all of the same colour then they will have the same elevation (line settings are mapped to a specific colour index rather than to a specific polygon). The Polygon tool solves this problem by changing polygons to new, unique colour indexes and at the same time setting them to unique elevations.

The procedure is quite simple, as follows:

1. Load the LWM bitmap into the Editor.
2. Click on the EDIT button to enable the editor tools.
3. Type the desired elevation into the SET ELEVATION box near the top right of the Editor panel.
4. Enable the check box just to the right of the POLY button. This enables the polygon elevation to be changed. If it is not checked the polygon's colour will be changed but it will keep its original elevation.
5. Click on the POLY button to select the polygon tool.
6. In the map view, left-click on the water polygon. Note that if you right-click it will delete the entire polygon!

The polygon is changed to a new, unique colour index and its elevation is changed to the set value. You can repeat this process for many other lake polygons, but it is limited by the total number of available colour indexes (about 250). Each time you use this tool the new colour index number is displayed. The actual displayed colours are determined by the palette of the bitmap blank.bmp.

Method 2: Elevation bitmap

The elevation of each water polygon can be set automatically by reading an elevation bitmap. The bitmap format is 8 bit grey scale and it can be any size providing the X and Y sizes are a multiple of 4. The elevation is proportional to the pixel brightness. If the project name is 'test' then the elevation bitmap should be named 'test_elev.bmp' and it should be placed in the bitmaps sub-directory. If you have created terrain mesh for the project in AutoAsm then the elevation bitmap will be suitable. In this case, copy the elevation bitmap from the terrain project sub-directory to the bitmaps sub-directory and rename it to e.g. 'test_elev.bmp', where 'test' is the project name.

To enable the elevation bitmap method, tick the BMP ELEV box on the Main panel.

To set this method up, click on SET. A small window opens with several options. The SCALE value sets the maximum elevation (for a pixel brightness of 255). For example, if you enter '5000' then the pixel brightness of 255 will correspond to an elevation of 5000 meters.

The four text boxes set the geographical bounds of the elevation bitmap, using the standard AutoAsm convention. The bounds of the elevation bitmap can be larger than the bounds of the main project.

The FROM MAIN and FROM TERRAIN buttons allow you to copy the bounds from the Main panel (i.e. the project bounds) or from the Terrain panel.

After enabling and setting up this method, read in the project bitmap in the usual way. The elevation bitmap will be read in and the lake elevations set accordingly when the LWM scenery is created (WRITE LWM button). You can run the scenery through LMWViewer to check that the lakes have differing elevations as required. Just place the pointer over a lake and LWMViewer displays the elevation.

# IMPORTING DATA

To create scenery AutoAsm always reads in data from a bitmap. However, several data formats can be imported. The import tools automatically convert the data to a standard LWM/VTP bitmap that AutoAsm can then read in in the usual way. The import tools are located on the Tools panel.
Note: for importing to work the blank.bmp bitmap must be present in the bitmaps sub-directory.

The supported file types are:

1 .dat (available at http://www.ngdc.noaa.gov/mgg/shorelines/shorelines.html)
2 .mp (also known as Polish format)
3 .bln
4 .ung

Note: several formats can be downloaded from the above site. Unfortunately the files always have the .dat suffix. The suffix informs AutoAsm of the file type, so you may have to change it to the correct type.

Basic procedure:

1. Copy the file to be imported into the AutoAsm import sub-directory.
2. Before importing a file you need to decide on the bitmap size. On the Tools panel, enter the X, Y pixel size into the O/P BITMAP SIZE text boxes. You could start with, say, 1000*1000 to speed up the initial setting up but before actually making the final scenery you'll probably need a higher resolution such as 8000*8000.
3. Enter the name of the file you want to import into the FILE NAME box. You can also browse for it. The file name must have the correct suffix e.g. 'test.bln', as the suffix informs AutoAsm of the data format.

Before importing the file the geographical bounds must be set. You could manually enter the bounds into the CENTRE POSITION and SIZE(DEGREES) boxes, but normally you will use the SET BOUNDS button to do it automatically.

4. Enable the LOAD ALL tick box to ensure that all lines and polygons will be checked.
5. Click on the SET BOUNDS button. After a few seconds the geographical bounds will appear in the four text boxes. You can use the SET MAIN to copy the bounds to the Main panel (the project bounds). Now we're ready to import the file.
6. Click on the LOAD button. Depending on the format, you will see an indication of the number of points as the data is read in.

When the data is imported you will see the lines and polygons that represent roads and shorelines etc You can immediately save the bitmap to a named file from the Editor panel. If you don't immediately see any lines, try zooming out or click several times on the FRAME button in the Editor panel (this gives a full-frame view of the bitmap).

## Parameters

The parameter settings are particularly useful when importing .mp files (Polish notation). Each line or polygon has a separate data block with a header that defines the data type e.g. 0x0001 (major road) or 0x0018 (stream). You can use the parameter settings to filter out only the lines or polygons that you want. For example, if you enter '0x0018' under 'LINES' while leaving the other boxes blank, then only streams will be loaded.

If you enable the LOAD ALL check box, then the parameter settings will be ignored and all lines will be loaded.

To load only selected types:

1. Disable (uncheck) the LOAD ALL checkbox.
2. Enter the types into the PARAMETER boxes (the first six are for polygons, the others are for lines). Leave blank any boxes that don't have type settings. Using the above example, if you enter '0x001' and '0x0018' into two boxes under LINES then only major roads and streams will be imported.

All lines of a given type will be of the same colour, so it would be easy to assign suitable line types in the LINE TYPES panel.

After importing the file you can view the lines and polygons in the map area, and you can save the bitmap from the Editor panel. You may need to experiment with the bitmap size settings. If you zoom in and carefully examine the lines and polygons, it will be fairly obvious if you need to increase the bitmap size. But if the bitmap size is larger than necessary more memory will be consumed and some operations (e.g. map redrawing) will be slower.

Normally the geographical bounds setting will be generated automatically and will cover the entire data set in the file. However, if you want to create scenery from a subset of the data, you can adjust the bounds settings manually e.g. by decreasing the size settings. If you decrease the size then the bitmap size could also be reduced.

If you have two data files that cover the same area (e.g. one with roads, the other with shorelines), you can merge the data as follows:

1. Set the bounds and import the first data file by clicking on LOAD.
2. Change the import file name to that of the second file.
3. To merge the second file, click on MERGE. This will import the second file without clearing the data from the first.

# CREATING LANDCLASS

You can cover your project area with new landclass, either by loading a landclass bitmap or painting new landclass directly over the LWM/VTP lines in the Editor. For a concise description of the landclass functions, refer to the Panels section.

A landclass or terrain mesh sub-project has a special structure. Each sub-project has a sub-directory inside the terrain sub-directory. The landclass bitmap is always named 'textures.bmp'. Therefore, if the sub-project is named 'test' then the full path to the bitmap might be c:\autoasm\terrain\test\textures.bmp. Note that the sub-directory can be used to generate both landclass and terrain mesh if it contains the two bitmaps textures.bmp and terrain.bmp.

A landclass bitmap must be 8 bit colour format. The colour index of any pixel directly specifies the corresponding landclass e.g. if the colour index of a pixel is 8 then the corresponding landclass type will be 8. Note that landclass bitmaps are fairly low resolution as each landclass 'pixel' is equivalent to a LOD13 square, roughly one kilometer across. Any pixels with colour index 254 will be transparent i.e. the existing default landclass will show.

To add landclass from a bitmap to an existing LWM/VTP project the procedure would be as follows:

1. Let's suppose the name of the existing project is 'test'. In the terrain sub-directory, add a sub-directory named 'test'. Place the landclass bitmap named textures.bmp in this directory (e.g. c:\autoasm\terrain\test\textures.bmp).
2. Load the project and then go to the Terrain panel.
3. In the LANDCLASS section, enter the geographical bounds. They would normally be the same as on the Main panel i.e. the LWM/VTP scenery so that the new landclass will cover the same area, but different bounds could be used if appropriate.
4. Because we're creating landclass only, tick the LC ONLY check box (enabled). Of course, TM ONLY should be unchecked. COPY TO FS should also be checked.
5. Enter 'test' into the PROJECT and OUTPUT FILE text boxes. The scenery file will then be test.bgl.
6. Check that the WORKING DIRECTORY and SCENERY BGL PATH are correctly set and then click on MAKE SCENERY.

In a few seconds the scenery bgl file is created and copied to FS. During this process several Microsoft tools are called and you will see their DOS windows briefly appear. Finally click on RUN FS. Flight Simulator will open with the aircraft positioned above the centre of the scenery area. You should see your new landclass scenery.

You can create a new landclass bitmap or edit an existing one in the Editor. To create a new landclass bitmap:

1. Load the LWM/VTP bitmap into the Editor.
2. In the Terrain panel, check that the landclass bounds are the same as for the main project.
3. In the LANDCLASS section, click on NEW. This creates a new blank bitmap.
4. Enable the DRAW LC check box. When checked, you can paint landclass directly onto the bitmap by left-clicking.

The SIZE text box controls the size of the landclass 'brush'. A value of one or zero causes each click to draw a single pixel. Values should be integers. As you increase the value the size of the brush increases so that a single click will cover multiple pixels with the selected landclass.

The TYPE text box sets the landclass type that will be painted. To paint landclass type 22, simply enter '22' into the box and click on the map where you want landclass 22. If SIZE is set to one, each time you click on the map a small filled circle indicates the landclass. The actual colour is determined by the palette of the blank_lc.bmp bitmap in the bitmaps sub-directory. If you increase the SIZE setting you can quickly fill large areas with landclass.

To disable landclass painting, uncheck DRAW LC. Now, if you click on the map, a thumbnail of the corresponding generic texture will be displayed at the top left of the map area. To save the landclass you just created, click on SAVE. To load the landclass bitmap, click on LOAD.

If you are creating both landclass and terrain mesh scenery, place both bitmaps (terrain.bmp and textures.bmp) in the 'test' sub-directory. Disable the LC ONLY and TM ONLY check boxes. When you click on MAKE SCENERY the landclass and terrain mesh scenery will be created and combined into a single bgl file named 'test.bgl'.

# CREATING TERRAIN MESH

You can create terrain mesh using a similar method as for landclass. The mesh elevation is defined by an 8 bit grey scale bitmap. The elevation of each point is proportional to the brightness of the corresponding bitmap pixel.

A terrain mesh sub-project has a special structure. Each sub-project has a sub-directory inside the terrain sub-directory. The terrain mesh bitmap is always named 'terrain.bmp'. Therefore, if the sub-project is named 'test' then the full path to the bitmap might be c:\autoasm\terrain\test\terrain.bmp. Note that the sub-directory can be used to generate both landclass and terrain mesh if it contains the two bitmaps textures.bmp and terrain.bmp.

To add terrain mesh from a bitmap to an existing LWM/VTP project the procedure would be as follows:

1. Let's suppose the name of the existing project is 'test'. In the terrain sub-directory, add a sub-directory named 'test'. Place the terrain mesh bitmap named terrain.bmp in this directory (e.g. c:\autoasm\terrain\test\terrain.bmp).
2. Load the project and then go to the Terrain panel.
3. In the TERRAIN MESH section, enter the geographical bounds.
4. Enter a suitable value into the LOD box. A value of 9 might be a good start. Higher LOD values will give higher mesh resolution, provided the bitmap has sufficient resolution. Note also that, if LOD 9 is chosen, all LOD9 squares that cover part or all of the specified scenery area must be completely filled with data from the bitmap, otherwise they will appear flat.
5. Enter a suitable value into the SCALE box. This value scales the height of the mesh. It specifies the height change in meters for a unit change of pixel brightness. For example, if SCALE = 10 then the elevation difference between pixel brightnesses 100 and 101 would be 10. The height corresponding to the maximum pixel brightness of 255 would then be 10*255 or 2550 meters.
6. Because we're creating terrain mesh only, tick the TM ONLY check box (enabled). Of course, LC ONLY should be unchecked. COPY TO FS should also be checked.

5. Enter 'test' into the PROJECT and OUTPUT FILE text boxes. The scenery file will then be test.bgl.

6. Check that the WORKING DIRECTORY and SCENERY BGL PATH are correctly set and then click on MAKE SCENERY.

In a few seconds the scenery bgl file is created and copied to FS. During this process several Microsoft tools are called and you will see their DOS windows briefly appear. Finally click on RUN FS. Flight Simulator will open with the aircraft positioned above the centre of the scenery area. You should see your new terrain mesh scenery.

If you are creating both landclass and terrain mesh scenery, place both bitmaps in the 'test' sub-directory. Disable the LC ONLY and TM ONLY check boxes. When you click on MAKE SCENERY the landclass and terrain mesh scenery will be created and combined into a single bgl file named 'test.bgl'.

# HOW AUTOASM WORKS

The input data is always in the form of a bitmap. AutoAsm can import other data formats. After importing a file it converts the data into a bitmap that can be edited, saved and read in again to create the scenery.

The bitmap must be 8 bit colour, Windows bitmap format (.bmp). For best performance the background of the bitmap should be of zero colour index. LWM and VTP features are represented by polygons and lines on the bitmap. All polygons and lines should be of single pixel width. This is particularly important for LWM polygons. The requirement for a perfect LWM polygon is simply stated: every pixel must touch exactly two other pixels of the same polygon.

The line colour is very important. Each line and polygon has a set of parameters (usually referred to as 'settings') which define how it will appear in Flight Simulator, for example width, texture type and elevation. More specifically, each set of parameters is assigned to a particular colour index. This means that all lines with the same colour will have the same settings. The colour indexes 250 to 255 are reserved, so there are 249 possible sets of parameters (1 to 249). Look at the bitmap TS_demo.bmp (in the bitmaps sub-directory) for an example.

After creating the bitmap the line parameters have to be defined. To do this, load the bitmap into the Editor (EDITOR panel). When it has loaded, select the LINE TYPES panel. If you have an existing project loaded, you will see a number of existing settings in the large text box. This is a collection of ready-made settings that you can use. You can add your own settings to the collection.

If you first select a line in the map display (just click on the line) and then double-click on one of the existing settings, those settings are assigned to the line. Alternatively, you could enter new settings into the small boxes at the top. When you're satisfied, click on UPDATE POLYGON to assign those settings to the line.

If you want to store for future use the settings you just created, click on ADD. This copies the settings in the small boxes into a new entry in the large box. REMOVE will delete the selected settings.

To check that the new settings have 'taken', click on the line and the settings appear in the STATUS slot.

If you exit AutoAsm and run it again, the lines will still have any new settings that you may have applied. But to make the new line settings permanent, be sure to click the SAVE button on the MAIN panel. This action saves all the settings to the project.

Projects are quite simple. Basic projects consist of three files:

1. The bitmap itself. If the bitmap is test.bmp then the project name is 'test'. This is what you would enter into the PROJECT NAME slot. The bitmap must be placed in the bitmaps sub-directory.
2. The settings file (test.ini). This has the settings that define all the text boxes and tick boxes. It must be placed in the settings sub-directory.
3. The data file (test.dat). This has the settings for all the individual lines and polygons, including the settings described above and object settings. It must be placed in the settings sub-directory.

Whenever you click on SAVE, all the project settings are saved to test.ini and test.dat (assuming the project name is 'test'). Whenever you click on LOAD, the settings are loaded from those two files.

Whenever the program is exited, all the settings are saved to autoasm.ini and autoasm.dat. They are re-loaded the next time the program is run. Therefore, whenever you run AutoAsm it will have the same settings as when you last exited it. Two exceptions to this are WORKING DIRECTORY and FLIGHT SIMULATOR PATH, because these do not change with different projects. These will only change when you specifically edit them.

When the line parameters have been set we're ready to read in the bitmap in preparation for creating the scenery. On the MAIN panel, click on READ DATA. The program reads the bitmap data into a single-dimension array. It then copies and converts the data into a two-dimensional array map(x,y). When this process is complete the pixel value of any point x, y can immediately be read from map(x,y).

The program then searches for the first pixel of any line or polygon. The search starts at pixel 0,0 and proceeds to the right, searching each line and progressing downwards (like a TV raster scan). Therefore, the first point of a polygon to be found is the highest (if there are two or more points at the same height then the left-most one will be found). For example, if the polygon is a square, then the point at the top left-hand corner will be found.

The program then traces along the line as it searches for adjacent points. As each point is found its entry in map(x,y) is marked to ensure it can't be found a second time. Its x,y co-ordinates are also converted to latitude and longitude and entered into the arrays.

When the first point of a LWM polygon is found, the program scans across from the point to the left-hand bitmap edge. It counts each LWM line that it encounters ( as a matter of interest, all LWM lines that it encounters have already been found and marked with the LWM MARKED colour - think about it!) This information is used to determine whether the polygon is water or an island. If it is an island, a parameter is changed so that the polygon will be traced in the counter-clockwise direction (water polygons are always traced in the clockwise direction).

When all the polygons have been found, further processing is required. For example, if it is an LWM polygon, two extra points have to be added in each LOD13 square (one where the line enters a LOD13 square and one where it leaves the square). Up to four additional points have to be added to complete the polygons within the LOD13 square.

When the read-in process is complete the map will be displayed. The polygon colours will be the same as those in the original bitmap.

If you click on a line its settings will be shown in the STATUS slot. The point's index number will appear in the INDEX box (just above the map). This represents its position within the data arrays. If you click the + button (just to the left of the INDEX box) a number of times, the INDEX number increments and the selected point (shown as green) will move along the line. If it is a water polygon the selected point will always 'walk' in the clockwise direction as you click +. AutoAsm uses the following convention: as you 'walk' along a water polygon, the water is always to the right. Therefore a water polygon will always go in the clockwise direction - in effect it is 'projecting' water into the polygon, which is appropriate for a lake or the sea.

However, if the polygon is an island (a polygon inside a polygon), then it goes in the counter-clockwise direction. It is still defined as a water polygon, but because of the counter-clockwise direction the water is 'projected' outside the polygon. The island appears as land, not because it is defined as land, but because the water is 'projected' outwards, and so the underlying landclass will appear.

You will see that if any islands are present they are coloured red. You will also find that they go counter-clockwise.

If an island is small and fits completely inside a LOD13 square without touching the side, then the geometrical processes used by AutoAsm do not work. For this reason such islands are detected and processed differently. Unlike normal island polygons, these islands are defined as land and in effect 'painted' over the water. This explains why small islands may appear green when viewed in LWMViewer.

If the AUTO FILL option is enabled, you will see that all LOD13 squares that lie inside water polygons with no LWM lines passing through them have a point in each corner, thus ensuring that large lakes and sea areas are correctly filled with water.

# PANELS

This section gives a functional description for all the buttons and readouts in the various panels.

## Fixed Panel

This panel is always visible and includes the map display.

*READ DATA* button: reads in a bmp image file and generates the map display.
*WRITE LWM* button: converts the map data to an LWM asm file in the working directory (working directory is set on the MAIN panel).
*WRITE VTP* button: converts the map data to a VTP asm file in the working directory.
*FINISH* button: saves the settings to autoasm.ini and exits AutoAsm.
*COMPILE* check box: when checked, calls bglc to compile the asm file into a bgl scenery file.
*TO FS* check box: when checked, copies the bgl file to Flight Simulator.
*AUTO RUN* button: automatically reads in the image file and creates the asm files (and scenery files if COMPILE enable selected).
*FILL* button: fills in LOD13 squares with water as required. If AUTO FILL (MAIN panel) is enabled, this is done automatically when reading in a bitmap.
*LWMViewer* button: calls LWMViewer (many thanks to Jim Keir). LWMViewer loads the scenery and displays it.
*FULL SCREEN* check box: if ticked, LWMViewer will be opened in full screen.
*RUN FS* button: calls Flight Simulator and places the aircraft in slew mode at the position selected in the AutoAsm map display ( left-click on a line or polygon to select the position ).
*DIAGNOSTICS WINDOW* button: displays diagnostics information, for example a list of objects added to a line or a list of bridges that were created. Click to open and click to close again.
*+ button* (next to INDEX slot): increments the selected point. Each time you click it the selected point will move one position clockwise (if it's a water polygon). The INDEX slot indicates the index number (actually the index position inside the program arrays). ++ moves the point in steps of ten. - and - - are similar, but decrement the index position.
*NEXT POLY* button: jumps to the next polygon or line. The ++ button to the left jumps in steps of ten polygons.
*STATUS* slot: indicates status or error messages as appropriate. If you click on a line in the map display it displays the parameter string for that line.
*INDEX:* indicates the index of the selected point inside the program arrays.
*POLYGON:* indicates the polygon number of the selected polygon and also the colour index number of the polygon.
*N8X N8Y:* indicates the LOD8 square number for the selected point.
*N13X N13Y GRP13:* The first two numbers indicate the number of the LOD13 square inside the LOD8 square (range 0 to 31). GRP13 indicates the number for each line segment that passes through a LOD13 square.
*VTPx VTPy X13 Y13:* the first pair of numbers are the co-ordinates of the selected point inside the LOD8 square (only used for VTP lines). The second pair of numbers are the co-ordinates for the selected point inside a LOD13 square (range 1 to 255).
*Array size LONG LAT:* the first number is the maximum index value used. The memory setting must be larger than this value or overflow in the arrays will occur. LONG and LAT indicate the geographical position of the selected point in decimal degrees.
*ZOOM* buttons: + increases the zoom for the map display. ++ increases by a larger amount. - and -- decrease the zoom in the same way.
Pan buttons: use these buttons to pan across the map display. + moves your viewpoint to the right while ++ moves a larger amount. The same applies to the - and -- buttons.
SAVE button: saves the current zoom and pan values

*RECALL* button: returns the map display to the same zoom and pan values as when the SAVE button was pressed or AutoAsm was last exited.

*CENTRE* button: centres the map display on the selected point. You can also centre the map display by clicking on the point while pressing the SHIFT key.

*BITMAP X, Y* readout: for the selected point, it indicates the corresponding pixel position on the input bitmap.

## MAIN Panel

*WORKING DIRECTORY:* enter the path to your working directory (usually it will be your main AutoAsm directory). E.g. c:\autoasm

*SCENERY BGL PATH:* where your scenery files will be placed in Flight Simulator e.g. c:\fs2004\addon scenery\scenery

*FLIGHT SIMULATOR PATH:* the main Flight Simulator directory e.g. c:\fs2004

*PROJECT NAME:* enter your project name here. If the project name is test then the bitmap must be test.bmp

*OUTPUT FILE:* the name used for the output files. Usually it will be the same as the project name. If the output name is test then the files created will be: test_lwm.asm test_vtp.asm test_lwm.bgl test_vtp.bgl

*LOAD button:* loads the previously-saved settings for the current project.

*SAVE button:* saves the settings for the current project. If the project name is test then the text files test.ini and test.dat will be saved to the settings sub-directory.

*READ DATA button:* reads in the image file (this button is duplicated here for convenience).

*ISLANDS* check box: if enabled, during bitmap read-in islands (polygons inside polygons) are automatically detected and processed accordingly.

*FIX PIXELS* check box: if enabled, during bitmap read-in some pixel errors will be automatically fixed. Note: if it is a 'green banner' bitmap (zero LWM errors) then this can be disabled.

*LWM ONLY:* if checked, only LWM lines are loaded from the image file. Useful when trouble-shooting LWM polygon problems.

*AUTO CULL:* when reading in an image, automatically culls points if the LWM 317 point limit is exceeded. This should normally be enabled.

*AUTO FILL:* when reading in an image, automatically fills in LOD13 squares with water as required.

*IMAGE SIZE:* indicates the image pixel size after a bitmap has been read in.

*CENTRE POSITION:* enter the geographical position of the centre of the input bitmap. Use decimal degrees, positive for east and north, negative for west and south.

*SIZE (DEGREES):* enter the geographical size of the scenery area covered by the input bitmap. X is the east-west size, Y is the north-south size. Enter in decimal degrees.

*READ-IN BOX:* this determines how much of the image bitmap will be read in. Imagine a rectangle drawn on the image - only lines inside the rectangle will be read in. The two X numbers set the LH and RH pixel co-ordinates respectively of the rectangle. The two Y numbers set the top and bottom pixel co-ordinates. This can speed up the read-in process considerably if you are only working on a part of the scenery.

To read in the entire image bitmap, set all four to zero or blank.

*SEARCH RADIUS:* if AutoAsm encounters a break in an LWM polygon, it will search a defined area for pixels with the same colour index (i.e. for the other side of the break). The value input sets the size of the box that will be searched. If you set 5, then a square 10 pixels across will be searched, centred on the last pixel to be found. There is a separate setting for VTP lines and LWM polygons. Note: if it is a 'green banner' bitmap (zero LWM errors) then the LWM SEARCH RADIUS is not required.

*INFO:* do not press this button.

*BMP ELEV* check box: if enabled (checked) the elevation for each water polygon will be read from an elevation bitmap (if the main LWM/VTP bitmap is test.bmp, the elevation bitmap must be test_elev.bmp, placed in the bitmaps sub_directory).

*BMP ELEV SET* button: opens a window for the bitmap elevation settings. The bounds for the elevation bitmap are set by the four text boxes (the format is the same as those on the Main panel). The FROM MAIN and FROM TERRAIN buttons allow you to copy the bounds settings from the Main panel or the Terrain panel. If you copy from the Terrain panel, the SCALE setting is also calculated and placed in the text box.

The SCALE setting scales the lake elevations that will be created. The pixel values of the elevation bitmap have the range 0 to 255. The SCALE setting specifies the maximum height. For example, if you enter 1000 then for any pixels on the elevation bitmap that have a brightness of 255, the corresponding lake elevation will be 1000 meters.

## LINE TYPES Panel

This panel is used to assign the various parameters to lines and polygons. When you have defined all the line types, remember to press PROJECT/SAVE (on the MAIN panel) to make the changes permanent.

Large text box on the left: this contains a list of FS2002-type line textures. Scroll down until you find the desired texture and click on it to select. Click on USE and the texture number will be copied to the 'type' text box to the right. You can also type in FS2004 texture numbers.

Large text box on the right: this contains a list of pre-defined settings that you can choose from (the palette). To apply settings to the selected line in the map display, double-click on one of the text lines. You can add a setting from the small editing boxes by clicking on the ADD button.

The small text boxes running along the top right are where you edit values as required. If you double-click on a line in the map display its settings will be copied into these boxes.

*'type'* text box: enter the desired line texture number or click on USE to copy it from the large text box on the left. If you enter a number in the range 1 to 255 then the corresponding landclass texture will be applied (for use with the VTP area polygon type).

*'water/land':* defines the polygon to be water or land (only used for LWM polygons). Click on the box to toggle the value.

*'lwm/vtp':* defines the line type. It can be 'lwm', 'vtp', 'both' or 'area' ('both' means it defines a LWM polygon and also a VTP line, usually a shoreline). Click on the box to toggle the values.

*'width':* sets the width of a VTP line in meters.

*'height':* sets the height for water. -9999 selects mesh-clinging water.

*'bridge':* if the value is a valid bridge type (1213 to 1261), the line will generate a bridge. A value of zero will remove any road that passes across water without replacing it with a bridge.

*'layer':* sets the layer number for the line or polygon. A higher value gives higher priority.

*'smooth':* sets the amount of smoothing applied to the line or polygon. A value of around 5 will give roads nice smooth corners while coastlines might have 3. Smoothing can reduce the raster-type effects caused by the finite resolution of the input image.

*'random':* sets the amount of randomness applied to the line or polygon. Roads would normally have zero random while coasts might have 20.

*'comments':* add a comment here e.g. 'gravel road', 'major road', 'island' or 'coastline'.

Note: when you click on a line in the map display, the parameters set by the boxes described will appear in the STATUS slot.

*ADD button:* copies the settings in the small text boxes to the large text box on the right (the palette).

*REMOVE button:* deletes a selected text line in the large box on the right (the palette).

*UPDATE POLYGON button:* copies the settings in the small text boxes to the selected line or polygon. (You can also double-click on a text line in the palette box to do this).

*NEXT POLY* button: selects the next polygon in the list.

*CLEAR* button: projects may accumulate settings that are not actually used for any of the lines or polygons. Use this button to scan the bitmap and to remove any settings that are not used. A list of the current settings can be seen in the diagnostics window.

# LOD8 SQUARES Panel

When working with complex bitmaps (e.g. those created with TerraScene), creating the asm files can take several minutes, particularly if there are thousands of roads. To save time, you can use this panel to define which LOD8 squares will be processed to asm files. You could also disable any LOD8 squares that are entirely filled with water, providing the existing default scenery already provides the water.

After loading in a bitmap you will see some of the slots in this panel have the LOD8 square numbers displayed. Each one has a check (tick) mark. Simply check the LOD8 squares you want to process into scenery. The order of the LOD8 squares in the scenery area is from left to right, starting with the top line and working down.

If you want to process them all you can check PROCESS ALL.

# OBJECTS Panel

You can link the creation of objects such as trees and buildings to VTP lines, for example as buildings along a road or trees along a coastline. It is similar to vector autogen, but more flexible. AutoAsm creates a scenery file or files that must then be loaded into Airport 2.60 (or Airport 3.0) for bgl creation.
You can also create dynamic objects that move along VTP roads, the most common application being moving road traffic.
On the left you will see two large text squares and a single text slot at the top. The upper large square is the Object Palette. Here you can store ready-to-use object definitions. You can assign multiple object definitions to a given VTP line. The **Toggle labels** button selects labels for static or dynamic objects as required.
The lower box shows the object definitions assigned to the selected line or polygon. If you double-click on a line in the map display the box will display all the objects assigned to the line.
The single-line text box at the top is where you enter parameters for a new object or edit parameters for an existing one. If you double-click on an object line in either of the large text boxes the object parameters will be copied to the top line for editing. If you click on **OK** then the edited settings will replace the line you originally double-clicked.
Parameters in the top line are separated by one or more spaces. Some parameters for static objects (e.g. offset) are split into two numbers, separated by a comma. The two numbers are a minimum and a maximum value. AutoAsm randomly chooses a value between the two, so that objects have a natural variation. If you set the two values equal then there will be no variation. Look at some of the object strings that come with the demo projects for examples.

Here's an example:
*buildings|cottage.api 5 3 2, 5 left 0.5, 0.9 2, 3 3, 8 20, 60 23, 66 10, 38 17, 20 cottage*

You can add any number of spaces so that the parameters form neat columns under each parameter label. The parameters for static objects are as follows:
*static object type:* this is the path and file name of the API relative to the API directory. The API directory is set by the 'object path' slot at the top right of the panel. An example should clarify this.
Suppose you want to store all your Airport API's at c:\airport\api. Enter this path into the 'object path' slot. Then, if you enter buildings\house.api under 'object type', the full definition for the API would be c:\airport\api\buildings\house.api
If, instead of entering a complete path to an API, you enter the reserved word 'building', a FS2000-style building will be created.
*density:* this sets the frequency at which the object appears along the VTP line. A suitable density for a telephone pole would be 50. Increasing the value increases the frequency (smaller spacing between the objects).
*group:* not currently used.
*offset:* the distance of the object from the VTP line. A setting of 5 to 10 would be suitable for a building that is to be close to a road.
*side:* determines which side of the road the object will be placed. Values are *left, right* or *both.*
*size:* sets the size of the object. The effect will depend on the actual object and is the same value that appears when you open an object in Airport.
*parameters:* general parameters used for expansion. There are six pairs of parameters. When using Gerrish Grey's trees, some of these parameters are used for setting tree types.
*comments:* enter any suitable comments here.

The parameters for dynamic objects are as follows:
*dynamic object type:* this defines the object, usually with a GUID number. For details, see the section on creating dynamic objects.
*density:* the number of instances of this object created on the VTP line. Objects are placed along the length of the line.
*speed:* the object speed in arbitrary units. A typical value for road traffic would be 4.
*offset:* the object's offset from the centre of the VTP line. A typical value for road traffic would be 3.
*side:* defines the offset to be to the right or to the left of the VTP line. Values are 'right' or 'left'.

*direction:* sets the direction of movement. Values are one or zero. If set to 1 (one) the object moves toward the end of the VTP line.

*spacing:* a larger value will tend to increase spacing between objects. A suitable value might be 10.

*altitude:* sets the flying altitude in meters for dynamic aircraft. If set to zero the object will move on the surface of the terrain mesh.

*comments:* enter any suitable comments here.

***Toggle labels*** button: toggles the labels at the top of the editing line as required for static or dynamic objects.

***OK*** button: to copy parameters from either of the large boxes to the top editing line, double-click on the text line. After editing the parameters, click the OK button to copy the edited settings back to the original line in the large box. If you edited settings in the POLYGON OBJECTS box, you must click the UPDATE POLY to make the edited settings apply to the selected VTP line.

***DELETE*** buttons: deletes the selected setting string in the box to the left.

***ADD*** buttons: copies a parameter string between the editing line and the two large parameter boxes. The direction of flow is indicated by the arrows. For example, the top ADD button will copy parameters in the top editing line to the OBJECT PALETTE box immediately below it.

***UPDATE POLY*** button: copies the object strings from the POLYGON OBJECTS box to the selected VTP line in the map display.

***WRITE FILES*** button: for static objects, creates the Airport scenery file or files. As objects are entered they are indicated by a red point on the map display. The Airport scenery file or files are placed in the working directory (usually the main AutoAsm directory). Note: Airport seems to have a limit on the number of objects. If a sufficiently large number of objects are created then they will be split into two or more Airport files.

For dynamic objects, creates an asm file, calls bglc to compile it and copies the resulting bgl dynamic scenery file to Flight Simulator.

**CALL AIRPORT** button: calls the Airport program with the scenery file or files already loaded. If more than one Airport file was created, multiple instances of Airport will be opened. On each instance that appears, click on the compile button and then close it.

## STATIC OBJECT SETTINGS

***STATIC OBJECTS*** tick box*:* tick if you want to create static objects when you click the WRITE FILES button.

***DENSITY BMP*** tick box*:* tick if you are using a density bitmap. The brightness level of the 8 bit monochrome bitmap defines the object density so for example you could crearte smaller numbers of objects along roads outside towns.

***VIS RANGE****:* enter the visibility range for objects in kilometers.

***CULL*** tick box: if ticked, the program will attempt to stop objects being placed on top of each other e.g. where two roads pass close together or at cross roads.

***AMOUNT****:* determines how much objects will be culled.

***QUALITY****:* determines the accurace of culling. The program creates an internal bitmap used for culling. Increasing this value increases the bitmap size and therefore the accuracy of culling.

## DYNAMIC OBJECT SETTINGS

***DYNAMIC OBJECTS*** tick box: tick if you want to create dynamic objects when you click the WRITE FILES button.

***DENSITY****:* determines the number of path nodes used to generate the dynamic object path. The value is a fraction of the total number of points on the VTP line. For example, if a value of 0.1 is entered then only one tenth of all the points will be used. This allows you to reduce the total number of path points (more than about 1600 path points will cause Flight Simulator to lock up).

***LENGTH****:* any VTP line with this number of points or less will not have dynamic objects assigned to it. Short VTP lines with dynamic objects look unrealistic as the objects are constantly appearing and disappearing. This is also useful for reducing the total number of path points.

***FILE SPLIT****:* this setting allows the dynamic scenery to be split into sections so that more dynamic objects can be shown. A value of 2 will split into four sections and a value of 3 will split into nine sections etc. Leave this text box blank if you do not wish to split the scenery into multiple files.

## EDITING OBJECT SETTINGS

You can move object strings in several ways:

Double-click on a line in the map display to copy its object strings to the POLYGON OBJECTS box.

Double-click on a line in either of the two large text boxes to copy the object string to the top editing slot.

Click the OK button: after double-clicking to copy an object string to the top editing slot, and when editing is complete, click OK to copy the edited string back to its original position.

Click on one of the ADD buttons. The arrows indicate the direction flow, for example the top ADD button will copy the object string from the editing slot into the OBJECT PALETTE.

To edit an object's settings:

1. Double-click on the line in the map display - the settings are copied to the POLYGON OBJECTS box.
2. Double-click on one of the POLYGON OBJECTS or OBJECT PALETTE's text strings - the string is copied to the top editing slot.
3. Edit the text string and then click on OK - the edited string is copied back to the POLYGON OBJECTS box.
4. If there is more than one text string in the POLYGON OBJECTS box, repeat steps 2 and 3 for the other strings as required.
5. When the settings in the POLYGON OBJECTS box are as required, click on UPDATE POLY to copy them back to the polygon.

# TERRAIN Panel

For more information, see the CREATING TERRAIN MESH and CREATING LANDCLASS sections.

This panel is used for creating terrain mesh and landclass scenery. The user can choose landclass values and 'paint' landclass directly over the LWM/VTP polygons.

The terrain input data is defined by a bitmap. The format is 8 bit, monochrome, Windows bitmap. The brightness of each pixel determines the elevation of the corresponding terrain point.

The landclass to be applied to the scenery is also defined by a bitmap. The format is 8 bit, colour, Windows bitmap. The colour index of each bitmap pixel directly determines the landclass index to be applied at the corresponding terrain point. It also determines the autogen types that will be placed. Generally the resolution of the textures/autogen areas in the scenery will be considerably less than that of the bitmap (a landclass 'pixel' is approximately a one kilometer square).

Some example textures:

| Colour index | Texture |
|---|---|
| 0 | water |
| 29 | deciduous forest |
| 2 | shrubland |
| 108 | city |
| 122 | snow |

For example, where you paint with colour index 108 you will get city textures and buildings.

The terrain creation process is as follows:

1. Create a new sub-directory inside the AutoAsm terrain sub-directory. The sub-directory should have the same name as the terrain project e.g. if the terrain project name is 'test' then the path might be c:\autoasm\terrain\test
2. Create the two bitmaps for terrain mesh and landclass and place them into the project sub-directory.
3. Enter the parameters into the TERRAIN panel and save the project.
4. Click on the MAKE SCENERY button to generate the terrain bgl file.

The common settings are as follows:

*PROJECT NAME:* enter the project name. It must be the same as the name of the project sub-directory.

*WORKING DIRECTORY:* this is a separate working directory for terrain. Generally it will be the default terrain sub-directory inside the main AutoAsm directory. An example: c:\AutoAsm\terrain

*SCENERY BGL PATH:* where the scenery bgl file is to be placed. An example: c:\fs2004\addon scenery\island\scenery

OUTPUT FILE: sets the name of the scenery bgl file e.g. 'test' will create the file 'test.bgl'.

Terrain mesh functions (in the TERRAIN MESH box):

CENTRE POSITION and SIZE (DEGREES): enter the geographical bounds in decimal degrees for the terrain mesh bitmap.

LOD: the level of detail setting for the mesh. A typical value might be 9. Provided the bitmap has sufficient resolution, the higher the LOD setting, the higher the mesh resolution. Depending on the chosen LOD value, the bitmap must completely fill all the LOD squares in the scenery area.

SCALE: this defines the height of the terrain mesh. It is the height in meters between two successive pixel brightnesses. For example, if SCALE = 20 then the difference in height between two pixels with brightness 120 and 121 will be 20 meters, and the height corresponding to a pixel brightness of 100 would be 2000 meters.

COPY TO LC: copies the terrain mesh geographical bounds to the landclass bounds.

Landclass functions (in the LANDCLASS box):

CENTRE POSITION and SIZE (DEGREES): enter the geographical bounds in decimal degrees for the terrain mesh bitmap.

COPY TO TM: copies the landclass geographical bounds to the terrain mesh bounds.

COPY TO MAIN: copies the landclass geographical bounds to the bounds on the Main panel (the LWM/VTP bounds).

DRAW LC checkbox: enables landclass values to be drawn directly onto the map display. Landclass display must have been enabled (e.g. by loading a landclass bitmap or clicking on NEW).

SIZE: sets the size of the brush used for drawing landclass values. If set to one or zero, each mouse-click draws one landclass 'pixel' (a square kilometer approximately). Increasing the value increases the number of landclass 'pixels' placed by each mouse-click.

TYPE: sets the landclass type to be drawn e.g. a value of 108 will draw city textures.

NEW button: creates a new, blank landclass map in memory. After drawing it can be saved as a bitmap. The map size is determined by the landclass bounds settings: each map pixel corresponds directly to one landclass 'pixel'.

LOAD button: loads a previously-created landclass bitmap. After loading, each landclass 'pixel' is shown as a small filled circle whose colour represents the landclass type (the actual colours are determined by the palette of the blank_lc.bmp bitmap in the bitmaps sub-directory).

SAVE: saves the landclass data to a bitmap named textures.bmp in the terrain project sub-directory.

MAKE SCENERY button: converts the terrain mesh and landclass bitmap data to a bgl scenery file.

TM ONLY checkbox: if enabled, the created bgl file will contain only terrain mesh data.

LC ONLY checkbox: if enabled, the created bgl file will contain only landclass data.

COPY TO FS checkbox: if enabled, the bgl scenery file will be copied to Flight Simulator (the file location is set by the SCENERY BGL PATH setting).

RUN FS button: opens Flight Simulator and places the aircraft in slew mode directly over the centre of the terrain scenery area.

# TOOLS Panel

FILL CHECK button: it may not always be obvious if a given LOD13 square has been correctly filled. This function will place a blue dot in each LOD13 square that has been filled.

MEMORY: sets the size of the arrays. If the setting is too small the program stops and prompts the user to increase the MEMORY setting. A value of 100,000 will suffice for many applications, but for complex 10,000*10,000 TerraScene images the setting may need to be 1,000,000 (one million) or more.

To change the setting, type in a new value. Then, on the MAIN panel, click on SAVE. The next time you read in an image the new memory setting will be used.

READ TS TEXT button: TerraScene creates a text file which includes the scenery bounds. When you convert the TS targa file to 8 bit bitmap and save it to the bitmaps sub-directory, also copy the corresponding text file to the bitmaps sub-directory. READ TS TEXT will then read the text file and set the geographical bounds on the MAIN panel.

MAP COLOURS sliders: use the three colour sliders to choose and background colour for the map and bitmap (editor) displays. The other two sliders set the brightness of the grid lines.

DATA IMPORT

The following file formats can be imported: .dat .mp (Polish format) .bln .ung

The polygon data is converted to a bitmap which can be edited and saved from the Editor panel. The bitmap can then be read in in the usual way to create scenery. AutoAsm uses the filename to determine the format, so it must be correct e.g. xxxx.mp for a .mp file. Current types would be indicated by filenames of the form xxxx.dat, xxxx.mp, xxxx.bln and xxxx.ung.

AutoAsm can import .dat and .ung coastline files downloaded from:

http://www.ngdc.noaa.gov/mgg/shorelines/shorelines.html

(note: the files are always of the format xxxx.dat, so if you download a .ung file you will need to change the filename to xxxx.ung before importing it into AutoAsm).

LOAD button: import the file specified in the FILE NAME text box. The file must be in the main AutoAsm directory. After import the data will be displayed as a bitmap which can then be edited and saved in the usual way (from the Editor panel).

MERGE button: if a file has been imported, further files that cover the same geographical area can be imported and merged with the original file.

O/P BITMAP SIZE (X and Y): these entries determine the size of the bitmap that will be created during import.

CENTRE POSITION (N and E) text boxes: sets the centre position of the bitmap that will be created during import. N sets the latitude and E sets the longitude. Enter the value in decimal degrees (negative for south or west).

SIZE (DEGREES) (X and Y) text boxes: sets the geographical size of the bitmap that will be created during import. Enter the values in decimal degrees.

SET BOUNDS button: reads the data file that will be imported and automatically sets the CENTRE POSITION and SIZE (DEGREES) settings. The values will be chosen so that all the data points just fit inside the bitmap.

SET MAIN button: copies the CENTRE POSITION and SIZE (DEGREES) settings to the corresponding text boxes on the MAIN panel.

PARAMETERS text boxes: each line and polygon type in the .mp format (Polish format) has a type heading e.g. 0x1, 0x2, 0x12 etc Enter the types you wish to import in these boxes. Some line and polygon objects may have the same 0x types, so there are six boxes for lines and six boxes for polygons. If LOAD ALL is not enabled, only the 0x types entered into the boxes will be loaded.

LOAD ALL: if this is enabled, all 0x types will be loaded (generally applicable only to .mp format).

## EDITOR Panel

The editor is a simple paint program with tools optimised for correcting bitmap errors. The panel also includes tools for automatically correcting bitmap errors.

INPUT BITMAP and OUTPUT BITMAP: enter the bitmap file names to be loaded or saved (without the .bmp extension - if the filename is test.bmp just enter 'test').
LOAD BITMAP and SAVE BITMAP: load or save the bitmap after editing.

VIEW button: there are two distinct displays: map (has grids) and bitmap (editor). This button toggles between the two displays. Note: if you read in a bitmap (using the READ DATA button) you will need to re-load the bitmap into the Editor.
LWM ONLY: display only LWM polygons.
QUICK DRAW: particularly when zoomed out, a bitmap redraw can take a few seconds. With this option enabled the display quality is reduced to give a quicker redraw.
GOTO XY button: centres the bitmap display at the pixel position set by the two text boxes immediately above it.
FRAME button: displays the entire bitmap. This is a toggle function, so if you click on it again it will return to the original view settings.
Note: if it is a large bitmap the full-frame display update will be slow. Enabling QUICK DRAW will help.

REPAIR BITMAP button: automatically scans the loaded bitmap and corrects some of the errors.

BITMAP EDGES: if a lake is cut off by the bitmap edge then the polygon will be incomplete, which may cause serious problems. This tool automatically detects this condition and draws lines along the bitmap edges to complete the lake polygons. The lines are drawn with colour index 251 (NO VTP) so that no shorelines will appear on the lines.
PIXEL 0,0 text box: if a bitmap has a single polygon then it could be a lake - or an island surrounded by sea. This setting specifies whether the top left pixel is land or sea. Click the text box to toggle the settings.

SEARCH button: automatically scans the bitmap for errors. When it finds an error it stops and displays the error (the pixel causing the problem is marked by a small cross-hair). The floating Editor panel is opened. The user then uses the Editor tools to correct the error. If AUTO JUMP is enabled the search tool will automatically start to search for the next error after the current error has been fixed.
AUTO JUMP tick box: if this is ticked the search tool will automatically start to search for the next error after the current error has been fixed. If you disable AUTO JUMP, you can use the NEXT button on the small floating Editor window to start the search for the next error.
DELAY text box: this adds a delay (e.g. half a second) before the search tool starts the next search after the current error has been fixed. It allows the user to see the result of any edits before the screen clears in preparation for the next error.

MAKE BRIDGES: automatically searches the bitmap for road sections that cross water and converts those sections to VTP-style bridges.
MAKE BRIDGES text boxes: defines the range of bridge ID numbers that will be used. The tool chooses random numbers inside the range. If the numbers are 1220 and 1250 then numbers will be randomly chosen from that range.
SEARCH: as the MAKE BRIDGES tool searches the bitmap and creates bridges, it writes the bridge co-ordinates into the DIAGNOSTICS WINDOW list. When the tool has finished the user can open the DIAGNOSTICS WINDOW to see the list. Double-clicking on a bridge entry will centre the display on that bridge, making it easy to inspect the newly-created bridges. However, on a later AutoAsm session the list will be lost. In this case, use SEARCH to re-create the list of all the VTP bridges on the bitmap.

NEXT button: after creating the bridges or using SEARCH, the DIAGNOSTICS WINDOW contains the list of bridge X, Y co-ordinates. NEXT will centre the bitmap display on the next bridge in the list, making it easy to scan through and inspect all the bridges.

UNSOLID button: some GIS programs (e.g. fGIS) can export bitmaps that can be used by AutoAsm. In many cases there will be solid polygons. Currently AutoAsm only uses line polygons. Use this tool to convert solid polygons to line polygons.

ADD NODES: node points are used to join VTP lines together, for example to form natural road junctions. This tool automatically scans the bitmap and adds yellow node points (index 253) at VTP line junctions.

## EDITOR

Note: the tools described above automatically scan the bitmap to perform their functions and require no user mouse clicks on the bitmap. The Editor tools (inside the EDITOR box) work like conventional paint program tools, that is, they require the user to click on the bitmap to specify pixel positions. For example, the POINTS tool requires the user to click on the bitmap to add a point at that position.

Where appropriate, the Editor tools can be used in two ways: left-click to add points or lines, right-click to delete points or lines.

EDIT button: enables the editor tools. If the editor is off, then clicking on the bitmap will not change the bitmap - if you click on a line its settings will be displayed in the STATUS slot. When you click on the EDIT button a small floating window with a subset of the editing tools appears. For convenience you can move this window close to the editing point on the bitmap display.

UNDO button: undoes any operation that affects multiple pixels. Currently there is only one level of undo.

EXIT button: disables the editor tools.

POLY button: this tool has two functions:

1. Left-click on the polygon to change its colour to the next unused colour.
2. Right-click to delete the polygon.

Changing the colour (left-click) allows you to give a polygon unique settings (because it now has a unique colour). If, after left-clicking to change the polygon colour, you change the elevation or the line width (LINE TYPES panel) then no other polygons will be affected (because no other polygons will have the same colour).

If you tick the box and enter an elevation value in the associated text box then the tool will automatically assign the elevation to the polygon. As the polygon now has a unique colour index, no other polygons will be affected.

To delete a complete polygon, right-click on it.

POLY tick box: enables automatic assignment of elevation when the POLY tool is used (left-click).

POLY text box: enter the elevation value here. When the POLY tool is used to give a polygon a new, unique colour index, the elevation value will be assigned to it.

NO VTP: if a polygon is defined as 'both' it will create a water polygon as well as a VTP line, usually a shoreline. In some cases (e.g. on the edge of the bitmap) it is required that part of the polygon will not generate a VTP line. To change a section of a polygon to 'no VTP':

1. Click on EDIT (if not already in Edit mode) and click on NO VTP to select the tool - 'NO VTP' is displayed in the top right of the Editor box and also in the small floating Editor panel.
2. Left-click on the beginning of the line section.
3. Left-click on the end of the line section. The colour of the section will change to indicate that it is now non-VTP.

Note: in some cases the wrong part of the polygon will be affected. In this case click on UNDO and then again click on the line section, but in the opposite direction.

NO VTP text box: determines the colour of the non-VTP line (red, green or blue). Click in the text box to toggle the selected colour.

POINTS button: adds or deletes points. Left-click to place a point on the bitmap, right-click to delete a point.

The colour is set by the current colour. The current colour can be chosen with the PICK COLOUR tool. If the POINTS tool is used during a search the colour will automatically be set to the colour of the polygon that has the error.

LINES button: add lines or delete line sections.

Left-click to add lines as follows:

1. Left-click on an existing pixel.
2. Move the mouse to a new position and left-click again. A line will be drawn between the two points. The colour is taken from the first pixel.

To draw more complex lines:

1. Left-click on an existing pixel.
2. Press the SHIFT key and keep it pressed.
3. Move the mouse to a new position and left-click again. A line will be drawn between the two points.
4. Again move the mouse to a new position and left-click. A line will be drawn between the previous position and the new position.
5. Keeping the SHIFT key pressed, click as many points as needed to draw the desired shape.

UNDO can be used to delete the previous drawn line section.

Right-click to delete sections of a line:

1. Right-click on the beginning of the line section.
2. Right-click on the end of the line section. The line section will be deleted.

Note: in some cases the wrong part of the polygon will be affected. In this case click on UNDO and then again right-click on the line section, but in the opposite direction.

PICK COLOUR button: selects the colour of an existing pixel. Simply left-click on the pixel.

SET COLOUR: sets the colour of a line or polygon to the current colour (use PICK COLOUR to set the current colour). Left-click on a line to change it to the current colour.

BACKGROUND: it is highly recommended that the background colour of a bitmap is colour index zero (some functions and bitmap drawing will be significantly faster). After selecting the BACKGROUND tool, simply left-click on a background pixel (i.e. any pixel that is not part of a line or polygon). The tool will automatically scan the bitmap. All pixels that have the same colour as the clicked pixel will be set to zero colour index.

# T U T O R I A L S

## TUTORIAL 1: MAKING SCENERY

In this tutorial we'll load the 'demo' project, create the scenery, check it with LWMViewer and then fly it in Flight Simulator.

The project consists of three files: the bitmap itself (in the bitmaps sub-directory) named demo.bmp and two text files that contain the settings (in the settings sub-directory): demo.ini and demo.dat. The bitmap must be a Windows .bmp file, 8 bit colour. It can be any size except that the X and Y sizes should be multiples of 4 (if not you'll get an error message that suggests the appropriate size). demo.ini contains the project settings (all the text boxes and check boxes that you can see). demo.dat contains the settings for individual lines and polygons.

1. Copy the scenery file test_area.bgl into your Flight Simulator addon scenery directory.

This is a square area of landclass placed off the east coast of Africa. As it's in the ocean there's no default scenery, so it's ideal for testing your scenery without having to worry about excluding the default scenery.

Note: if you don't want to use the test area, you'll need to adjust the various project settings to place the scenery on land.

2. On the MAIN panel, click the small browse box to the right of PROJECT NAME. In the bitmaps sub-directory, select demo.bmp and double-click (or select and click on OK). 'demo' appears in the PROJECT NAME slot. Alternatively, you could type in 'demo' directly.

3. Click on SETTINGS / LOAD.

This will load the settings for the demo project. You will notice that the CENTRE POSITION is N10, E62, which is the centre of the test area.

4. Type the path to the desired FS scenery location into SCENERY BGL PATH e.g. c:\FS2004\addon scenery\scenery and then click on SAVE to make the change permanent.

5. Click on READ DATA.

This will read in the bitmap and display the lines and polygons. You can use the pan and zoom buttons to take a detailed look. If you left-click on a line its settings will be displayed in the STATUS slot. The INDEX slot (just above the map area) indicates the point number (i.e. the index into the arrays). The POLYGON slot indicates the polygon number and also the colour index number. (See Advanced Topics for more information on the various data readouts).

You'll notice that all the islands are coloured red (including islands inside lakes). AutoAsm automatically detects island polygons (i.e. polygons inside polygons) and processes them accordingly.

Now we'll create the scenery. You'll notice that both COMPILE and TO FS are checked, so that after the ASM files are created the program will call bglc.exe and then copy the created bgl files to Flight Simulator (but the SCENERY BGL PATH setting must be correct).

6. Click on WRITE LWM.

This will create the LWM scenery file (water polygons) and copy it to FS. As each scenery section is processed it is marked on the map display, thus giving a clear indication of progress. The bglc DOS window will open and quickly close. This is a small project, so it will only take a second or so.

7. Click on WRITE VTP.

This will create the VTP scenery (roads and shorelines etc). As with LWM, progress is marked on the map display. A progress line is also drawn across the top of the map display. After the bglc DOS window has opened and closed the process is complete ( a message in the STATUS slot confirms this).

The scenery has been created. A simple summary of the process:

1. Choose a project name and load its settings.
2. Read in the bitmap.
3. Create the scenery with WRITE LWM and WRITE VTP.

8. Click on the LWMViewer button (if the option is enabled with the check box it will open full-screen).

You can clearly see the water polygons forming the lakes and coastlines, with the VTP roads and shorelines. Everything looks good - but if there are errors it'll be very obvious in LWMViewer. Now we'll run it in Flight Simulator.

9. Close LWMViewer and then click on the map display to select a point where you want FS to open.
10. Click on RUN FS.

AutoAsm will call FS and then close. When Flight Simulator opens you will find yourself in slew mode exactly above the point you selected. Fly around and take a good look at the scenery. You'll see that a number of VTP bridges have been created. By default there's not much control over the actual bridge types (this is a Flight Simulator peculiarity, I hasten to add!). In the BRIDGES section of the manual you'll find a technique that allows you to select the exact kind of bridges you want.

Before finishing, we'll give a quick demo of the search function.

11. Close Flight Simulator and run AutoAsm.
12. Select the Editor panel. INPUT BITMAP shows 'demo'. Click on LOAD to load the bitmap.

The bitmap looks the same as in your paint program apart from one important difference. All pixels that are grey or have zero colour index are shown with the background colour instead of black. A lighter background colour usually makes line polygons more visible. You can change the background colour to anything you wish - go to the TOOLS panel and play with the sliders.

You can use the pan and zoom buttons to control the bitmap view. As in the map view (READ DATA), if you click on a pixel its colour index and settings are shown in the STATUS slot.

We'll now do a search for errors.

13. Click on SEARCH.

The tool scans the bitmap as it searches for errors. This is a 'perfect' bitmap, so there are no errors. You'll very quickly see a green banner and a message informing you that there are no errors. When you work on new bitmaps to remove all errors, the aim is to obtain the green banner.

# TUTORIAL 2: Setting up a project

The previous tutorial came complete with the project already set up. This tutorial shows you how to create a project from scratch. We'll call this project 'new'.

1. In your paint program create an 8-bit colour bitmap with resolution 512*512.
2. Choose a non-grey colour with index in the range one to 200 and draw a closed polygon with single-pixel width.
3. Save the bitmap into the AutoAsm bitmaps sub-directory with the name 'new.bmp'.
4. In *PROJECT NAME* enter 'null' and click on *LOAD*. This loads the null project. Most of the settings are blank!
5. Enter the settings for the new project as follows:

*PROJECT NAME:* enter 'new' as we're going to create a project with that name.

*WORKING DIRECTORY:* your main AutoAsm directory e.g. c:\program files\AutoAsm. If you previously set this it will still have the correct setting after loading the null project.

*SCENERY BGL PATH:* where you want to place the scenery in Flight Simulator e.g. c:\fs2004\addon scenery\scenery

*FLIGHT SIMULATOR PATH:* the path to the main Flight Simulator directory e.g. c:\fs2004

*OUTPUT FILE:* generally it will have the same name as the project, but it can be different, for example test. Enter 'new'. In this case the asm files will be new_lwm.asm and new_vtp.asm. The bgl files will be new_lwm.bgl and new_vtp.bgl.

*CENTRE POSITION:* this defines the geographical centre point of your scenery. Enter 10 and 62 (the test area off the coast of Africa).

*SIZE (DEGREES):* the geographical size of your scenery in the X and Y direction. Enter 0.5 and 0.5 for X and Y.

*READ-IN BOX:* set these four to zero (or blank).

*SEARCH RADIUS:* set both to 3.

*AUTO CULL* and *AUTO FILL:* these should be checked (ticked).

*LWM ONLY:* this should be unchecked.

You don't need to enter anything into IMAGE SIZE.

6. Now click on SAVE. This saves the data to two files in the settings sub-directory: new.ini and new.dat. To re-load the settings later, simply type 'new' into PROJECT NAME and click on LOAD.

When you click on EXIT to exit the program AutoAsm saves autoasm.ini and autoasm.dat to the settings sub-directory and it re-loads the settings the next time you run the program.

# READING BITMAPS

Bitmaps are the prime method of importing scenery data into AutoAsm. For LWM and VTP import the image format is 8 bit colour, Windows bitmap (.bmp). You can use any reasonable bitmap size (I've tested it up to about 10,000*10,000, which is a hundred million pixels). The X and Y sizes should be multiples of 2 and 4 respectively. If the size is incorrect you will get an error message during read-in. It will also suggest the correct X,Y size that the bitmap should be resized to.

All lines and polygons *must* be coloured and not grey, that is, at least two of the R, G and B components must be different. All grey lines are ignored. This means that you can annotate the image with text or diagrams provided you use grey, black or pure white. The colour indexes 251 to 255 are reserved for special functions, so all normal lines and polygons should have colour indexes in the range zero to 250.

In the next tutorial we'll draw some simple polygons and lines and show how to convert them into lakes and roads.

# TUTORIAL 3: Reading a TerraScene Bitmap

I’ve included an edited LWM/VTP TS bitmap which is cut down in size to speed things up a little.

In AutoAsm, type ts_demo into the PROJECT NAME and click on LOAD. You will now have appropriate settings for this bitmap. Type in settings in the three slots at the top of the MAIN panel that are appropriate for your installation (WORKING DIRECTORY, SCENERY BGL Path and FLIGHT SIMULATOR PATH). The working directory is most likely your main AutoAsm directory. Click on SAVE (on the MAIN panel) to save the settings.

Enable LWM ONLY. With this setting VTP lines will not be loaded.

Check COMPILE and TO FS so that they are enabled - this will allow the scenery files to be copied to FS.

Set CENTRE POSITION to N10  E62

Click on READ DATA. After a few seconds you will see the water polygons displayed in the map area. Because there are no VTP lines you can pan and zoom quite quickly. You will see that the islands are a different colour. Their colour index is 254.

Now disable LWM ONLY and read in the image again. It’ll take a bit longer and you’ll notice that there are a LOT more polygons. When it’s loaded you’ll see all the roads. Sometimes the roads can almost hide the water polygons.

Before creating the scenery we’ll be a little more adventurous. We’ll edit the water polygon height setting.

Select the LINE TYPES panel. Double-click on a water polygon and you’ll see the line parameters appear in the small boxes at the top. Change the height parameter to 1000 and, with the water polygon still selected, click on UPDATE POLYGON. Then select one of the island polygons and again click on UPDATE POLYGON. You could click on SAVE (on the MAIN panel) to save the changes to the project. But if you don’t, the next time you run AutoAsm it will still have the changes you just made.

Click on WRITE LWM to create the LWM asm file. As  COMPILE and TO FS are checked the asm file will be compiled by bglc and the bgl file copied to Flight Simulator. Sections of the map display are highlighted in green as they are processed.

Click on WRITE VTP to create the VTP scenery.

When it’s completed click on LWMViewer to check that all is as it should be. You will see that the islands are shown in green. As they have the island 254 colour index, AutoAsm thoughtfully changed them to land polygons.

Finally select a point on the map display and click on RUN FS. In a few seconds you will be hovering in slew mode right above the point you selected.

# TUTORIAL 4: SETTING UP TERRASCENE

This tutorial provides a detailed procedure for setting up a TerraScene project that will generate suitable line polygons.

1. With TerraScene running, select **Project / New** to create a new project. Input a name and click on **Next.**
2. Choose to drag a box on the main map and click **Next.**
3. Choose the Base Fly!2 generic textures and click **Next.** *We'll set up the 'textures' required for AutoAsm in a little while.*
4. Click on **Finish.**
5. Draw a box to select your scenery area, using the zoom function for a more detailed view. When you've defined the correct area, click on **Project / Save**. *Now we have to set up the render options so that TerraScene generates line polygons that can be used by AutoAsm.*
6. Click on **Options.**

*There are six classes of lines / textures: **Land Types (day), Land Types (Night), Land Types (Water Mask), Lines (day), Lines (night), Lines (Water Mask).** We will only use **Lines (day**), so first we need to disable the others:*

7. Under **Options,** select **Land Types (day).** Click on **Set all to None** and then click on **Done**. Similarly, select the other five and click on **Set all to None**. Now everything is disabled!
8. Select **Options / Lines (Day).**

*You will see a list of all the possible line types. Currently they are all disabled, so we'll have to enable just the ones we want to use.*

9. Click to select the first line type: **Class 1 Roads**.
10. Click on **Solid Color**.

*We will *not* be using any textures in TerraScene, particularly as Fly! textures could not possibly work in Flight Simulator. We will assign suitable FS textures to VTP lines in AutoAsm later. For this reason, solid colours (i.e. not textures) are appropriate.*

11. Click on the colour swatch and choose a suitable colour. *The same colour will appear in AutoAsm.*
12. Set the pixel width to 1. *This is essential. AutoAsm expects to read in single pixel-width polygons. If it gets anything else it will complain bitterly! Important note: if you need to change the pixel width then you *must* click on Update, otherwise the change will be lost.*
13. Switch off **Antialias**.

*Again, AutoAsm expects single pixel-width polygons. For a given polygon, every pixel must be of the same colour. This will not occur if antialiasing is selected.*

14. Switch off 3D. *We certainly don't want 3D polygons!*

*To summarise, the settings for each line type must be:*
*Solid color*
*Width (pixels) = 1*
*Antialias off*
*3D off*

15. Repeat this process for **Class 2, 3** and **4** roads and **streams**.
16. Repeat this process for the three water types**: Ocean & Bay Coastline (land side), Lake Coastline (Land side) and River Coastline (land side).** Leave all the other water settings disabled.

*If you use different colours for the three shoreline types you may find that TerraScene draws a single water polygon using two or all of these colours. This breaks the rule that all the points of a given polygon must be the same colour - if it occurs ( and it will!) then it is equivalent to two or more breaks in the polygon. Or, looking at it another way, there are two or more separate polygons, each with a large break. Of course this can be fixed by editing the bitmap. But, unless you have a specific need for different water types, I would suggest setting the three shoreline types to the same colour (note that they must be *exactly* the same!)*

*For the purposes of this tutorial, set these three items to exactly the same colour.*

*Now all the line types we're going to use for this tutorial are set up. Of course, in practice you can choose the lines you want, for example Ramps, connectors, transitions or railroads.*

17. Click on each line type in turn and check that the settings are correct as summarised above. All lines not required should be set to **None (don't draw).**

*These settings ensure that TerraScene will create simple, single pixel-width polygons with the specified colours.*

18. When you're satisfied everything is correct, click on **Done** and then click on **Project / Save** to make these changes permanent.

19. Choose **Options / Save Render Options** and save with a suitable name, such as _LWMand_VTP. *This is useful, as at any time in the future you can load these settings back when you want to generate this combination of lines with a later project. You could create new settings for VTP-only lines and LWM-only lines. Funnily enough, the files AutoAsm_both.trp, AutoAsm_vtp.trp and AutoAsm_lwm.trp do exactly this. But in practice you'll probably want to create trp files for other line combinations. Instead of going through the tedious process just described, you could have chosen **Options / Load Render Options** and immediately loaded AutoAsm_both.trp. But doing it the hard way is good practice, don't you think?*

*Now we're ready to create the targa image filled with polygons and lines.*

20. Choose **Render** and again **Render**. *You will be presented with the render options.*

21. For day scenery, choose **generate.** For the other three (**Elevation mesh, Night scenery and Water mask**), choose **skip.**

22. Switch off **Slice with FlyEdit**. *Now the only thing enabled on the left-hand side is **Day scenery**.*

23. Accept the default 2500 pixels. *Now we're ready to render the targa image.*

24. Log onto the Internet.

25. Click on **Render now!** Sit back and relax. Or make a cup of coffee or watch your favourite TV show. *Depending on your system and Internet connection, it'll take twenty minutes or more. When it's finished the image file project name.tga will be in the TerraScene output directory.*

26. Load the targa file into a paint program such as Paint Shop Pro.

27. Reduce the colours to 8 bit / 256 colours (in PSP select Colors / Decrease Color Depth / 256 Colors).

28. Save the image to the AutoAsm bitmaps directory. *The format must be Windows bitmap (.bmp).* In PSP select File / Save As…. Choose Windows bitmap (.bmp) from the drop-down menu and enter the path and file name. *The path will then be similar to c:\AutoAsm\bitmaps\tutorial.bmp.*

# TUTORIAL 5: CORRECTING BITMAP ERRORS

*In this tutorial we'll use the AutoAsm editor's repair tools. Together they attempt to comply with the one Golden Rule for LWM polygons: EVERY PIXEL MUST TOUCH EXACTLY TWO OTHER PIXELS OF THE SAME COLOUR.*

*The REPAIR BITMAP tool automatically scans the bitmap and fixes three kinds of problems:*

1. *Line breaks. It will fix line breaks up to the size set by SEARCH RADIUS by drawing a single line across the break. If it encounters a break larger than this then it will stop and wait for you to take appropriate action.*
2. *Corner pixels.*
3. *Water polygons whose edges are broken by the edge of the bitmap.*

*After the repair tool has completed its pass you will then run the SEARCH tool. It will automatically find any errors that the repair tool was unable to fix. Errors might be small breaks, a corner pixel or a pixel that touches too many other pixels. When SEARCH finds an error it displays that part of the polygon on the screen and states what it thinks the problem is. The user can then edit the polygon to fix the problem. Left-click to add points and right-click to delete points. If AUTO is enabled, as soon as you have fixed the error the tool searches for the next error and then displays it. When the bitmap has been searched you will see a message giving the number of errors. Repeat until you get a search run with zero errors (a run with zero errors is indicated by a green banner).*

*When the SEARCH tool finds an error it uses a small cross-hair to indicate where the error is. However, action may need to be taken on an adjacent pixel and not the one directly indicated by the cross-hair. For example, it may indicate a pixel that touches more than two other pixels of the same colour. Usually one of the other pixels needs to be deleted, and not the one indicated by the cross-hair.*

*We'll now use the REPAIR BITMAP and SEARCH tools. errors.bmp has plenty of errors to be fixed!*

1. Type 'errors' into PROJECT NAME and click on LOAD. *This will load in the 'errors' project.*
2. Change the entry in WORKING DIRECTORY to suit your system and click on SAVE.

Set SCENERY BGL PATH as required e.g. c:\fs2004\addon scenery\scenery. This determines where the scenery files will be placed in Flight Simulator.

Check that WORKING DIRECTORY and FLIGHT SIMULATOR PATH are correct. If you set them up on a previous tutorial then they should still be correct.

3. Click on READ DATA to read in the bitmap. *You will see there are plenty of errors including line breaks. If you open the DIAGNOSTICS WINDOW you will see plenty of breaks and some fatal breaks listed. You can also see spurious water squares in the map display. Finally compile the scenery by clicking WRITE LWM and look at it in LWMViewer. It's a mess, quite frankly.*
4. Go to the Editor panel and check that INPUT BITMAP shows 'errors'. Click on LOAD BITMAP. *You will see the bitmap displayed. There are breaks and many polygons are broken at the edge of the bitmap. You can use the zoom and pan controls to take a closer look.*
5. Click on REPAIR BITMAP. *After a few seconds the map is transformed. Many breaks have vanished and most of the spurious/redundant pixels have vanished.*

6. Check that PIXEL 0,0 is set to land (just below the BITMAP EDGES button). *To toggle between land and water, click in the text box. The polygons are water and therefore the top left pixel is land. After completing this tutorial you could run through it again, but this time select water. Now the scenery will be filled with water and the polygons will be islands.*
7. Click on the BITMAP EDGES button. *Red lines are automatically added to complete polygons where they are cut off by the edge of the bitmap. These are NO VTP lines. You could choose a different colour by clicking in the NO VTP text box near the top right of the Editor panel.*
8. Starting at the top left corner, scan around the bitmap going clockwise and check that the red lines are correct.

9. There's an error near the bottom left corner (pixel co-ordinates 21,511). Four pixels run along the bottom of the bitmap but they don't touch the edge. Click on UNDO to remove the edge lines and then add a pixel so that the polygon ends on the bitmap edge (click on EDIT, select LINES, click on the left-most pixel and then click just below).

10. Click on BITMAP EDGES again and continue checking around the bitmap edge. There's another error at pixel co-ordinates 0,400. Click on UNDO and then add a pixel so that the polygon terminates on the bitmap edge.

11. Click on BITMAP EDGES again. *This time there should be no errors. Now we're ready to search for the other pixel errors.*

12. Enable the AUTO JUMP tick box and enter a value of one into the DELAY text box. *When you have fixed an error the tool will wait one second before automatically scanning for the next error. This allows you to see the change caused by your editing. When you gain more experience you may want to reduce this setting or even set it to zero to speed things up. Alternatively, if you disable the AUTO JUMP tick box, the tool will wait until you click on NEXT (on the small floating window) before searching for the next error.*

13. Click on SEARCH. *The tool will find the first error and display it. It also prints a message in the STATUS slot: 'Bitmap X, Y: 23, 0 Too many points / corner point: Right-click to delete.'*

14. Right-click on the pixel on the extreme right of the top line of the small polygon. *It was a redundant pixel and right-clicking on it deletes it. The pixel to its left (and indicated by the cross-hair) is a corner pixel. Right-click on it and it will go away. The tool immediately jumps to the next error. The next polygon has exactly the same error.*

15. Delete the pixel on the right and delete the corner pixel. *The tool jumps to the next error which, strangely enough, has the same error. Now you should know what to do! The next green polygon has a corner pixel, but if you deleted it there would then be a break.*

16. Delete the green pixel on the extreme left of the small polygon. *The next blue polygon has the same error.*

17. Delete the pixel on the left. *The next polygon has a corner pixel at the top right.*

18. Delete the corner pixel.

19. Continue until all the errors are fixed. *The indicated y co-ordinates indicate how far you have progressed through the bitmap. If you make a mistake the tool will take you back for another go, or it will be picked up on the next pass. If you delete too many points you may create a break. In that case, left-click to add points. If you're not quite sure what to do, remember the Golden Rule!*

20. When the STATUS slot informs you the search run has completed, click on SEARCH again to do another pass. Repeat until you get a search pass with zero errors (it will be indicated by the green banner).

21. After getting the green banner, check that OUTPUT BITMAP shows 'easy_repaired' and click on SAVE BITMAP.

22. Go to the MAIN panel and change the PROJECT NAME to easy_repaired (the repaired bitmap you just saved).

23. Click on READ DATA. *After read-in you should have a perfect display. Note that the islands are correctly shown in red. All the breaks have vanished. The DIAGNOSTICS WINDOW lists no breaks at all.*

24. Compile the LWM scenery and check the result with LWMViewer. *Everything looks fine. The bitmap has been successfully corrected.*

*It's very important to understand the errors you will see. There are three main error types as follows:*

1. *Small breaks. These are breaks that the repair tool missed - or even created when deleting corner pixels! The break may also be caused by a VTP line crossing the polygon. In this case, check the actual colours carefully. In this project the LWM lines are green. Just left-click in the gap to repair the break.*

2. *Corner pixels. These are pixels that form a right-angled corner. Examples (the x is the corner pixel):*

```
o       o
xo      ox      xo
                o
```

*When this occurs, at least one pixel touches three other pixels, which is invalid. Remember, for a perfect polygon every pixel touches exactly two other pixels of the same colour.*

*In this case, right-click on the x pixel to delete it.*

3. *Too many pixels. Again, this occurs when too many pixels touch each other. It could be caused by two polygons of the same colour that touch each other, or when two sides of the same polygon touch. When this occurs there will probably be one or more corner pixels as well. Simply right-click the little devils until they go away. In some cases you may need to both add and delete pixels to correct the problem. Remember, the criterion for a perfect polygon is that every pixel touches exactly two other pixels of the same colour. With AUTO JUMP enabled, as soon as you have cleared the problem the editor will start searching for the next error. If you, say, deleted too many pixels and created a break, the editor will return to give you another chance to fix it.*

# TUTORIAL 6: Adding objects

You can add objects linked to VTP lines rather like vector autogen. AutoAsm creates scenery files for use with Airport 2.60, so you will need to have Airport installed.
First we'll add some FS2000-style buildings along a road.

1. Choose a suitable bitmap and set up the project settings. For test purposes a small bitmap with just a few water polygons and roads would be suitable.
2. On the MAIN panel click on READ DATA to read in the bitmap.
3. Go to the Objects panel.
The upper large text box is the Object Palette - it's a collection of object settings that can be 'dropped' straight onto lines.
4. If there are any entries in the Object Palette delete them. To delete an entry simply click on it to select it and click on the DELETE button just to the right of the Object Palette box.
Now we'll add our own setting.
5. Enter the following line into the single-line text box at the top:
building 1.7 1 3, 5 both .7, 1.5 8, 30 8, 30 8, 30 8, 30 1, 1 1, 1 building

You can copy and paste into the slot. Just above the slot you'll see the headings 'object type density group' etc. Insert or delete spaces so the groups of settings fall just under the corresponding heading. For example, '3, 5' should be just under 'offset'.

The settings are as follows:
object type: building (enter the full path to the api for the object. 'building' is a special case).
density: 1.7 (the number of objects for a given length of the line)
group: 1 (currently not used)
offset: 3, 5 (the object's distance from the line)
side: both (determines whether the object is to the right, to the left or both)
size: .7, 1.5 (sets the object size)
parameters: 8, 30 8, 30 8, 30 8, 30 1, 1 1, 1 (extra parameters, e.g. for Gerrish Grey's trees or FS2000-style buildings).
comments: building

You'll see that several settings are separated by commas. The program randomly chooses a value between the two limits so any amount of variation can be specified. For example, for each building its offset (the distance from the road) may be any value between 3 and 5. Even if the extra parameters are not required you should still provide six pairs of numbers separated by commas.
The 'object type' entry is normally a path to the api file that defines the object. The 'object path' text box at the top right of the panel defines where your Airport api's are stored e.g. c:\airport\api
Suppose you have a sub-directory named 'buildings' and the api file name is 'cottage.api'. In this case your entry under 'object type' would be 'buildings\cottage.api'. The full path would then be c:\airport\api\buildings\cottage.api
To see some example object settings, load the project TS_demo (it was in the original AutoAsm zip). Of course, you could create the scenery and the objects, but you would need to substitute api files that exist on your system.
The 'side' setting has three possible values: left, right or both. If you want to add objects along a shoreline, then you must use 'left', or otherwise the objects will be in the water!
Now we'll copy the settings into the Object Palette so they can be used at any time in the future.

6. Click on the top ADD button.
The settings appear in the Object Palette. You can see that the arrows indicate the desired flow direction. Now we'll apply the settings to a line. We first need to copy the settings into the Polygon Objects text box (the lower large one). You can copy multiple settings into this box. These settings can then be applied to a selected line.
7. Click on the settings in the Object Palette to select and then click on the lower left-hand ADD button.

The settings are copied to the Polygon Objects box. From the arrows, you can see that you could also have clicked the right-hand ADD button: this would copy the settings from the top slot, which have the same values. In general, for each ADD button the arrows indicate the direction flow.

8. Click on the line in the map display to select it and then click on the UPDATE POLY button. Repeat as required for any other lines with different colour indexes.

This copies all the settings in the Polygon Objects box to the selected line.

9. Click on WRITE AIRPORT FILE.

This creates the Airport scenery file in the main AutoAsm directory. Each object that has been created is marked on the map display by a red dot. Look carefully and ensure you can see the red dots.

Before creating the scenery we'll edit the object settings: let's increase the density to 5. First, a couple of things to remember:

1. Double-click on a line in the map display: copies the settings for that line into the Polygon Objects box.

2. Double-click on an entry in either of the large text boxes: copies the settings into the editing box at the top.

10. Delete all the entries in the three text boxes (select an entry and click on the appropriate DELETE button - for the top text box you can use BACKSPACE). Now all the text slots are empty.

11. Double-click on a line in the map display.

Its object settings are copied into the Polygon Objects box.

12. Double-click on the entry in the Polygon Objects box.

The object setting is copied into the top editing box.

13. Change the density setting to 5.

14. Click on the OK button to update the setting that you double-clicked in the Polygon Objects box.

15. Click on the UPDATE POLY button (the line in the map display must still be selected).

The new, edited settings are copied to the line.

16. Click on the WRITE AIRPORT FILE button.

The Airport scenery file is created in the main AutoAsm directory. The file name is set by the OUTPUT FILE setting on the MAIN panel. For example, if OUTPUT FILE is 'test' then the file name will be test_apt.apt. If large numbers of objects are created AutoAsm splits the Airport files. In this case they would be test_apt.apt, test_apt_1.apt, test_apt_2.apt etc.

17. Click on the CALL AIRPORT button.

It runs Airport and loads the scenery file just created. If the scenery was split into more than one file, an instance of Airport is created for each file.

18. Click on the Airport compile button to create the scenery file and then close Airport. If more than one instance of Airport was opened, click the compile button on each instance and, when compilation is complete, close that instance of Airport. Repeat for the other instances of Airport.

Some key points:

1. Double-click on a line in the map display: copies the settings for that line into the Polygon Objects box.

2. Double-click on an entry in either of the large text boxes: copies the settings into the editing box at the top.

3. You use the ADD buttons to move settings downwards - the arrows indicate the direction of flow.

4. The OK button copies the edited version back to the entry you originally double-clicked.

5. UPDATE POLY copies the object settings in the POLYGON OBJECTS to the selected polygon.

6. You can apply multiple object settings (e.g. buildings and trees) to a polygon by copying the settings into the POLYGON OBJECTS box.

# TUTORIAL 7: IMPORTING DATA

AutoAsm can import several data formats including Polish format. The import functions are located on the Tools panel. During import the data is converted to a standard LWM/VTP bitmap which can then be edited and converted into scenery in the usual way.

In this tutorial we'll go through the steps required to import a Polish format data file (file type .mp). GPSMapEdit can be used to convert other data sources to Polish format.

1. After downloading or creating your .mp file copy it into the AutoAsm import sub-directory and name it 'test.mp'.
2. Go to the Tools panel and select the test.mp file (click on the small button to the right of the FILE NAME text box and select the file). The entry will be 'import\test.mp'.
3. In the O/P BITMAP SIZE text boxes, enter 1000. This will create a bitmap size of 1000*1000.
4. Enable (check) the LOAD ALL check box. This ensures that all lines in the file will be read in.
5. Click the SET BOUNDS button. This will read in all the data points and choose bounds that will just fit the data.
6. Click the SET MAIN button. This copies the bounds settings to the Main panel.
7. Click the LOAD button. After the data is read in the bitmap is displayed in the lower map window. If the view is zoomed in you may not see anything. Zoom out until the lines appear. Alternatively you could go to the Editor panel and click on FRAME. This adjusts the zoom so that the entire bitmap is visible. If you click FRAME a second time the zoom returns to the original value.

If no lines appear in the map display after importing, first check that, during the import process, the number of data points is listed successively in the lower window. If no numbers are printed out then there may be something wrong with the input data file. Also check that the file blank_lc.bmp is present in the AutoAsm bitmaps sub-directory.

All the lines are displayed in different colours. This is useful to look at the data but not so useful for creating scenery. We now need to select the line types that will be loaded. Polish format files may include an enormous number of different line types, many of which are of no use for scenery creation (e.g. town boundaries).

Polish format divides the data into lines and closed polygons. They are further divided into line types such as lake shores, roads and streams. For example, small lakes may be denoted by 0x0041.

8. Disable the LOAD ALL check box. Now only selected lines will be loaded.
9. In the P1 slot underneath POLYGONS, enter a suitable designation ('0x0041' or similar). Ensure that all the other slots beneath POLYGONS and LINES are blank. You can find the line designations in GPSMapEdit or by inspecting the input data in a text editor.
10. Click the LOAD button. After reading the file only the chosen line type is displayed.
11. Add another line type and click LOAD. You will see that all the lines of a given type are the same colour so that suitable line settings can be assigned to the various line types.
12. Choose which line types you require for scenery generation and enter them into the slots, and then click LOAD. The map display will display all the chosen line types. If everything looks okay we'll need to choose the final bitmap resolution. Unless the scenery covers a small area, 1000*1000 will probably be too small.
13. Increase the O/P BITMAP SIZE e.g. to 2000*2000 and click LOAD to import. Examine the pixels and decide whether the resolution is high enough. Increase the bitmap resolution and import until the pixel displays accurately represent the input data with no loss of accuracy.
14. From the Editor panel, save the bitmap with a suitable name (enter the name but without the .bmp extension).
15. Go to the Line Types panel and assign suitable line settings. When you've finished, go to the Main panel and save the project. If the bitmap is test.bmp then the project name must be 'test'.
16. With the bitmap still loaded in the editor, use the editor tools (e.g. REPAIR BITMAP and SEARCH) to correct any pixel errors (Tutorial 5 has examples of these procedures). When you obtain the green banner, save the bitmap in readiness for creating scenery.
17. From the Main panel, click on the READ DATA to read in the data from the bitmap.
18. Click on the WRITE LWM and WRITE VTP buttons to create the scenery files. Tutorial 1 illustrates this process in detail.

# TUTORIAL 8: BUILDING AN ISLAND

This final tutorial uses a number of AutoAsm features to build an imaginary island off the coast of Africa as follows:
1. Create an area of landclass in the sea to provide some background textures.
2. Create the LWM water polygons and VTP roads from the islands.bmp bitmap.
3. Create terrain mesh to add some hills and a mountain.
3. Add buildings linked to the roads to form some small towns.

The bitmaps are already provided, so all you need to do is go through the process of creating the scenery. You could then try modifying the bitmaps (e.g. add another volcano) to build the island of your dreams.

The main project is called 'island' and the LWM/VTP bitmap is therefore island.bmp. You will also find a sub-directory named 'island' inside the terrain sub-directory. This contains two bitmaps named textures.bmp and terrain.bmp. textures.bmp contains the landclass data while terrain.bmp contains the terrain mesh data.

We'll start by creating the area of landclass.

1. On the Main panel, type 'island' into the PROJECT NAME box and click the LOAD button to load the project settings.
2. Select the Terrain panel and set WORKING DIRECTORY as appropriate. It should be the terrain sub-directory e.g. c:\autoasm\terrain
3. Set SCENERY BGL PATH to define where the bgl scenery files will be placed inside Flight Simulator. All other settings will be appropriate as they were loaded with the project.
4. Enable the LC ONLY check box. This ensures that the scenery file will contain only landclass data.
5. Click on MAKE SCENERY. After reading in the textures.bmp bitmap AutoAsm calls the SDK tools to create the scenery file. As COPY TO FS was checked the scenery file will be copied to FS.
6. Click on RUN FS. The program runs Flight Simulator and places you over the centre of the scenery area (off the east coast of Africa). You should see a flat area of land with landclass defined by the bitmap.

Optionally you could modify the landclass as follows:

Go to the Editor panel and click on LOAD BITMAP to load the LWM/VTP bitmap. Go to the Terrain panel and click on LOAD (inside the LANDCLASS box). Each landclass 'pixel' is indicated by a small circle whose colour represents the landclass. If you click on the map you will see a thumbnail of the actual texture. To paint new landclass values, choose a landclass value in TYPE, choose a brush size in SIZE and finally enable the DRAW LC check box. You can now 'paint' the landclass value over the bitmap. When you've finished painting, click on SAVE to save your changes to textures.bmp.

Now we'll add the LWM/VTP scenery to define the coastlines and add roads.

7. Go to the Main panel and click on READ DATA. After the LWM and VTP data has been read from island.bmp you will see the island coastline and some roads.
8. Click on WRITE LWM to create the LWM data. As COMPILE and COPY TO FS were enabled the data is compiled to a scenery bgl file and the file copied to Flight Simulator.
9. Click on WRITE VTP to create the VTP data.
10. Click on LWMViewer to inspect the LWM/VTP scenery. If it looks good exit LWMViewer. Click on a polygon and then click on RUN FS. Flight Simulator will open with you placed above the point you selected. You should now see the correct coastline and some roads.

Now we'll add some hills and a volcano.

7. Go to the Terrain panel and disable the LC ONLY check box. With both LC ONLY and TM ONLY disabled then the scenery file will contain both the landclass and terrain mesh data.

8. Click on MAKE SCENERY. Both the landclass and terrain mesh data will be placed into one bgl file and the file copied to Flight Simulator.
9. Click on RUN FS. You will be placed at the centre of the scenery area. Now you will see some hills and a volcano.

Finally we'll add some buildings to make the island look as if it is inhabited. For this to work you need an installation of Airport for Windows, version 2.60 or later. We'll use the standard FS2000-style buildings, which are basically textured boxes. The advantage of this type of building is that no special textures or macro files are required.

10. On the Main panel, click on READ DATA to read in the LWM/VTP bitmap and then go to the Objects panel.
11. Double-click on a road line in the map display. The object parameters for the line appear in the POLYGON OBJECTS box. The text string begins with the reserved word 'building', indicating that it will create a FS2000-style building.
12. Ensure that airport path is correctly set e.g. c:\airport
13. Click on MAKE SCENERY. You will see many red dots appear on the map display, all of them aligned on roads. Each dot represents a building. During this process one or more Airport scenery files are created in the main AutoAsm directory.
14. Click on CALL AIRPORT. This opens the Airport program and loads all the scenery files that were created in the previous step.
15. In Airport, compile the scenery and then delete the scenery window. Repeat until you have compiled and deleted all the scenery windows.
16. Click on the map display and then click on RUN FS. In Flight Simulator you will see many buildings placed close to the town roads, giving the impression of small towns.

As you fly over the scenery you'll see that the lakes don't fit into the terrain at all well. In this final procedure we'll use the terrain mesh bitmap to automatically align the lake elevations to match the terrain.

17. Copy the file terrain.bmp in the terrain\island sub-directory into the bitmaps sub-directory and rename the file to island_elev.bmp
18. On the Main panel, click on the SET button (beneath the BMP ELEV label). This opens the settings for the elevation bitmap. Check that the bounds are the same as on the Main panel and SCALE is set to 3060 (this value indicates that bitmap pixels with the value 255 will generate a terrain elevation of 3060 meters).
19. Enable (check) the BMP ELEV check box. Now, when the LWM/VTP bitmap is read in, the island_elev.bmp bitmap will also be read in. This will enable AutoAsm to automatically assign lake elevations according to the local terrain mesh elevation.
20. Click on the WRITE LWM button to create the LWM scenery. If you then check the scenery with LWMViewer you will see that all the lakes have different elevations. Finally check the scenery in Flight Simulator by clicking on the RUN FS button. You will see that the lakes fit the terrain mesh quite well. Even the small lake in the volcano's crater is at the correct height.