FS Earth Tiles v1.0

User Documentation

Sections

Advanced Documentation

Sections

Shorts:

FSET	= FS Earth Tiles	(main exe / application)
FSEM	= FS Earth Masks	(add on exe / application)

User Documentation

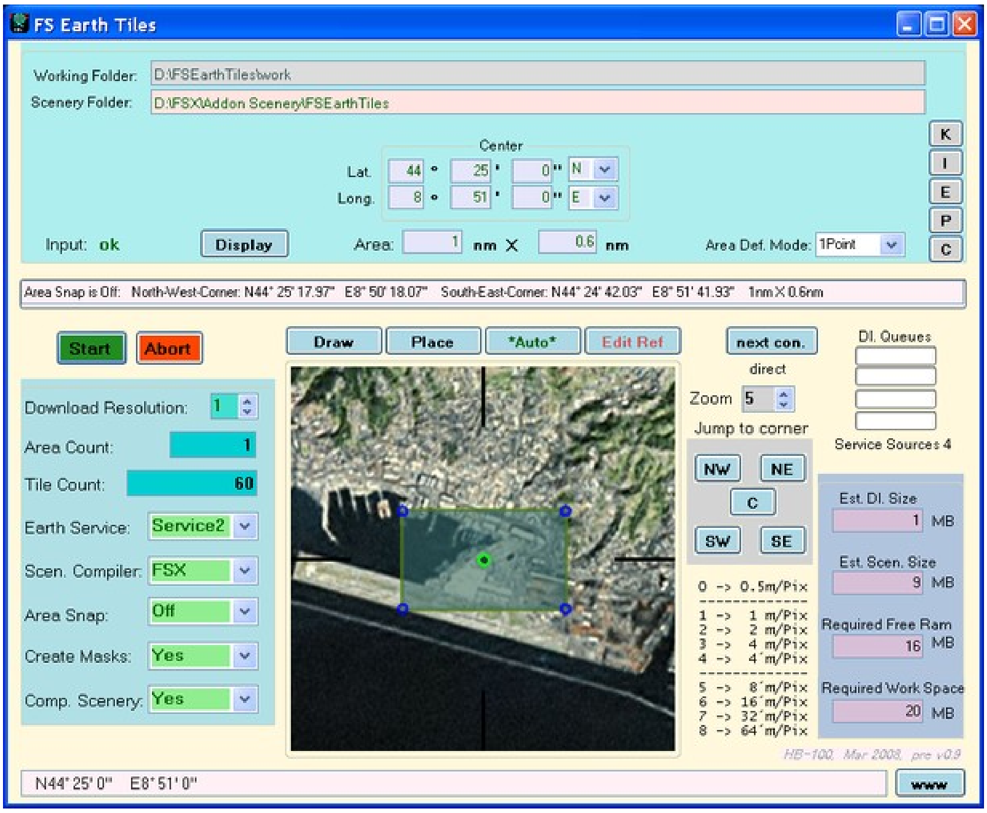
1. Introduction

Congratulation on choosing FS Earth Tiles! (*I always wanted to write this sentence once!* :))

FS Earth Tiles is a Flight Simulator tool I started to develop for me and you, inspired by the cool Tile Proxy project, helping you to get photo-textures as static add-on scenery into the Flight Simulator by accessing earth services.

Important!

All earth services have copyright on their material! That means you may not distribute scenery you create that includes such material. You are also not really allowed to download such material and store and use it in FS or elsewhere without the permission of the services.



Working Folder: D:\FSEarthTiles\work
Scenery Folder: D:\FSX\Addon Scenery\FSEarthTiles

Center
Lat. 44 ° 25 ' 0 '' N
Long. 8 ° 51 ' 0 '' E

Input: ok
Area: 1 nm X 0.6 nm
Area Def. Mode: 1Point

Area Snap is Off: North-West-Corner: N44° 25' 17.97" E8° 50' 18.07" South-East-Corner: N44° 24' 42.03" E8° 51' 41.93" 1nm X 0.6nm

Download Resolution: 1
Area Count: 1
Tile Count: 60
Earth Service: Service2
Scen. Compiler: FSX
Area Snap: Off
Create Masks: Yes
Comp. Scenery: Yes

Zoom 5

Service Sources 4

Est. Dl. Size 1 MB
Est. Scen. Size 9 MB
Required Free Ram 16 MB
Required Work Space 20 MB

```
0 -> 0.5m/Pix
-------------
1 ->  1 m/Pix
2 ->  2 m/Pix
3 ->  4 m/Pix
4 ->  4´m/Pix
-------------
5 ->  8´m/Pix
6 -> 16´m/Pix
7 -> 32´m/Pix
8 -> 64´m/Pix
```

HB-100, Mar 2008, pre v0.9

N44° 25' 0" E8° 51' 0"

2. Installation

1. **Create a folder "FSEarthTiles" for this application and unzip the package into this folder.**

2. **Open the file FSEarthTiles.ini and complete the Service configurations!**
For legal reason's they are only preconfigured. Read what is written at the beginning of that file. The internet is a very helpful source to figure out valid configurations. So use it and do an active search. Do NOT post and asking me for a configuration! And don't be lazy asking others to do your work. It is simple enough to figure out a valid configuration!

3. **Make sure .NET 2.0 (or newer) is installed.**
On all modern updated XP and Vista Systems this should be the case. On older systems or rarely updated systems you might need to visit the Microsoft pages and download the .NET 2.0 framework. (If you don't have .NET installed at all on application start you get a very ugly non interpretable Microsoft error)

3. Start

- Simple double click on FSEarthTiles.exe

Alternative 1: Drag an individual FSEarthTiles configuration file (Partial FSEarthTiles.ini) on FSEarthTiles.exe. Settings in this individual configuration file that you drag on FSEarthTiles.exe will overwrite settings of the FSEarthTiles.ini default file that will always be loaded in first. The individual configuration file can be reduced to the overwrites only but always require the Section -Tags.

Example content of an individual configuration file:

```
[FSEarthTiles]
AreaDefinitionMode = 1Point
AreaSizeX          = 7.6
AreaSizeY          = 6.3
CenterLatitude      = 44.25grad north
CenterLongitude    = 8.75grad east
AutoStartDownload = Yes
AutoExitApplication = Yes
```

Alternative 2: You can start FSEarthTiles.exe with command line arguments. Check the file CommandLineParametersReadMe.txt for the commands.

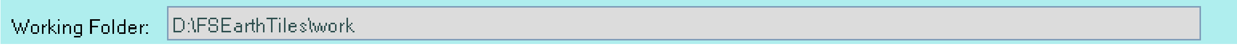
Alternative 3: You can start FSEarthTiles.exe with dragging a .kml File on the exe. The KML-File should contain a polygon or line (no single points) with the label "**Area**". The minimum and maximum coordinates of all the points of the polygon or line specifies the area.

4. Generating your very first test scenery

1. **Start FSEarthTiles** (double click on FSEarthTiles.exe)

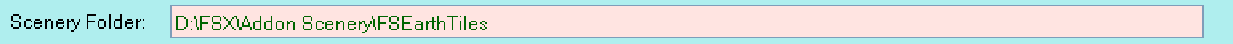
2. **Check that you get a similar picture like the one in the Introduction**. If the Airport view is missing or you see an error displayed like "can not access"... then either your services configuration in the FSEarthTiles.ini file is wrong (Check chapter 2. Installation point 2) or your internet connection is not working. (You need a working internet connection for this of course)

3. **Enter a working folder**. (A folder where FSEarthTiles will store intermediate files required for the scenery generation. The folder will be created if non-existent.
Attention. The default is a folder on the D: Drive! If you have no D: drive change this to C:



Working Folder: D:\FSEarthTiles\work

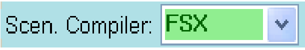
4. **Enter a scenery folder**. (A folder where FSEarthTiles will store the build scenery. The folder will be created if non-existent including the two sub folder's scenery and texture).
Attention. The default is a folder on the D: Drive! If you have no D: drive change this to C:



Scenery Folder: D:\FSX\Addon Scenery\FSEarthTiles

Note that you can also change this folders in the FSEarthTiles.ini file. If you change it there you have them already set right for the next start of FSEarthTiles.

5. **Select your Scenery Compiler** with the ***Scen. Compiler*** switch (FSX or FS2004)



Scen. Compiler: FSX

6. **Press Start** and enjoy the show. (Don't close the windows that opens.. they have work to do and will close when done)

7. Once FSEarthTiles is finish (Status: Done.) close it and start your Flight Simulator and add the scenery (The scenery is in the scenery folder you entered in FSEarthTiles in step 4).

Refer to your Flight Simulator documentation or help if you don't know how to add sceneries in the Flight Simulator. For FS2004 (or FS9 as some say) you need to restart the Flight Simulator after every scenery addition. In FSX you can continue right after the scenery addition.

8. **Choose Free flight:** Airport Sestri in Genoa (Italy), 44°25' north 8°51' east

Take off (best with helicopter) or switch to satellite view and if all went well you should see a little peace of photo texture scenery. :)

5. Generating your individual scenery

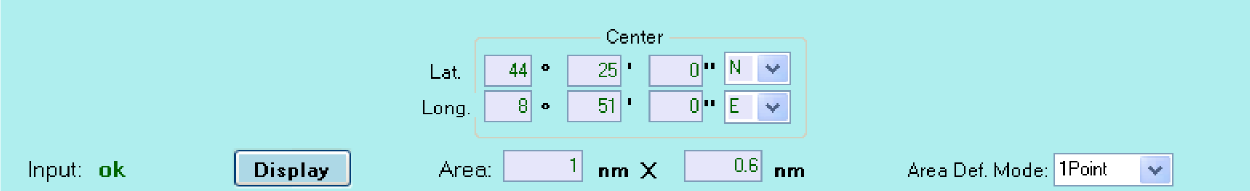
Once you created your very first test scenery described in Chapter 4 you should be ready to build your own scenery.
The best is to start with GoogleEarth or ww.flashearth.com and check out for your location first.
Next you start FSEarthTiles and specify the working and the scenery folder. It's recommended to specify own work folder and scenery folder for every scenery you create so it will never conflict with old sceneries and you can more easily find the scenery and its source files again.

Now you have 4 ways to define your location / area you want to download and install as scenery.

First way

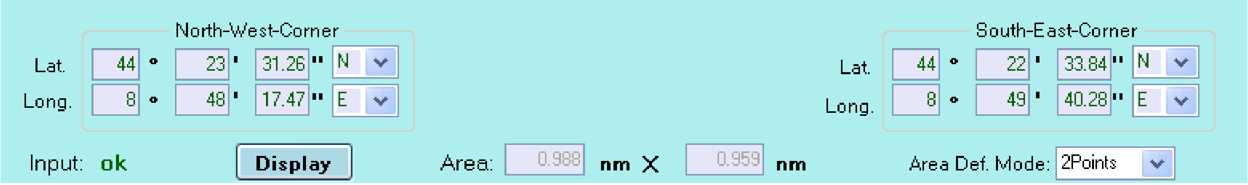
The first way is by entering the specific coordinates and this again you can do in 2 ways depending on the ***Area Def. Mode*** selector setting you can:

1Point mode: You enter the center coordinates of the area of interest in the ***Center*** filed and right below the size (***Area*** field).



Center
Lat. 44 ° 25 ' 0 " N
Long. 8 ° 51 ' 0 " E
Input: ok
Area: 1 nm X 0.6 nm
Area Def. Mode: 1Point

2Point mode: You enter the North-West-Corner coordinates of the area of interest and the South-East-Corner coordinates.



North-West-Corner
Lat. 44 ° 23 ' 31.26 " N
Long. 8 ° 48 ' 17.47 " E

South-East-Corner
Lat. 44 ° 22 ' 33.84 " N
Long. 8 ° 49 ' 40.28 " E

Input: ok
Area: 0.988 nm X 0.959 nm
Area Def. Mode: 2Points

Press the ***Display*** Button once you have entered the coordinates to see the whole area displayed in the Display below

Note: The world coordinate format is **hours**, **minutes** and **seconds.**

1 hour = 60 minutes, 1 minute = 60 seconds.

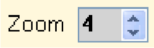
If you happen to meet coordinates like 44 25.9' then this means 25 point 9 minutes and that is equal 25 minutes 54 seconds (0.9*60sec). If this is too complicated then enter the whole 25.9 in the minute field.

Please learn to know world coordinates before you post rubbish like you can not enter the plane position in FSET like I read in a foreign forum once.

Second way

The second way is by simply drawing the Area of interest into the display.

To do this first use the Zoom selector and zoom out a little so you can see more. (higher numbers)



Zoom 4

Navigate (**left click in display and drag the map**) to your location of interest and zoom out so that you have the full area you want to download in the display.
You can do a double click to center the map on a certain position.

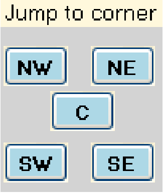
Then click on the ***Draw Area*** button

Your cursor changes into a cross. Move it to the top left corner of the location you want to download and **left click and hold.**
Move your cursor to the right corner of the location of interest and release.

Now you have drawn your area. If you want to modify it simple click and hold in one of this Rings and drag it around. (blue ◎ , green◎)

If you want to see the Area cantered and fit into the display again the simplest way is by clicking on the ***Display*** Button any time you like.

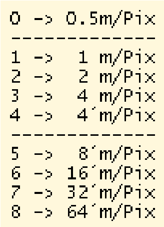
You can also use this keys that simple centers either one of the 4 corners of the area or the area center in the display.



Once you have drawn your Area you should select the "Download Resolution" level,

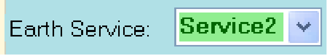
Download Resolution: 1

A resolution Level Table equates to what your Scenery resolution will be in the Flight Simulator you see displayed on the Tool:



```
0 -> 0.5m/Pix
-------------
1 ->  1 m/Pix
2 ->  2 m/Pix
3 ->  4 m/Pix
4 ->  4´m/Pix
-------------
5 ->  8´m/Pix
6 -> 16´m/Pix
7 -> 32´m/Pix
8 -> 64´m/Pix
```

You should check if the service has Tiles available with such a resolution by setting the Zoom Level equal to the Download Resolution Level!
You can try out a different Service. (when you have configured them). You will see that the textures have different looks and different quality.



Earth Service: Service2

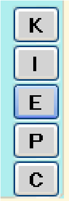
Select your Scenery Compiler if you haven't done this fix in the FSEarthTiles.ini configuration file already.

Then when ready start the Download .

Third way

The third way is by passing a file to FSET that specifies the coordinates of the area. (PartialFSEarthTiles.ini or AreaKML.kml)
The File can be passed by dragging it on the FSEarthTiles.exe to start the Application or later at run time by dragging it directly onto the FSET coordinate input field group.

In case you have named the file PartialFSEarthTiles.ini or AreaKML.kml and placed it into the Working folder you can also load it by using the grey keys
I (import .ini) and K (import .kml)



The Keys are:
K = Import AreaKML.kml file located in the Working Folder
I = Import PartialFSEarthTiles.ini File located in the Working Folder
E = Export PartialFSEarthTiles.ini File to the Working Folder
P = Paste FSET-Input data from the clipboard into FSET
C = Copy FSET-Input data to the clipboard.

The Copy and Paste includes the information of the specified folders, whereas the Import and Export buttons only transfer the Input Coordinates.

Fourth way

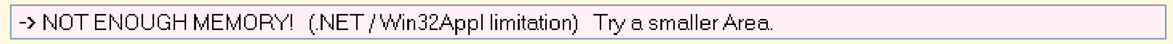
The fourth way is dragging data from one running FSET-Application into a second FSET-Application.
To do this you simply click in the Coordinate Input Group, hold the mouse and move to the second FSET-Application.
This will copy all GUI settings from one FSET into another and is very helpful if you want to work with multiple FSET's running at the same time.

The coordinate data's that become copied are the Snapped coordinates and NOT the input coordinates. You can drag and drop the data with a simple click on the coordinate input group to one and the same FSET Application. In this case the Input coordinates become set to the Snapped coordinates.

The best way to clone FSET between one and another FSET Application is
1) a short simple click with the mouse into the coordinate input group of the source FSET to set Input coordinates = snap Coordinates
2) a click and hold, drag over to the second FSET Application and release to copy/clone all the FSET data to the second FSET.

6. Area and it's Reference Area

When you experimented a little with downloading an Area in earlier pre versions you probably already have seen this message by now:



-> NOT ENOUGH MEMORY! (.NET / Win32Appl limitation) Try a smaller Area.

FSEarthTiles assembles the texture of the area in one large bitmap in the memory. If in the past you drew a large area and choose a high resolution (=low resolution level) the bitmap became so large that it can not be held in your memory.

Required Ram 10404 MB

FSEarthTiles really tries to allocate the texture in a pre work step and only when the system says there is no space it reports that it can no be done. .NET limits a large part before the memory is used all up on XP 32Bit systems. On a Vista 64 Bit you can download larger areas. The limit there is on the general 32Bit application limit (FSEarthTiles is 32Bit compiled) which is about 2GByte for data.

Another good indicator to see how close you are to a limit instead of the required ram is the Tile count

Tile Count: 806

A Tile is a unit used by the earth services and specifies a photo texture with a fixed size of 256x256 pixel usually.

The way out of this memory limit is to work with multi areas. That is something I never intended to do but was requested by a lot users.
There are two possible thinkable ways to do this:

1) Draw a large area and auto split it into smaller areas.
2) Draw a small Reference Area and assemble a large area with single areas of the size of the reference area.

Method 1, as cool as it is for the user, is very troublesome as soon as you do not have freedom in how you split the area. For FS2004 for example we are bound to handle/create areas of Flight Simulators LOD-Grid LOD13.
You could think of splitting everything into a LOD13 as smallest unit but with a large spectrum of resolution / zoom you either create thousands of thousands of very small area's on the one end of the spectrum or you hit your memory limitation just with one such unit on the other end. It is also hard to ensure precise Area boundaries.

Method 2 is a lot better from a developers standpoint because by having the user defining a reference area that fit's within it's memory, the way to assemble a large Area is all determined and trouble free.

Therefore FSET pre v0.8 (and earlier) ways was the two step:
1. User defines a Reference Area that can be processed.
2. User draws the large area that is then assembled of Reference Areas

Like v0.8 and earlier FSET v0.9 and later is all based on assembling and not splitting. It holds one Reference Area and the desired Area.

I recognized that for a lot users the Assembling process is uncommon and users did not understand how this works.

In a big step to make it more user friendly I took the time to implement an Auto Reference Area mode.

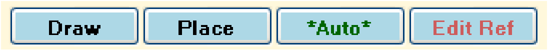
With that it is now possible in v0.9 and later to simple draw your large area of interest and FSET auto splits that Area.

(Yes I said I never would do that. But I did so many things in FSET I never wanted to do because they all mean a lot of work)

The Auto Reference Area mode

If you start FSET you will be in the Auto Reference Mode.

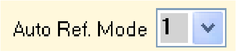
This looks like that:



The *Auto* is the sign that you are in the Auto Reference Area Mode.

In this mode you can simple draw your large area and start the download. A users dream.

If you choose FS2004 as Compiler your AreaSnap is set to LOD13 (instead off) and you will see an additional Switch (left button of FSET)

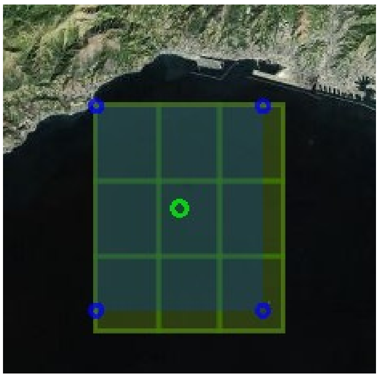


Auto Ref. Mode 1

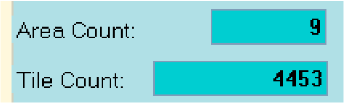
With this you can select a Auto-Reference Mode:
There are 4 Modes:

1 - Maintains the Input Coordinates best. Snaps to the Grid. (Area Snap is always to the next larger) But this one can produce long Strips. (a prime number AreaSnap Unit count can not be split)
2 - Allows enlargement of the Area to +1 more Snap Grid Unit.
3 - Maximum Freedom. This comes closest to the AreaSnap off and produces for your memory the most optimal Areas-Units. (Large Area Units are better for FSX)
4 - Resolves/Splits the Area completely into AreaSnap Units. The Reference Area becomes equal to 1 AreaSnap unit.

An Area assembled of largest possible Reference Area's (AreaSnap off):



This is your Area which in my example consists of 9 simple (reference) areas.
You also see them totalled up.



Area Count: 9
Tile Count: 4453

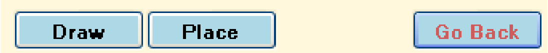
If you press start

FSEarthTiles will start processing each one of this 9 Areas one after another. It completely finish the scenery of each area before starting to download and work on the next simple area.

The Edit Reference Area mode

If you feel the need to edit the Reference Area Manually you can click on this Button

The Button Bar will change to:



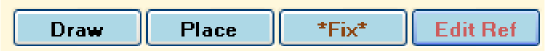
and sign that you are now in the Reference Area Edit mode.

You can simple draw or edit the Area as you do it with the large are of desire.

If you finish you click back and you will be in the Fix Reference Area mode.

Important: Make sure you leave the Edit Reference Mode before you start the download by pressing "start" else it will only download the Reference Area!

The Fix Reference Area mode

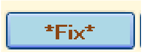


This is the Sign that you are in the Fix Reference Area mode. You can get to here from the Edit Reference Area Mode or when you click on the Reference Mode Button.

In this mode the Reference Area is frozen and the area you draw will be assembled out of this fixed Reference Area. (Similar to v0.8)

There is another way to define a Reference Area without using the Edit Reference mode:

In the Auto Reference Mode you can simple click on the Reference Mode Button button which will change to indicating you changed to the Fix Reference Area mode and the actual by the Auto mode evaluated Reference Area becomes a frozen fix Reference Area.

To switch back to the Auto Reference Mode simply click on the  reference mode button.

Auto-Ref-Mode   Fix-Ref-Mode

<---->

toggle

7. Water

For water creation you need a Vector Drawing Tool.
For FSEarthTiles you can use

Google Earth (4.2 Beta) http://earth.google.com/

and/or

Inkscape (0.45.1) http://www.inkscape.org/

The basic concept is to draw 2 poly lines (path) that enclose the transition from land to water.

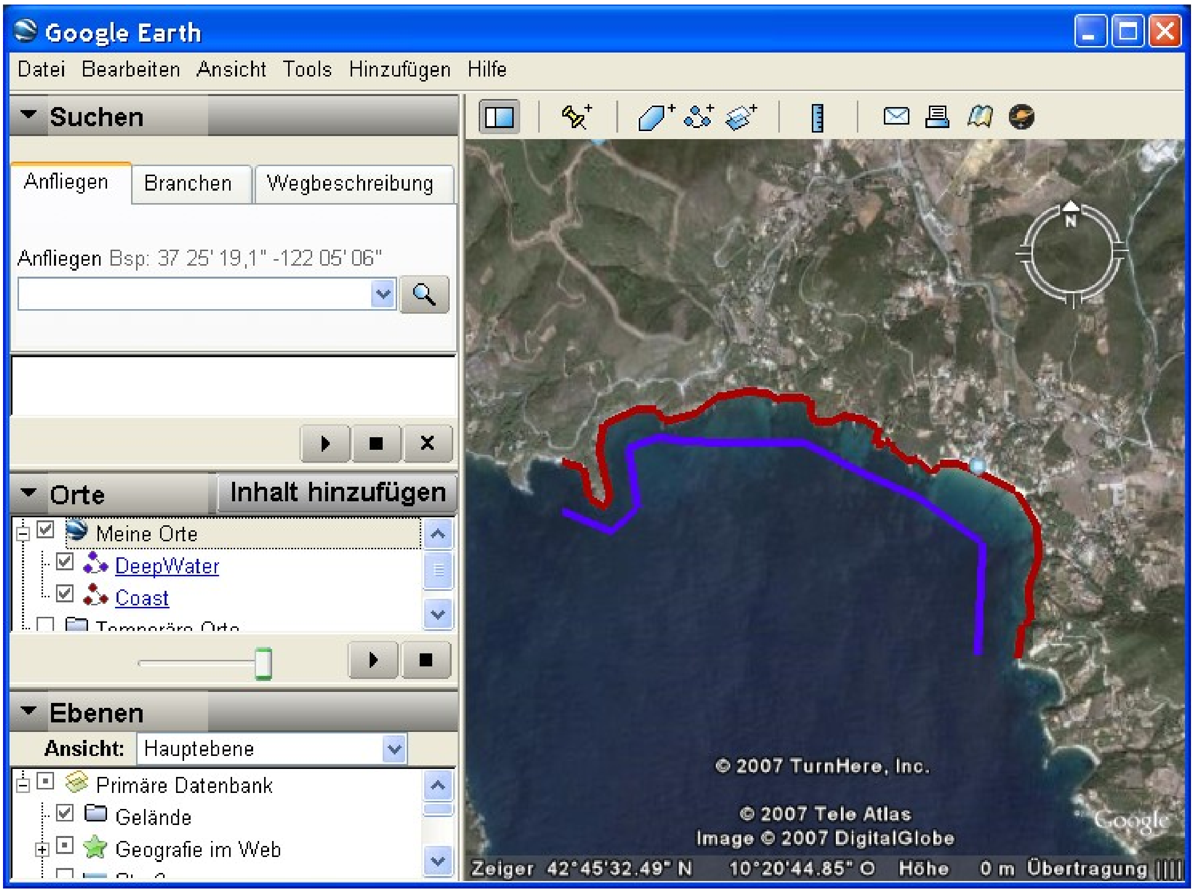
One poly line you have to label **Coast** and you should draw it as exactly on the coast as possible.
The other poly line you have to label **DeepWater** and this one you draw where deep water should start. This line can be coarse.

You can draw more than one such poly line for each type. Whereas in GoogleEarth you can give them exactly the same name on Inkscape a unique name
is required. You can add anything you like to the label name to make it unique. Example Coast1 Coast2, DeepWater1, DeepWater2...

The poly lines may be closed and become a polygon. FSEarthTiles will process poly lines and polygon's. Internal there is no difference because all that is processed are the single lines within a poly line or polygon. (with v0.8)

Google Earth

Google Earth you start **BEFORE** you start FSEarthTiles.
You draw the water transition lines of the WHOLE area in advance. If you plan to download Multi Areas you have to cover the whole Multi Area.



Make sure The Folder is selected that contains all your poly lines. In my case "Meine Orte"
Then save the places (File -> save places in..) **in the FSEarthTiles Work Folder** you are going to use later.

The File Name has to be: **AreaKML.kml**

Important: You need to select FileType: Kml (*.kml) ! The default google setting is Kmz.

Now start FSEarthTiles and set the working folder.
Make sure Create Masks is set to Yes.
Then start your download.

(FSEarthTiles is preconfigured to run with the .kml File so there are no configuration changes required.)

Inkscape

Inkscape will be started from FSEarthTiles after **EVERY** downloaded area.

To do this you need to change the FSEarthTiles.ini configuration file to:

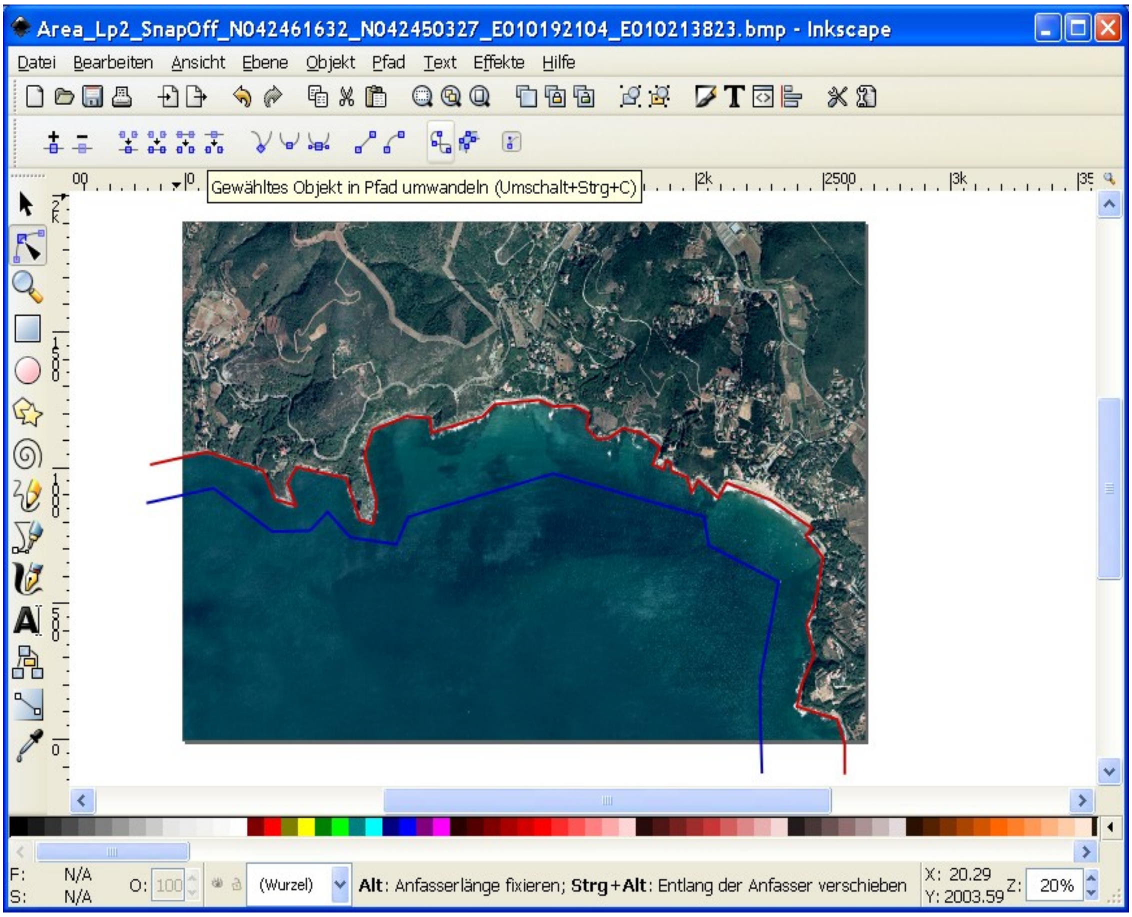
```
#--- Third Party Tools ---
UseAreaKMLFile                  = No
UseScalableVectorGraphicsTool   = Yes
ScalableVectorGraphicsTool      = D:\Inkscape\inkscape.exe
```

You have to enter your path where you installed Inkscape.
You can also leave the UseAreaKMLFile to Yes but when it finds a AreaKML.kml file it will be used then also (mixes both together)

Start FSEarthTiles.
Make sure Create Masks is set to Yes.

Then start the Download.

After Downloading FSEarthTiles will fire up Inkscape where you draw the Coast and the DeepWater line.
Give them unique names by adding numbers. (Object properties, ctrl-shift-o)
It's a good idea to check if it accepted the name by reopening the properties again. If it doesn't accepts the name (not unique) you don't see this immediately.



Area_Lp2_SnapOff_N042461632_N042450327_E010192104_E010213823.bmp - Inkscape

Gewähltes Objekt in Pfad umwandeln (Umschalt+Strg+C)

F: N/A
S: N/A
O: 100
(Wurzel)

Alt: Anfasserlänge fixieren; **Strg+Alt**: Entlang der Anfasser verschieben

X: 20.29
Y: 2003.59
Z: 20%

Make sure you draw the Line a good way outside the Bitmap also!
If you cut too short the water land detection might go wrong.

When drawn go to file and save as.
Important in version 0.45: Delete the .bmp from the file name!

Then save it. It should save the same name but with a .svg suffix (auto add clicked on) to the correct working folder.
You can let the default Inkscape .svg selection. It works with that.

Close Inkscape windows and FSEarthTiles continues with starting FSEarthMasks that creates the water bitmap out of the .svg file.

Water and FS2004

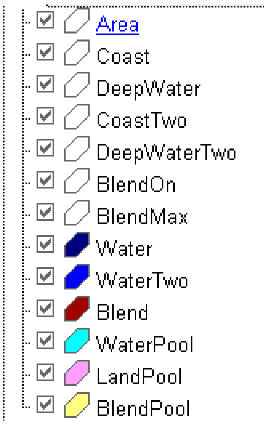
FS2004 does not know Transitions.
Water creation for FS2004 is handled as all the Transition = Water.

FS2004 also does not know Blending. Blending is ignored.

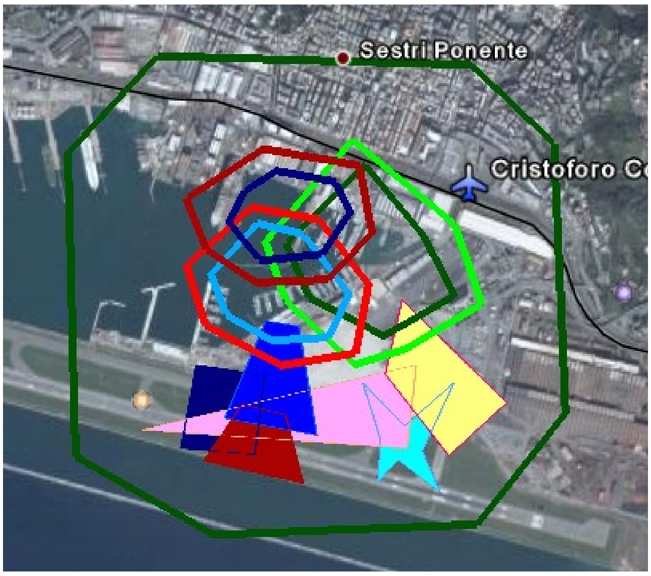
Other KML Commando-Lines

Note: Check for some .KML examples that possible comes with FSET.

With v0.9 you have a number of more Commando-Lines Label you can use for FSET.:



- ☑ Area
- ☑ Coast
- ☑ DeepWater
- ☑ CoastTwo
- ☑ DeepWaterTwo
- ☑ BlendOn
- ☑ BlendMax
- ☑ Water
- ☑ WaterTwo
- ☑ Blend
- ☑ WaterPool
- ☑ LandPool
- ☑ BlendPool



Coast and **DeepWater** are only one pair of Lines marking a Water-Transition.

With v0.9 you have 2 more such Transitions:

CoastTwo and **DeepWaterTwo** best used for lakes, rivers and such.

BlendOn and **BlendMax**. This is used to Blend a texture out.

Blend means that the original FSX background is looking through. When there is water you have water. When there is land you will see land.
It's a Transparency you can start from 0 to 100%.

Note that also the Water Transition by FSEarthTiles/FSEarthMasks default is set up as Blend/Transparency and not using Water Reflection.
The water looks much better and has a smoother transition if you do it by a Blend/Transparency effect and don't use Water-Reflection. (digital on/off and color changing over lighting)

This means the water transition relays on a good FSX mesh.

Area is of interest for FSEarthTiles main Application only and specifies the Area Coordinates (min and max of all points coordinates in the polygon or line) you want to download.

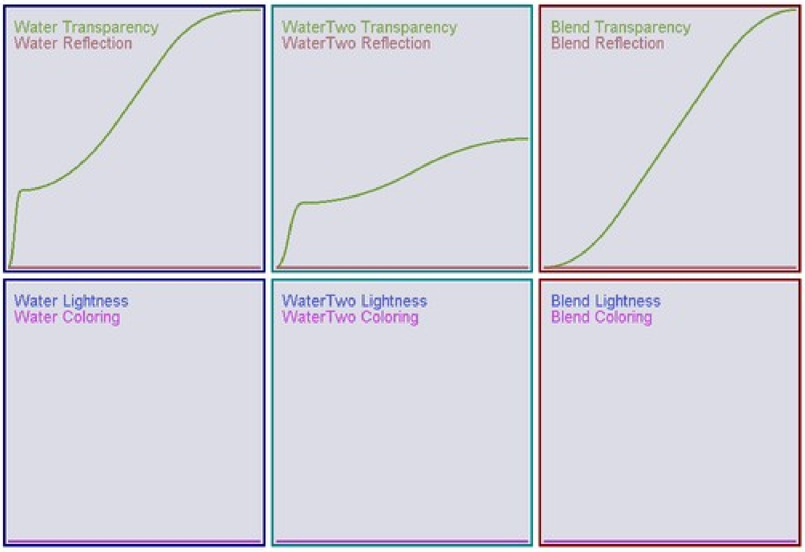
WaterPool, **LandPool** and **BlendPool** are an alternative way to mark Water, Land or Blend Areas with just one polygon (no open lines!).
Different to the Transition Pools have a constant value! I.e. all Full Water, Full Land, Full Blend.

Pools are always dominant (higher priority than transitions) and the order you draw them is important also. The most recent is drawn on top of the one before.

The Transitions are on the lowest Level and all the 3 kinds share the same level with equal priority to each other. Here at the end the just at one point higher value in a transition will dominate.

The Transitions also have a sort of Pool polygon's that can be used although on the same and lowest level as the transitions.
This Transition-pools are **Water, WaterTwo** and **Blend**. The value they have is the Value of the specific Transition end.

The Transition function as well as the Pool Values can be completely customized. For this you have to check the parameters in the FSEarthMasks.ini.
There is also a parameter that generates a Plot graphic(TransitionPlotGraphicBitmapFile.bmp) in the work folder to see the Transition functions.



I have no time at the moment to explain this parameters.
Just one thing you need to know and this is that this is a function that consists of 3 S whereas the first S and the Last are bound to the Entry Value 0 and exist Value 1. (1 = 100%)

If you want a constant value for a whole transition you need to deactivate the first and the last S and only use the middle one! To do that you specify the first connection point between the first and the second S to x = 0 and the second connection point between the second S and the third S to x = 1.0

Re-Coloring Water in a Water-Transition

Like you can see in the Plot you can also create Transition functions for recoloring of the water. (parameters in the FSEarthMasks.ini)

Important: Recoloring is only active when you let FSET/FSEM create a Summer Bitmap! This way the original Source Bitmap coloring will not be altered and the process can be repeated later.

I don't think it is so useful but well if you need it you can do it.

8. Seasons

You can create Seasons with FSET/FSEM. This is done by altering the color pixel for pixels. The routines are original from Thomas M. and their original purpose is to create seasons for the Switzerland Pro scenery. I included them into FSET and expanded spring,night and adapted them a little.
Season creation works but you can not expect too much. A main problem is that the service's tiles are compressed (.jpg) format and in Hard Winter for example you will see jpg rectangle artefacts. Also the color between the services and regions vary strongly so that the color detection and altering doesn't always fit so well.

9. C# Scripts! FSET can be adapted, also by you!

Don't like the generated Seasons Bitmaps? Or do you want to include a special FS commando in the Info? With FSET pre v0.9 I included C# Script. This is a third party freeware C# Library by Oleg S. allowing to handle C# code as a Script at runtime.
You will find .cs files in the FSET folders. This are C# Scripts. If you open them and look at them you will discover that this is C# code. Different to a C# project where the code needs to be compiled ahead of time, this .cs Script Files will be read in and compiled at run time.

Activate the C#Scripts handling by setting the parameter

```
UseCSharpScripts      = Yes
```

in the FSEarthTiles.ini and the FSEarthMasks.ini files.

Then do your changes in this .cs Script File codes. The next time you start FSET and/or FSEM the Scripts with your changes become read in compiled and used.

Most data you need are either passed as parameter's or you find them in a class such as EarthConfig in the FSEarthTiles project. To see what such a class contains you should check with the FSET Source files. But a lot of things can be done without having to look for additional parameters in the sources.

10. The story of FSEarthTiles development

When FSX came out the disappointment was big. Not only the hardly tweak able performance was bad, but also the look! Now I know some will disagree when reading this.
But honestly, do you really think the default look of FSX is in any way acceptable especially if you compare it with FS2004 with 3 years improvements form the FS Community?
It almost looks like that 3 years improvement is wasted. The FS Developers did not take notice of any improvements in the scene ignored it and compare their new product with their old one instead with what is wanted. It's so simple: What makes a difference between a Flight Simulator we like and one we don't like!? In most part it is flight-physics, performance and look!

Flight Physics: Not much to complain here except the A321. With or without EFCS it flys not like an Airbus.

Look: FSX default textures are the worst I've ever seen. Even worse than FS2004 defaults. The bigshock I got when I saw that they covered Europe with a desert textures (or with what is their idea of a desert texture) in autumn! You can create the best product but you kill it all with a bad look. What had it cost you to create sharp quality textures?

Performance: A disappointing performance doesn't improve the acceptance of a product. It doesn't helps if you say the performance will be acceptable with new hardware in 2 or 3 years. We want to fly now with this product. In 3 years you bring out a new product anyway.

As disappointing as FSX was at first sight I also saw the potential it had to at least bring back a good look and this was even before I knew about Tile Proxy and so I left FS2004 and switched to FSX. All what was required is: 1 Tweaks to make the displayed texture sharp again. 2. new default texture files.

I had moderate success at first, nothing really helped. It all changed once when I heard about the Tile Proxy project that during the flight online streams photo textures of earth services into the Flight Simulator. I was impressed how it all looked better with this textures. But this better look had a price. It was too slow for flying with a Jet. Also the LOD radius was recommended to set to minimum which reduces the sharpness you can get and I had to pause a lot also on slower planes to get the scenery display catch up. And then there were this Vista limitations which made using Tile Proxy troublesome. (F8 on boot, User access right settings). And there was this long long scenery loading time when you started a flight.

In short it was not feasible for my commercial Jet flying needs. I like to fly commercial Jet's not at last because FSPassangers made it a lot more enjoyable.
My Idea was to fly from A to B and that requires detailed ground texture at the point A and at the point B that also can handle the speed.
It was clear to me that what I want can only be archived with a static scenery and not with an online

streaming.

When I saw the Tile Proxy configuration I was surprised to see that the earth services had their material simple accessible with every Web Browser. There is no encrypting or anything it's all open there in the public. The only point left to do was to figure out the mapping of the Tiles (1 Tile= a peace photo texture of a fix size in pixel: 256x256) on the globe. This took me 1-2 day to figure it out. From then on I knew it was possible to write a program to create static sceneries out of this earth services tiles. FSEarthTiles pre Version v0.1 that already could to this was out after 8 days developing using C#.

I didn't know it then but today I have the look that makes FSX looking way ahead of FS2004 now. I could tweak the sharpness and with FSEarthTiles I can replace the ugly default textures with great looking photo-textures. It really makes a big big difference. Just the bad FSX performance is still a handicap to really enjoy it. It will be solved once when a 6Ghz PC is out because that is what is required for 20fps (AI + AutoGen + bloom + water).

```
//--------------------------------------------------------------------------
//            FS Earth Tiles pre V0.99      HB-100 July 2008
//
//         written / programmed by HB-100       (copyright HB-100)
//
//
//       This program and the source code is dedicated to the
//                Flight Simulator Community
//
//   and it's tireless effort to transform the Flight Simulator into an
//   awesome experience for everyone!
//
//
//        The program and the code may be used free.
//              No limitations from my side.
//
//
//    Thanks to the following active contributors during the development:
//            Steffen I.   (Expansions/Patches/Service3)
//            Wolfram R.   (Feedback/Ideas/Testing)
//            Christian B. (Hadn't started this project when there
//                          had been no Tile Proxy)
//            Antoine D.   (FS Discussions/Ideas exchange)
//            Thomas  M.   (CH Pro Seasons Routines)
//            Oleg S.      (C# Script Library)
//            Kerry
//            Jojo
//            fly-a-lot  and all the people of the FSPassengers forum
//
//--------------------------------------------------------------------------
```

Advanced Documentation

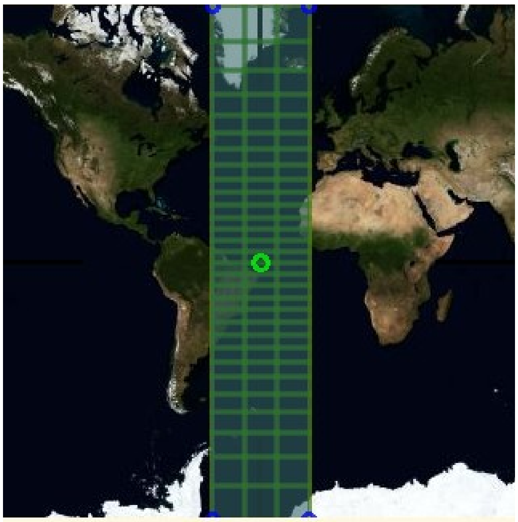
11. Earth Services and Earth Projection (Mercator)

FSEarthTiles is designed to work with Earth Services that use the Mercator projection. With the Mercator projection the Earth surface is projected onto a flat map.
The Mercator projection is by far not the only earth projections that exists. For some reasons that I don't know they choose this projection as their base. Maybe it is because it is a often used projection.

A Mercator projection map of the Earth looks like the picture in this link:

http://de.wikipedia.org/wiki/Bild:Mercator-projektion.png

A similar view can be generated with FSET:



The Areas in this grid here all have the same Latitude (north border – south border) and Longitude (west border -east border) coordinate difference.

Like you can see the Longitude part of the Grid is in a constant raster whereas the Latitude part is stretching toward the earth poles.

In fact the stretching grows exponential toward infinite at the poles. This means that the earth poles itself can NOT be covered by a Mercator projection map. The service coverage stops at a certain Latitude. The exact value where the map is cut in Latitude is extremely important to be able to map any Tile of an earth services back into an exact earth location. The reason is that this cut value is used to normalize the Mercator projection. It took me a day to figure out the exact value by systematic narrowing it. To my surprise the value turned out to be PI (in the Mercator space which transform to about 85.05 and some more digits degree Latitude). PI is a very elegant value to cut/limit the Latitude of a Mercator Map because it goes hand in hand with the maximum value you get when you transform the Longitude maximum value that is +180 degree into the radiant system → PI. I am not in the mapping business but because of this elegant value I guess that it is most probably a standard normalizing method of common Mercator maps.

More to the Mercator projection you can find here:
http://en.wikipedia.org/wiki/Mercator_projection

and in the FSET source code package in the EarthMath class (file EarthMath.cs) of FSEarthTile.

But for those impatient that like formulas here is for example how the transformation formula for Latitude Phi(in radiant) into Mercator value (not normalized) y looks:
y = 0.5*ln((1+sin(Phi))/(1-sin(Phi)))

Knowing the used transformation of the earth services is one thing. Knowing how the photo map is

stored and accessible the other.
Now you know Tile Proxy opened the door for this project in a way.
The key element was to see that the photo map is open accessible for everyone with a browser and an internet connection. It's not protected. It is just an URL (a Web page address).

Let me remind you here again that also when it is not protected and open accessible that the photo material has a copyright. The earth services holds the rights to it!

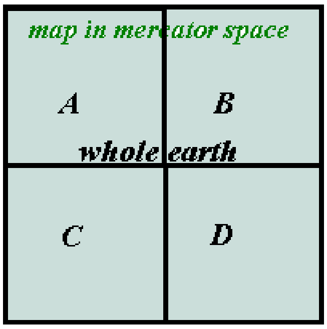
In the URL you give an address code of a specific Tile location in the Mercator space.
(In the Mercator space it is a ordinary raster)
One Tile as mentioned already is a piece photo texture of 256 x 256 pixel. It comes as a .jpg file.

The services have multiple map's or layers stored with different resolution (meter/pixel). This layers is what you have as zoom and download resolution level.

In the Top level (in FSET it is zoom Level 18) the whole Earth Map is one single Tile.

map in mercator space
Top Level
whole earth
=1 Tile

In the next Level (zoom lvl17) the number of Tiles becomes doubled in each dimension 1 Top Tile -> 4 lower Tiles



The whole earth is now covered with 4 Tiles. Each Tile is still 256x256 pixel that makes 512x512 pixel in total. With have a better resolution (meter/pixel) now

And this doubling or quadruple in Tile numbers goes on for every level we go downward.
At Level 1, where we have about 1meter/pixel and an interesting resolution for the Flight Simulator, the Tile count is extremely large! It is 4 ^ (18-1) = 17'179'869'184 Tiles.
If we consider the memory for one true color Tile in native format: 256*256*3 = 196'608 Bytes
Then we get 17'179'869'184 * 196'608Bytes = 33'77'699'720'527'872 Bytes

Seeing this number it is simple to grasp that you can not have the whole world with 1meter/pixel resolution in your Flight Simulator.

There are two ways how the Tile addressing is done.

One is starting at the Top Level and then going down Level for Level where on each Level you check in which quadrant Tile of the quadruple your wished position is. If we assign a Letter for each quadrant Tile like above A,B,C,D we can create an address code by doing the level step down play.

Example we want to find the Tile that covers Genoa (default location)
Its in Europe that is quadrant B so we note B. One level down we figure it is in the south west quadrant of B. That would be a C. We now have BC. And this way we go deeper and deeper and get something like:

BCACCDBACBCA ..

The result is the address of that Tile that covers the location Genoa. As more letters we add as better resolution (meter/pixel) we get. Of course we cover less and less of a region with that high resolution Tile then also and you have to load more and more Tiles for the same region.

Now this can be different from service to service where you have to enter this address in the URL. For this in FSET is a place holder %s in the service URL you enter in the FSEarthTiles.ini file.
This %s becomes replaced with the code.
Instead the letters A,B,C,D for the quadrant coding services can use different letters 0123 for example..
That's what you specify in the ServiceCodeing parameter in FSEarthTiles.ini

The other Tile addressing method services use is a more direct addressing.
Instead going down Level for level you specify the level of the map direct with a parameter in the address (URL) example z=7 for zoom level 7

On a given Level you know exactly how many Tiles you have to cover the whole world. That's why you can put a simple X/Y Tile number coordinate grid on it to address the Tiles.
An example Address then can be x=1230,y=4019,z=7

Where the origin of the Tile numbering coordinate grid is depends on the service.

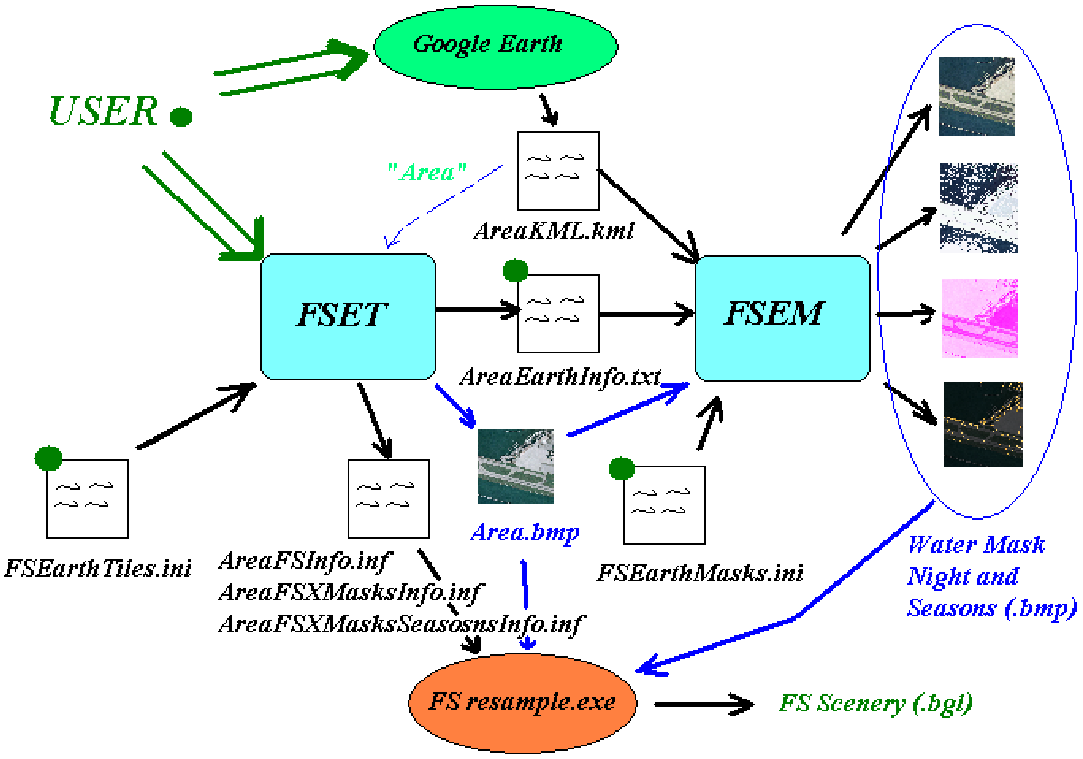
Interestingly is the I FSET internal use this system to address Tiles and it almost mapped exactly on one of the services that lynx later added.

This Tile Address code x=,y=..is than also placed somewhere in the URL. For FSET the same place holder is used: %s However the identification for such a service is different. With ServiceCodeing = xyz is told to use such a specific addressing. In this type of addressing the services usually are very different.

That means if you want to add more services with x,y,z codeing you need to generate a recognition ServiceCodeing = myrecognition (anything that is not 4 letter long and not xyz) and code it yourself in the TileCodeingScript.cs Script. (Activate the Scripts in the ini file for this).

Don't hesitate to do this. It is simple and the scripts are meant to do exactly such!

12. Files and File flow



The picture looks a little overloaded I have to confess. But it is worth to study it.

The simplest case is the USER starts FSET, draws the Area and starts the download.
And the rest is done automatic.

If USER wants Water then the USER starts Google Earth first and creates the AreaKML.kml file with the Water information and maybe an area poly line. Then the USER starts FSET eighter imports the AreaKML.kml file or draws the Area and starts FSET. And the rest is done automatic.

If the User wants Seasons the User has to select them first in the FSEarthTiles.ini

The User can also do adaptation in the FSEarthMasks.ini File and in the AreaEarthInfo.txt File that is created by FSET and delivered to FSEM.

The AreaEarthInfo.txt contains the information for the FSEM what to generate. For example what Seasons. Or for what Flight Simulator.

There is also a PartialFSEarthTiles.ini file not drawn here. That one is created on the E button (export) and can be read again in with the I Button (Import). It is the same format as the FSEarthTiles.ini file and indeed you can specify everything you like that you can in the FSEarthTiles.ini configuration file. It's a full configuration file like FSEarthTiles.ini just with a specific name for the E and I buttons.

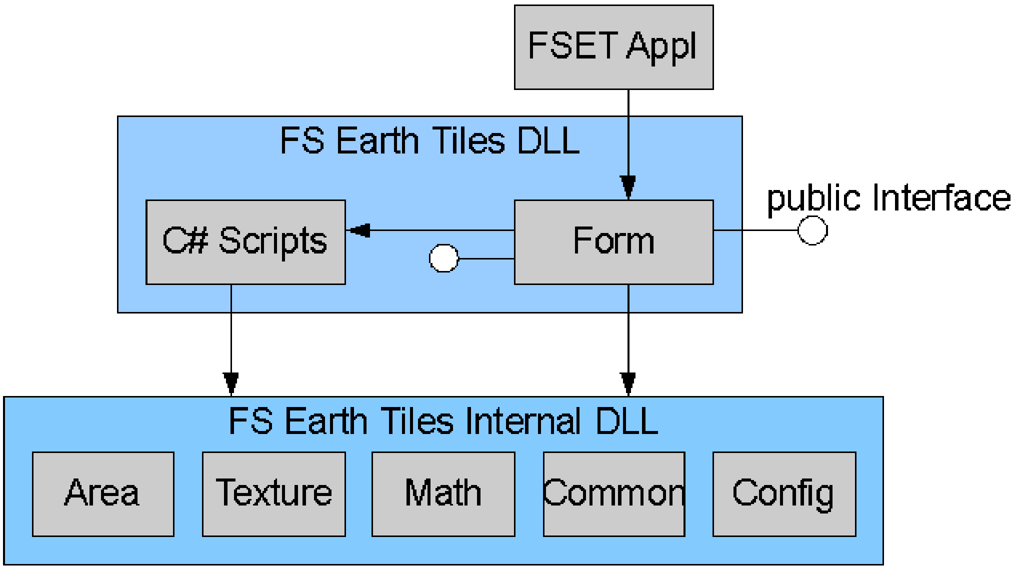
Note that you can also drag and drop an FSEarthTiles.ini file or that PartialFSEarthTiles.ini file on FSET at runtime. That can be very practical if you want to try out different configurations.

You can also drag and drop the AreaKML.kml file on FSET at runtime to define the Area if it has an with Area tagged poly line.

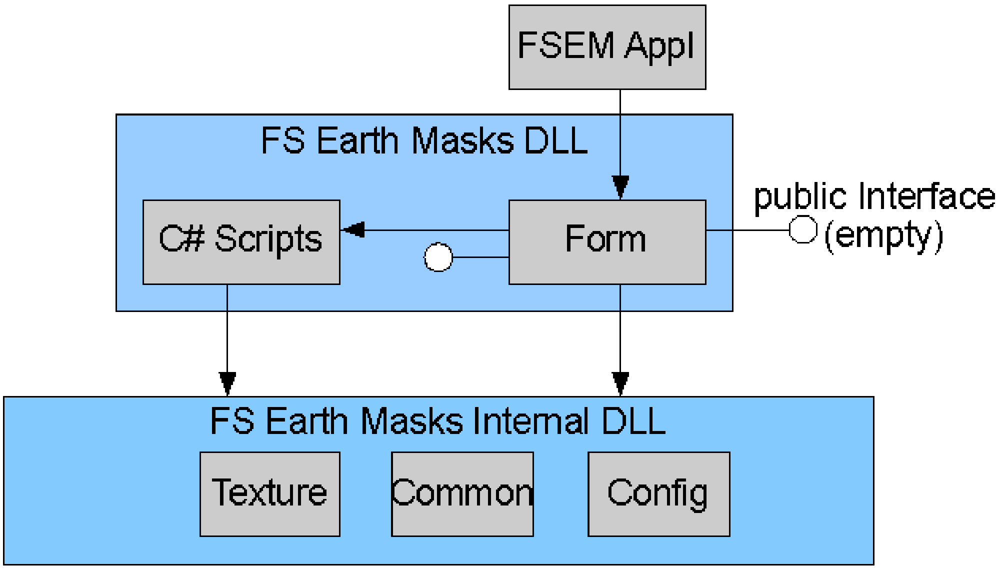
It's also possible to drag and drop the .kml file or the .ini file on the FSEarthTiles.exe to start FSET with using them.

13. FSET and FSEM Internal Structure

FS Earth Tiles Application



FS Earth Masks Application

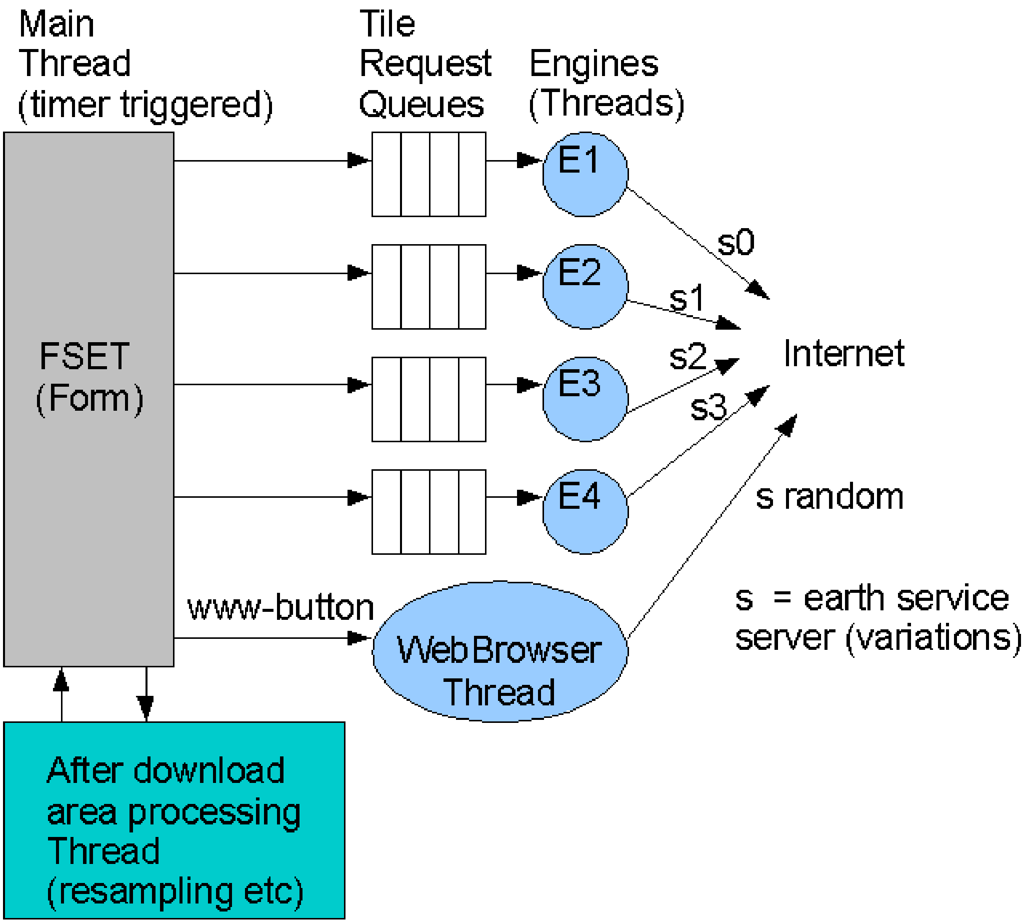


Original it was not my intention to work with C# Libraries (DLL's). If it is not a large project where such a capsuling makes sense then using library makes the thing only less handy. However with including of Oleg's C# Script it was set that I have to create at least one shared DLL (FS Earth Tiles Internal DLL) in order to make it possible to access FS Earth Tiles common information structures/types and data. Creating one DLL you can as well create two. Once started...you know. So the Main DLL (FS Earth Tiles DLL) capsules FSET for the outer world. With the public Interface bound on FSET's Form (Dialogue Window/GUI) and main control code FSET can be steered and ordered to download Areas by own build applications. You only need to include the FS Earth Tiles DLL into your application to do that.
An example how this is done you can find in the source file FSEarthTilesDLLTester.cs

There is also an Internal Interface bound on FSET's Form. This is used to give back status information to the form that will be displayed in the status bar from threads others than the main thread.

Like you can see the FS Earth Masks Applictaion has got the same structure.

FS Earth Tiles Threads and Engines


Main Thread (timer triggered)
Tile Request Queues
Engines (Threads)
FSET (Form)
E1
E2
E3
E4
s0
s1
s2
s3
Internet
s random
www-button
Web Browser Thread
s = earth service server (variations)
After download area processing Thread (resampling etc)

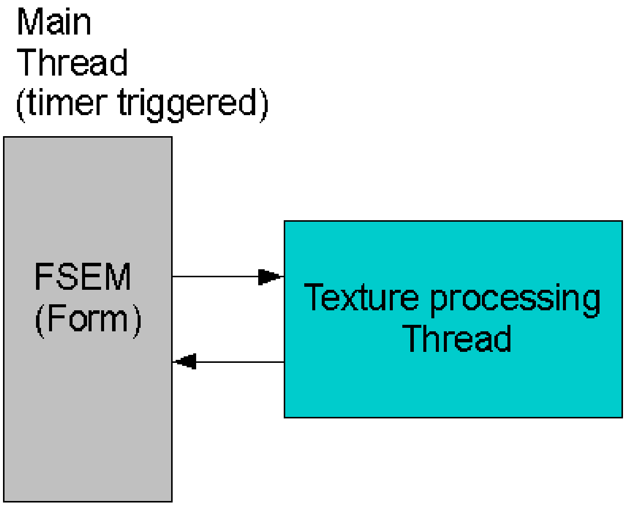
Next to the main thread (Form) with an set-up timer (heartbeat) that calls a timer event handling method every 25ms FSET contains 4 internal self running tile request and getter engines. There are usually waiting idle (AutoResetEvent) till FSET puts them some Tile requests into it's FIFO queues.
Some Earth Services have more than one server/URL (mostly 4) with the same Tile coverage.
In FSEarthTiles.ini you can specify this with the parameter “ServerVariations”. If you specify 4 variations each engine becomes assigned to one and is serving one of the server.
FSET is polling the engines for the results. On every timer event 25ms it checks the engines if they have a Tile ready.

If the user decides to use the Web Browser FSET stops filling the Engine queues and leads every Tile request over the Web Browser. The Web Browser has no queue and is equal one engine in capacity. This means the Tile flow rate will usually drop a factor 4 if you use the Web Browser.

Once all Tiles for one area are downloaded and in the memory FSET starts and after download thread that does the after processing including starting FS Earth Masks and the FS scenery compiler (resample.exe). This way the main thread is free from heavy calculations that would block/freeze the Window on the desktop by non reacting on user events any more.

For developers:
Note that C# Express Studio 2005 seems to have trouble with debugging threads proper. Therefore for debugging it is helpful to run this after work in the main thread. To do this you can set the mDebugMode variable to true. FSET however will not continue after the after work has been done because in the timer event method it is waiting for the thread end that never become started.

FS Earth Masks Threads



Again the thread design structure for FS Earth Masks is equal to the one of FSET.
There is a Main thread (Window GUI) with a timer sending an event every 25ms and an (after-)work thread doing all the time intense calculation to ensure the FSEM window is not frozen during that time.
There is also a mDebugMode variable you can set top true for debugging.

14. FSET parameters (File: FSEarthTiles.ini)

The FSEarthTiles.ini file contains 3 sections or tabs.

[Service*] where the * is a number of **0** till **9**
[ProxyList]
[FSEarthTiles]

The **[Service*]** tag is for the connection configuration to earth service,
The service configuration is the one thing that you **have** to complete before FSEarthTiles really works because that configurations are on legal reasons left in a template form only.
Please read the header of that file and do research on the internet to figure out earth services you can connect and working configurations. I want to remind you here again, that Tiles of earth services are under copyright by the services.

[Service1]
The Service Section or Tag. You can create up to 9 services.

ServiceCodeing   **= qrts**
The Service Tile address codeing. If you write 4 letters or numbers than this signs to use the quadrant method where every quadrant is specified by one of the letters or numbers you put here.
If you write **xyz** them the 3D coordinate addressing (x=, y=, z=) will be used. Note that this 3D coordinate addressing is very service depend and it might be required to write an own codeing in the C# scripts. For this just invent a unique Label different to xyz and everything but not 4 letter or numbers long.
The Address will be inserted in the URL for the place holder %s in the next parameter.

ServiceUrl   **= http://EarthServiceServer1.example.com/tiles/%s.jpeg**
The URL (Web address) template for the Tile access of an earth service. (Here it is just an example)
The **%s** becomes replaced with the coded address of the Tile.

Referer   **= http://EarthServiceBaseServer.example.com**
Some Web Pages on the internet want to know from what web page you called them. If you don't tell them they block their content. I found no earth services where this information is required. And so this is by default marked as comment only with a **#**

UserAgent   **= ABrowserOrApplicationName**
Same as for the Referer some web pages would like to know with what Browser or Application you access. You may or may not tell them just as you like. It is usually not a required information and therefore marked as comment only with a **#**

ServerVariations **= ServiceServer0,ServiceServer1,ServiceServer2, ServiceServer3**
Whenever an earth service offers more than one server as Tile source which all have the same Tiles accessible you should make use of that and specify this server here. You can add up to 4 server.
This is a replace for a part of the URL in the ServiceUrl. FSET checks if it can find any of the 4 names, here ServiceServer0, ServiceServer1, ServiceServer2 and, ServiceServer3 in the ServiceURL string.
If so it will create 4 URL's with the variations and assign 1 of the 4 URL to each of its internal Tile gathering Engines. If every of the 4 Engines accesses an own Server it is speed optimal for you and load optimal for the earth services also. That's why you should do this whenever this possibility for an earth service is given.

[ProxyList]
You can give a list of proxies. On the FSET GUI you can click through it with the “next con” button.
My original idea was to go through the list automatic. But I didn't implement an automatic.
Note that you can connect to whole proxy net's with just one address. That's why I don't saw an automatic really required and spare the time.

A proxy address looks like that: 127.0.0.1:8118
whereas 127.0.0.1 is the IP of the proxy and 8118 the port of the proxy..

Use the word **direct** to mark a direct connection without using a proxy. With the word direct you have the same behaviour as you have with no Proxy list. → a simple direct connection.

[FSEarthTiles]
Every other initialisation or configuration parameter of FSEarthTiles has to be below this Tag.

WorkingFolder = D:\FSEarthTiles\work
Here you specify the working folder for a scenery creation. The working folder is only required for FSET to do the work it is not required for the flight simulator once the scenery is complete build and can be deleted. However you might want to keep the one or the other intermediate file so you can do changes and recompile a scenery to a later time. The directory and subdirectory will be created by FSET if nonexistent.
Attention. If you don't have a D: drive you need to replace the D: with a C: because FSET can not create a folder on a non existing drive. All windows computers have C: drive. But most people work with a D: drive for data storage and so do I. It is general recommended not to use the windows boot drive (C:) as data and work drive.

SceneryFolder = D:\FSX\Addon Scenery\FSEarthTiles
The directory where you want the scenery be generated. Best if you check where your flight simulator add on sceneries are installed and set that directory here. The directory and subdirectory will be created by FSET if non existent. A scenery and texture directory in that directory you specify here will also be created.
Remember that when the scenery generation is completely that you still have to add the scenery in the Flight simulator. Check your Flight Simulator manual for how to do this if you don't know how.

DownloadResolution = 1
Initial Download Resolution Level. Check the Table of the FSET window for the Level to resolution relationship. On standard the setting 1 corresponds to 1 meter/pixel. A good resolution.
(Note that FS2004's max scenery texture resolution is 4meter/pixel. You can download a higher resolution but it will be converted in a 4meter/pixel resolution by the FS2004 scenery compiler (resemple.exe)
Allowed values for the DownloadResolution are **-4** till **+8**

StartWithService = Service2
Initial selected earth service.

SelectedSceneryCompiler = FSX
The selected scenery compiler **FSX** or **FS2004**

AreaSnap = Off
The Snap-Grid of the Area. Allowed values are: **Off, LOD13**, **LatLong** and **Tiles**
FSX works with all of them.
For FS2004 you have to select LOD13 !!! else you get a water border around your scenery. That's a FS2004 restriction.
Off means completely freedom. LOD13 is a Fight Simulator FS2004 and older Grid System.
LatLong snaps the Area on whole minutes in Latitude and Longitude. That makes it simpler for you to keep track of an area coverage if you plan to attach more scenery in an region later.

CreateAreaMaskBmp = Yes
If you want to work out water then you need to set this to yes. If it is set on yes then FS Earth Masks Application will be started that creates a water mask bitmap. If you have created an AreaKML.kml file with google earth the information will be used to create the water mask bitmap. If not an empty water mask bitmap will be created. You also need to set this switch to Yes when you want to generate Seasons and Night Bitmaps. This are also generated by FS Earth Masks.

CompileScenery = Yes
That starts the Flight Simulator scenery compiler to generate the scenery. (.bgl)

AutoReferenceMode = 1
For AreaSnap other than Off the Auto Reference Area System can work in 4 different modes.

1 = match the exact Area dimension. Your drawn Area border will be hold at all cost.
2 = allows to violate the border a little ans snap to the over next grid line if this allows a more optimal reference area size.
3 = maximum freedom in increasing the Area size. This forms the best (x/y relation) and largest reference Area and behaves closest to AreaSnap=Off. I recommend to use this mode whenever possible.
4 = the reference area will be the smallest unit in the AreaSnap Grid.

Zoom = 4
Initial Zoom level. Allowed values -4 till +18

AreaDefinitionMode = 1Point
Initial Area definition mode: **1Point** or **2Points**

AreaSizeX = 1
AreaSizeY = 0.6
The Area dimension in nautical miles [nm] for the 1Point area definition mode.

CenterLatitude = 44deg 25min 00sec north
CenterLongitude = 8deg 51min 00sec east
The centre of the Area for the 1Point area definition mode

NorthWestCornerLatitude = 44deg 24min 00sec north
NorthWestCornerLongitude = 8deg 52min 00sec east
SouthEastLatitude = 44deg 26min 00sec north
SouthEastLongitude = 8deg 50min 00sec east
The area corner coordinates for the 2Points area definition mode.

TextureUndistortion = PerfectHighQualityFSPreResampling
You can choose between: **Off**, **Good**, **Perfect**, **PerfectHighQualityFSPreResampling**

The texture undistorted is important for scenery accuracy! You should never set it to off !!!
That a texture distortion is required has to do with the mercator projection the earth services uses. Every Tile we get from the earth services is distorted with the the mercator projection. The whole bitmap of the area we get by assembling the tiles in memory is in Y direction mercator distorted to a high degree. This distortion is a s larger as higher (North-south)the Area is. The effect is unfortunately not marginal and unless you create only very slim Areas (not more than LOD 13 heights) you have to correct it.

The selection **Good** is doing just that, correcting the mercator distortion in Y.

The selection **Perfect** is also correcting the little X error you get by specifying the area Longitude coordinates within a Tile pixel. This is usually the case since you don't know where you cut. With the correction in X and Y you avoid a double pixel between two Areas you download. However this costs a lot more time than just the Good mode.

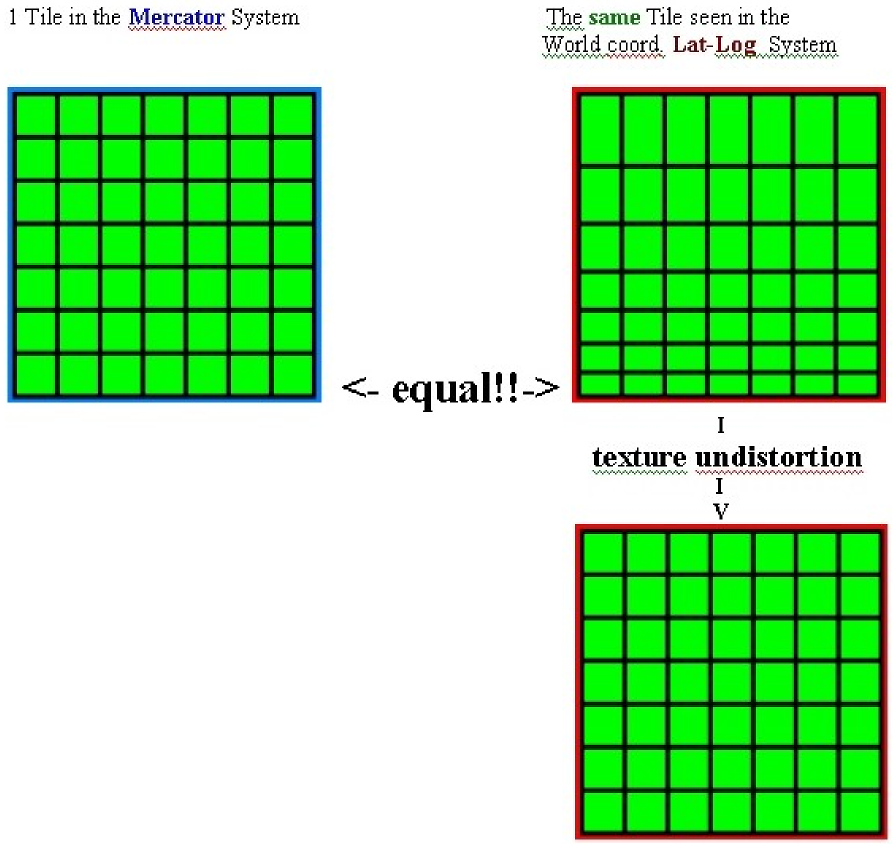
The **PerfectHighQualityFSPreResampling** does equal to the Perfect mode correct the X and Y distortion of a texture. But it does more than that. It resamples (The pixel count/Bitmap size changes) the whole texture into Flight Simulators native scenery resolution (The LOD system) and this with a high quality resampling methode. The method or Filter used is Lancos 3. If you have a paint program that offers this methode to resize a picture you will notice that this is one of the best resizeing methodes when it comes to quality. The reason why I integrated a resampler in FSET is because the Flight Simulators own resample distorts the texture quality to a high degree. This is especially the case if you choose Gauss (default). Gauss is an unsharpening filter and makes textures unsharp. Why anyone wants unsharp textures I don't get. Textures become unsharp in FS when displayed more than enough already, it's almost not to bear how much everything is filtered into unsharpness in modern software.

The PerfectHighQualityFSPreResampling mode cost not more time than the Perfect mode.
It cost somewhat more Memory size (The Auto reference Area are a little smaller) but that's a fair price for the quality you get or better keep. The Preresampled texture is feed to the FS resample with the point sampling mode. FS resample has a lot less work to do then and is faster. It just picks the pixel 1:1 from our high quality pre resampled textures.

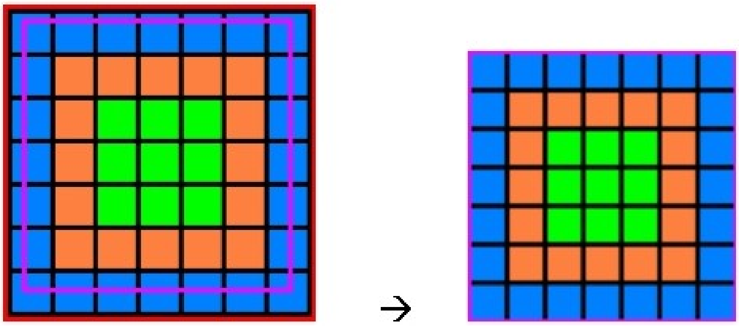
I recommend to have it always set to PerfectHighQualityFSPreResampling mode.

Illustration:

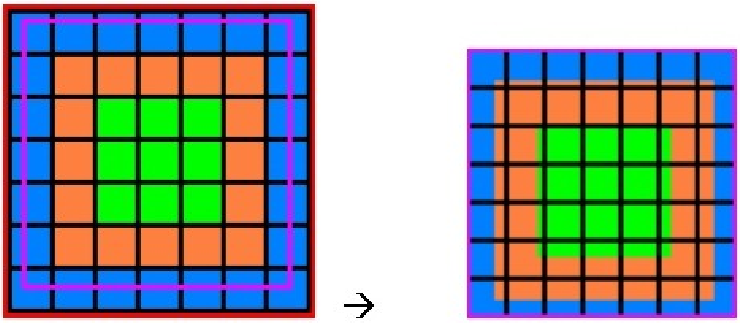
The Flight Simulator expects Textures with a ordinary pixel raster in the World coordinate Lat-Long System.


1 Tile in the **Mercator** System
The **same** Tile seen in the
World coord. **Lat-Log** System
<- equal!!->
I
texture undistortion
I
V

Good (also valid for Off)

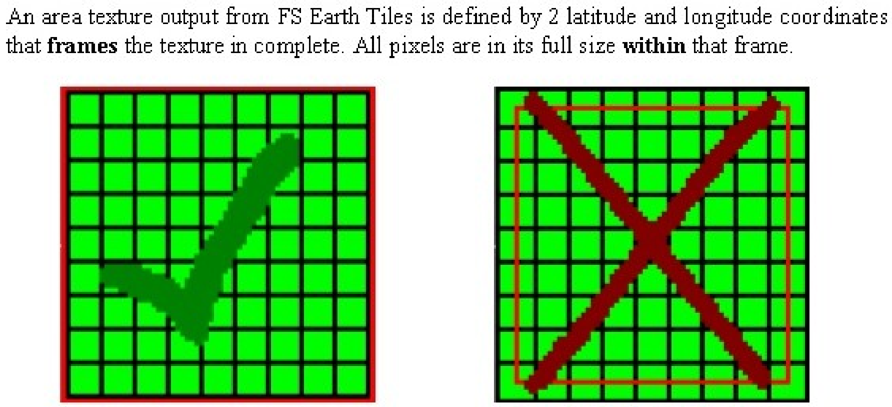


Prefect



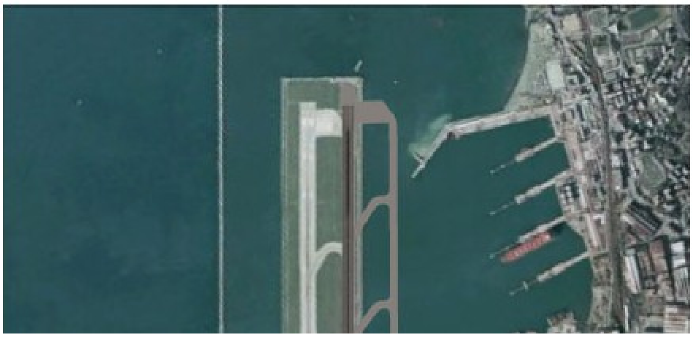
The **PerfectHighQualityFSPreResampling** mode does the same as Perfect. But the pixel count changes.

In this connection you should also know the output of FS Earth Tiles:


An area texture output from FS Earth Tiles is defined by 2 latitude and longitude coordinates that **frames** the texture in complete. All pixels are in its full size **within** that frame.

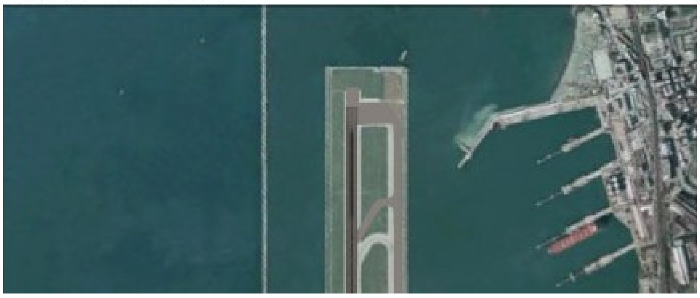
Here is an example what happens with undistortion off. I believe the Area height (north-south) was 40nm

Texture Undistortion Off:



And here it is undistorted

Texture Undistortion Good:

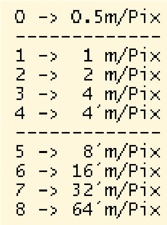


FSPreResampleingAllowShrinkTexture = Yes
Allows to shrink the texture size when resampling instead enlarging. When your Harddisk space is limited then shrinking the texture is more optimal in term of space/quality.
Example: you want a scenery resolution of 1m/pixel.

Depending on the location (latitude) of the Area of interest the earth services due to mercator do not offer FS native 1meter/pixel resolution but 1.5meter/pixel or 0.75meter. Now we can download less Tiles choosing the 1.5meter/pixel and enlarge the texture to 1meter/pixel resolution or download more Tiles with 0.75meter/pixel resolution and shrink it to 1meter/pixel for the same area.
Of course the shrinking leaves you with the better quality because the source had more pixel better resolution. The disadvantage is you have to download more Tiles and sometimesthe service simple do not offer the higher resolution that you can do the luxury shrinking.

FSPreResampleingShrinkTextureThresholdResolutionLevel = 3

This is best explained with the table you see on FSEarthTiles.
A threshold value of 3 means everything up to Download Resolution level 3 becomes shrink. Of course only when FSPreResampleingAllowShringTexture is set to Yes.



```
0 -> 0.5m/Pix
-------------
1 ->   1 m/Pix
2 ->   2 m/Pix
3 ->   4 m/Pix
4 ->   4´m/Pix
-------------
5 ->   8´m/Pix
6 ->  16´m/Pix
7 ->  32´m/Pix
8 ->  64´m/Pix
```

In the Table you see that Level 3 becomes shrink (no apostrophe) to a 4 meter/Pixel scenery.
Level 4 is above the Threshold value and becomes enlarged (with apostrophe) to a 4 meter/Pixel scenery.

Brightness = -6.0
Contrast = +12.0
Color brightness and contrast correction in percent.
You should set the Brightness value to equal minus a half of the Contrast value.
In cities it is recommended to have a high contrast so that the shadows between the houses give a 3D look. If the shadows aren't black enough the 3D effect is not happening.

DownloadFailTimeOut = 1800
In seconds.
When there were no successful Tile access anymore within this time. (1800 sec) then FSEarthTiles will abort the process, give up the download and does not complete a scenery.
Note that when you work with Proxy Nets or resting your internet connection (with a script) you need to set a large enough time.

MaxDownloadSpeed = 20
Tiles/sec
That is the speed limit with which FSEarthTiles is allowed to download the Tiles. You can reduce the speed but not increase it. For fairness reason and to reduce the tension for the services I put in a hard coded max limit of 20tile/sec in FSET. That is fast enough. Really. The after work costs much more time. You know you download with max speed if the tiles are flowing and the Engine queues stay empty.

MaxMemoryUsageFactor = 1.0
This is an important parameter. On startup FS Earth Tiles detects how much memory is available and adjusts the auto reference area size accordingly. It has to take care that all processes in the area processing chain (includeing FSEarthMasks) are able to process the large area bitmaps. You can reduce the Memory usage and the Auto Reference Area size by making this factor smaller.
Be aware that if you start other application next to FSEarthTiles that you steel memory away from your system and it might cause FSET area processing to stop with an out of memory error.

WebBrowserRecoverTimer = 15
in seconds
Whenever a Tile request lead to the Webbrowser is redirected to another address or the user enters manual addresses during the process the WebBrowser engine enters a blocking by waiting on the result that will never come. After waiting this time (15sec) the Tile request will be cancelled and and new request will be send to the Browser.

WebBrowserNoTileFoundKeyWords = Not Found,HTTP Error 404,Bad Request

There is no real error handling possible with the WebBrowser because there is no proper access feedback. This makes it required to analyse the WebPage context in order to decide if an access was successful or not. Whenever a key word in the above list is detected on the page that appears on a Tile request the Tile is handled as “No Tile Found.”
Note that there are Windows System internal Error pages that appear and error pages of the services. So you need to select the key words well. Internal pages often mean that your internet connection simple broke and then you don't want “No Tile Found” for this event but a retry. So only add key words when you are sure that is a page that appears from a service that marks a “No Tile Found” and the word does not appear on your systems internal pages. In doubt don't add it.
All this trouble/mess mess you don't have with the FSET internal engines.
That is really only the disadvantage of using a Web Browser for downloading.

UseCSharpScripts = No
When you want to use C# Scripts then set this to yes. If this is set on Yes the Scripts will be read in and compiled and used at runtime each time you start FSEarthTiles.

KeepAreaInfFile	**= Yes**
KeepAreaMaskInfFile	**= Yes**
KeepAreaMaskSeasonInfFile	**= Yes**
KeepAreaEarthInfoFile	**= Yes**
KeepSourceBitmap	**= Yes**
KeepSummerBitmap	**= Yes**
KeepMaskBitmap	**= Yes**
KeepSeasonsBitmaps	**= Yes**

This are files that will be generated in the work folder to compile and create the scenery.
If you have less Harddisk space or you are sure you don't every want to edit and recompile/regenerate the scenery manual **you can consider to let FSEarthTiles always delete the files or some of them**. **The HD size usage during the processing will shrink significant if you do so** because FSEarthTiles then will only keep the files required for building one Area at a time in the work folder.

FS2004KeepTGAs = Yes
If this is set to yes FSEarthTiles will store the temporary TGA's for a FS2004 scenery generation in a subfolder of the texture folder of the Scenery. I believe you need the TGA's for editing something you can not without them in FS2004. But don't ask me what. Autogen perhaps?
The scenery generation for FSX does not go the way over creating TGAs. There I know you dont need them every. It's all in the .bgl. File. The Autogen you do with the bgl.

UseInformativeAreaNames = Yes
Instead this Area.bmp
it creates a name with the world coordinates in it like this
Area_Lp2_SnapLatLong_N042470000_N042400000_E010050000_E010170000.bmp
Honest I don't know what happens if you set this to No.. does this still work then!?
It's such an old thing and it should always stay on yes.

ShuffleTilesForDownload = Yes
Well as the name says it requests the Tile in a random order from the services.
Hm, I thought I have left this on No. I am surprised to see it's on yes by default. But it doesn't play any role.

ShuffleAreasForDownload = No
The same for the Areas. If you want to download them in a random order you can.

AutoStartDownload = No
As soon as FSEarthTiles has started it will start the Download if you set this to Yes.

AutoExitApplication = No
As soon as FSEarthTiles has finished the Scenery generation it exits if this is set to Yes.

OpenWebBrowsersOnStart = No
Opens the Web Browser right on FSET's start.

ShowDisplaySelector = No
Aehm.. better don't touch. The selector allows you to switch the display into an old single Tile display mode. It is only of interest if you want to study Tiles. But you can do that with the Web Browser as well now. I don't know if this single Tile display mode is still operating safe.

SuppressPitchBlackPixels = No
Well sometimes it is helpful to exclude a specific Color of a texture. If you set this to Yes there will be no pixel in the area source texture (the one generated by FSET directly) with the Color Red=0, Blue=0,green=0. I thought it could be required for something.

CompileWithAreaMask = Yes
Should stay on Yes.
It decides what .inf file is send to the FS resample. The one with the (Water-)Mask or the one without.
An old parameter.

UseLODLimits = Yes
MinimumDestinationLOD = 13

With this two parameters and settings you advice FSEarthTiles to write the .inf file for the FS resample the way that it creates at least a LOD 13 texture. This way you can avoid that FSX is switching to it's own texture in close plane proximate when your downloaded texture resolution is gross (level 4 and above). FS resample has a setting for this but it may only be inserted when you don't have a high resolution texture or it limits your super cool highresolution to LOD13. Not what you want! Therefore FSET has to steer this and FS resemplers sdk renders once more unusable.

CompressionQuality = 100
The Compression Quality you like FS resampler to compile the scenery with.
Don't be fooled. CompressionQuality 100 is NOT a lossless compression as they state in their sdk. A blatant lying of them. Some people like to set this to 80 to safe HD space. However I suffer from lost texture quality already with the setting 100.

FSXSceneryCompiler = resampleFSXSP2.exe
FS2004SceneryCompiler = resampleFS2004.exe
FS2004SceneryImageTool = imagetoolFS2004.exe
The FS resample tools. Make sure there are all in the FSEarthTiles folder and not one missing.

FSEarthMasksTool = FSEarthMasks.exe
FSEarthTiles add on Application for generating Water Masks and seasons.
Make sure it is in the FSEarthTiles folder and not missing.

CreateWaterMaskBitmap = Yes
CreateSummerBitmap = No
CreateNightBitmap = No
CreateSpringBitmap = No
CreateAutumnBitmap = No
CreateWinterBitmap = No
CreateHardWinterBitmap = No
The Bitmaps / Textures FSEarthMasks shall generate for you and include in the scenery.
Note that seasons use a lot HD space and lead to very large .bgls (reduce the reference area size).
Also the FS resample takes a lot time to compile scenery with seasons. The FS2004 resample might run in a memory if you dont reduce the reference area size or the memory usage factor of FSET. I have observed the FS2004 resample.exe trying to load every bitmap of the season in in advanced what I guess can not go well with the limited memory (32Bit appl have 2G limit) when a single Bitmap is already 0.5GByte. Note that this are simple pixel color faked seasons. The earth service do not offer seasons. Therefore don't expect too much. It's just a teaser.

The Summer Bitmap plays a special role. Usually the source texture, the one you downloaded, is used as the summer bitmap. When you set CreateSummerBitmap to yes then a copy of that Bitmap will be done where FSEarthMasks can do color adaption/changes. The original Source Bitmap is never touched.
Whenever you want FSEarthMasks to do color correction be it general or in transitions you have to set CreateSummerBitmap to yes!!!

BlendBorders = No

On request of some user I inserted the option to let automatic create a blend out border (a Mask) of the scenery. The border size is specified in FSEarthMasks.ini. Blending an Area smoothly out or over to the standard scenery looks nice but isn't so handy if you plane to cover more of the region with photo texture later.

UseAreaKMLFile = Yes

Advices FSEarthTiles or more correct FSEarthMasks (parameter will be passed to it) to use of a KML file (AreaKML.kml) that you create with Google Earth and has to be placed in the working folder.

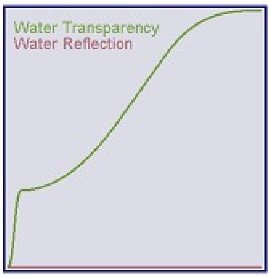
UseScalableVectorGraphicsTool = No
ScalableVectorGraphicsTool = D:\Inkscape\inkscape.exe

As an alternative to Google Earth you can create a .svg file for the water information using a freeware vector tool like inkscape.
(attention: as an exception you have to specify the full path of the tool exe here and not just the exe name.)
Different to google earth were you have to create the water information file in ahead, this tool (inkscape) will be started for every just downloaded area.

15. FSEM parameters (File: FSEarthMasks.ini)

#Helpers

CreateTransitionPlotGraphicBitmap = No
If you set this to Yes FSEM creates a transition function plot bitmap with the file name TransitionPlotGraphicBitmapFile.bmp in the working folder.



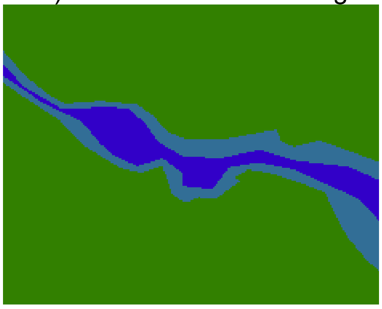
(Coast) (DeepWater)

This helps you in tuning the parameters and creating own transition function.
The transition functions are applied between the two lines you draw. Far left is the Coast- Far right the DeepWater line. The green line shows that the transparency rises very fast at the Coast to 1/3 of the maximum and then smoothly rises to full transparency in an S form till to the DeepWater line.
There is no Water Reflection applied,

Unfortunately there exist only a transparency (blending) and (reflection)channel in FSX for water generation. You don't have a default FS water channel as you had it in FS2004. The default water channel became the reflection channel. But reflection and default water is something completely different. Adding reflection makes your texture extremely bright, next that you only have an on /off choose (1bit) Reflection renders unusable. Therefore I don't use it as default. Instead reflection we make use of the transparency/blending channel and blend over from our texture to FSX's default texture. Here at least you can do this more smooth. It's not just an on/off but has 4 bits. (16 steps), The rest have to be done by dithering. FSEM does that. Using transparency however requires that there is water in the default FS scenery below our scenery in order to have water. And this requires an accurate Coast mesh.
If you you don't have an accurate coast in FSX's default itself or through an add on the way to generate one by yourself is unfortunately very troublesome. Basicly you have to delete the coast vectors in FSX for a whole large region (LOD11) and create new ones where you do not have the datas of the old one. And this you have to do with an SDK that does everything to make it unusable for a normal user.
Why they made it that impossible to generate FSX default water at any place is beyond my horizon of grasp. If you really want to try and adjust the default scenery coast in order to have proper water I recommend to check the "FSX KML" project. I am told with this you can do that more or less.
I bounced my head at the next wall and said to me once again well done microsoft.

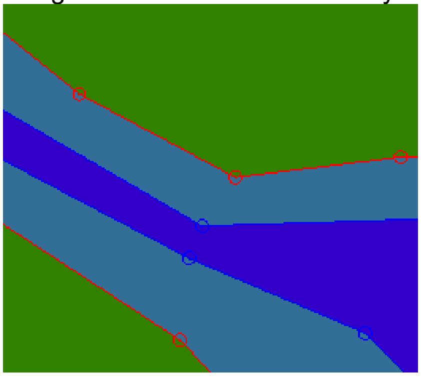
TransitionPlotGraphicSizeFactor = 1.0
You can make the Transition plot bitmap larger or smaller to see more details. Allowed range: [0.5 ... 10.0]

CreateCommandGraphicBitmap = No
When you set this to Yes FSEM creates a bitmap/mask where you can see the land(green), water(dark blue) and water transition region(bright blue) next to other transitions. Here is how it looks for a river.



Of course you only get this when you have created an AreaKML.kml with the water lines.

ShowVectorsInCommandGraphicBitmap = No
When you set this to Yes then also the vectors (water lines) will be drawn including the joints points. This is useful when anything went wrong in the water/land detection. For example if there is a hole, a line missing or a joint open, the water can flood (there is a flood fill algorithm work) the land.
The .NET isn't that great when it comes to drawing lines connected to each other without holes. After fixing holes after holes I was very close to program even the line drawing function my self.



#General

UseCSharpScripts = No
If you want to work with C# Scripts, for example creating your own season routines, you need to set this to Yes that FSEM reads them in, compiles them and uses them at runtime.
It only reads in and compiles the files that are really used. If you don't create that season it wont read in that C# Script for it.

UseReversePoolPolygonOrderForKMLFile = Yes
Some Tools like Google Earth store the Polygon order backward in the file (the last first). So a reversing of the order is required to place the right one (the last drawn) on top of the others.

BlendBorderDistance = 50.0
If you set BlendBorders = Yes in FSEarthTiles.ini then this is the distance in Pixel or the wide used for blending the borders.

CreateWaterForFS2004 = Yes
This affects FS2004 only. If this is set to yes (it should always stay on yes) then the bitmaps if compiled for FS2004 will be expanded with a 4. Channel (32 bit) that holds the water information for FS2004. Water and transition becomes marked as water there. FS2004 does not know transition (blend/transparency).

MergeWaterForFS2004 = No
If the Bitmaps already contains a water channel, the new water information will be merged in instead overwriting the context if you set this to yes. This is only of use if you manual build the season again with manual water changes directly in the bitmap.

SpareOutWaterForSeasonsGeneration = No
If set to yes then the water in the texture will not be altered through the season routines.

NoSnowInWaterForWinterAndHardWinter = Yes
If set to yes then no snow will be created on the water parts in Winter and HardWinter.

WaterResolutionBitsCount = 1
This is the bit resolution of the reflection channel (water). FSX only knows on/off so this is 1 bit.
If you create a smooth transition for the reflection FSEM applies dithering to approximate the value.
This however looks usually bad. You should create smooth transitions for the reflection channel but better full on/off. You can also set this to 8 (8bit) so no dithering will be done. The water then simple kicks in at a threshold level only known by FSX. (It seems to be1/4 and not ½)

BlendResolutionBitsCount = 4

The blend channel has 4 bits (16 steps) how I figured with trying it out.
That is just enough that dithering to boost it to the full 8 bits(256 steps) works well and still looks good.
Although the dithering is done very precise and correct (check the generated Masks) with FSX you still can see the 16 steps if you look at the water in some specific weather condition an direction. That's a Fsx thing.

```
# ---- General, applied on whole Texture---

GeneralLighteness      = 1.0
GeneralBrightness      = 0.0
GeneralContrast        = 0.0
GeneralColoringRed     = 0
GeneralColoringGreen = 0
GeneralColoringBlue    = 0
```

You can do general color correction with FSEarthMasks. But this will only work if you let generate a Summer Bitmap (**Set CreateSummerBitmap In FSEarthTiles.ini to yes**)

Reason that a Summer Bitmap is required for color changes: The source bitmap has to stay unaltered so the process can be repeated manual later anytime.
A Summer Bitmap cost HD space and is really only used if you want to change the color. Therefore it is set to no as default.

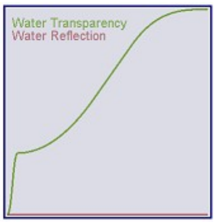
GeneralLighteness is a multiplication factor.
New PixelElementColorValue = ThisFactor *   PixelElementColorValue;

GeneralBrightness and GeneralContrast are the same like in FSEarthTiles.
Unit: percent. Best is to make BrighnessValue = - 0.5 * ContrastValue.

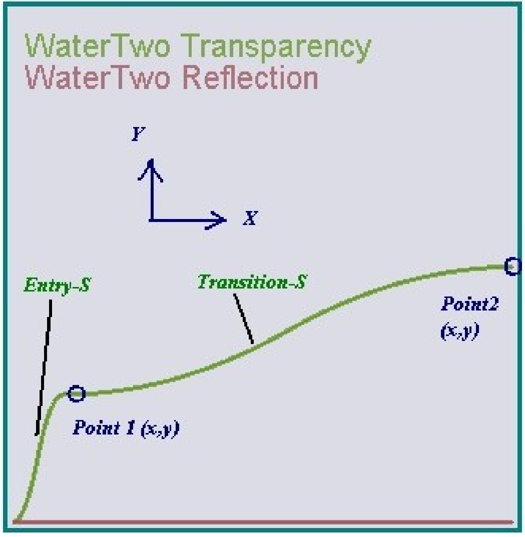
With GeneralColoringRed, GeneralColoringGreen and GeneralColoringBlue
You can simple add a value to the specific Color component. (A color component value goes from 0 to 255)

```
# ---------- Water Transition One (Coast, DeepWater, Water)  ---------------

WaterTransitionTransparencyEntrySFunctionFirstHalfOrder                  = 2
WaterTransitionTransparencyEntrySFunctionSecondHalfOrder                 = 2

WaterTransitionTransparencySFunctionToSFunctionConnectionPoint1x = 0.05
WaterTransitionTransparencySFunctionToSFunctionConnectionPoint1y = 0.3

WaterTransitionTransparencyTransitionSFunctionFirstHalfOrder             = 2
WaterTransitionTransparencyTransitionSFunctionSecondHalfOrder            = 3

WaterTransitionTransparencySFunctionToSFunctionConnectionPoint2x = 1.0
WaterTransitionTransparencySFunctionToSFunctionConnectionPoint2y = 1.0

WaterTransitionTransparencyExitSFunctionFirstHalfOrder                   = 2
WaterTransitionTransparencyExitSFunctionSecondHalfOrder                  = 2

WaterTransitionTransparencyFlipFunction                                        = No
WaterTransitionTransparencyUseEntrySFunctionDistanceLimit                      = Yes
WaterTransitionTransparencyStretchTransitionSAndExitSFunctionToFillAnyGap = Yes
WaterTransitionTransparencyEntrySFunctionDistanceLimit                  = 25
WaterTransitionTransparencyEntrySFunctionLinearSlopeBegin               = 0.5
WaterTransitionTransparencyEntrySFunctionLinearSlopeEnd                 = 0.5
WaterTransitionTransparencyTransitionSFunctionLinearSlopeBegin          = 0.4
WaterTransitionTransparencyTransitionSFunctionLinearSlopeEnd            = 0.6
WaterTransitionTransparencyExitSFunctionLinearSlopeBegin                = 0.5
WaterTransitionTransparencyExitSFunctionLinearSlopeEnd                  = 0.5
WaterTransitionTransparencySFunctionToSFunctionConnectionPoint1Slope      = 0.0
WaterTransitionTransparencySFunctionToSFunctionConnectionPoint2Slope      = 0.0
```

The above values give a transition that looks like this:



First we need to work out some basics to understand it and we do this with the Water2Transition.
All transitions are build the same way and have the same parameters. So we can pick just one.

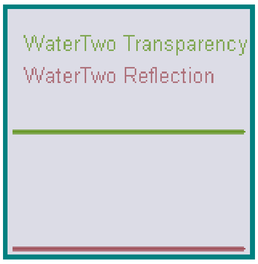


(CoastTwo) (DeepWaterTwo)

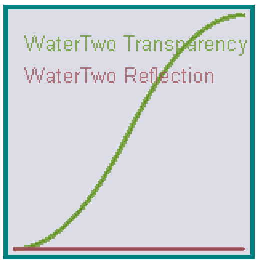
The whole transition function takes place between the two lines CoastTwo and DeepWaterTwo.
We put an X/Y coordinate system on it with x=0.0 where the CoastTwo line is. The DeepWaterTwo line is at x=1.0. The base is y=0.0. (or off) And the top of the graphic is y=1.0 (full/max value/transparency)
Now a transition consists of 3 sections (functions).
An Entry-S section/function.
An Transition-S section/function
An Exist-S section/function
This 3 sections have connection points.
Between the Entry-S and the Transition-S is the connection Point1
Between the Transition-S and the Exit-S is the connection Point2

The location of this connection points you specify with this:
Water2TransitionTransparencySFunctionToSFunctionConnectionPoint1x = 0.1
Water2TransitionTransparencySFunctionToSFunctionConnectionPoint1y = 0.25
Water2TransitionTransparencySFunctionToSFunctionConnectionPoint2x = 1.0
Water2TransitionTransparencySFunctionToSFunctionConnectionPoint2y = 0.5
This are the most important parameter/value. You can to a large degree just adjust this 2 connection points to build the transfer function you like without touching anything else.

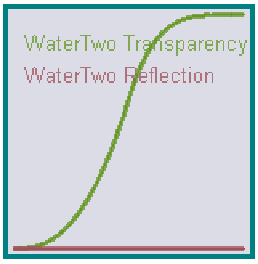
If for example I set Point1 to (0, 0.5) and Point 2 to (1.0, 0.5) I get a constant transition with the value 0.5



Or I can set Point1 to (0,0) and Point2 to (1.0,1.0) and get a transition with just the Transition-S going in an S from from 0.0 to 1.0



Now we can also change the Form of the Transition-S
Here fore we have this parameters
Water2TransitionTransparencyTransitionSFunctionFirstHalfOrder = 2
Water2TransitionTransparencyTransitionSFunctionSecondHalfOrder = 2
This tells FSEarthMasks to use second order (parable) as start and stop of the S-function.
We can make it 2.5 and 4.5 and see what happens:
Water2TransitionTransparencyTransitionSFunctionFirstHalfOrder = 2.5
Water2TransitionTransparencyTransitionSFunctionSecondHalfOrder = 4.5

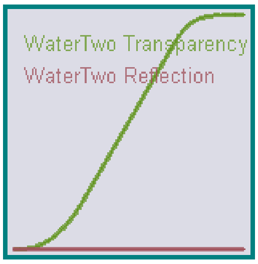


clearly the form changes. 4.5 is a high order.
Now we are not just happy. The whole transition is all bown everywhere. But we want to have a linear part also.

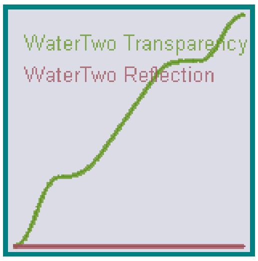
For this we have those parameters:
Water2TransitionTransparencyTransitionSFunctionLinearSlopeBegin = 0.5
Water2TransitionTransparencyTransitionSFunctionLinearSlopeEnd = 0.5
That means between 50% and 50% of the Transition-S function we have a linear part.
That makes it inactive.

We want to have a linear part between 30% to 70% of the Transition-S function:
Water2TransitionTransparencyTransitionSFunctionLinearSlopeBegin = 0.3
Water2TransitionTransparencyTransitionSFunctionLinearSlopeEnd = 0.7

and now we get this:



For the next step we change the connection Points P1 to (0.2,0.3) and P2 to (0.8,0.8) to have all 3 sections (Entry-S,Transition-S,Exist-S)
That looks like this:

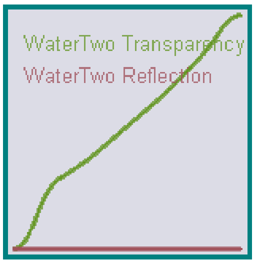


Funny I know. But it's only an example to explain the things.
One thing sticks out here. The plateaus at the connection point P1 and P2. It would be good if we don't have to connect flat but give a slope. For this we have this Slope parameters for every connection point one.

Water2TransitionTransparencySFunctionToSFunctionConnectionPoint1Slope = 0.0
Water2TransitionTransparencySFunctionToSFunctionConnectionPoint2Slope = 0.0

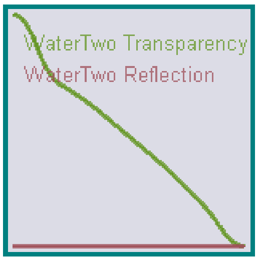
Now the Slope is defined as DeltaY divided through DeltaX.
If you rather like to deal with angle's calculate the slope value with
slope value = tan(SlopeAngle), where tan is the tangence.
For 45degree the slope value is 1.
For 30degree the slope value is tan(30deg) = 0.58

With
Water2TransitionTransparencySFunctionToSFunctionConnectionPoint1Slope = 0.6
Water2TransitionTransparencySFunctionToSFunctionConnectionPoint2Slope = 1.0
we get this:



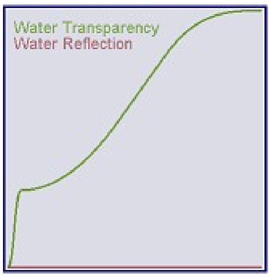
Negative slope will work as well.

Water2TransitionTransparencyFlipFunction = No
Setting this parameter to Yes Flips the function head over:



Water2TransitionTransparencyUseEntrySFunctionDistanceLimit = Yes
Water2TransitionTransparencyStretchTransitionSAndExitSFunctionToFillAnyGap = Yes
Water2TransitionTransparencyEntrySFunctionDistanceLimit = 25

Now this is something special but yet important.
Lets go back to our original Water-Transition function used for the water (one)



(Coast)   (DeepWater)
It consist of 2 sections. The Entry-S and the Transition-S.
The entry S is mainly here to create an Offset. A pre transparency that seems required for FSX. You see it is kept very small. We could right start with an offset and let the Entry-S away but that will make the vectors you draw visible because then you have a hard step from zero transparency to that offset. With the Entry-S we make sure it is a smooth entry. The problem now is that when the DeepWater line is far away the whole transition becomes far stretched and the offset is build far away of the coast. That doesn't looks good any more. Therefore we need to limit how far the Entry-S function my stretch to keep the transparency offset close at the coast independent of how far out the DeepWater line is.
This we do with setting this to
Water2TransitionTransparencyUseEntrySFunctionDistanceLimit   = Yes
and give a maximum stretch distance limit:
Water2TransitionTransparencyEntrySFunctionDistanceLimit   = 25
This Limit is in meter in case of a KML file and in pixel in case of a .svg file!

If we don't stretch the Entry-S function above that limit any more we create a gap between the Entry-S and the Transition-S and Exist-S section. The gab becomes filled with a constant value, the last value of the Entry-S section. That means we get a plateau. This is not that bad but also not the best thing.
To avoid this we set this parameter to yes:
Water2TransitionTransparencyStretchTransitionSAndExitSFunctionToFillAnyGap = Yes
This tells FSEM to stretch the Transition-S and Exist-S more than usual to close the gap. It's like that the point 1 is adapting dynamical in X depending on the Coast to DeepWater distance to create no gap.
This way we get our smooth pre defined transition form back with no gap and can have an offset close to the coast.

I suggest to let this always on Yes and Yes and only adapt the DistanceLimit.

We are done with the transition function parameters-
You see understanding the transition function parameters isn't that complicated. Don't hesitate to play with them and figure out the for you best water transition. The tastes are very different.
I like large transitions, as larger as better, other don't.

#---

You probably figured that there are transition curves you can determine:
Transition, Reflection, Lightness and Coloring

Whereas the first 2 first are channels that are feed in a separate Bitmap (AreaMask.bmp) to FSX (alternative to a second bitmap you could create a Tiff with 5 channels) the two others are no channels but alter the color of the texture itself.

This is only working when you let FSEarthTiles create a Summer Bitmap (CreateSummerBitmap = yes) because the policy is not to alter the source bitmap in order that the process can be repeated manual later.

There are additional parameters that work in connection with the Lightness Transition and the Coloring transition.
WaterTransitionLighteness   = 1.0
WaterTransitionBrightness   = 0.0
WaterTransitionContrast   = 0.0

This is similar to the global values with the different that the strongness of this values are managed by the Lightness Transition function.

```
WaterTransitionColoringRed   = 0
WaterTransitionColoringGreen = 0
WaterTransitionColoringBlue  = 0
```

This is also similar to the global values with the difference that the strongness of this values are managed by the Coloring Transition function.

I believe the less will use this coloring. I self don't use it. However you can. Just don't forget that you need to set the CreateSummerBitmap in FSET's ini to yes that this works.

```
# ---------- WaterPool (WaterPool)  ----------------

WaterPoolTransparency  = 1.0
WaterPoolReflection      = 0.0
WaterPoolLightness       = 1.0
WaterPoolBrightness      = 0.0
WaterPoolContrast         = 0.0
WaterPoolColoringRed   = 0
WaterPoolColoringGreen = 0
WaterPoolColoringBlue   = 0
```

Next to the Transitions you have Pools. A pool is just a closed polygon with specific and constant values. Pool simplifies the drawing of water because you do not have to draw 2 lines. However you will have no smooth transition and see the vectors because of sudden changes.
Pools are hierarchical. The last you draw over covers the one before.

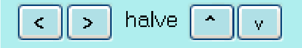
```
# ---------- Seasons ----------------
HardWinterStreetsConditionOn                        =  Yes
HardWinterStreetConditionGreyToleranceValue      =  8
...
```

This are just parameters used for generating fake seasons bitmaps.
What they do is best seen in the C# Seasons Scripts itself.
Mind you that the Seasons is just an add on to this project rather than something really worked out.
However you have a mighty tool in hands through the C# Scripts. It allows you to program your own seasons bitmaps creation. (don't forget to set UseCSharpScripts to yes)

16. Unsorted Information

Area halfing Keys / FSET parallel working

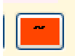
In the 2 Point definition mode you have this keys that allows you to half the Area.



This is helpful if you like to run multiple FSEarthTiles application parallel.
(1 FSEarthTiles application requires 1 core and 2GByte of memory)

When you have set up everything in one FSET you can simple click in the blue field and drag and drop the settings into a second and third and fourth FSET. (quadcore, 8GByte)
Now you can set each FSET's Area to a quadrant of the full Area and download the whole region in a quarter of the time that you would need with one only.
But please consider the impact it has on the earth services when you do massive downloading like that over a longer time.

Panic Button



The panic Button has only effect on the Web Browser (www button).
In case the Web Browser is hanging in an endless attempts to get a specific Tile of the service you can push this button to set the Web Browser into the panic mode that make it pass the blocking.
In the panic mode the Browser accepts everything as Tile regardless what is the root (for example broken connection)
The Panic Mode becomes automatic rested as soon as the download for one area is complete.

Captcha

A Captcha is an offer in a web browser to enter a certain code that is displayed in an input field to identify as human. Only if you do that the service will give free further access to its content.
If an earth service requires a captcha than you can only further access it with the integrated Web Browser because only the Web Browser is dealing with the captcha. FSEarthTiles itself is not dealing with captcha. If the page that is displayed offers you no captcha then you need to check with your cookies. The cookie is the key and you usually get it from the main pages of a services when you visit it.
I don't want to say more here.

The AreaMask Bitmap

The AreaMask Bitmap is an 'invention' from me.
The problem in FSX is that you have to hold 2 additional Channels (transparency, reflection) to the usual 3 color channels (red, green, blue). That makes it 5 Channels. But a Bitmap can only hold 4 Channels in maximum. Now you either have to use the .tiff format instead bitmaps format which is not that far distributed and with every program readable like the bitmap format or you store the 2 channels in a separate bitmap. The last has the advantage that multiple sources (seasons) can share this 2 channel. Within FSEarthMasks the 2 channels transparency (red) and reflection(blue) are created positive. That means 1= on, 0= off. This is stored together with the green channel containing a grey color representation of the Area texture. Before saveing the Bitmap becomes inverted because FSX requires a negative logic 0=on, 1 =off for the reflection and the transparency channel.
This creates the pink look at the end.

The idea was to invert it again for manual water work in the mask. However I've seen that some have started to work with the inverted Bitmap directly. By 'coincidence' (the invert of the red transparence channel) the water looks blue there.

Optimal FSX settings

In order to get the best quality look (sharp) for the ground textures I can recommend the following settings:
(also check: http://iblueyonder.wordpress.com/2007/05/31/fsx-relief-for-the-blurries/)

in FSX.cfg:

```
[Display.Device.xxxx.0]
MipBias = 6
```

and in the NVIDIA driver: Negative LOD Bias set to Clamp

This makes the look sharp.

If you have a fast dual core or even better a quad core I highly recommend to also set the LOD_RADIUS to 7.0 in the FSX.cfg:

```
LOD_RADIUS = 7.0
```

This ensures that it is not only sharp just exact below the plane.

Unfortunately this LOD_RADIUS = 7.0 setting you have to repeat whenever you adjust the settings in FSX. The FSX slider only knows a value of 4.5 in maximum.

Whenever possible create sceneries with a resolution of 1meter/pixel (level 1) or better. That gives a good look. Above 1 meter/pixel the look in FSX becomes fast not satisfying.

The difference in the look between a resolution of 1meter/pixel compared to a 4meter/pixel in FSX is really significant. FSX texture quality depends on high resolution.