Solipsis
Building Guide

1. System configuration

Minimal system configuration

- 1Gb of physical memory
- 250 Mb of free disk space
- Windows® XP or Vista
- DirectX® 9.0 or OpenGL® 2 compatible graphic card with 128Mb memory
- Internet connection

Recommended system configuration

- 2Gb of physical memory
- 250 Mb of free disk space
- Windows® XP or Vista
- DirectX® 9.0 or OpenGL® 2 compatible graphic card with 512Mb memory
- Internet connection

2. Links

lexiExporter plugin   http://www.ogre3d.org/wiki/index.php/LEXIExporter

oFusion exporter plugin   http://www.ofusiontechnologies.com

3. General requirements

A. Softwares/Plugins

- Your favorite 3D modeling software (3DS Max®, Maya®, Blender®)
- Your favorite Ogre exporter plugin (LexiExporter for 3DS Max®, oFusion for 3DS Max®).

  For instance lexiExporter is exporting correctly avatars, oFusion has some troubles with bounding box computed/exported into the .mesh file.
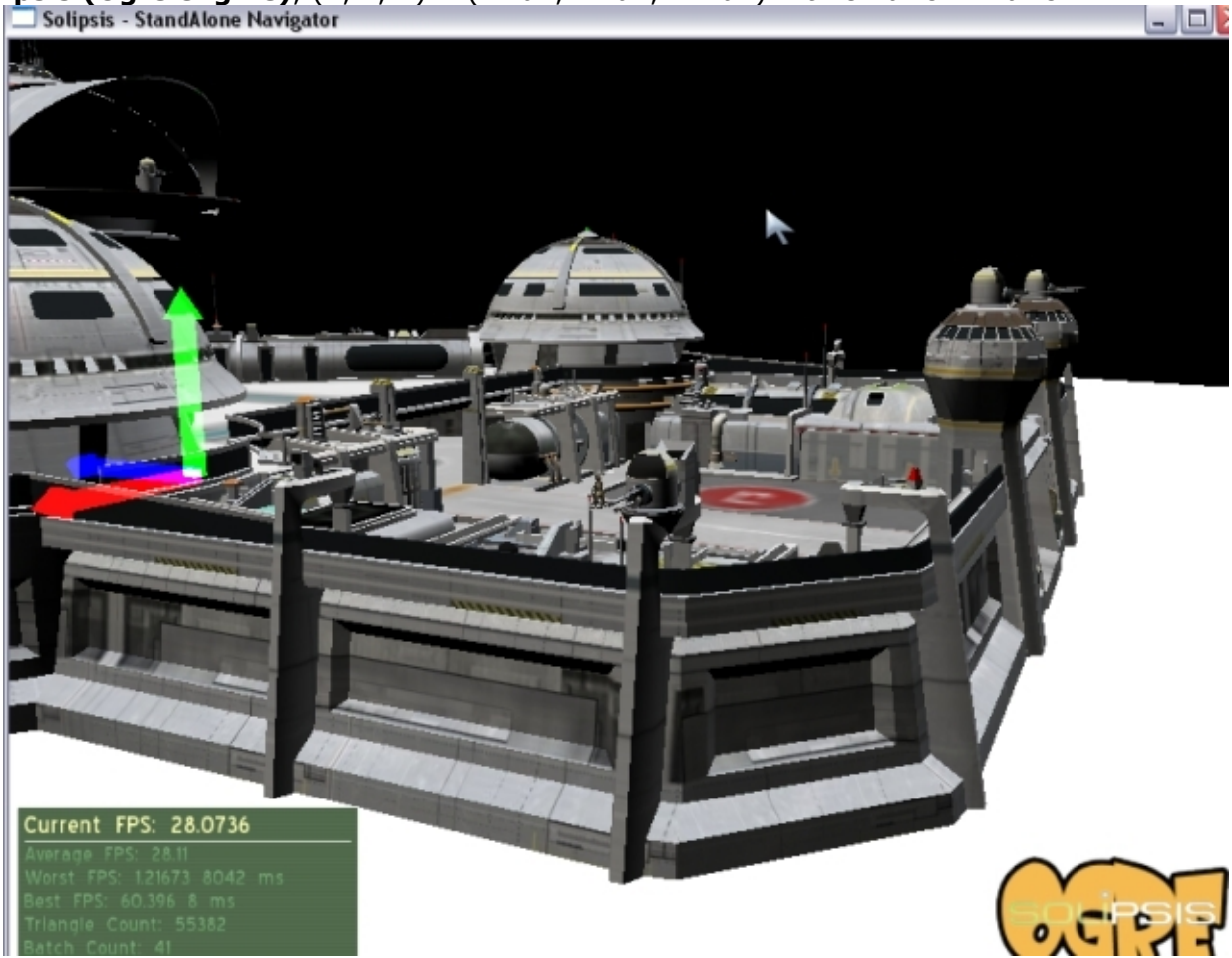
B. Axis/Scale

Models must respect the standard meter unit (scale).
(0, 0, 1) is 1 meter vector on the UP-axis (Z in 3DSMax)

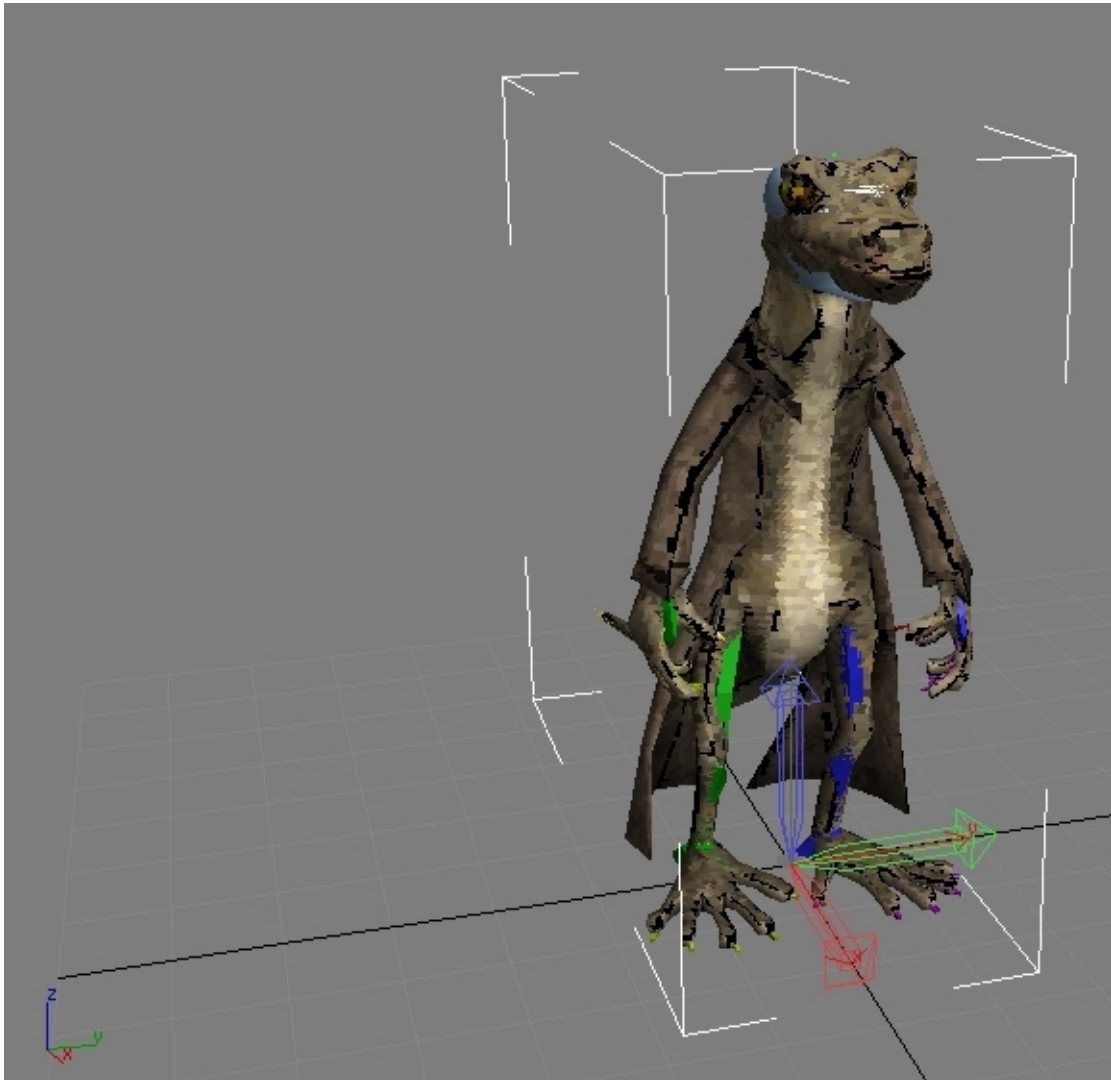In **3DS Max**, (X*max*, Y*max*, Z*max*) with UP axis = Z axis
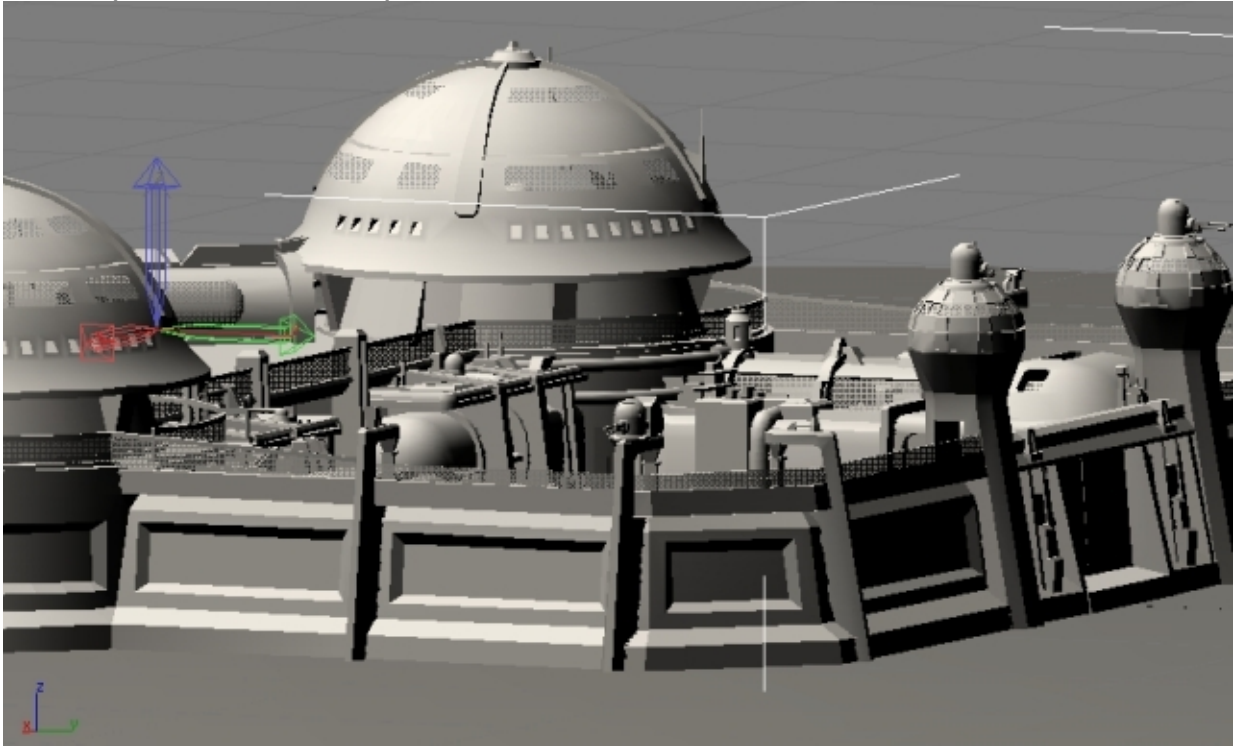
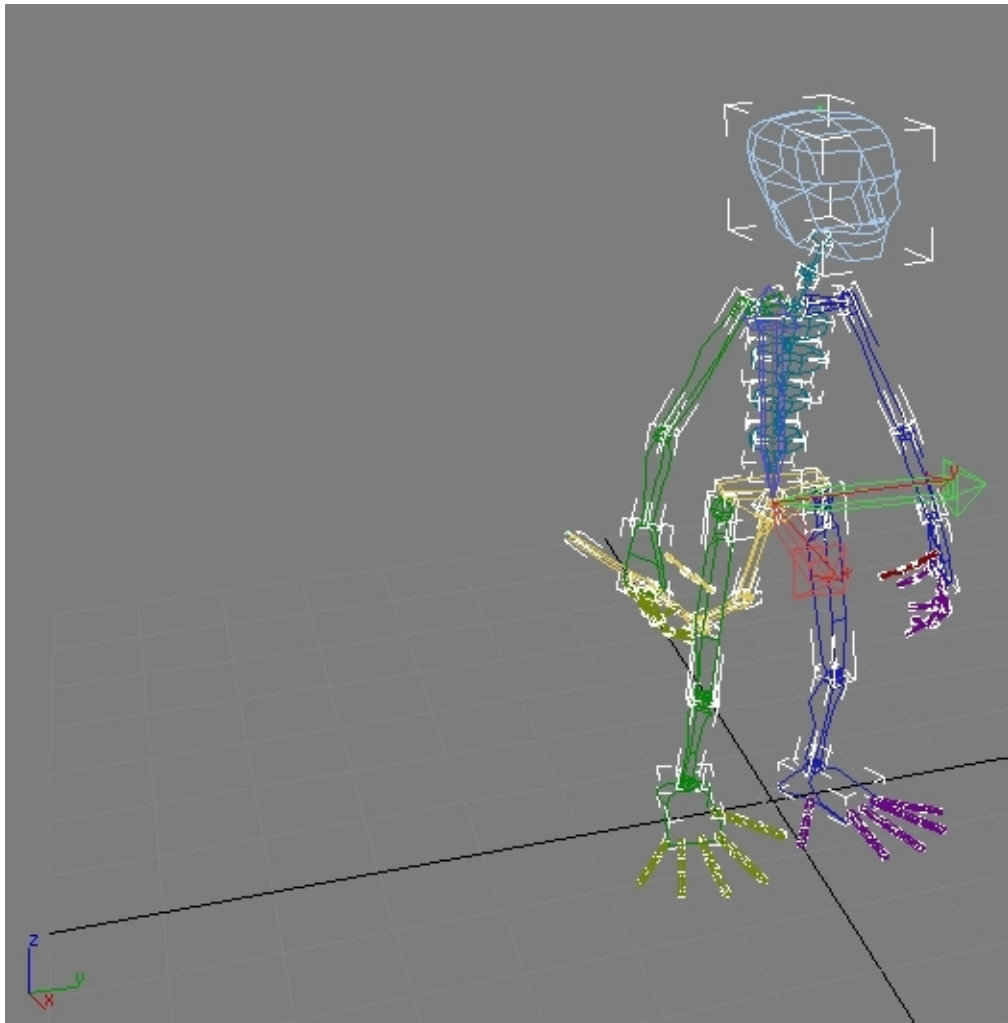In **Solipsis (Ogre engine)**, (X, Y, Z) = (X*max*, Z*max*, -Y*max*) with UP axis = Y axis

4. Avatar building

A. Requirements

- Avatar must respect the standard meter unit (scale), basic avatars are about 1.5 m.
- Avatar is facing the X axis ie. Eye vector=(1, 0, 0)
- Pivot on mesh must be centered and just below feet


- Pivot on skeleton must be centered on the hip joint

- All animations are embedded into the same .max file, each animation will be described into the exporter plugin dialogs. Animations are : Idle, Walk, Run, Fly, Swim

oFusion users note : Take care about initial pose of skinned mesh according to the rigged skin because oFusion take this pose into account on bounding box computation
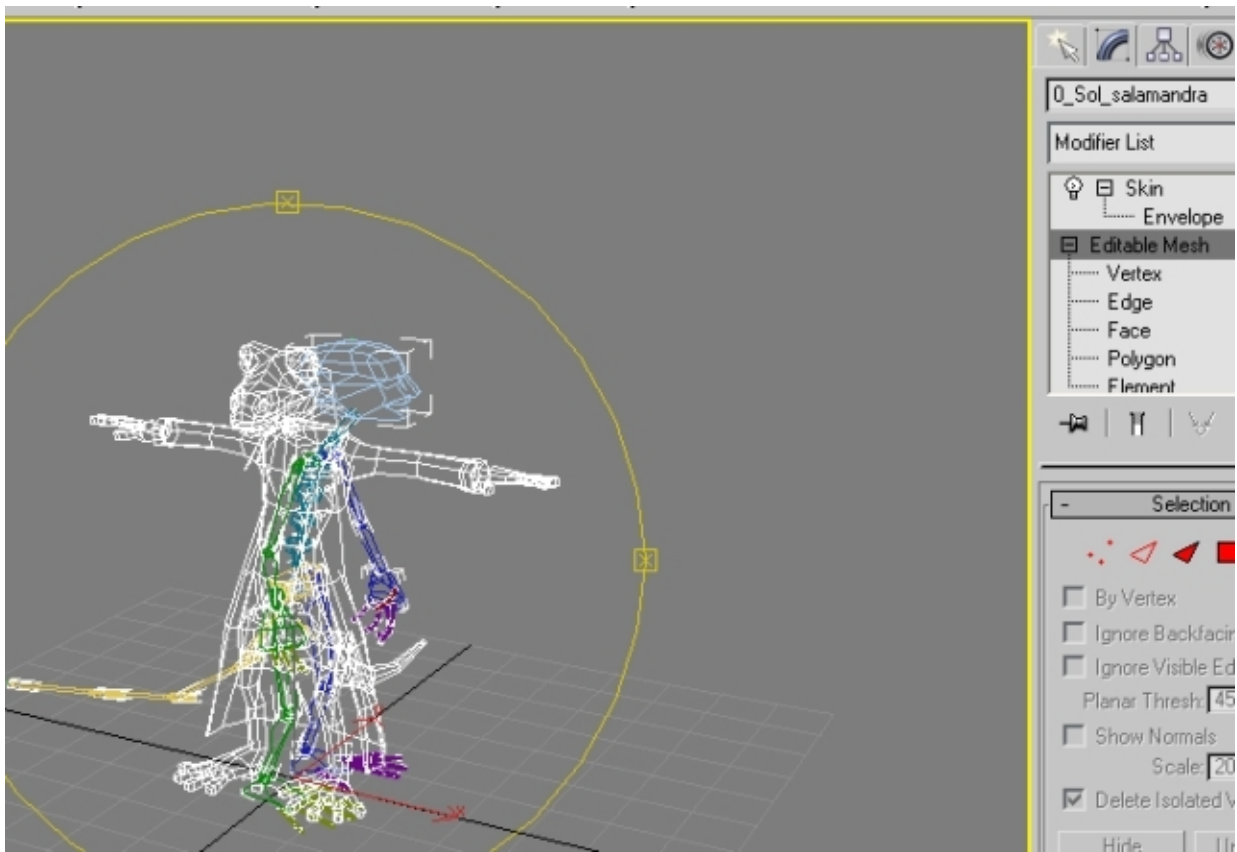
B. Step by step

1. Create your avatar

mesh, materials, textures, skinning, rigging, animations ...

2. Export your avatar (eg. with lexiExporter in 3DS Max) into 1 dedicated directory

You will have all associated files (mesh, skeleton, materials, textures)

3. Build your Solipsis Avatar File (.saf)

The SAF file is a simple ZIP file renamed in SAF and contains all exported files (<myavatar> stands for the filename of your avatar) :
  • <myavatar>_edition.mesh : the OGRE mesh file
  • <myavatar>_mesh.skeleton : the OGRE bones/animations file
  • <myavatar>_edition.material : the OGRE material file
  • .bmp, .jpg, ... : textures files

Example Maxime.saf avatar file contains :
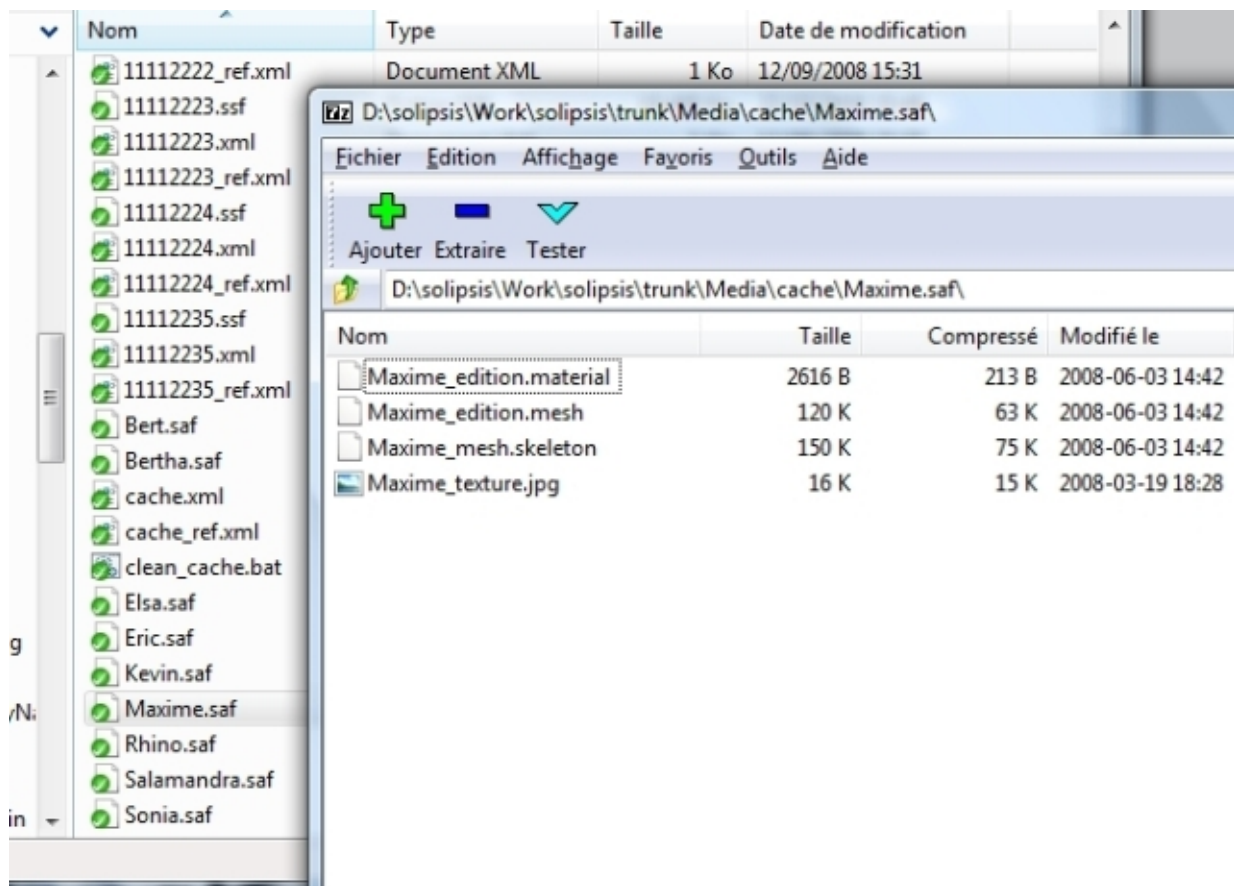
C. Integration

Simply copy your SAF avatar file into your local Solipsis Navigator directory, commonly into C:\Program Files\Solipsis\Media\cache. Restart your navigator and you will see your avatar into the Avatar editor list.

5. Scene *building*

A. Requirements

- Scene must respect the standard meter unit (scale)

B. Step by step

1. Create your scene

meshs, materials, textures ...

2. Create the collision mesh of your scene

floor, walls, stairs, ...
Keep it as simple as possible, avoid too much tesselation
Give it a specific name (COLLISION_mesh for example)
Choose the position where new incoming avatars will be dropped (the entry gate)

3. Export your scene (eg. with oFusion in 3DS Max) into 1 dedicated directory
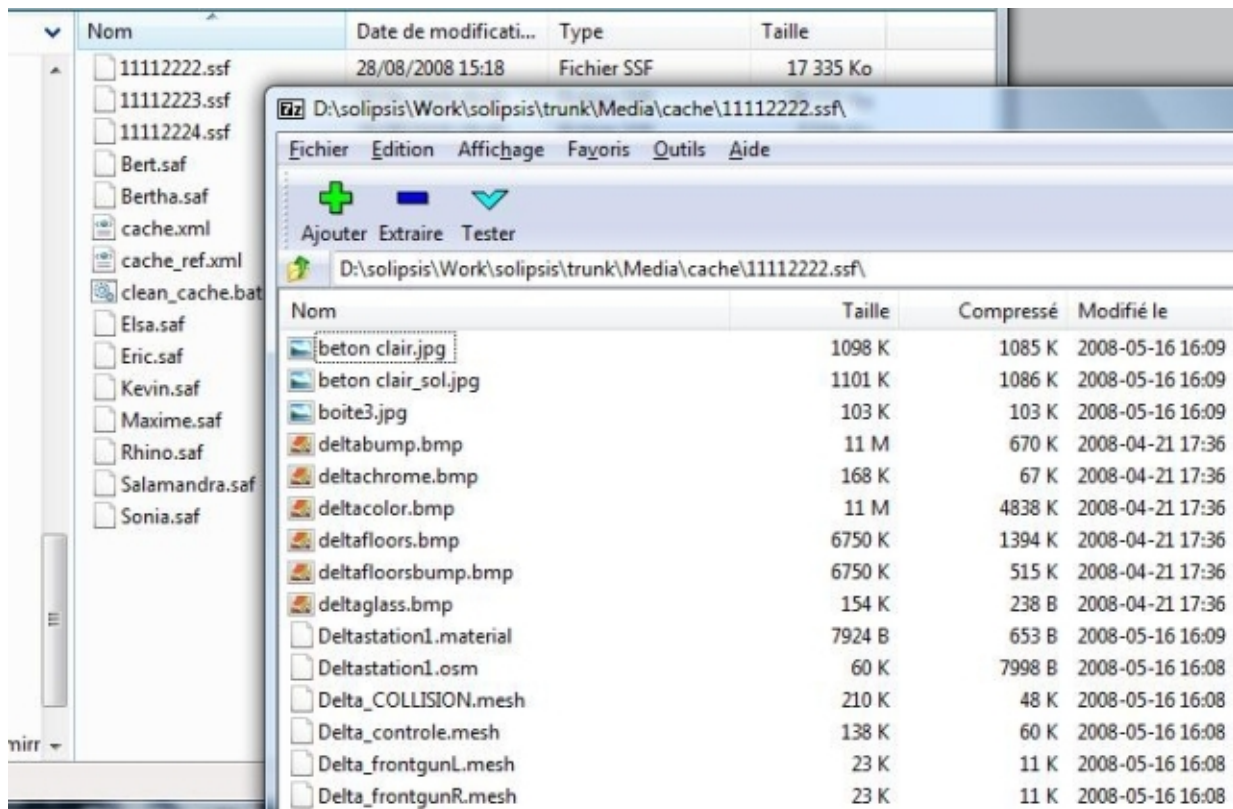
You will have a .OSM file with all associated files (meshs, materials, textures)

4. Build your Solipsis Scene File (.ssf)

The SSF file is a simple ZIP file renamed in SSF and contains all exported files :
- .osm : the oFusion scene file
- .mesh : OGRE mesh files
- .material : the OGRE material file
- .bmp, .jpg, ... : textures files

Example 11112222.ssf scene file contains :

C. Integration

1. Create the XML scene description file

The final step is the creation of the XML file describing your scene.
In this example the XML file 11112222.xml looks like :

```xml
<node>
  <entity uid="11112222" owner="00000001" type="1" name="Deltastation1" version="00000000">
    <position x="18" y="-58" z="133" />
    <orientation x="0" y="0" z="0" w="1" />
    <aabb>
      <min x="0" y="0" z="0" />
      <max x="0" y="0" z="0" />
    </aabb>
    <content>
      <sceneContent>
        <entryGate gravity="true">
          <position x="17" y="-50" z="115" />
        </entryGate>
      </sceneContent>
      <lod level="0">
        <sceneLodContent mainFilename="Deltastation1.osm" collision="Delta_COLLISION" />
        <files>
          <file name="11112222.ssf" version="00000000" />
        </files>
      </lod>
    </content>
  </entity>
</node>
```

11112222 is the unique node identifier (*nodeId*) and 11112222 is the unique entity identifier
(*entityUID*).
Owner is the node identifier of the creator (avatar).
Adjust if necessary **position** and **orientation** of the scene.
Define the **entryGate** attribute : position of new incoming avatars and the flag indicating if the gravity
is activated.
Define the **mainFilename** attribute with the OSM file present in the SSF archive.
Define the **collision** attribute with the name of the collision mesh present in the SSF archive.

Finally define the SSF archive file into the **files** tag.

2. Server administrator adds the SSF file
The server administrator can now integrate the scene into the dedicated RakNet server. He must copy the SSF file and the <nodeId>.xml file into the server Media/cache directory of the server.
To simplify clean up of cache, copy the <nodeId>.xml file into <nodeId>_ref.xml file and update the clean_cache.bat.
Finally launch the RakNet server with argument -s <nodeId> and the dedicated port -p <port>. To define it into the WorldsServer, edit the file WorldsServer\uiworlds.html and insert your new server like :

eg. uiworlds.html for RakNetServer.exe -p 8556

```
...
selectWorlds.innerHTML += "<option value='127.0.0.1:8556'>MyScene</option>";
...
```

6. Solipsis Object File (SOF)

A. Build your Solipsis Object File (.sof)

The SOF file is a simple ZIP file renamed in SOF and contains some exported files (mesh, images, videos, ...) or some URL links for other medias (www).
The SOF file contains also a XML file which describes many things :

- the creation properties (*<properties>*).

- the model and the deformations applies on it (*<model> + <transformation>*).

- the material description like colors, transparency, UN coordinates for the textures applied... and the list of textures which can be used to cover this model (*<material> + <texturelist>*).

- we can also found the world position, orientation and scale for this model, plus a tree of parent / children description for any assembled models (a chair for example) (*<threeD>*).

- finally, there another information like for the animation or for other any extra content (*<animations>, <extras>, <content>*).

Example : `SDxAPujP-5G3KOwPHQCurzv_00010000.sof` object file contains :

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- @version: 040611 -->
<SOLObject>
  <properties>
    <modelerversion Name="1.0.1" />
    <objuid Uid="SDxAPujP-5G3KOwPHQCurzv_00010000" />
    <objname Name="Cube" />
    <objtags Name="" />
    <objdesc Name="" />
    <objowner Name="MrXYZ" />
    <objcreator Name="MrXYZ" />
    <objcreationdate Name="ven.  mar 2009 - 11:56:10" />
    <objgroup Name="" />
    <objrigths mod="0" cop="0" />
    <objparent Uid="NULL" />
  </properties>
  <model>
    <primitive Name="BOX" />
    <transformation>
      <transfo type ="3" valueX="1279" valueY="1279" valueZ="1279" />
    </transformation>
  </model>
  <material>
    <matambient r="0.321569" g="0.592157" b="1" a="1" />
    <matdiff r="0.321569" g="0.592157" b="1" a="1" />
    <matspec r="0" g="0" b="0" a="1" />
    <matshin value="0" />
    <matopac value="1" />
    <matdblside value="2" />
```

```xml
        <texturescroll u="0" v="0" />
        <texturescale u="1" v="1" />
        <texturerotate value="0" />
        <texturelist>
            <texture Name="myPicture.jpg" currenttexture="true">
            </texture>
            <texture Name="WWW_SDxAPujP-5G3KOwPHQCurzv_00010000Texture"
currenttexture="false">
                <textureExtParamsMap>
                    <param Name="frames_per_second" Value="0" />
                    <param Name="height" Value="512" />
                    <param Name="plugin" Value="www" />
                    <param Name="query_flags" Value="1" />
                    <param Name="sound_params" Value="3d 0 10" />
                    <param Name="url" Value="www.google.fr" />
                    <param Name="width" Value="512" />
                </textureExtParamsMap>
            </texture>
            <texture Name="VLCTexture0" currenttexture="false">
                <textureExtParamsMap>
                    <param Name="frames_per_second" Value="25" />
                    <param Name="height" Value="240" />
                    <param Name="mrl" Value="myVideo.avi" />
                    <param Name="plugin" Value="vlc" />
                    <param Name="query_flags" Value="2" />
                    <param Name="remote_mrl" Value="" />
                    <param Name="sound_params" Value="3d 10 20" />
                    <param Name="vlc_params" Value="" />
                    <param Name="width" Value="320" />
                </textureExtParamsMap>
            </texture>
        </texturelist>
    </material>
    <threeD>
        <objposition x="-266.888" y="-5.79461" z="392.821" />
        <objorientation w="-0.837719" x="0" y="0.546102" z="0" />
        <objscale x="1.28" y="1.28" z="1.28" />
        <physics col="0" grav="0" />
    </threeD>
    <animations>
    </animations>
    <extras>
    </extras>
    <content>
    </content>
</SOLObject>
```

About the <texturelist>, only one texture can be used in the same time and this one is notified by the currenttexture="true" tag's property. The different medias and links are basically an URL or the file name for a picture or video file (texture & SWF tabs). You can also diffuse (VLC & VNC tabs) a personal stream like your webcam, your favorite music clip or why not your desktop to show a presentation for example!
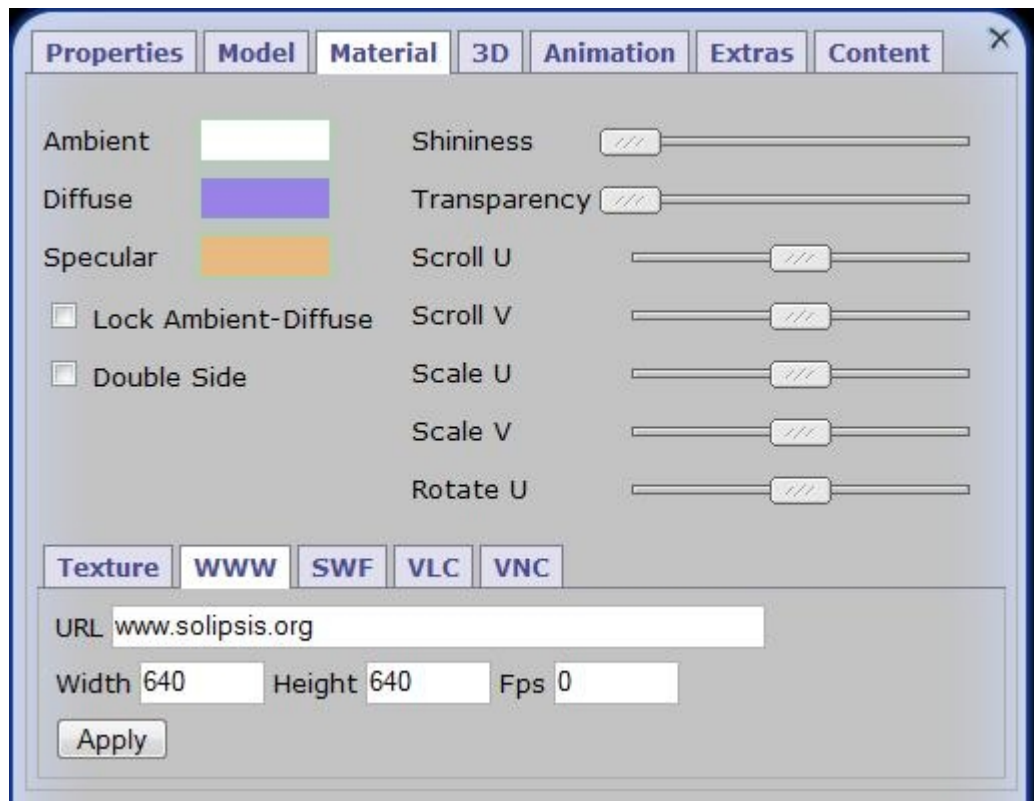
For all of those textures, you have to specify the size (height & width), the update frequency and the audible distance for an embedded video. The update frequency defines the maximum updates per seconds, and if you enter 0, it will be updates only when you'll interact with this media (like moving a scroll bar for example).

For the VLC media and specially with the current example, near the object the sound volume is at 100%, at 15m it's about 50% and after 20m there is no more sound.