

Spintires:MudRunner™ modding guide (Beta)

30 October 2017

Contents

Creating a mod for Spintires:MudRunner™	2
Step 1. Install free Spintires:MudRunner™ Editor in Steam® and create a mod.....	2
Step 2. Create truck '.X' mesh file.....	5
Step 3. Create truck '.XML' mesh file.....	8
Step 4. Create truck file	14
Step 5. Customizing the truck	17
Step 6. Test the truck in the game.....	22
Step 7. Publish mod	25
Importing Spintires-2016 mod into Spintires:MudRunner™	26
Step 1. Create new mod	26
Step 2. Import Spintires-2016 mod content.....	26
Appendix A. Using 3dsMax2012 to create X files	27
Appendix B. General information on textures.....	29
Appendix C. General information on XML inheritance.....	30
Appendix D. Attributes introduced since Spintires-2016	32
Creating a level for Spintires:MudRunner™ (Beta).....	34
Step 1. Create the source level files	34
Step 2. Edit the level	37
Step 3. Test the level in the game	41
Appendix A. Importing Spintires-2016 terrain.....	44

Creating a mod for Spintires:MudRunner™

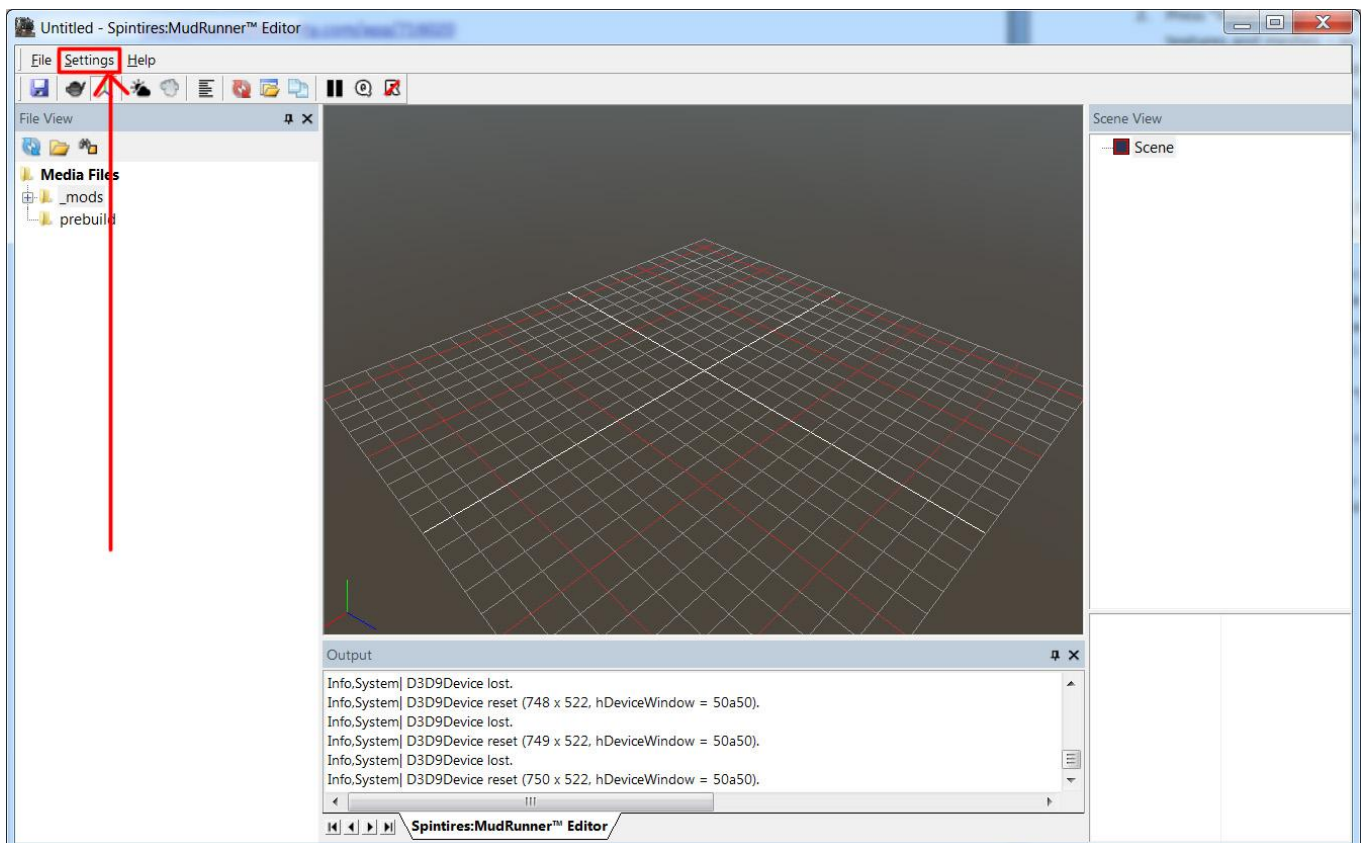
Step 1. Install free Spintires:MudRunner™ Editor in Steam® and create a mod

The Editor is located in the Steam Library>Tools>Spintires: Mudrunner – Editor Beta

For more info on modding, follow the link:

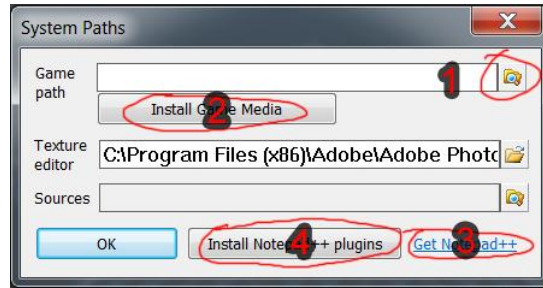
<http://steamcommunity.com/app/675010/workshop/>

Upon first launch of Spintires:MudRunner™ Editor, that's what you should see:



Spintires:MudRunner™ Editor first launch

You need to configure Spintires:MudRunner™ Editor first. Go to Settings>Paths.



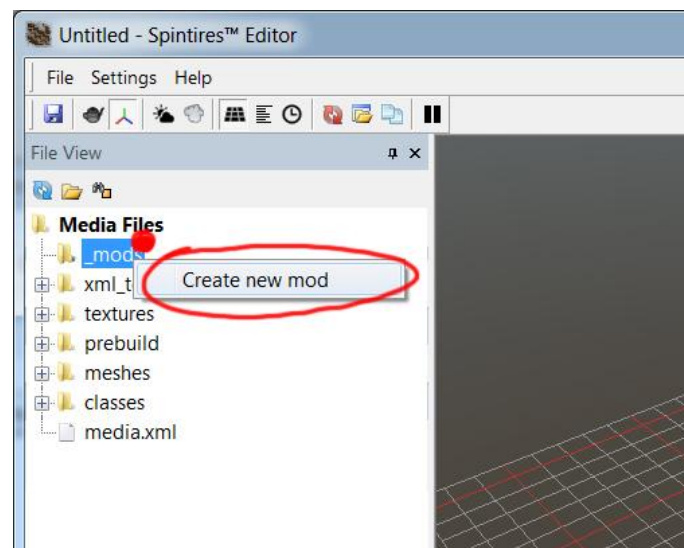
Configuring Spintires:MudRunner™ Editor.

1. You need to have Spintires:MudRunner™ installed! Select path to the installed game (by default “.\Steam\SteamApps\common\Spintires MudRunner”).
2. Press “Install Game Media” – this extracts game files to local Spintires:MudRunner™ Editor folder, so you can use them to build your mods.
3. It is also recommended to download and install Notepad++ for XML editing. Just follow the link “Get Notepad++”.
4. After Notepad++ is installed, press “Install Notepad++ plugins” to install SpinTires plugins and follow the instructions. A set of useful functions would become available in Notepad++, including:
 - Quick open any game file (ALT-SHIFT-O)
 - Goto XML node within file (ALT-SHIFT-D)
 - Quick comment/uncomment XML (ALT-SHIFT-C / ALT-SHIFT-U)
 - XML file format/sort (ALT-SHIFT-F)

After you click “Ok” to close “System Paths” dialog, “File View” pane would get populated by file items.

Tip: make sure Steam® is currently running (you should see Steam® overlay in bottom right corner at startup)

Right click “_mods” subfolder in “File View” and select “Create new mod” in context menu. This will create a new Steam® Workshop mod, which won’t be visible to anyone but you (until you publish it later).



Creating new Steam® Workshop mod

If mod is created successfully, a new subfolder would be created, with name consisting of 8 to 16 alphanumeric lowercase letters. This name is unique Steam® Workshop item ID, which is used to identify your mod.

Basic mod concepts

A mod is a set of files within directory with mod ID name – which is in turn within special directory called “_mods”.

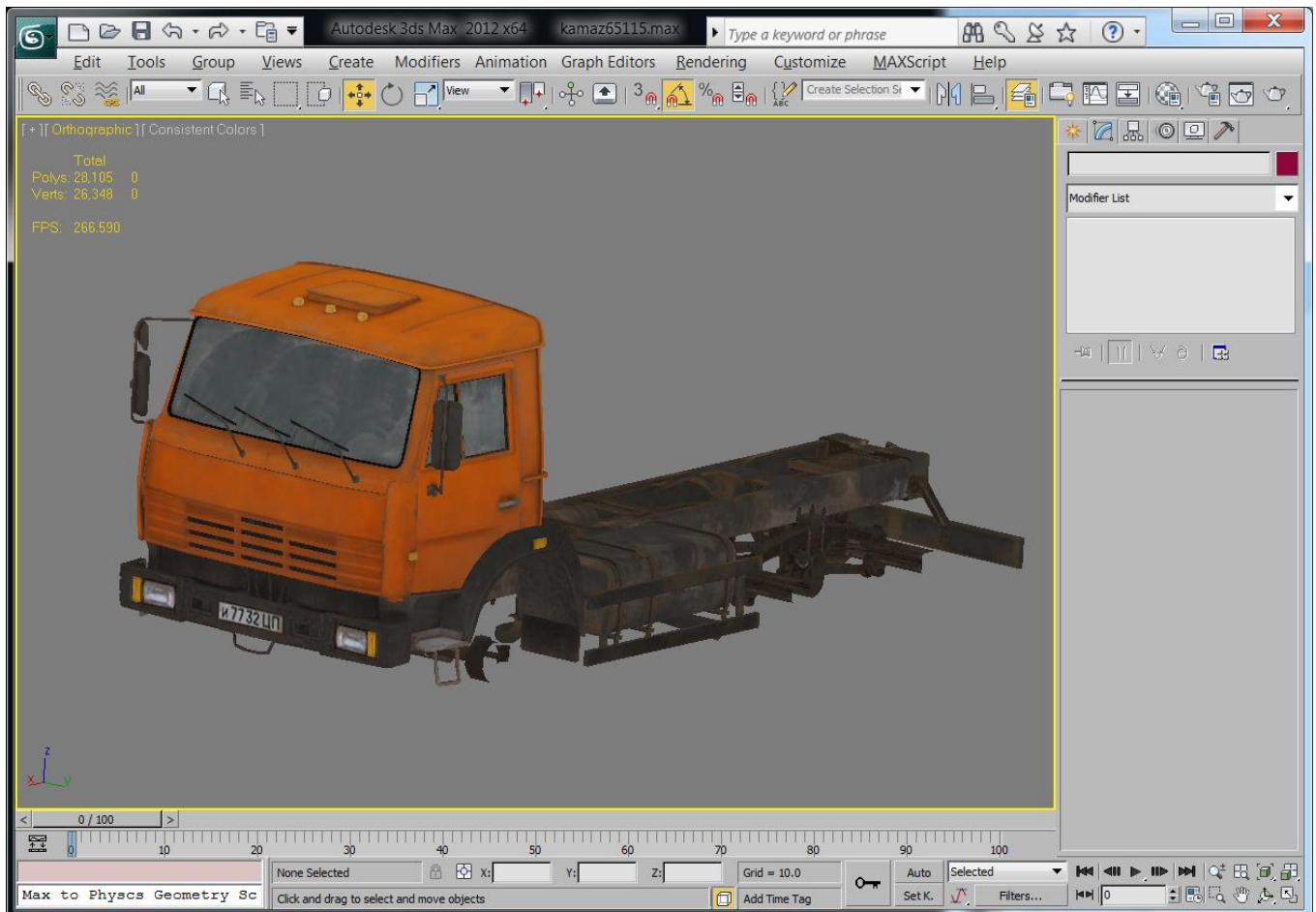
Each time you launch Spintires:MudRunner™ Editor, directories for all mods that you own are automatically created within “_mods”. But you need to populate them with files yourself before publishing. When you choose to publish the mod, those files would be automatically converted to game-friendly formats, renamed and packed – and then sent to Steam® Workshop. So no one but mod author would see original files.

Step 2. Create truck 'X' mesh file

You don't always need to create a new mesh for new truck – sometimes you just want to mod the textures or properties of existing trucks (which is perfectly ok) – but in this example, we are creating new truck from scratch.

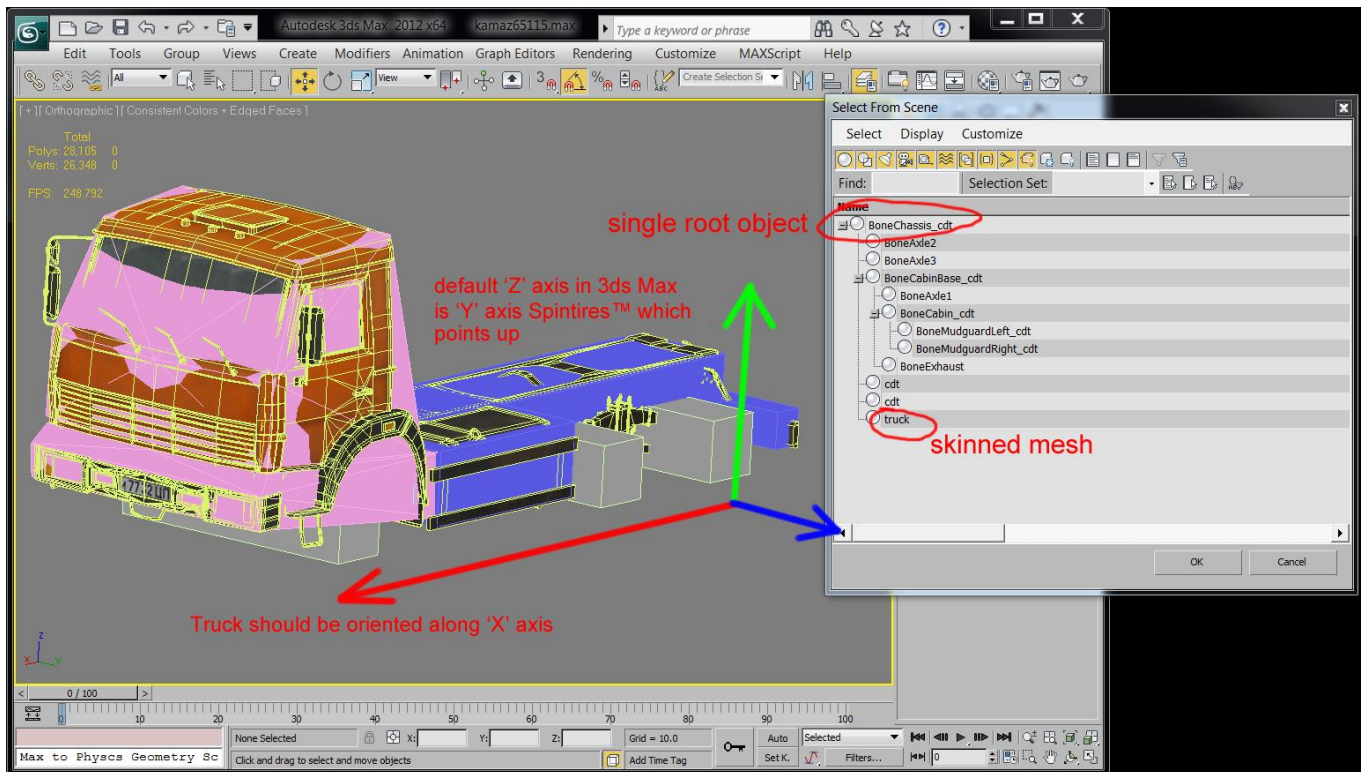
Spintires:MudRunner™ uses “.X” mesh file format – which is an old DirectX file format.

So you have to get a converter for whatever modeling software you are using. In our case its 3ds Max 2012 – and recommended converter is **kW X-port** (See Appendix A)



3ds Max 2012 view of the truck mesh we will be converting to Spintires:MudRunner™

Make sure to setup your mesh properly:



Note that there should be single root object. Also note that 3ds Max Z axis is Y axis in Spintires:MudRunner™ (and Z axis in Spintires:MudRunner™ is Y axis in 3ds Max)

Tip: Make sure all exported objects have identity scale (X:100.0, Y:100.0, Z:100.0).

Make sure all exported object's pivots are default (3ds Max: Hierarchy tab -> Reset Pivot).

Make sure root object and truck mesh itself (in our case, "truck" object) have identity rotation (X: 0.0, Y:0.0; Z:0.0).

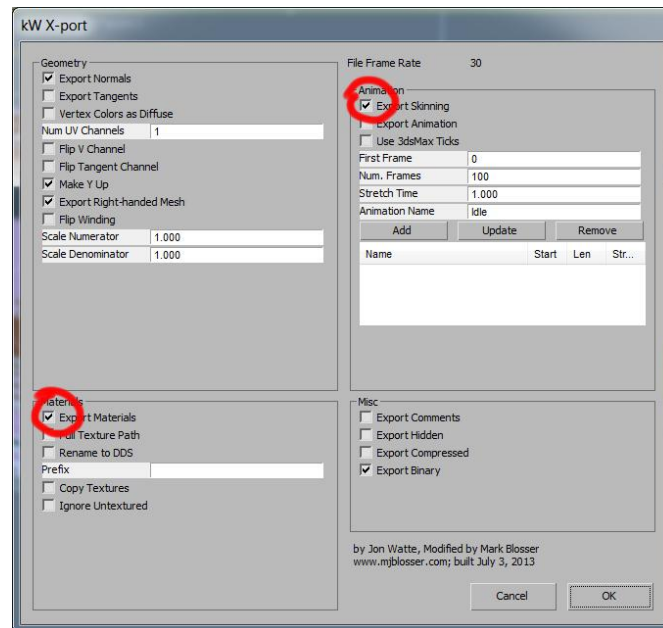
3ds Max: to assure it does, use Utilities tab -> Reset XForm modifier)

Note: truck should be aligned along 'X' axis. Typical default alignment axis is 'Z'. 'X' instead of 'Z' is a legacy issue in Spintires:MudRunner™

At least one truck object (subobject) should be physical. A physical object must have collision model. Spintires:MudRunner™ automatically generates collision model for objects with "_cdt" name postfix. So those objects won't be visible in game, but they can have physical bodies attached to. At the same time, if an object has child objects which are called "cdt" – it uses their geometry as its own collision model. Objects called "cdt" are not accessible in game on their own.

After truck mesh setup is complete, export .X file to “Media_mods\[mod name]\meshes\trucks”

3ds Max: Select all exported objects, click “Export -> Export Selected” and pick “kW X-port (*.X)” in “save as type” combobox.

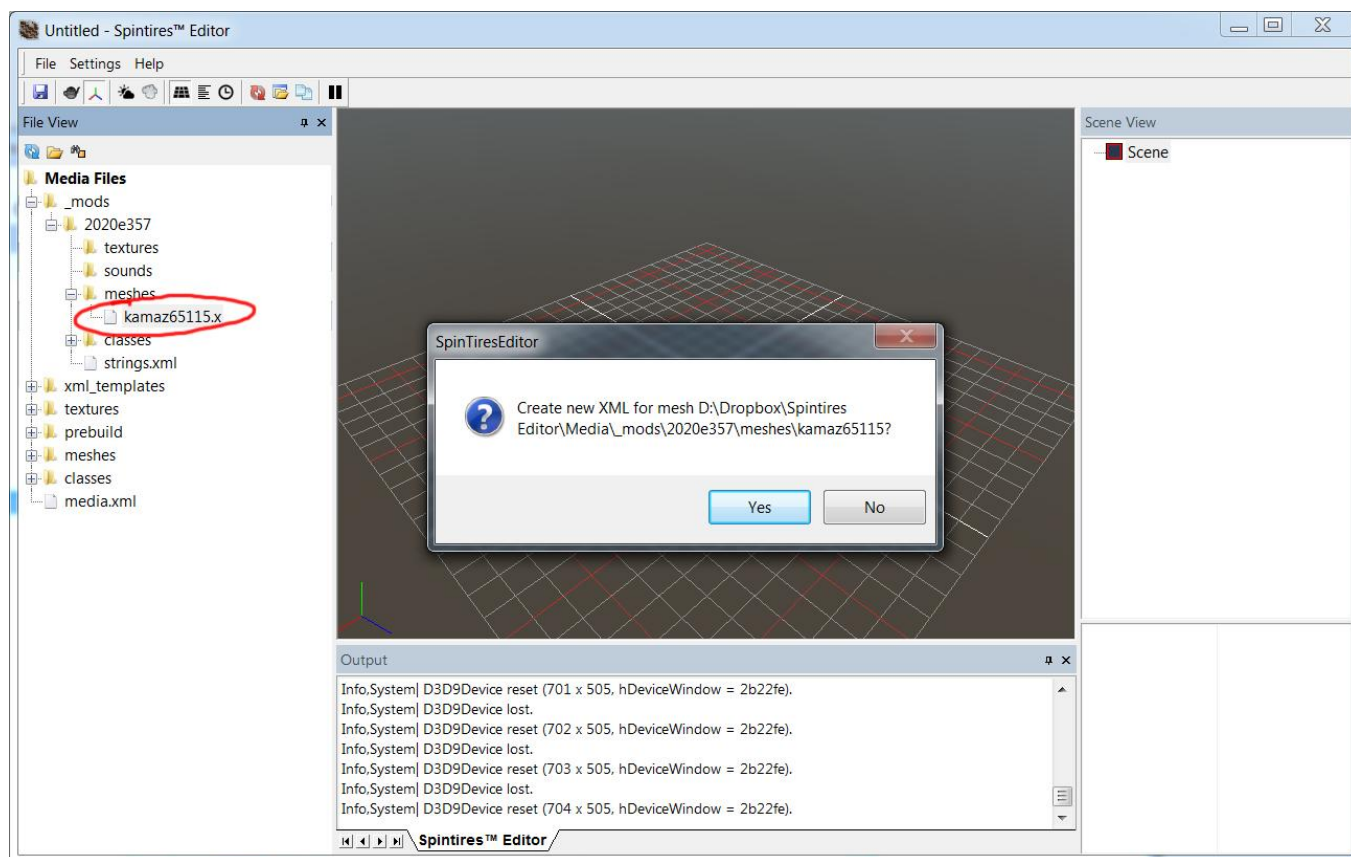


kW X-port exporter settings for 3ds Max 2012

Step 3. Create truck 'XML' mesh file

Each '.X' file in Spintires:MudRunner™ has paired '.XML' mesh file – which describes textures and game-specific mesh properties.

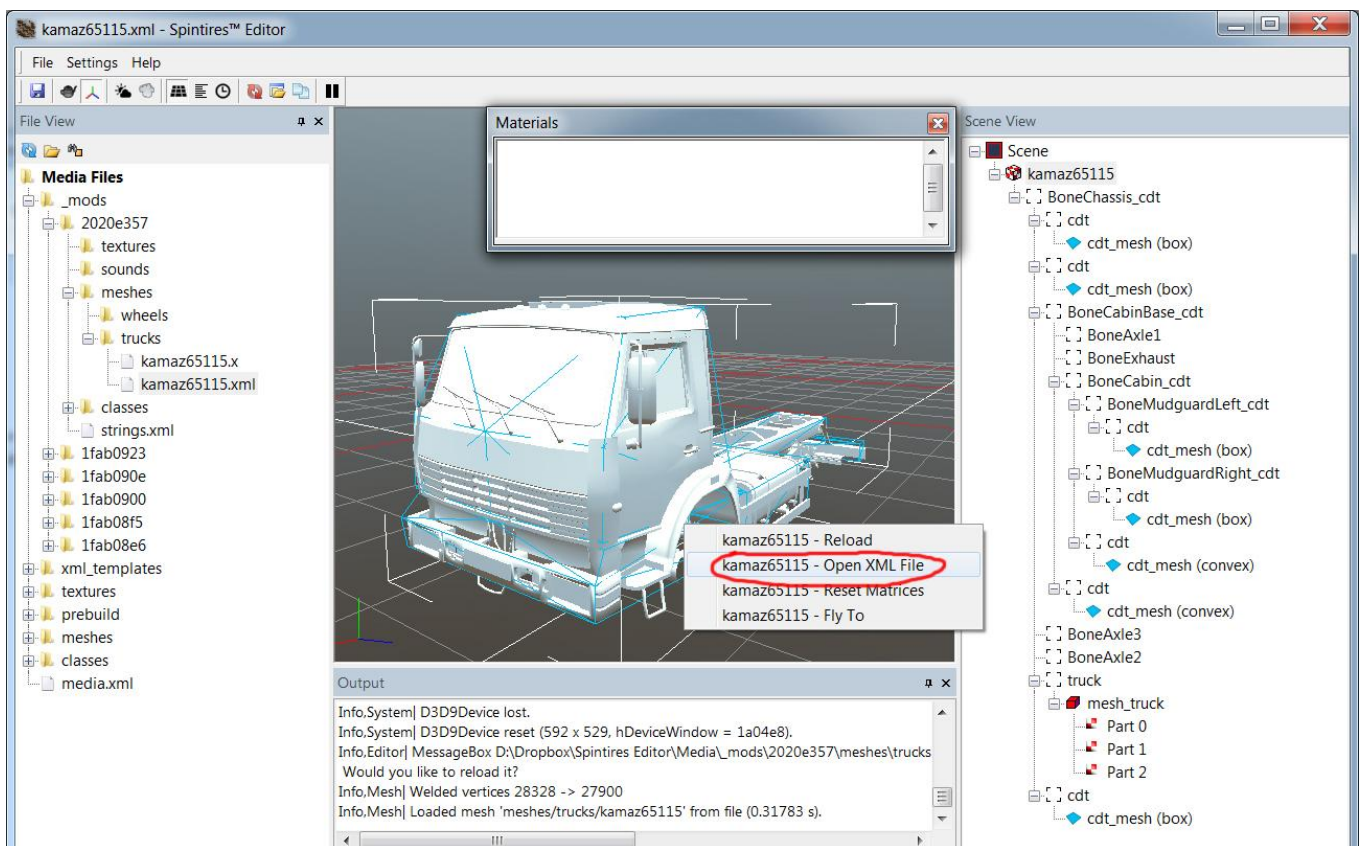
Navigate to newly created '.X' file in "File View" and double-click it:



A prompt to create new XML for '.X' file

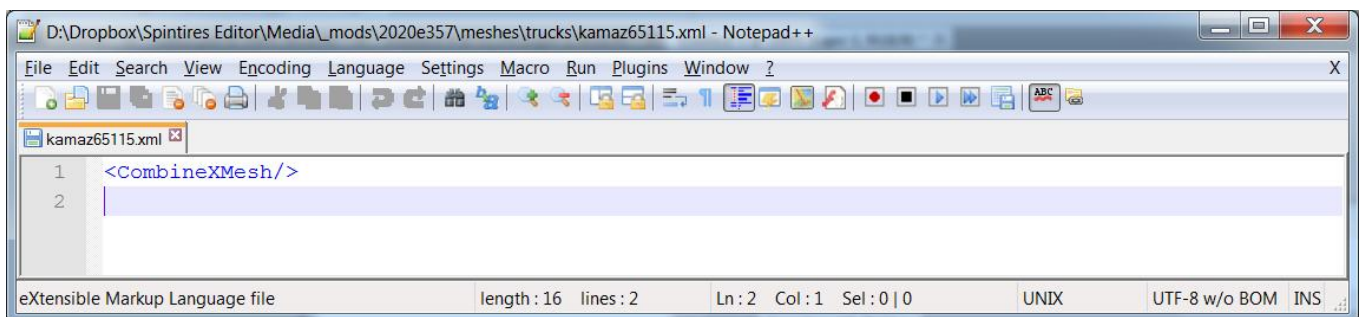
Technically, you can have multiple XML meshes referencing single '.X' file. But in general case you should have XML with the same name as '.X' file (leave suggested name as default).

Open new XML file in Spintires:MudRunner™ Editor. Right click main view and select “Open XML File” in context menu.



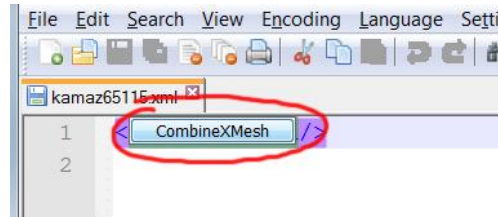
Spintires:MudRunner™ mesh file view. Notice mesh structure in “Scene View”. You should verify it before going any further. Any object (frame) you are planning to attach physical body to should have at least one “cdt” child (collision model). Collision models themselves are visualized with teal wireframe. Collision model type is part of the name in brackets. It can be either box, convex, or mopp. “Box” is the best performing collision model – while “Mopp” is by far the slowest – so make sure any collision models that can be represented as boxes are actually “Boxes”.

We assume you are using Notepad++ for XML editing, that you have “.xml” extension associated with it and that you have Spintires:MudRunner™ Editor plugins for Notepad++ installed (as described above). Given that, “Open XML File” would automatically launch Notepad++ with given XML mesh file contents:



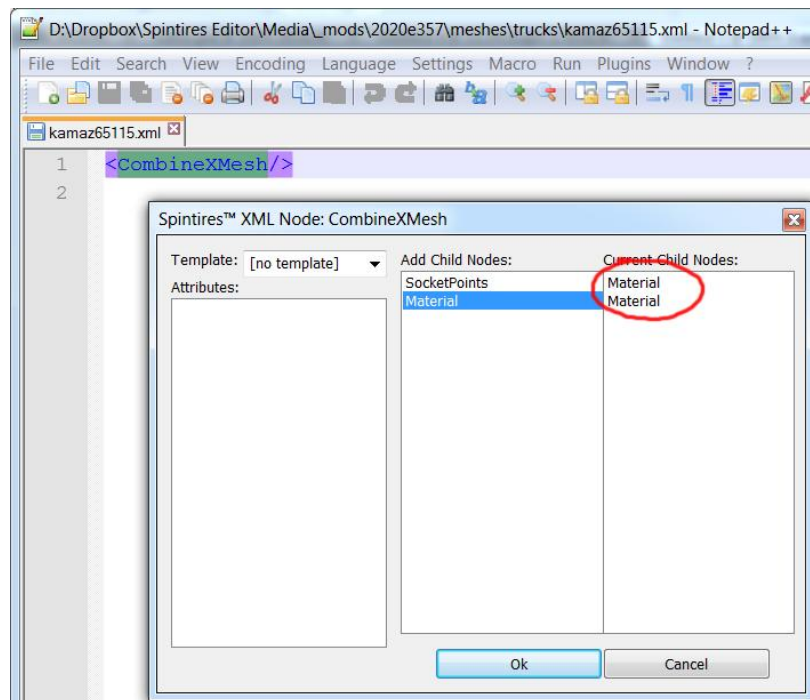
Notepad++ showing XML empty mesh file

Double-click on “CombineXMesh” (XML root element) – it should turn into a button:



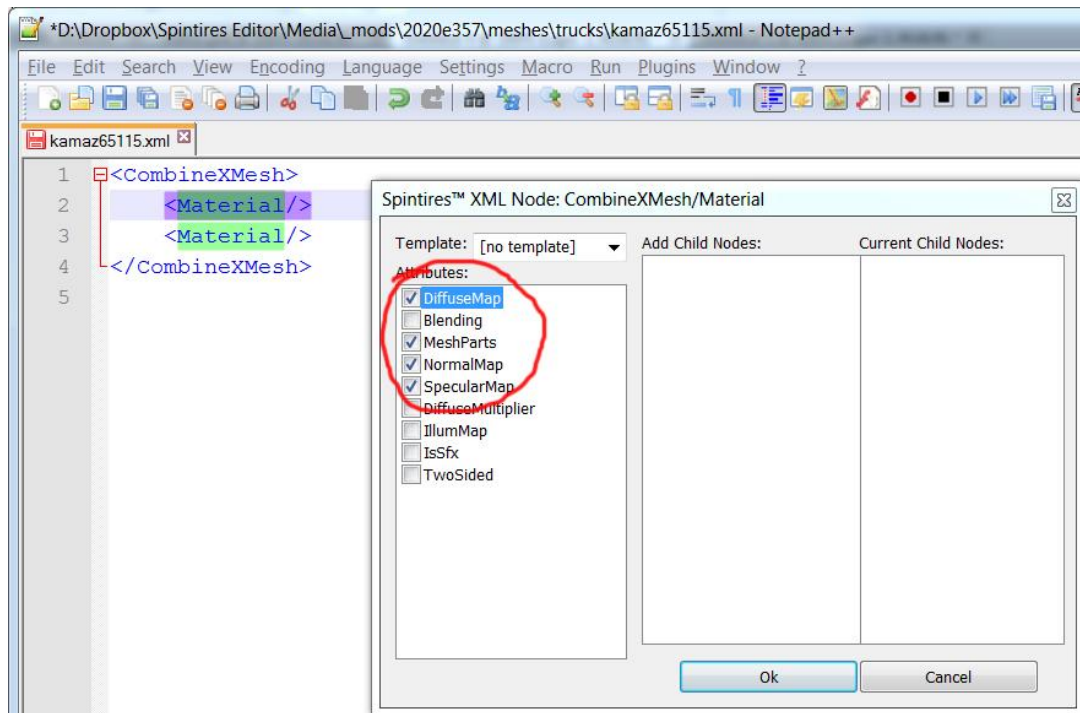
XML node name turns into a button

Click on “CombineXMesh” button. Node editing dialog should open up. Add appropriate amount of materials. (“material” describes part of mesh with specific texture, blending and surface shading)



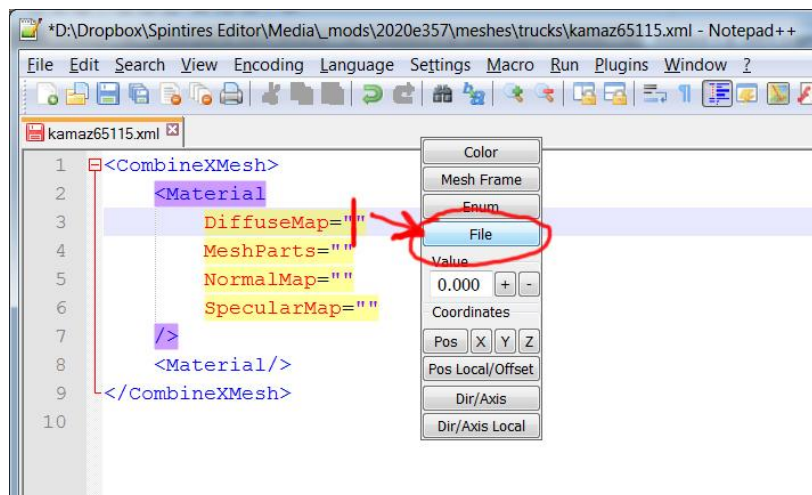
Dialog shows available child nodes and current child node. Double click on element on the list to add/remove them.

After “Material” child nodes are added, double-click them and add appropriate attributes:



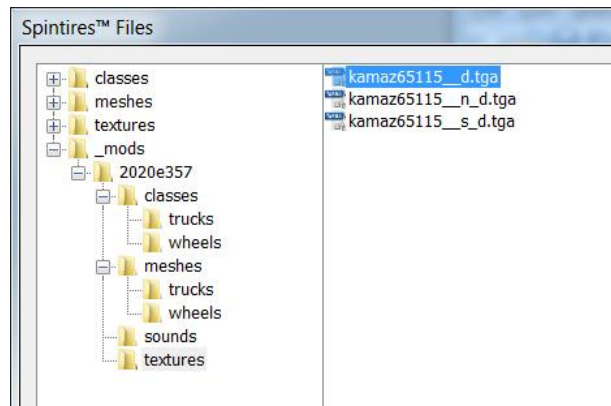
Select attributes for materials. DiffuseMap, NormalMap and SpecularMap are texture files. MeshParts describes part of the mesh which material is applied to – and is required unless there is single material in a file.

After you add DiffuseMap and other attributes, place cursor at the attribute values (between commas). Attribute-editing dialog should show up. Click “File” button and browse for the texture file:



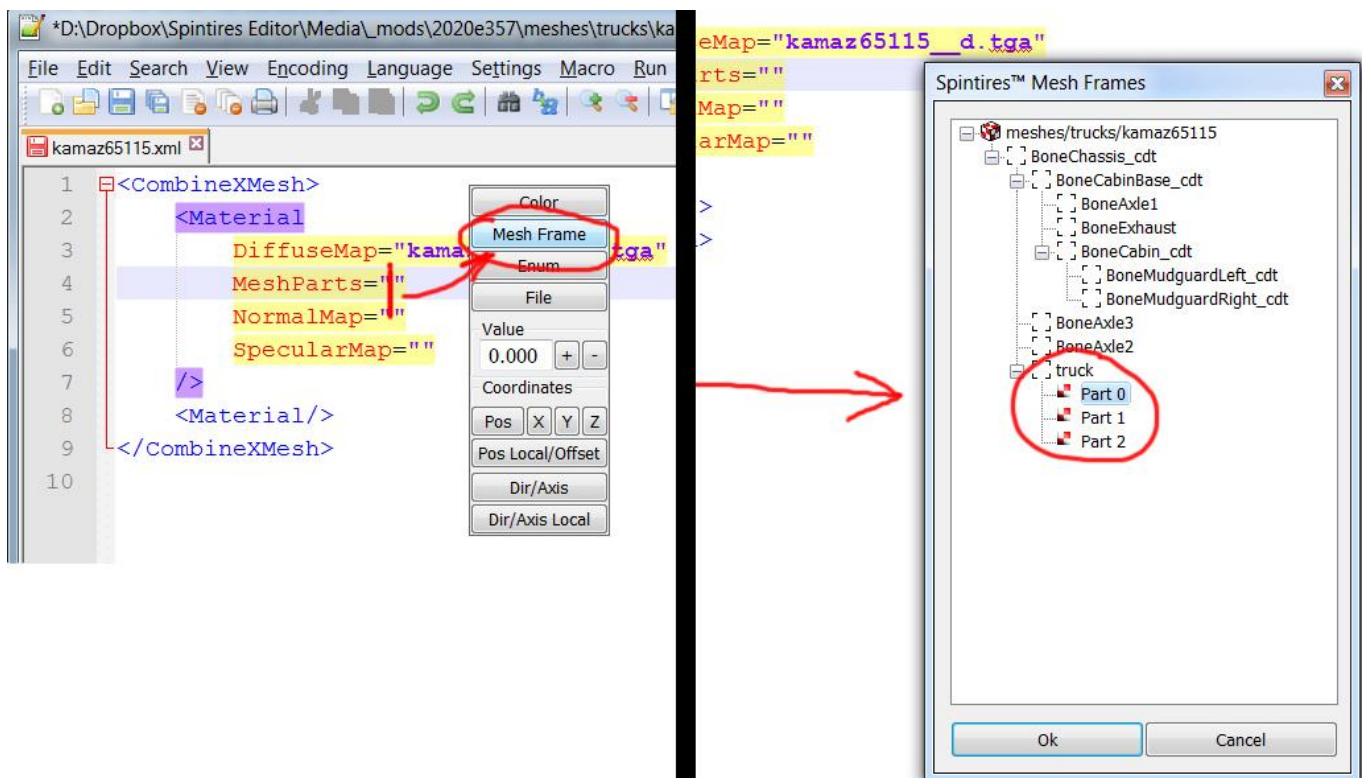
Attribute-editing dialog provides set of useful tools for XML editing

Pick texture file with Spintires:MudRunner™ file browser:



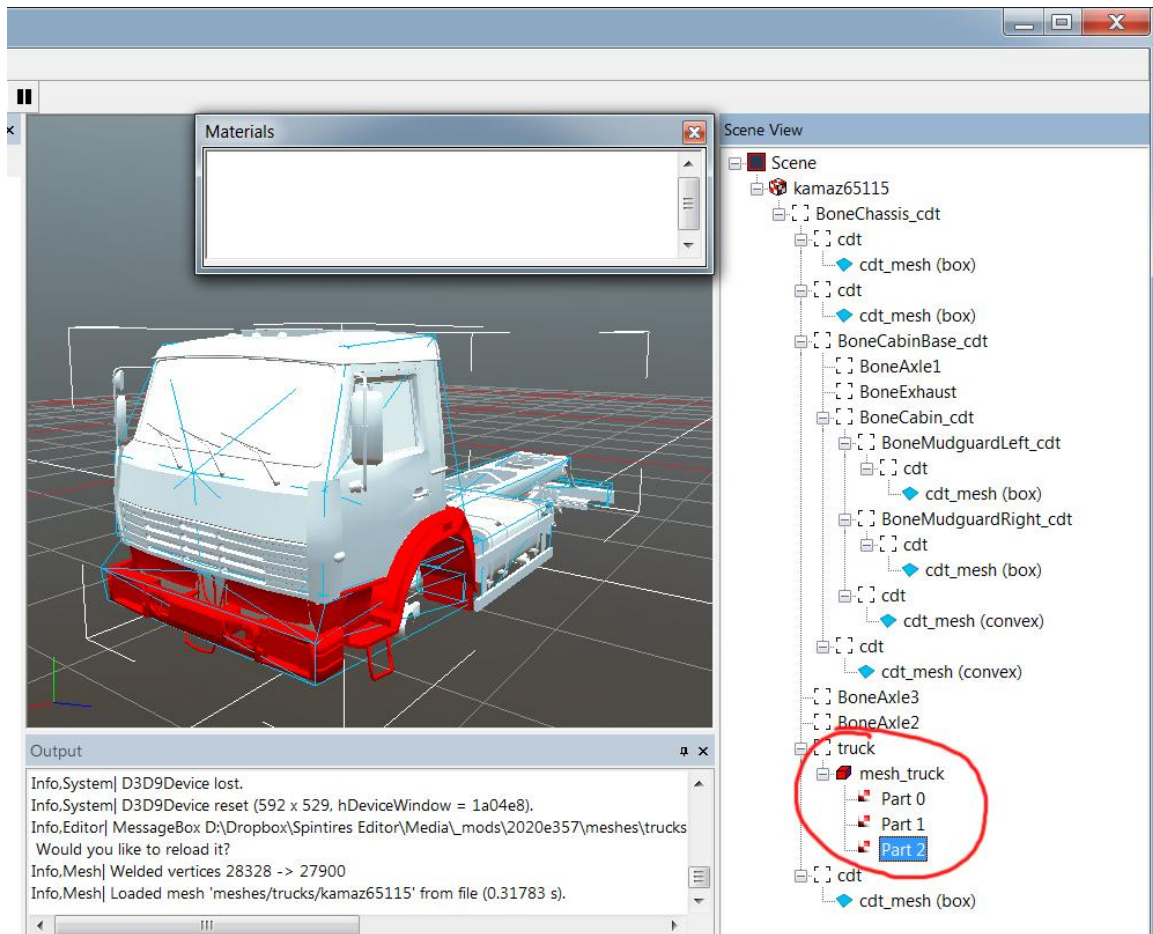
Spintires:MudRunner™ file browser. Note – when texture is referenced, only its sub-path under “..\Media\textures\” is stored. If mesh file (XML) is referenced, its extension is omitted and its sub-path under “..\Media\meshes\” is stored. If class file (XML under “..\Media\classes\”) is being referenced, its extension is omitted and its sub-path under “..\Media\classes\[class type]” is stored. Spintires:MudRunner™ file browser automatically extracts the right sub-path for the file you selected.

Use attribute-editing dialog to pick “MeshParts” for the material. Click “Mesh Frame” buttons and select one or more mesh part:



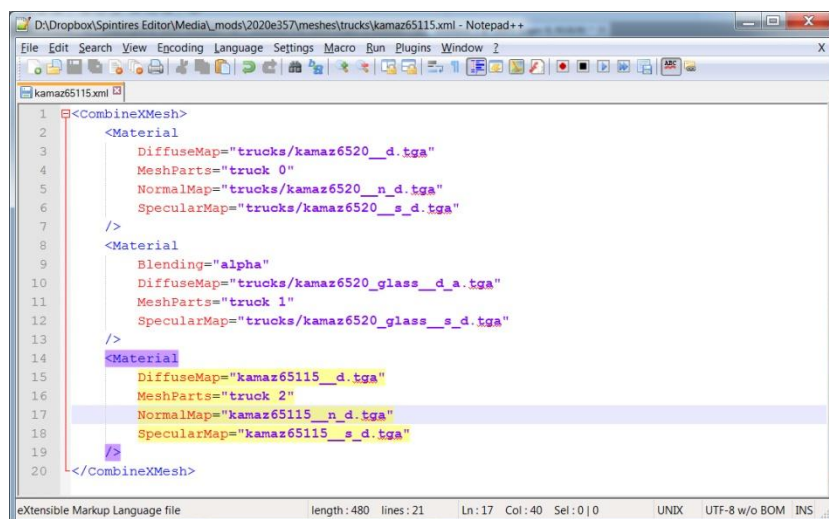
Spintires:MudRuuner™ mesh frame browser. It only works correctly after you specify correct mesh XML file within your truck XML file.

Use Spintires:MudRunner™ Editor to figure out which mesh parts should use which materials:



Spintires:MudRunner™ Editor show mesh parts. A mesh file ('.X') may contain multiple meshes. Each mesh might be split into multiple materials. A mesh part is identified by mesh (parent frame) name and an index (starting 0)

In the end XML mesh file should look similar to that:

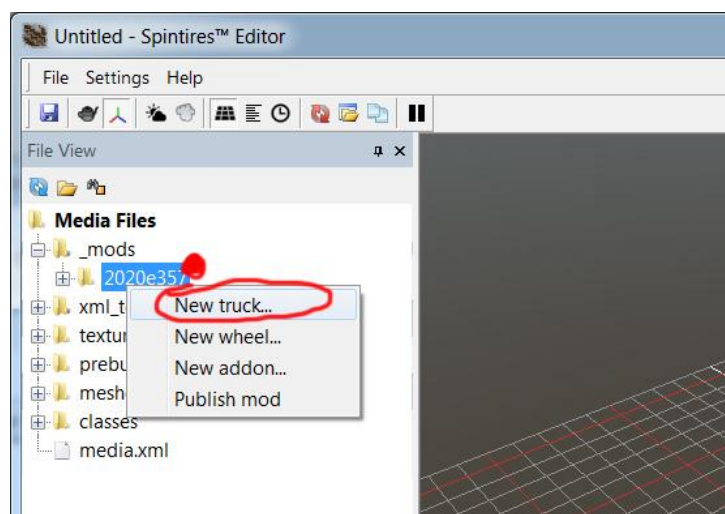


Typical view of a XML mesh file

Tip: Use ALT-SHIFT-F after text changes to standardize XML sorting/formatting.

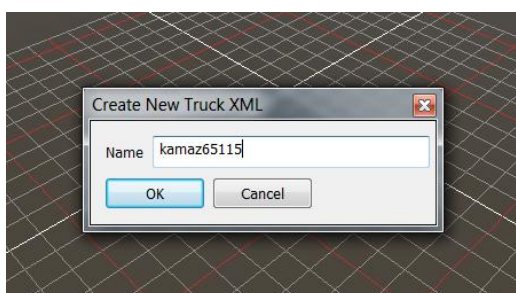
Step 4. Create truck file

Right click mod folder created at Step 1 and select “New truck...” in context menu:



Creating new truck

Pick the name for your new truck (it is advised to consist of alphanumeric lowercase letters and “_”)

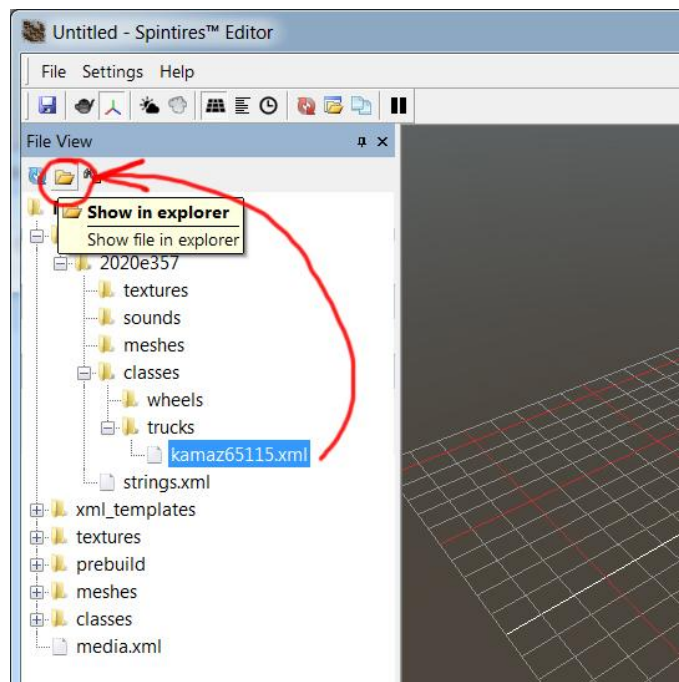


Naming new truck

Right now, Spintires:MudRunner™ mod may contain several (or none) trucks, wheels and addons. In future, ability to create map mods would become available.

After you choose a name, a new XML file should be create at [mod name]\classes\trucks.

Select it and pick “Show in explorer” in toolbar:



Show a file in Windows Explorer

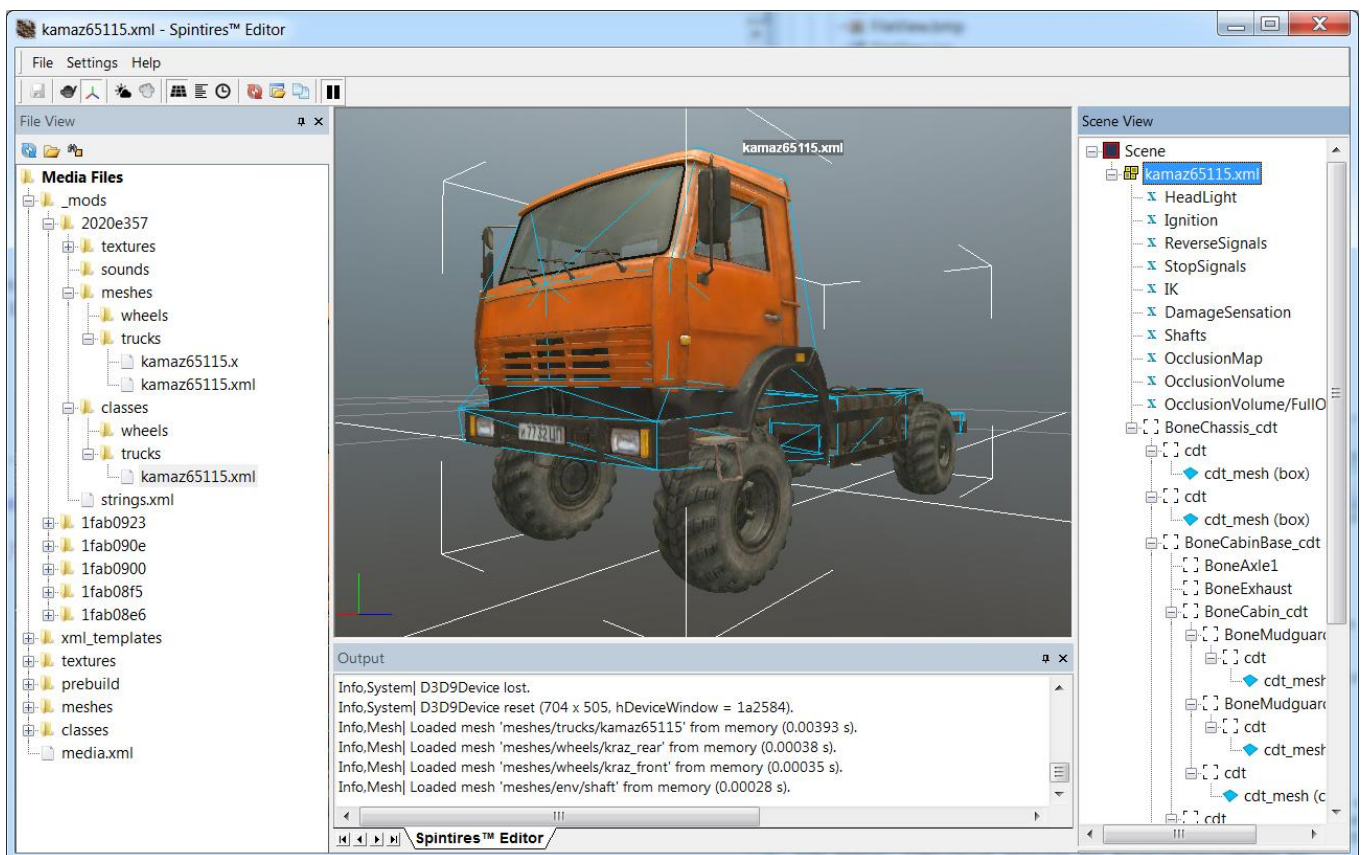
Open truck XML file in Windows Explorer. Spintires:MudRunner™ Editor automatically creates a template truck file which you now need to customize.

First off, select truck mesh file as an attribute value to “Truck/PhysicsModel/Mesh”. Use Spintires:MudRunner™ tools to pick a file and instead of “[truck mesh]” you should have “trucks/kamaz65115” (as an example).

Next, select wheel classes (some of default wheel classes are listed in “Media/classes/wheels” folder) – and replace “[rear wheel class]” and “[front wheel class]”. Note that those are XML classes – not meshes.

After that, verify “ParentFrame” attributes point to correct mesh frames (sub-objects).

With those modifications done, you can open truck XML file in Spintires:MudRunner™ Editor and it should look similar to that:

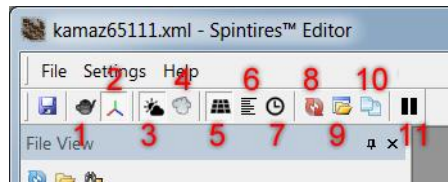


Default mod truck with minimal modifications

Next step is customizing truck parameters.

Step 5. Customizing the truck

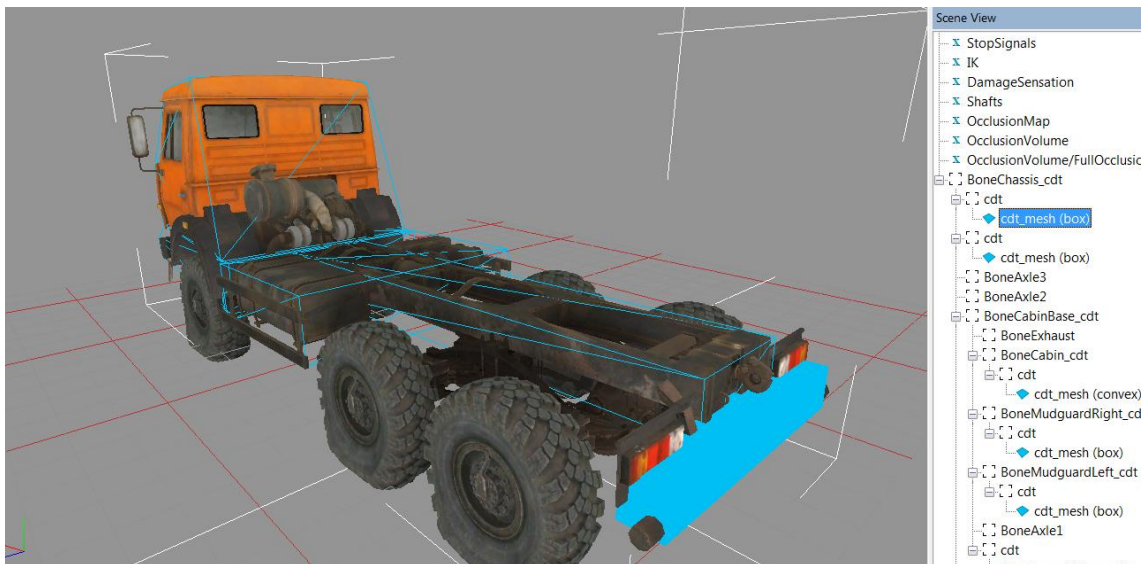
Process of creating the truck essentially boils down to editing truck XML text file. But there are a number of tools to assist you with that:



Spintires:MudRunner™ Editor toolbar. Each action has a hotkey (hover mouse to read tooltip)

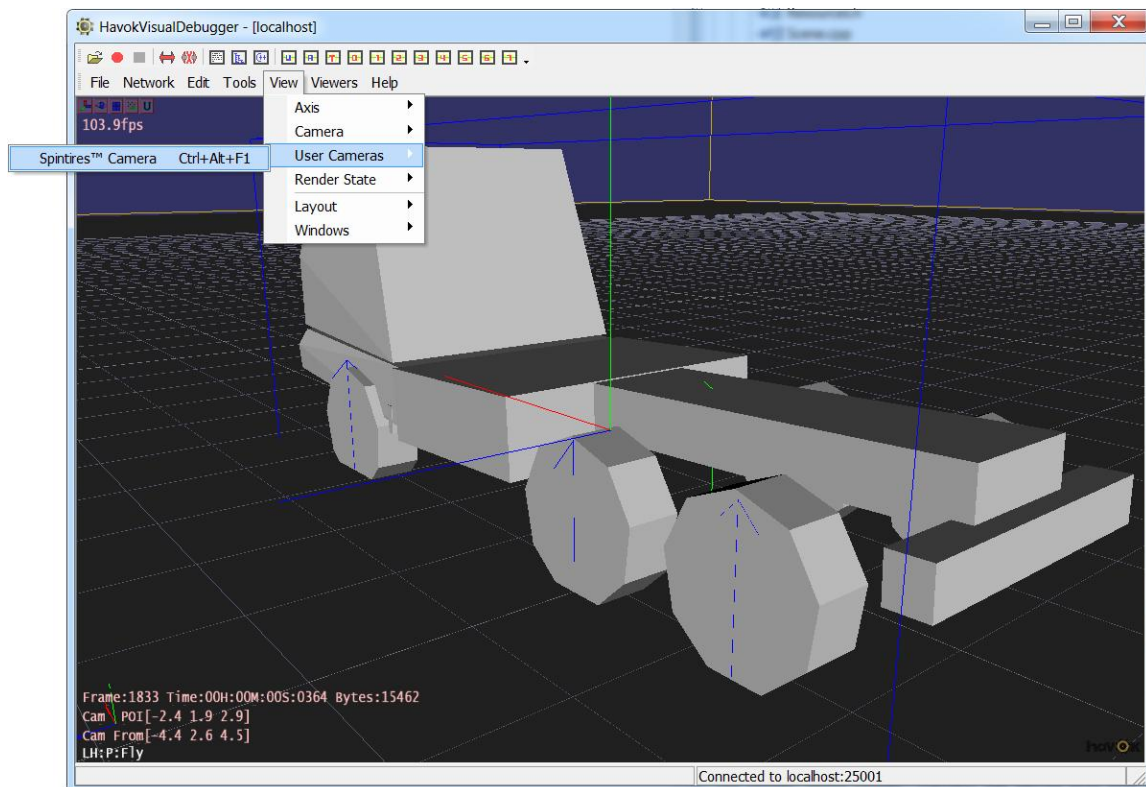
1. Wireframe mesh view
2. Disable edit when you only want view but not change (for instance, object positions in mesh view)
3. Lighting mode: directional or ambient-only (helpful for verifying textures and occlusion volumes)
4. Enable/disable fog. Not used for creating truck mods.
5. Show/hide scene grid
6. System statistics
7. System timers
8. Reload resources (textures and meshes)
9. Open Media file. An important assist to navigate files of your mods.
10. Duplicate object. Not used for creating truck mods.
11. Start/stop physics simulation. Use this to test trucks physics within Spintires:MudRunner™ Editor.

You can also use Havok® Visual Debugger to test your physics setup. To do so, open truck XML file within Spintires:MudRunner™ Editor, then start the simulation (toolbar button 11):



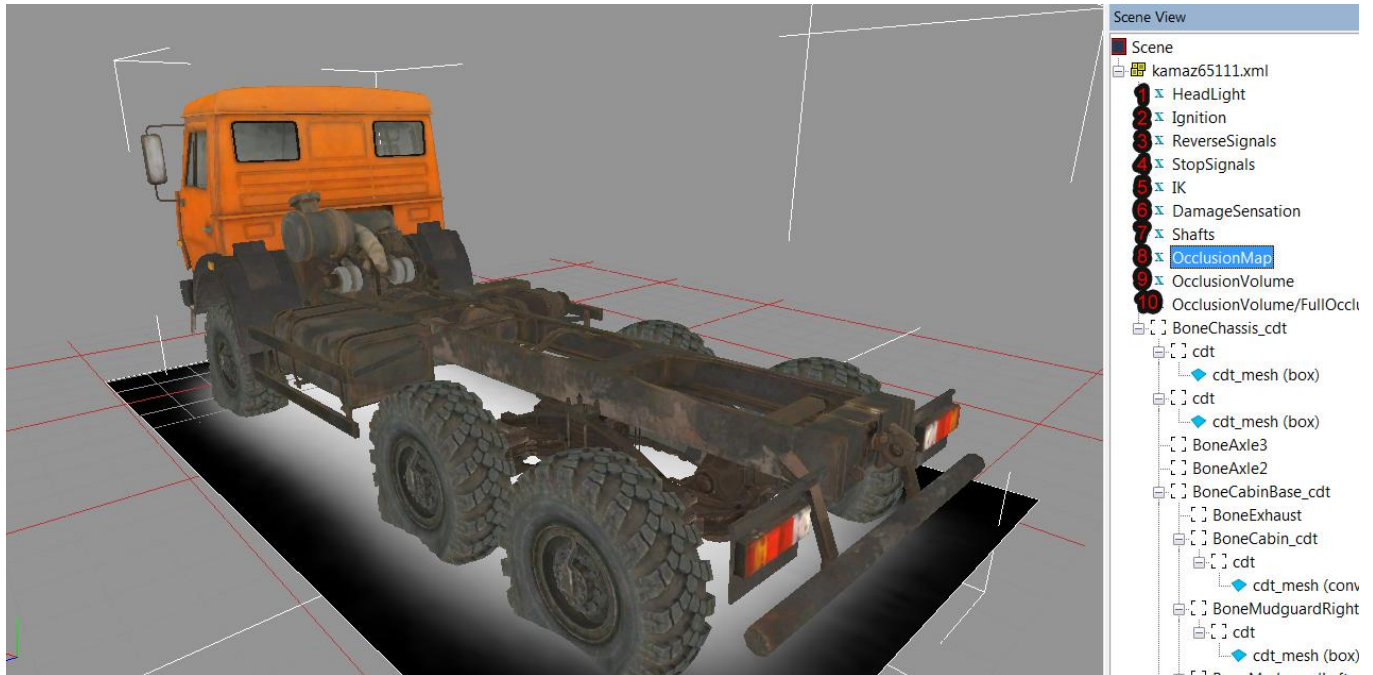
Spintires:MudRunner™ Editor shows collision information for objects that physics bodies can be attached to.

Then launch Havok® Visual Debugger (hkVisualDebugger.exe) – and connect it to Spintires:MudRunner™ Editor (if it doesn't auto-connect, Network > Connect > localhost). You can then use View > User Cameras > Spintires:MudRunner™ Camera to align Visual Debugger's camera to Editor's camera:



Havok® Visual Debugger view. You can hold “Space” and then drag mouse to pull objects around and test your physics setup. You can also view constraints information.

Spintires:MudRunner™ Editor is essentially a truck viewer:



Truck view in Spintires:MudRunner™ Editor

1,2,3,4 (HeadLight, Igniting, ReverseSignals, StopSignals) – allows you to trigger and view sections from Truck/ModelAttachments and TruckAddon/ModelAttachments XML.

5 (IK) – shows IK setup for TruckAddon/ControlledIK and TruckAddon/AutomaticIK

6 (DamageSensation) – shows area of the truck the game damage is tested against. This area is also used to test water overflows.

7 (Shafts) – positions of drive shafts of your truck.

8 (OcclusionMap) – a texture projected onto terrain beneath the truck to simulate light occlusion.

9 (OcclusionVolume) – an area of the truck that truck self-occludes.

10 (OcclusionVolume/FullOcclusion) – used instead of OcclusionVolume if addon with attribute "IsChassisFullOcclusion" set to "true" is installed.

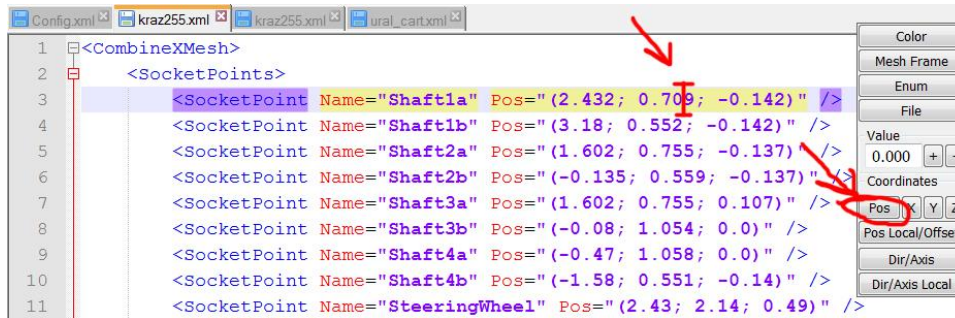
Spintires:MudRunner™ Editor works in pair with Notepad++ Spintires:MudRunner™ plugins:

Example 1. Adjust positions of drive shafts

1. Open MESH XML of the truck you want to add drive shafts to

Tip: Use ALT-SHIFT-O

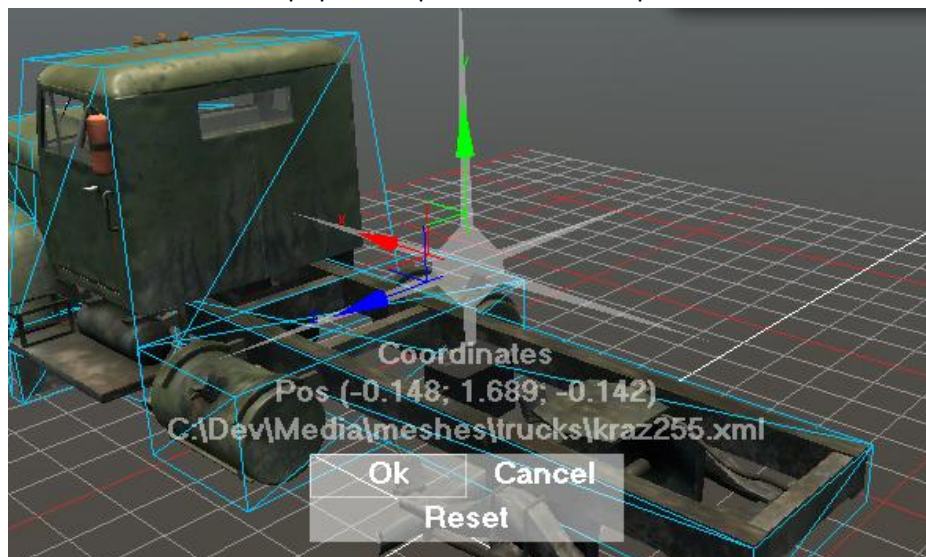
2. In context menu, select “Open XML File”
3. In Notepad++, add “SocketPoints/SocketPoint” node if there is none, then put a cursor within commas of “Pos” attribute value, the press “Pos”



```
1 <CombineXMesh>
2 <SocketPoints>
3 <SocketPoint Name="Shaft1a" Pos="(2.432; 0.709; -0.142)" />
4 <SocketPoint Name="Shaft1b" Pos="(3.18; 0.552; -0.142)" />
5 <SocketPoint Name="Shaft2a" Pos="(1.602; 0.755; -0.137)" />
6 <SocketPoint Name="Shaft2b" Pos="(-0.135; 0.559; -0.137)" />
7 <SocketPoint Name="Shaft3a" Pos="(1.602; 0.755; 0.107)" />
8 <SocketPoint Name="Shaft3b" Pos="(-0.08; 1.054; 0.0)" />
9 <SocketPoint Name="Shaft4a" Pos="(-0.47; 1.058; 0.0)" />
10 <SocketPoint Name="Shaft4b" Pos="(-1.58; 0.551; -0.14)" />
11 <SocketPoint Name="SteeringWheel" Pos="(2.43; 2.14; 0.49)" />
```

Spintires:MudRunner™ Notepad++ plugins

4. Spintires:MudRunner™ Editor should pop back up and show world position of select shaft node:

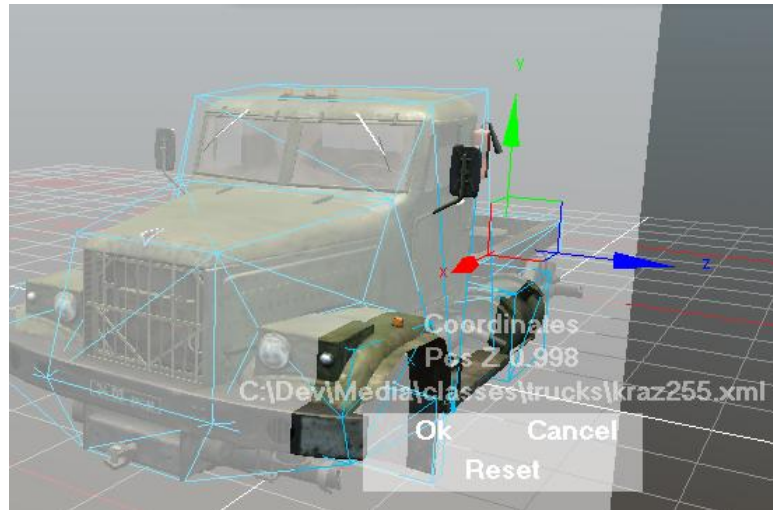


Spintires:MudRunner™ Editor allows editing of world positions

5. Use 3d control to change position of shaft node – OR – click the mesh to snap to point of intersection. Press “Ok” to save coordinates in Notepad++, “Cancel” to cancel the editing and “Reset” to snap back to original input.

Example 2. Adjust OcclusionMap

1. Open MESH XML of the truck you want to add drive shafts to
Tip: Use ALT-SHIFT-O
2. In context menu, select "Open XML File"
3. In Notepad++, add "Truck/OcclusionMap" node if it's not there yet then put cursor within commas of "HalfSizeZ" attribute value, the press "Z" (next to "Pos")



Spintires:MudRunner™ Editor allows editing of individual components (X,Y,Z) world positions

4. Use 3d control to change width of OcclusionMap.

In the same manner, you can edit directions.

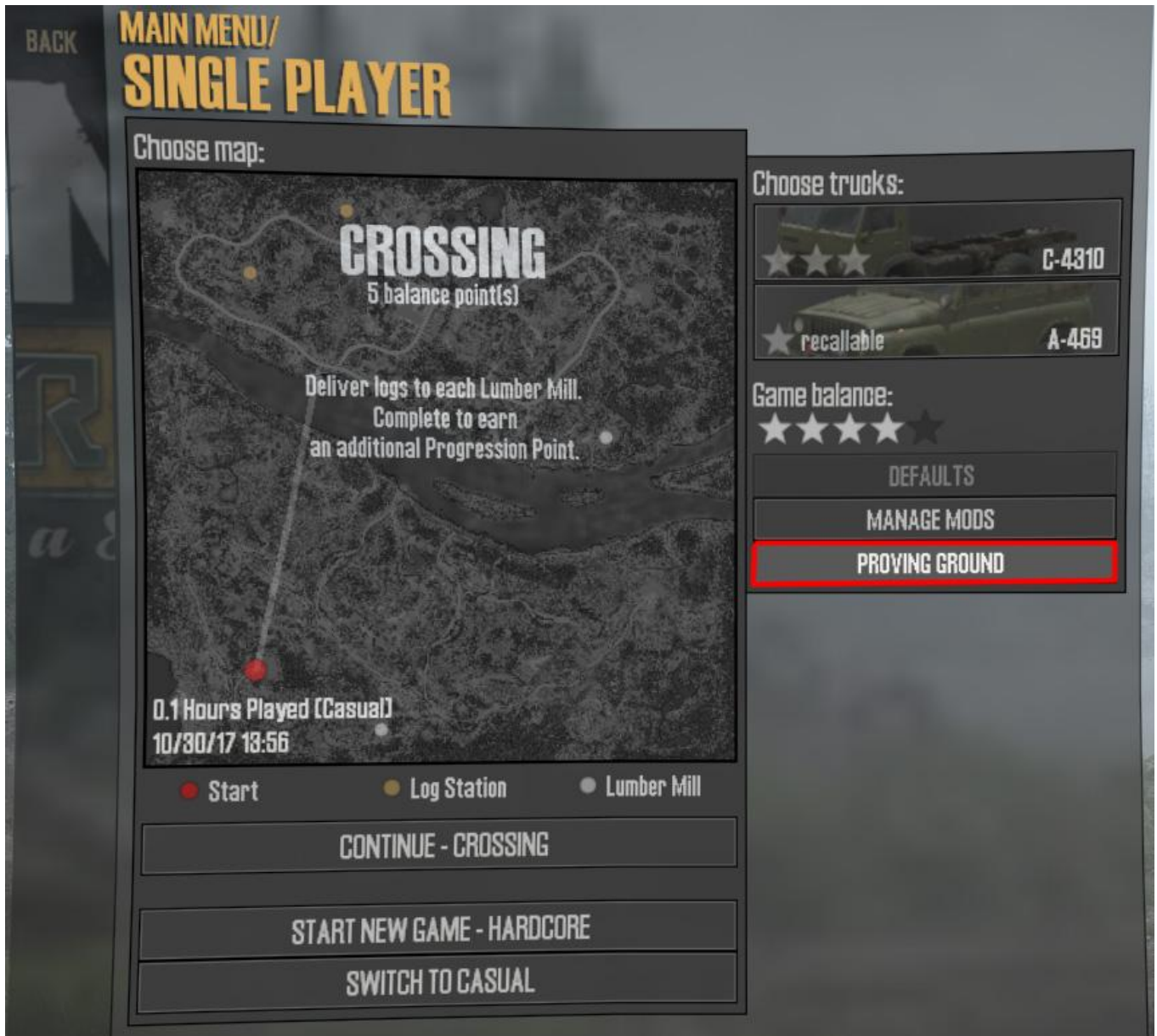
You can also edit local directions and positions (which are relative to position of other objects). To do so, first select a sub-object in Spintires:MudRunner™ Editor, the press "Pos Local/Offset" or "Dir/Axis Local in Notepad++ Spintires:MudRunner™ plugin dialog.

Step 6. Test the truck in the game

After you have tested your truck with Spintires:MudRunner™ Editor, it is advised you test it in game.

Launch Spintires:MudRunner™ (you have to own it), make sure Steam® is active.

In Main Menu, press “Proving ground” to launch proving grounds map:



Spintires:MudRunner™ “proving ground” mode

“Proving ground” is a special map where you can spawn, reload and refill your trucks at any time. It also allows you to spawn trucks and addons from un-published mods (in all other cases only published mods that you have subscribed to are available).

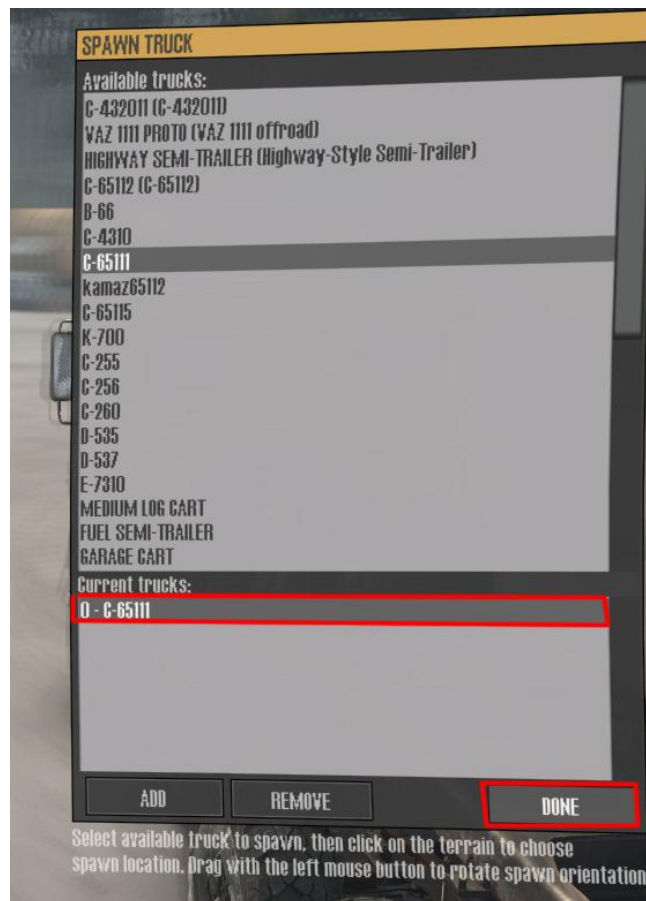


Proving grounds allows you to use development tools.

Use "Spawn" to select and add your new truck:



Proving grounds allows you to spawn and switch between trucks



Switch to spawned truck

Proving grounds has a number of locations you can teleport your truck to. Test your truck in various environments and various addons configurations before going to next step. You can also use Havok® Visual Debugger and connect it to Spintires:MudRunner™ to test your physics setup.

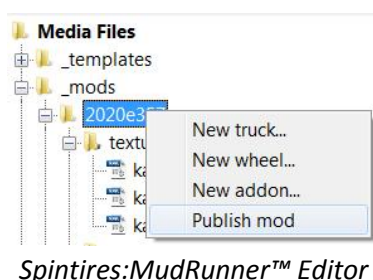
Step 7. Publish mod

You should provide friendly names for all the trucks and addons in your mods. To do so, add nodes like that

```
<String Id="kamaz65111" Value="Type C-65111" />
```

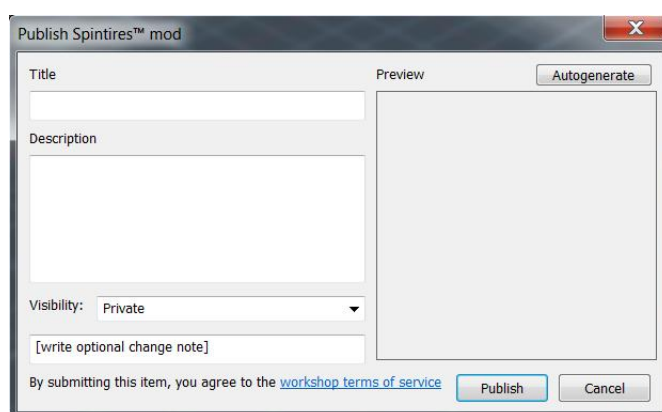
in special file called "strings.xml" located in root folder of your mod. Replace "kamaz65111" with a filename of your truck/addons, and give it a name in "Value" attribute. Mod localization is not currently supported so it's recommended to use English for all titles.

After you have named all your assets and tested them in both Editor and Game, launch Spintires:MudRunner™ Editor and right click your mod in "File View":



Spintires:MudRunner™ Editor

Publishing dialog would pop up:



Publishing Spintires:MudRunner™ mod to Steam® Workshop

Enter Title and Description for your mod. Select Visibility (by default, "Private" means no one but you will see your mod). Press "Autogenerate" to create preview for your mod. You can then edit auto-generated picture which is located at "Media_mods\[mod name]\preview.png". It is advised to retain as much of auto-generated design as possible to make mod icons look uniform.

If it's the first time you submit item to Steam® Workshop, you should agree to terms of service by clicking the link "workshop terms of service".

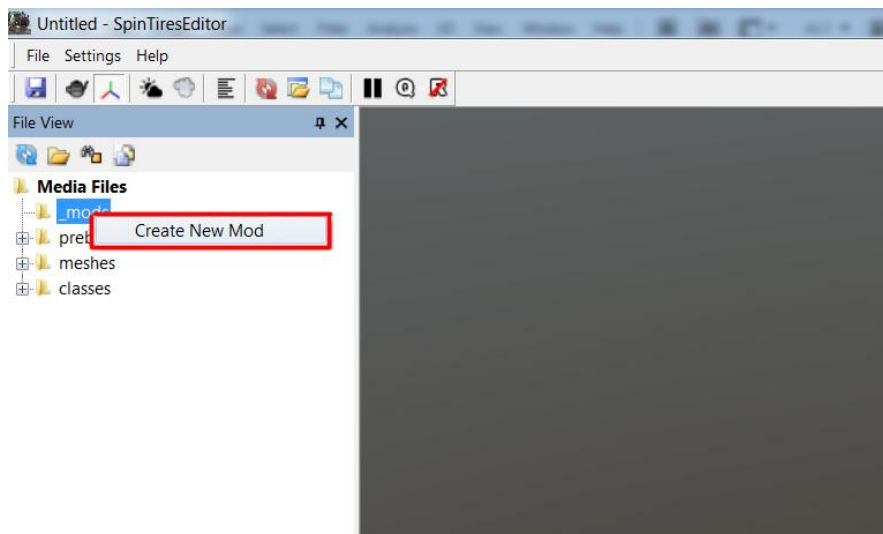
Provide optional change note and press "Publish". A progress bar dialog would pop-up. After uploading process is complete, mod would be visible at <http://steamcommunity.com/app/675010/workshop/>

For examples see "Samples/mod_addon" and "Samples/mod_truck_inherited" in Spintires:MudRunner™ install directory.

Importing Spintires-2016 mod into Spintires:MudRunner™

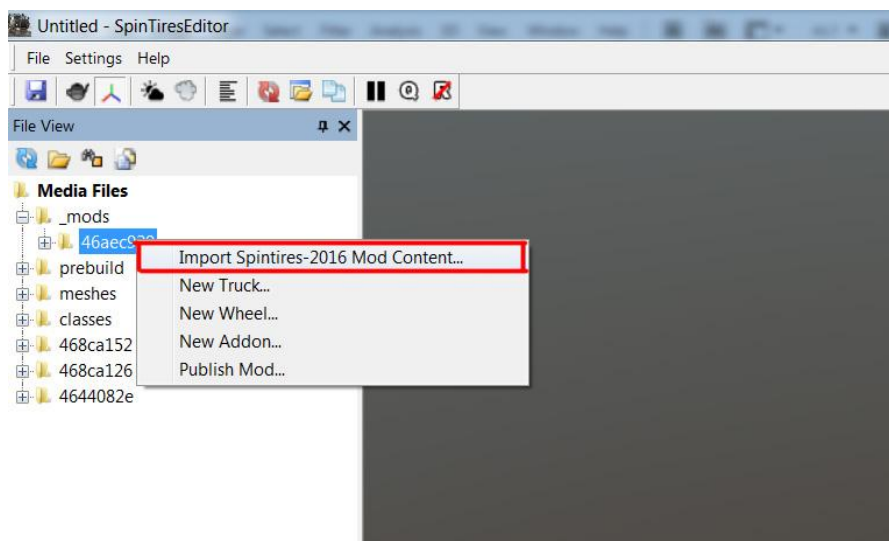
Step 1. Create new mod

Repeat Step 1 from “Creating a mod for Spintires:MudRunner™” section:



Step 2. Import Spintires-2016 mod content

Right-click new mod and select “Import Spintires-2016 Mod Content...”.



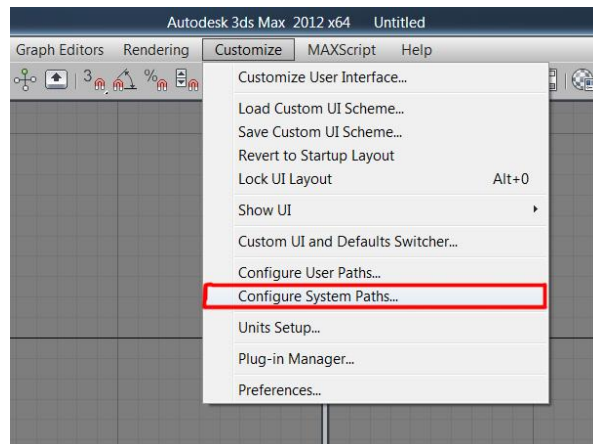
Browse for the mod content and your folder with Spintires:MudRunner™ will get populated with converted files. There are still 2 things you need to do by hand after that:

- From the XML mesh file of your truck, remove “SteeringWheel” socket:
<SocketPoint Name="SteeringWheel" Pos="(2.43; 2.14; 0.49)" />
Move “Pos” to your XML class file (TruckData/Driver):
<Driver ... SteeringWheelPos="(2.43; 2.14; 0.49)" />
- If you used inheritance in your XML files, double-check the parent files, as some of them have changed and it wasn't taken into account when converting content!

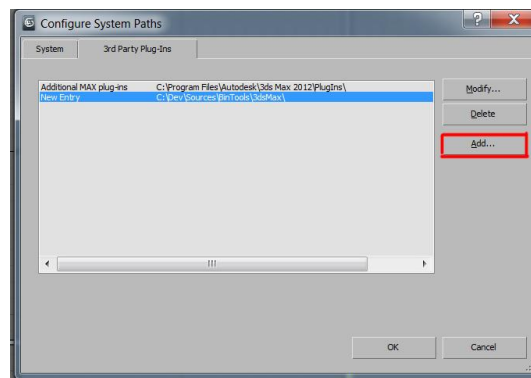
Appendix A. Using 3dsMax2012 to create X files

Install “X” file export plugin: 3rdParty/3dsMax2012/kwxport-1.5.0-win32.msi or 3rdParty/3dsMax2012/kwxport-1.5.0-x64.msi (or a different plugin of your choice)

Install Spintires:MudRunner™ 3dsMax script plugin (“CombineExportX.ms”):

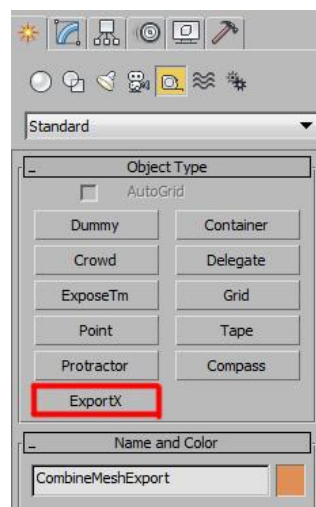


Follow a standard 3dsMax procedure when installing script plugins, go to “Customize>Configure System Paths...”

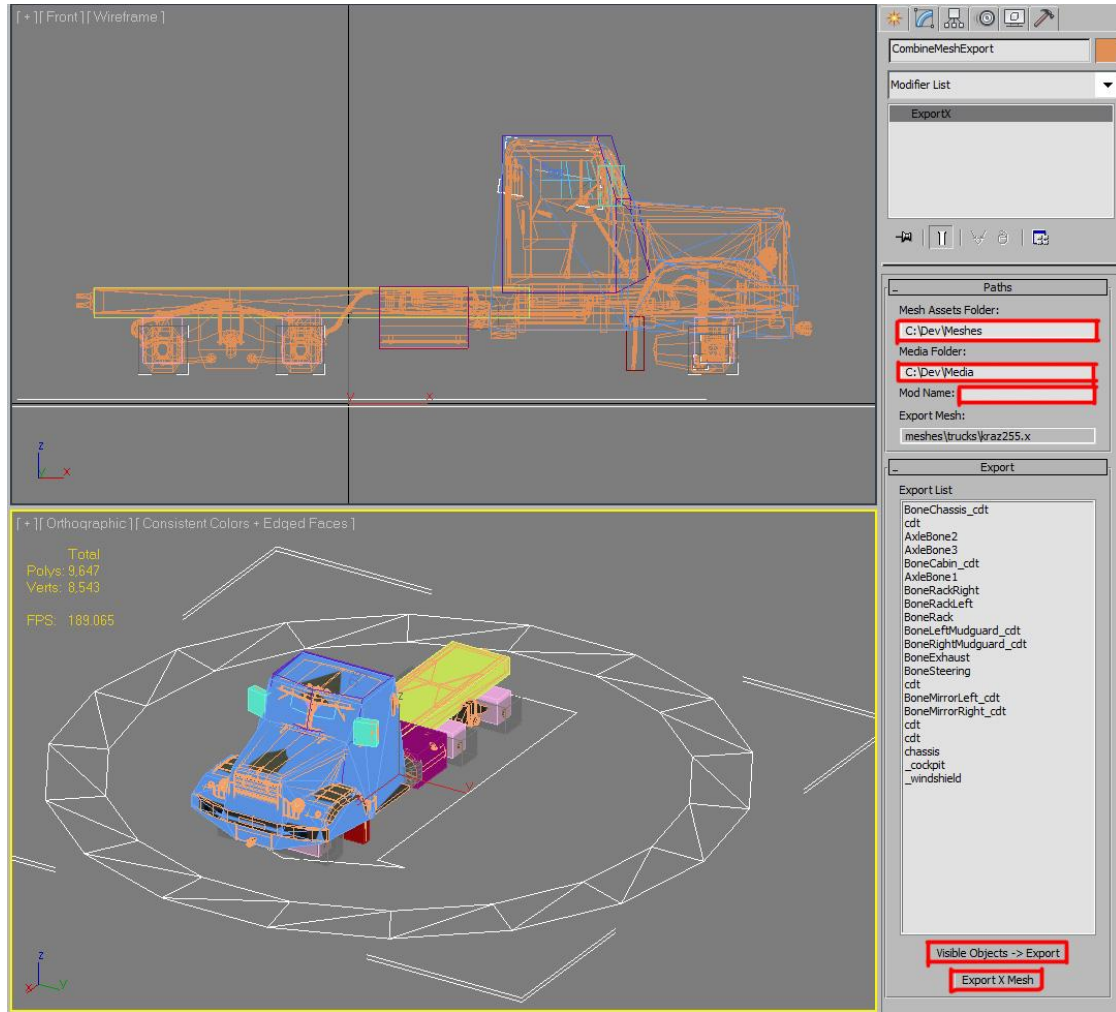


Add a path to “CombineExportX.ms”. Restart 3dsMax.

You should be able to create a “Helper” called “ExportX”:



Select “ExportX” and click anywhere in viewport as this helper doesn’t have a position.



Configure Spintires:MudRunner™ 3dsMax plugin:

1. Specify a “Mesh Assets Folder” path (root directory to your .MAX files) – a “source root” folder.
2. Specify “Media Folder” path (a directory in the Spintires:MudRunner™ Editor folder) – a “destination root” folder. **Make sure the subdirectory of your MAX file relative to “Mesh Assets Folder” is the same as subdirectory of your destination X file compared to [“Media Folder”][“Mod Name”]\meshes.**
3. Specify mod name (8 to 16 alphanumeric lowercase letters) that your truck belongs to.
4. Leave only the objects that needs to be exported to X files visible (hide helpers and unused meshes) and press “Visible Objects -> Export” (you only need to do this once)
5. Press “Press Export X Mesh”. X file will be saved to [“Media Folder”][“Export Mesh”]. **Note that to re-export the X file, you only need to press this button now.**

Appendix B. General information on textures

Spintires:MudRunner™ can read wide variety of texture formats – but is recommended to use lossless texture formats (TGA is good choice), because no matter what original format was, all textures are automatically converted to “DDS” file format.

Original mesh or textures won't be part of published mod!

It is advised to always use lowercase alphanumeric names for all assets, the only allowed special symbol is “_”.

Some properties of texture file are described by a postfix of its filename (all symbols past double “_” symbol):

_cube – file would be automatically converted to cubemap texture.

_vol – file would automatically converted to volume texture.

_n – file is a normalmap and would use be compressed accordingly. Mipmap chain would be generated automatically

_s – file is specularmap and would be converted to 1-channel texture (only R channel or original texture). Mipmap chain would be generated automatically

_uncmp – file won't be compressed – so texture quality would be better but size would be 4 times larger.

_d – file is a diffusemap and would be compressed accordingly. Mipmap chain would be generated automatically.

_a – file contains alpha-channel (if filename doesn't have this postfix, A channel is omitted)

But in general, you should name your files like that:

[name]__d.tga for diffuse textures.

[name]__d_a.tga for diffuse texture with alpha-channel (transparency or alpha-kill).

[name]__s_d.tga for specular map textures.

[name]__n_d.tga for normal map textures.

Appendix C. General information on XML inheritance

Spintires:MudRunner™ XML files are typical XML files that any software can read, but Spintires:MudRunner™ uses special rules while reading data from XML files.

Concept 1. Templates

Spintires:MudRunner™ XML template is used to avoid copy-pasting. For instance, wheels of particular truck are mostly similar – but each is described by its own XML node. To avoid copying attributes in each node, a Wheel template is created in a special section of file called “**_templates**”:

```
<_templates>
  <Wheel>
    <RearWheel
      ConnectedToHandbrake="true"
      Type="kraz_rear"
      HardpointOffsetY="-0.244"
      SuspensionMin="-0.4"
      SuspensionStrength="0.02"
      Torque="default"
    />
  </Wheel>
</_templates>
```

Note: there are global template files with predefined templates. They can be included by adding Include="trucks" (for instance) attribute. Those files are all stored in “\Media_templates” folder.

And then each Wheel node uses this template (attribute called “**_template**”):

```
<Wheel _template="RearWheel" HardpointX="-0.49" />
<Wheel _template="RearWheel" HardpointX="-0.49" RightSide="true" />
<Wheel _template="RearWheel" HardpointX="-1.93" />
<Wheel _template="RearWheel" HardpointX="-1.93" RightSide="true" />
```

Note: XML node can only use template which was defined as a child of node with the same name. Notice Wheels are using “RearWheel” template which is a child of node “Wheel” in “_templates” section.

So this is the same as copying all attributes to each individual XML node. A node that uses template can override any attribute that is part of the template. Node hierarchy is copied as well as attributes by templates system.

Concept 2. Inheritance

Spintires:MudRunner™ XML file can inherit another file:

```
<_parent File="kamaz6520" />
```

Note: ".xml" extension is omitted

This is like a global template for entire file. So all attribute and nodes are loaded from file "kamaz6520.xml" and then nodes and attributes from current file (if any) are merged on top (overridden).

It makes it easy, for instance, to maintain multiple trucks that differ slightly from each other. Inheritance also affects game behavior of a truck or truck addon:

- Steam® game achievements work with all inherited trucks. I.e. if truck B has
`<_parent File="A" />`
Then player can get all A achievement with B
- Spintires:MudRunner™ truck addon requirements and excluded addons are transferred all inherited addons.
I.e. `ExcludeAddons="A"` also excludes B if B has `<_parent File="A" />`

And requirement

```
<RequiredAddon Types="A" />
```

Is fulfilled by B if B is inherited from A

- Spintires:MudRunner™ load addons inherit all properties (achievements and types of object loads break into)

`_inheritRemove`

If a node of inherited XML file has `_inheritRemove="true"`, then it won't be copied from parent file. For instance if your truck mod has 4 wheels instead of 6, you add 2 Wheel nodes with `_inheritRemove`.

`_noinherit`

If a node of inherited XML file has `_noinherit="true"`, then its attributes and child nodes won't be copied from parent file. For instance if your truck mod has a completely different wheel, you use `_noinherit` to prevent parent file from overriding any of your parameters.

Appendix D. Attributes introduced since Spintires-2016

Seethrough engine:

Truck/TruckData/EngineIconMesh="env/engine_default" (built in meshes are "engine_small", "engine_default", "engine_big")

Truck/TruckData/EngineIconScale="1.3" (size of the seethrough engine mesh)

Seethrough fluid:

Truck/TruckData/

<LimitedFluid

Center="(0.8; 0.829; 0.843)"

LengthX="0.891"

RadiusY="0.297"

RadiusZ="0.297"

Type="Cylinder"

/>

Or

Truck/TruckData/

<LimitedFluid

Center="(-0.6; 0.952; 0.778)"

SizeX="0.863"

SizeY="0.268"

SizeZ="0.541"

Type="Box"

/>

Steering wheel and steering rack:

Truck/TruckData/

```
<SteeringWheel Frame="BoneSteering" />
<SteeringRack
    Frame="[rack frame]"
    FrameSteerLeft="[frame LEFT]"
    FrameSteerRight="[frame RIGHT]"
    RackHalfSizeZ="[rack length along Z]"
/>
```

Cockpit camera parameters:

Truck/TruckData/

```
<Cockpit
    _template="Default"
    ControlsDiffuse="g(62; 64; 59)"
    ControlsPos="(2.63; 2.13; 0.608)"
    ViewPos="(2.0; 2.38; 0.49)"
    WindshieldDiffuseTexture="trucks/kraz255_glass__d_a.tga" (the same texture as windshield in mesh)
/>
```

Note that steering wheel positioned is moved from the mesh XML to truck class XML :

Driver/SteeringWheelPos.

For cockpit view to work properly, you also need to create a mesh called “_cockpit” in your truck mesh file. For the windshield to work properly in cockpit camera mode, you need to duplicate the windshield mesh and call it “_windshield”. You can increase the mesh tessellation of “_windshield” so wetness/mud overlays are applied more evenly (but don’t over-tessellate!) (see Samples/3dsmax2012_vehicle/kraz255.max)

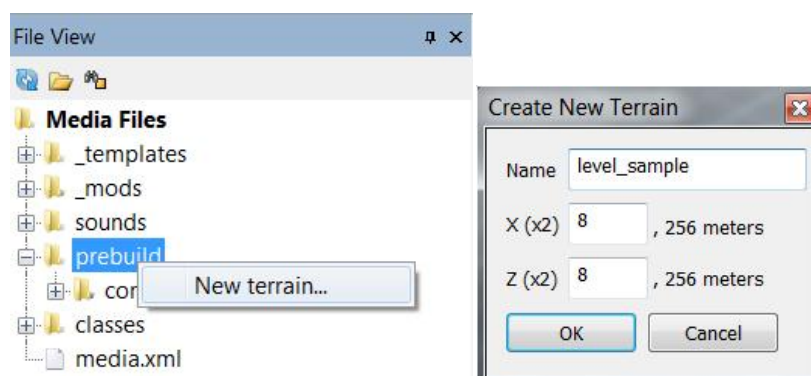
Creating a level for Spintires:MudRunner™ (Beta)

Please note: user levels are not yet fully supported in the game, we are working on it!

Step 1. Create the source level files

Complete Step 1 from section “Creating a mod for Spintires:MudRunner™”

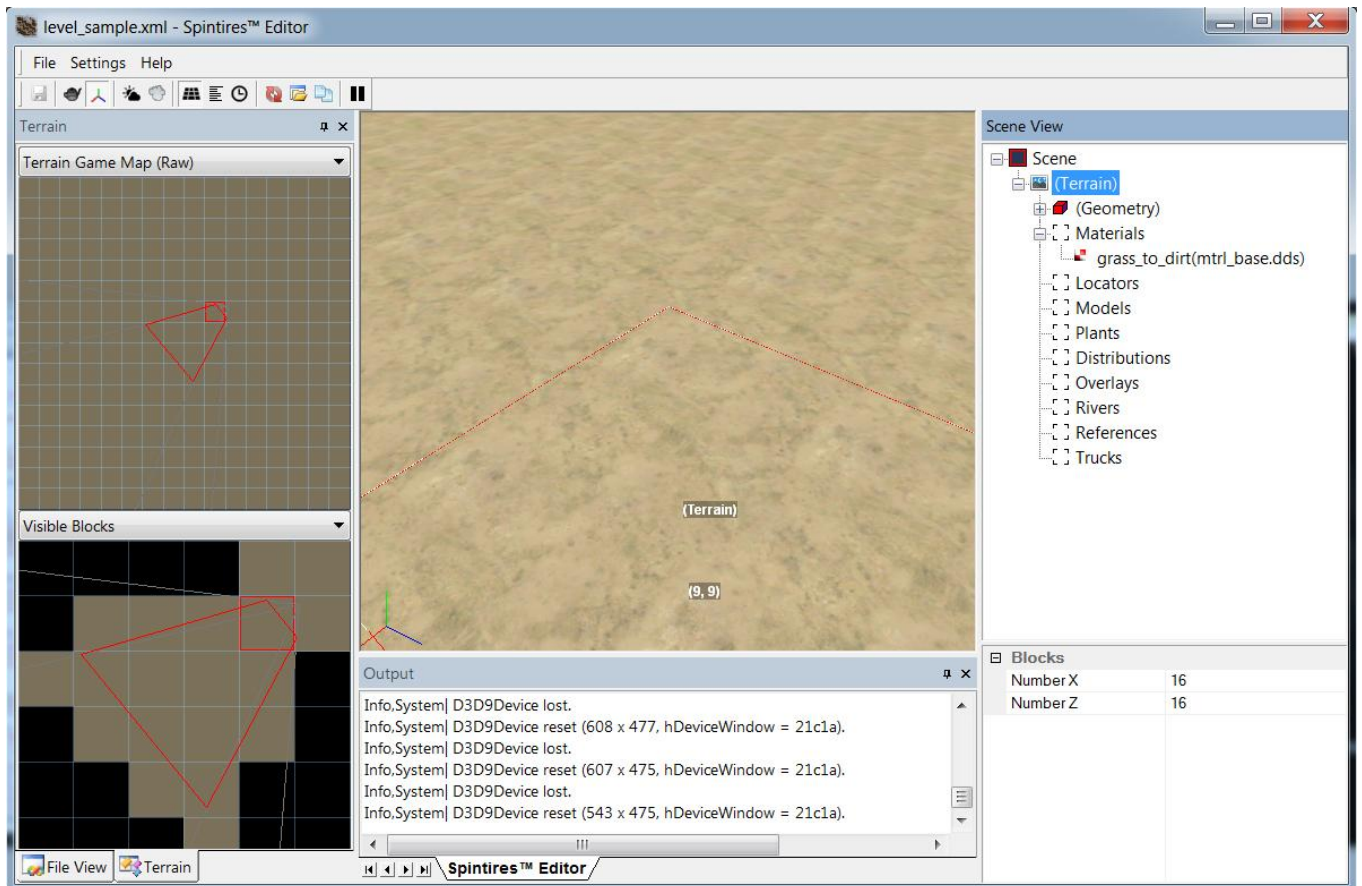
Right click “prebuild” subfolder in “File View” and select “New terrain...” in context menu.



Pick the name for your new level (it is advised to consist of alphanumeric lowercase letters and “_”). Only levels starting with “level_” will be listed and accessible in Spintires:MudRunner™ game. X(x2) and Z(x2) are half the amount of “blocks” level consists of. Each block is 16x16 meters, and there are maximum 64x64 blocks (1024x1024 meters level).

This creates an XML file within “prebuild” folder and a subfolder with the same name. This is what called level “source” files. Spintires:MudRunner™ game uses “compiled” version of the level, which is an STG file – which you will be prompted to create.

After that you are presented with a view of empty level:



Empty Spintires:MudRunner™ level.

Spintires:MudRunner™ works with XML, not STG file. So to open existing Spintires:MudRunner™ level, double click its XML in "prebuild" folder.

Basic levels concepts

Spintires:MudRunner™ level is represented by a pair of files (".stg" and ".dds" with the same name) in "Media\levels" folder. ".stg" is level data, ".dds" is a map (icon) level texture.

Spintires:MudRunner™ level is created by Spintires:MudRunner™ Editor level from "source" files located within "prebuild" folder: XML file with level name and a set of textures (maps) in subfolder with level name. Maps are ".dds" textures that can be edited by any software that support "dds" format.

After level is "built", files from "prebuild" are no longer used – and should not be shipped with the level.

Spintires:MudRunner™ level generally consists of following objects:

- Heightmap, which defines surface of the terrain. "(Geometry)" in Scene View. Tint map "(Tint)" is a texture that describes which part of terrain are soft (those would be highly by dark or yellowish color). "(Mud)" describes mud deformations of the level.
- Rivers: any number of curves (splines, a series of 3d nodes). Each curve node has a "width" attribute. Curves define location and flow direction of water. During build process, all rivers are merged into a continuous water surface. "(Water)" shows "Flow" map (speed of flow) and "Foam" (static water foam). "(Mud)" child of "(Water)" blends 2 water types (which are specified in properties of each River)
- Materials: terrain textures, and associated properties ("concrete" can't be deformed, "grass" doesn't spawn dust when truck drives, etc.). Default material would be automatically created for each level, with map called "mtrl_base.dds".
- Locators: define a game event that happens when truck drives over its area (Watchpoint, Objective, Garage, etc.)
- Models: assets like houses or bridges
- Plants: individual instances of plants. Difference between Plants and Models is that Models don't have winch points and are much slower to process (but they can be large and highly detailed)
- Distributions: a texture (map) and a set of "brushes" that place plants over terrain. Map has 3 channels, 1 (RED) is density of plants, 2 (GREEN) is scale of plants, 3 (BLUE) is auxiliary information (grid) not used by the Editor. "Brush" is one of brushes listed in "prebuild\common\brushes.xml" – it describes plants and their attributes.
- Overlays: 2d curves that describe texture objects like roads (each node have with – and there can't be more than 1 texture overlay at any given terrain block) and geometric objects like cliffs.
- References: instances of other levels can be placed, copied and rotated – which can hugely speed up production of big levels.
- Trucks: Information for the game about where and how to spawn trucks when player starts to play the map.

Note: there is no easy way to share your own models and plants right now. This functionality is not yet fully supported.

Step 2. Edit the level

A video tutorial:

<https://www.youtube.com/watch?v=T9xkejPJVpg>

Camera controls

Left click – select object

Left double click – fly to object

Right click – context menu

Mouse wheel – zoom in/out relative to object under cursor

Left mouse drag – rotate around object under cursor

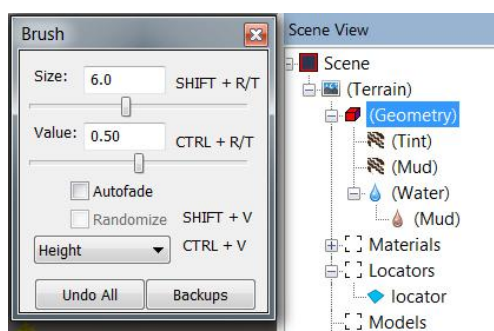
Ctrl + Left mouse drag – strafe camera

Shift + Left mouse drag – pan camera

Alt + Left mouse drag – rotate around center of selected object

You generally start by creating heightmap – by either painting it within Spintires:MudRunner™ Editor or exporting from 3rd party software like World Machine (R channel in file “_height.dds”).

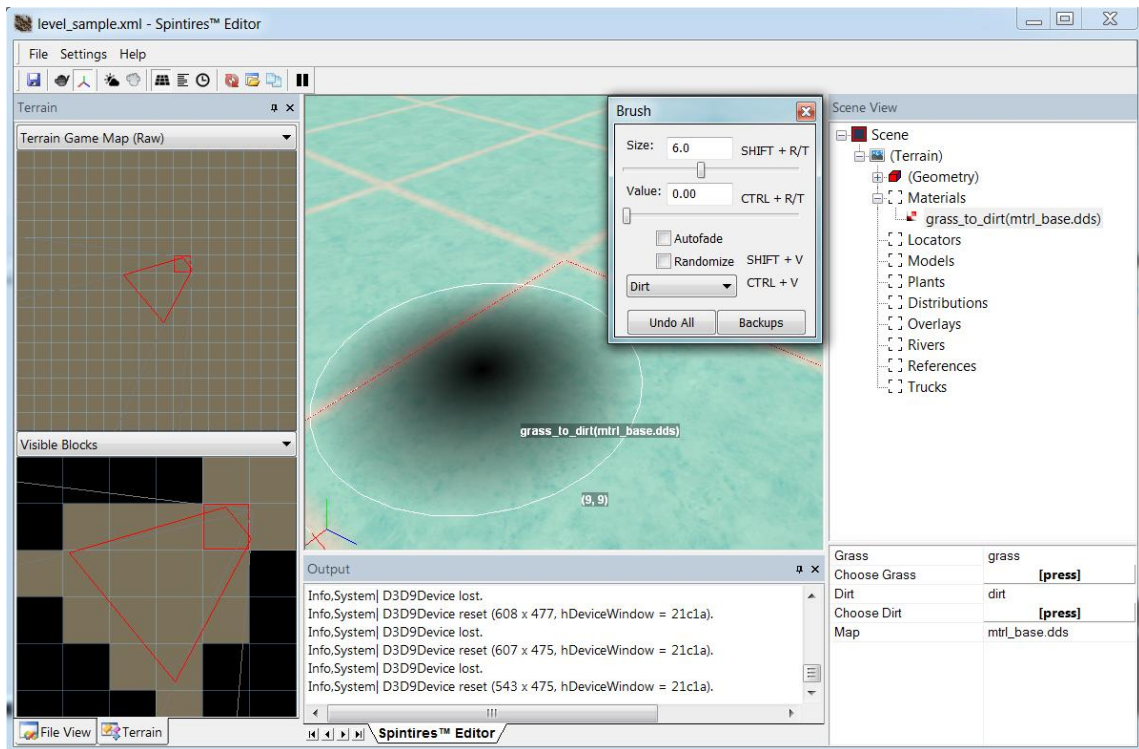
Click to select (Geometry) in Scene View to starting editing heightmap with Spintires:MudRunner™ Editor Brush:



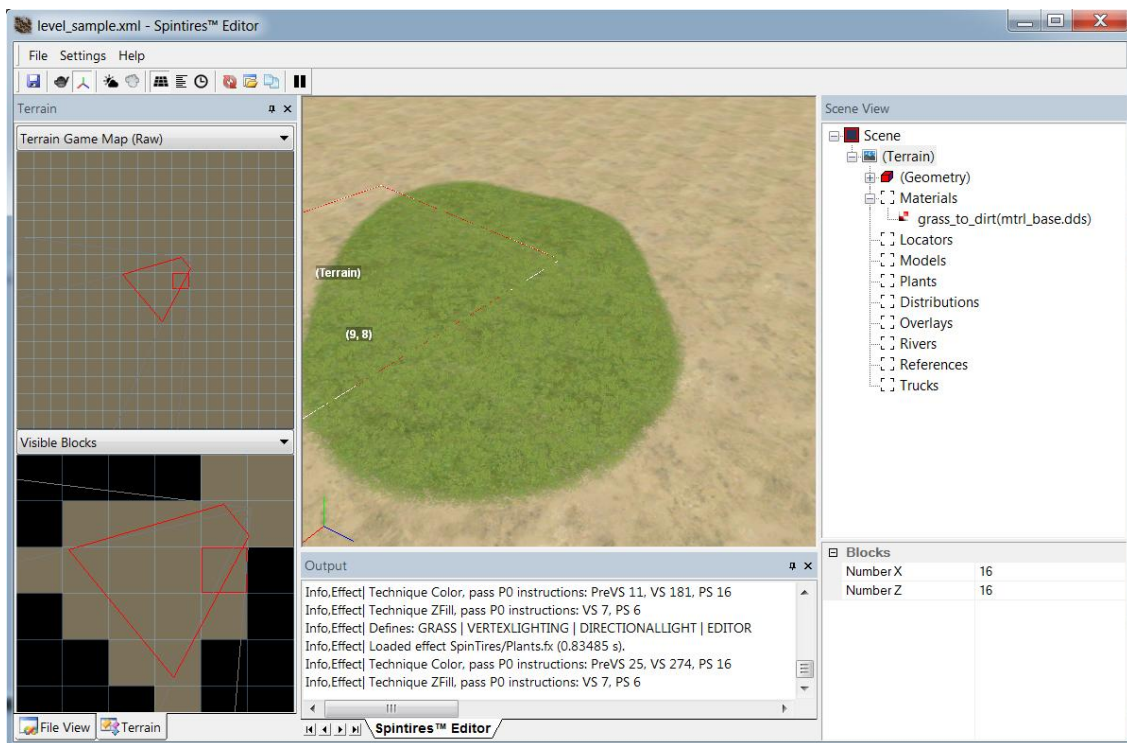
Brush dialog. To effectively use it, memorize the hotkeys. “Undo All” restores map you were editing to the state before you started the editing. “Backups” open a folder where local backups for the maps are automatically generated and stored.

Note: there is no real Undo/Redo functionality in Spintires:MudRunner™ Editor!

While Brush dialog is active, Mouse Right click paints the map instead of opening context menu. Left click anywhere on terrain (deselect “(Geometry)”) to save your changes, update terrain and exit Brush dialog.

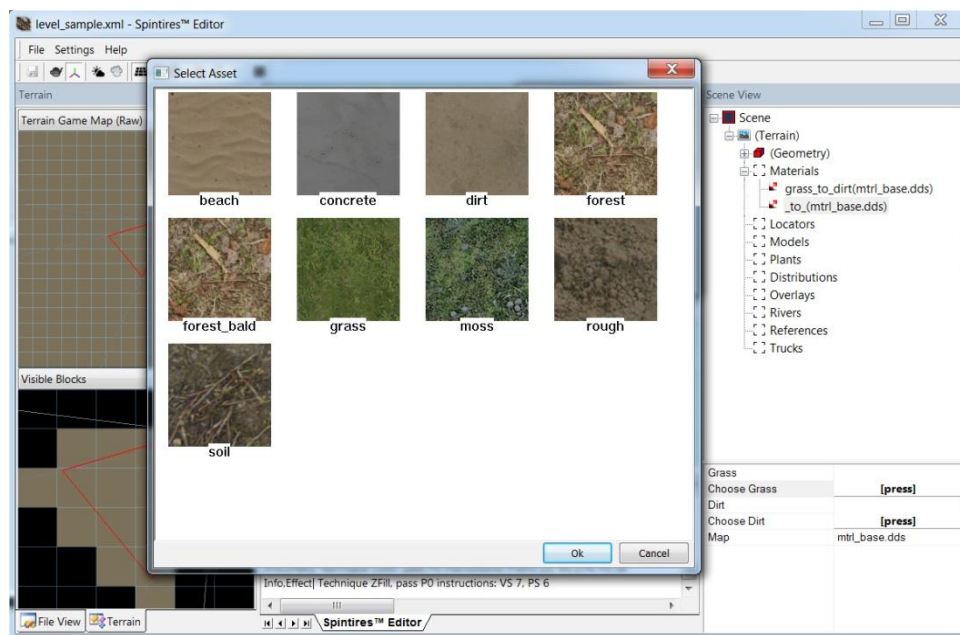


In the same way Brush paints heightmap, it also paints Material maps, Distribution maps, Tint map, Water flow/foam map and Water mud map. “Randomize” generates noise while painting. “Autofade” automatically adds trail to brush swipes that fade to zero opacity.



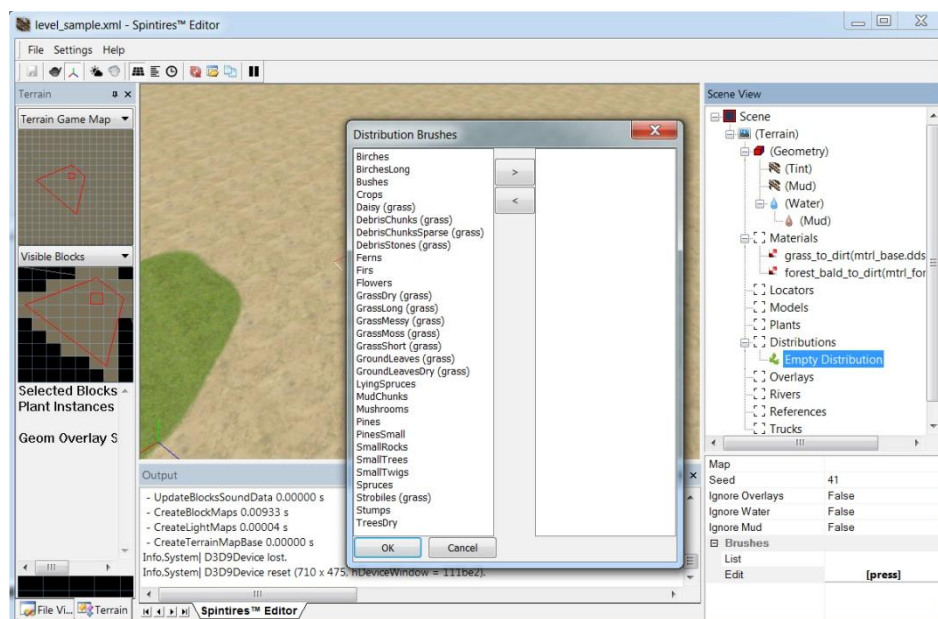
Grass material automatically places grass onto terrain.

You can pick terrain materials, models and plants with assets browsers (by clicking appropriate buttons in properties sections beneath Scene View):

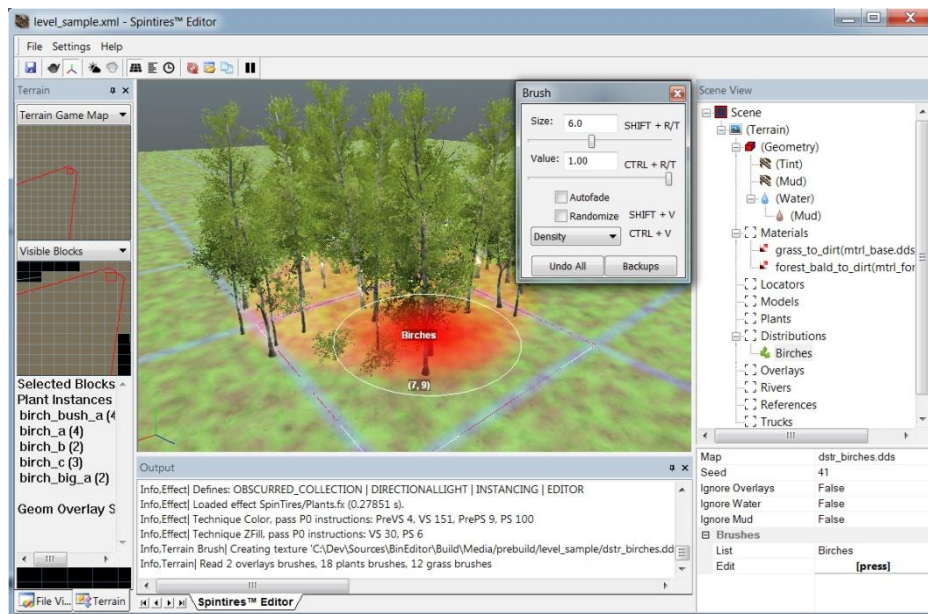


Terrain material browser.

Create a Distribution with context menu (click on terrain or “Distributions” in Scene View):

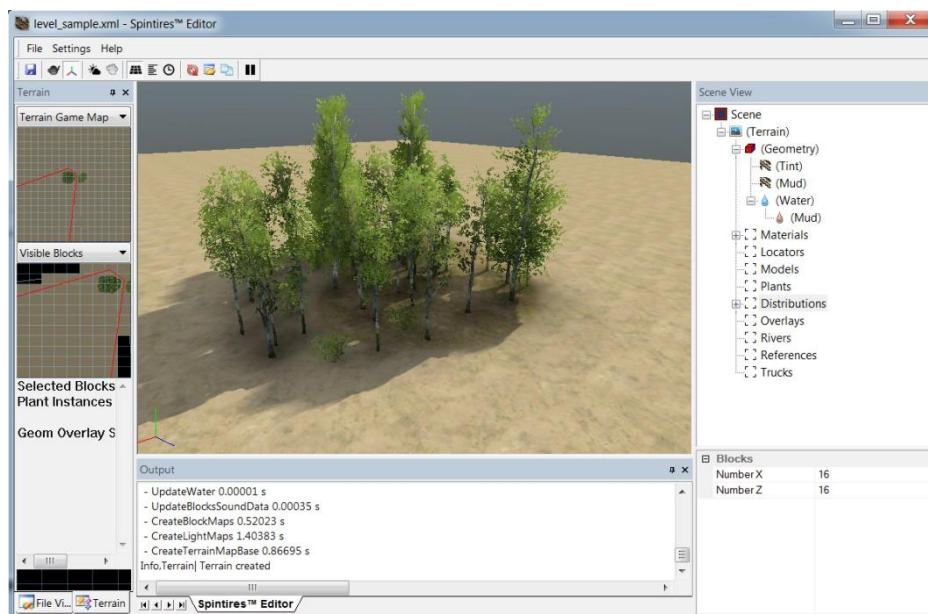


Plants distribution brush browser.



Use Brush to paint Distribution map.

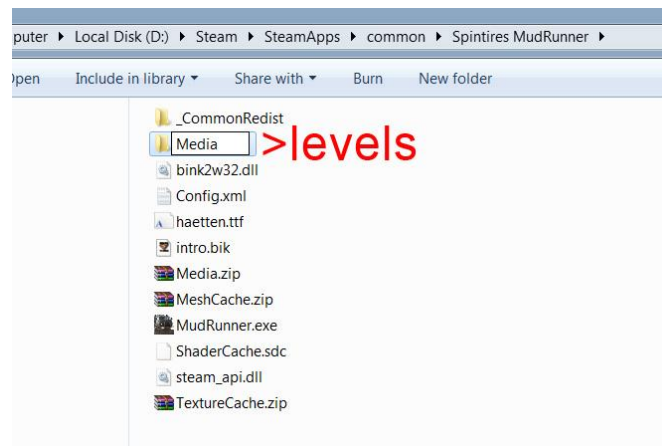
After painting is done, select “Recreate Terrain” (alternatively, select “Recreate Selection” or “Recreate Visible” so that building is faster). This is a final build step and it should be always done before shipping the level.



“Built” view of a Distribution.

Step 3. Test the level in the game

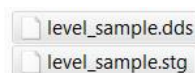
Create a folder “Media\levels” in the folder with installed Spintires:MudRunner™ game:



Close the game if its running, open Config.xml located in “C:\Users\[username]\AppData\Roaming\SpinTires MudRunner”, and add a line “<MediaPath Path=“Media” />” after <MediaPath Path=“Media.zip” />:

```
1 <Config Version="20151004">
2   <System
3     Height="1379"
4     Maximized="true"
5     Width="2560"
6     Windowed="true"
7     WorkingDirectory="C:/Dev/Sources/Bin/build_steam"
8     Adapter="0"
9   />
10  <MediaPath Path="Media.zip" />
11  <MediaPath Path="Media" />
12  <MediaPath DoPrepend="true" MountPoint="_t/" Path="TextureCache.zip" />
13  <MediaPath DoPrepend="true" MountPoint="_m/" Path="MeshCache.zip" />
14 </Config>
15
```

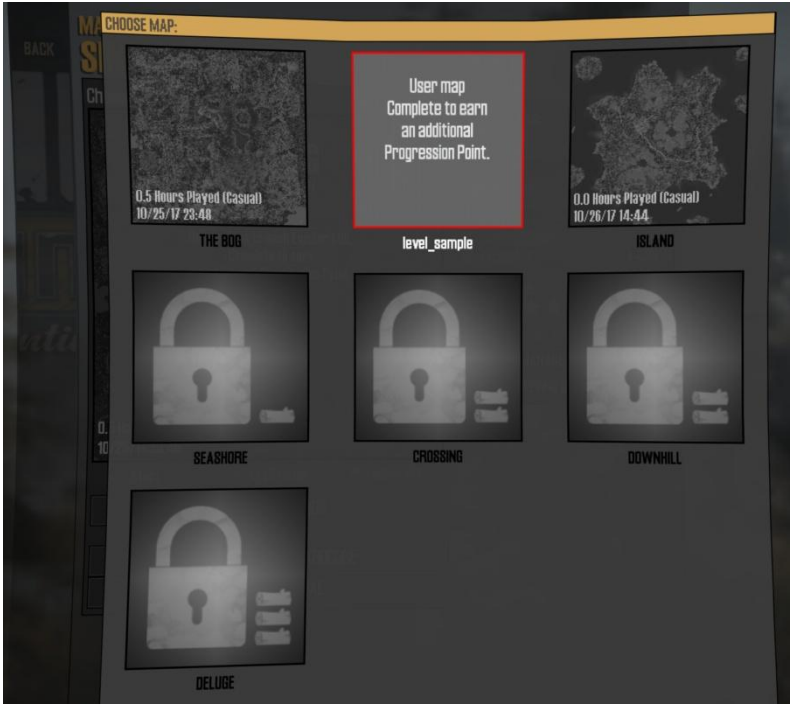
Copy “.stg” and “.dds” files from “Media\levels” folder of your Spintires:MudRunner™ Editor to a “Media\levels” game folder (that you created):



Pair of files that represent Spintires:MudRunner™ level.

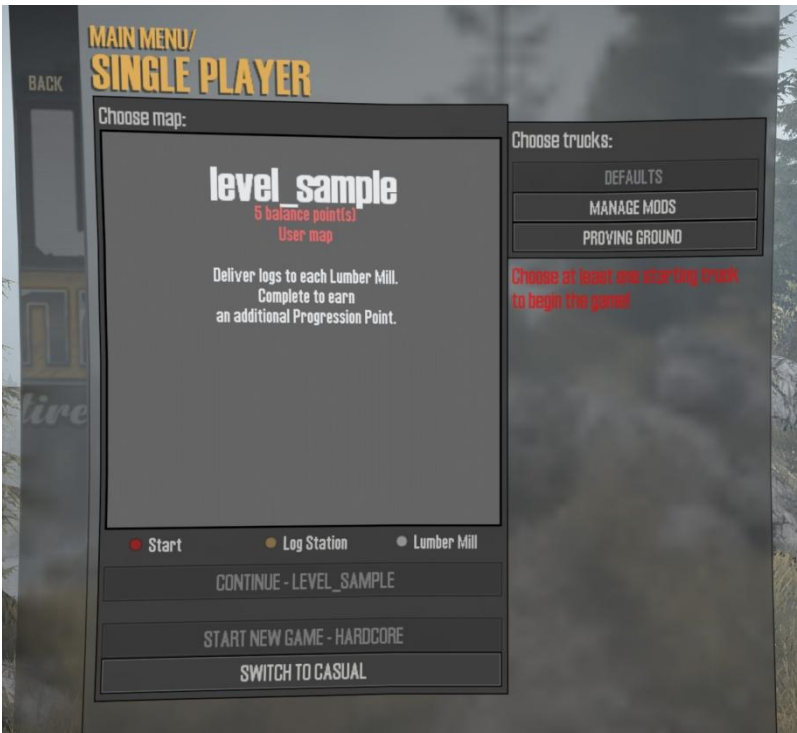
Note: you can use symlinks to avoid copying files

You should now be able to see the level in game (game doesn't need to be reloaded):



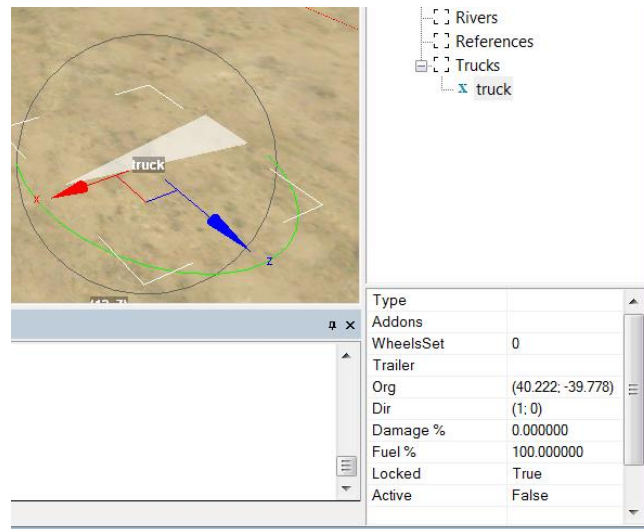
Spintires:MudRunner™ levels.

But you won't be able to play it unless you add "trucks" to the map (in Spintires:MudRunner™ Editor):



Can't start Spintires:MudRunner™ game with no trucks.

Add at least one truck:



A truck on the level. Type, Trailer and Addons are essentially filenames (without extensions). If Locked, can't be accessible in game unless player drives nearby. First among "Active" trucks is the one player starts his game with. In future, a user-friendly dialog that places trucks on the map should be part of the Editor.

A total sum of "unlocked" trucks balance points +1 = maps balance points. Any garages near first "active" truck (the one player starts with) are automatically unlocked.

Note: levels can't be uploaded and shared with Steam® Workshop yet.

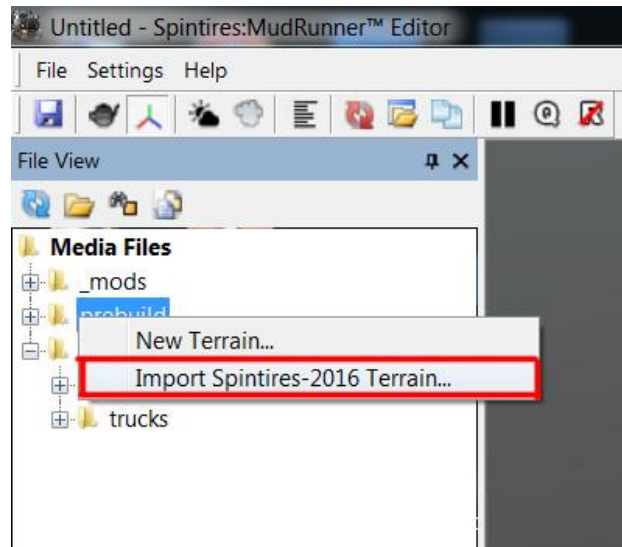
Examples of working in earlier Spintires:MudRunner™ Editor versions which are still relevant:

<https://www.youtube.com/watch?v=i3zoeHkDgiM>

https://www.youtube.com/watch?v=_zb0hQetax0

Appendix A. Importing Spintires-2016 terrain

Right click the “_prebuild” folder and select “Import Spintires-2016 Terrain...”



Browse for the source terrain files. A new terrain would be created with its name ending with “_imported2016”. You can then rebuild terrain to create .STG and .DDS level files.