BRIAGE User Manual

- Now in 4D! -

Documentation - EN - v3.20.835

Table of contents

Export

Export mode

Visible children

Hierarchy processing

Export skinned geometry

Verbose

Main Object

Main object center

Main object rotate

Texture path mapping

Textures directory

Texture strings to replace

Settings application order

Custom keywords

Bind / Unbind

arg1

Scroll UVs

Animate UVs

ZBufferMode

Transparency

Z Offset

Visible distance

MipLODBias

FilterMode

Viewfacing

BackFaceCull

Two sided

Unlit

Pre-DPP

Cast shadows

1 Foreword

1.1 Thanks

Thanks a lot to:

- Dom107
- JADsHome
- Karma99(Victory Works)
- Mike Adams
- Hugo_B
- Jáchym Hurtík

for their remarks, tests and help around the exporter!

All of this wouldn't have been possible without Dom107's exporter anyway.

1.2 Personal thoughts about the exporter

This new exporter is being worked on since November 2015. I am very proud of the result and having been part of the great adventure of this exporter which contributes to the successful DTG's Train Simulator 20xx series (I'm not affiliated to Dovetail Games in any ways).
I still don't know why anybody made this before, but I'm happy it's here for you now.

I began working on this because using the old exporter was too extensive in time and resources compared to the results. I also wanted it to be more ergonomic for newcomers to 3D and complex TS asset authoring.
I learnt Python on-the-job for this. After 2 months I got something more useful through blender's custom properties, this was the V1.5 . More internal versions came by while I was getting better and hence quicker with Python. More and more features got implemented too until it may finally be named a “V2.0”. Some corrections and hundreds of revisions later, there was finally something with most of the features I wanted and a stable core. That's where few of you began hearing of it on *UKTS* or *RailSim-fr.com* some talented beta-testers applied and two months later quite everything was finished thanks to their remarks!
Thanks to them you can now use a mostly bug-free exporter! (according to test's extrapolations....)

Why so late? I wanted it be the best I can give to the community. I typed this thing alone in my bedroom all day and everyday for months, and used it as an excuse to cast aside as much people as I could to stay alone in my little PC world (which was very effective). And finally, this was really hard sometimes. You don't believe me? -Ask my father or try yourself to develop something based on someone else work in a language you don't understand!-

Fun fact, I always thought I was alone in this development ship, but in facts, there where lots of people supporting me. Sparse PMs or half-weekly phone calls: everything and everyone weighs in the scale! They don't belong in the “special thanks” section but they will recognize themselves while reading this (if they ever read it... hahaha). This was for me a rewarding programming and typing experience as much as an enriching human experience in the end.

My lasts words about this are that I gladly wish this software will be as useful to you as it was to me.

1.3 Where does that name comes from?!

BRIAGE is an acronym for ***B**lende**R** **I**ntermediate **A**nimation and **G**eometry **E**xporter* which is the full name of this software.

It's also an old French word which designates the last step of work where a dough or pastry is homogenized and packed using a wooden bar or a mechanical device. I found this to be an appropriate enough fact for that name to be kept compared with others I came up with so far.

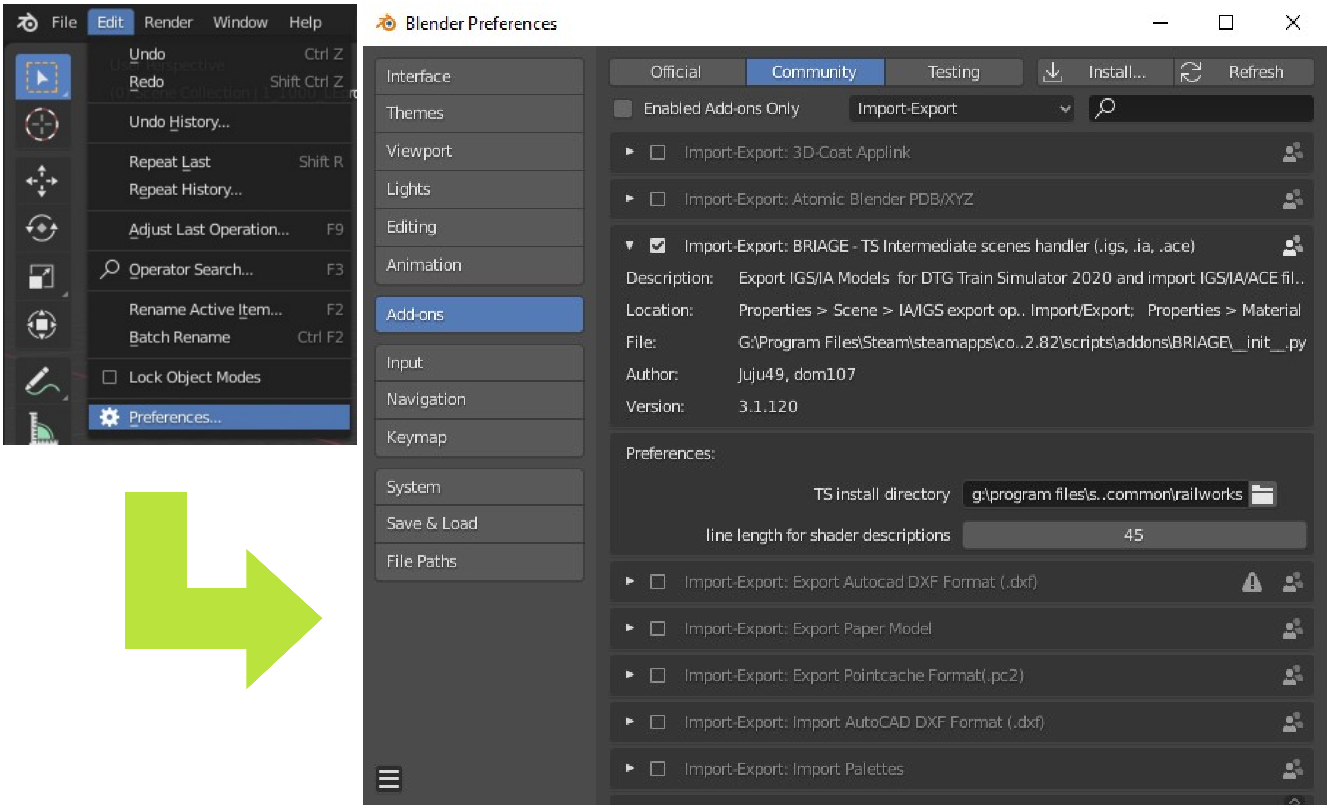
1.4 Why did it took so long?

Life happened, and it will happen again! But I always end up writing something in here in the end...

2 Installation

2.1 Install files

1. Select your installation method:
   - With Windows Explorer:
     1. Install BRIAGE Add-On by simply copying "`BRIAGE`" folder in Blender "scripts\addons" sub-directory (for example: `D:\Program Files\Blender Foundation\Blender\2.82\scripts\addons`).
     2. Start Blender to open the default Blender file.
     3. Open Files > User Preferences...
     4. go to Add-ons tab.
     5. Look for "BRIAGE - TS Intermediate scenes handler (.igs, .ia, .ace)".
        It is advisable to filter add-ons for this purpose. Select:
        - Supported Level: "Community"
        - Categories: "Import-Export"
   - With Blender Filesystem explorer
     1. Start Blender to open the default Blender file.
     2. Open Edit > Preferences...
     3. go to Add-ons tab.
     4. Click Install from file button. A file browser opens.
     5. Reach "`BRIAGE.zip`" file and confirm.
2. The installed Add-On must be enabled before it can be used. Simply place a check mark in the Enable Add-On box . The new export functionalities are now integrated into Blender.
3. Display add-on informations by clicking the white triangle near the checkbox.



☑ Import-Export: BRIAGE - TS Intermediate scenes handler (.igs, .ia, .ace)

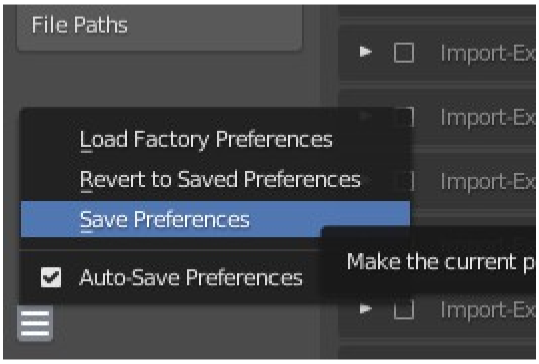
Description: Export IGS/IA Models for DTG Train Simulator 2020 and import IGS/IA/ACE fil..
Location: Properties > Scene > IA/IGS export op.. Import/Export; Properties > Material
File: G:\Program Files\Steam\steamapps\co..2.82\scripts\addons\BRIAGE\__init__.py
Author: Juju49, dom107
Version: 3.1.120

Preferences:
TS install directory g:\program files\s..common\railworks
line length for shader descriptions 45

4. Specify TS main "`\Railworks`" folder path (where TS executable file is located) if it is not already populated automatically. This is used to reach shader information.
   If not set, it will try to regenerate it on each load from system and steam informations. However this last technique might fail in some cases and an error will be displayed in material IGS settings shader description box.

5. Specify a line length in characters per line for shader description box text if necessary. This option allows you to make sure you can read shader descriptions properly from Blender UI.

6. Click on "Save Preferences".



2.2 Uninstall files

1. Start Blender to open the default Blender file.

2. Open Edit > Preferences...

3. go to Add-ons tab.

4. Look for "BRIAGE - TS Intermediate scenes handler (.igs, .ia, .ace)".
   It is advisable to filter add-ons for this purpose. Select:
   - Supported Level: "Community"
   - Categories: "Import-Export"

5. Simply uncheck its Enable Add-On box. Add-on's functionalities are now withdrawn from Blender.

6. Select the same as your installation method:

   - With Windows Explorer:
     1. Delete "`BRIAGE`" folder in Blender "scripts\addons" sub-directory (for example: `D:\Program Files\Blender Foundation\Blender\2.82\scripts\addons\BRIAGE`).
   - With Blender Filesystem explorer
     1. Display add-on informations by clicking the white arrow near the checkbox.
     2. Click "remove" button

3 Presentation

3.1 Contents

Exporter side is made of 2 main parts:

- IGSE module: IGS format export (model structure)
- IAE module: IA format export (animation)

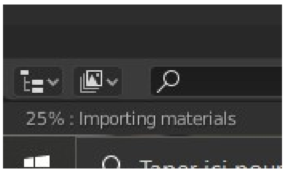
Importer side is made of 2 main parts too:

- IGSI module: IGS format import (model structure) and SACE format import (old textures)
- IAI module: IA format import (animation)

You, the user, influence its behaviours through Blender's interfaces mainly located in the properties area:

- **Material tab panels (see 6)**
- **Scene tab panels (see 4)**
- User Preferences Add-ons tab panel (see 2)

- Status bar on long tasks displays progress informations



25% : Importing materials

Example files are provided in the package for you to study.

3.2 What is processed from Blender?

✔ Objects
+positions
+rotations
+main object colour

✔ Meshes
+materials
+vertices
+faces
+custom normals
+active Vertex colours (unlike in Blender only one colour is saved per vertex), see 8.7.3
+Vertex groups weights (if linked to bones through armature modifier), see 8.3

✔ Bones
+positions
+rotations
+deform bindings up to 4 bone influences per vertex

✔ Meta objects, Curves (if geometry is found), Texts and NURBS can be exported too as regular objects if they have 1 material assigned. They are auto-converted during export to regular objects and your scene stays untouched.

✔ All dupli and instances counterparts of the preceding statements

✔ Animations of positions and rotation (deformations with bones)

3.3 What is processed to Blender?

- ✔ Objects
  +positions
  +rotations
- ✔ Meshes
  +materials
  +vertices
  +faces
  +custom normals
  +active Vertex colours (unlike in Blender only one colour is saved per vertex), see 8.7.3
  +Vertex groups weights (if linked to bones through armature modifier), see 8.3
- ✔ Bones
  +positions
  +rotations
  +deform bindings
- ✔ Animations of positions and rotation (deformations with bones)
- ✔ ACE textures top MIP is packed inside as png

3.4 Usual workflow

The principle to create an exportable model is as follows:

Name objects

An object must follow the TS20xx strict naming rules for LODs (see in 11.3).

If an object doesn't follow these rules, some processing is made to try to set the name as required by the game (see in 11.3).

Texture meshes

A material usually needs several texture slots, their content depends on the shader used (see in 7.1, 11.1, 11.2).

Adjust some pivots

Outside specific TS20xx or animations requirements, it is advisable to centre the pivot of each object on its geometry.

Among the specific requirements of TS20xx, the pivot of the main body of a car or locomotive must be centred with z = 0 at the rail contact (See example in 4.7). The bogie pivot must coincide with its rotation axis and the bogies naming convention must be followed with bo01 as the front bogie and bo02 as the back bogie, in the case of a rolling stock with 2 bogies.

Setup material attributes in the Blender material tab

In Blender, you can select shader names in a dropping menu, use custom ones or even use the old still supported way through a material's first texture slot name. Theses two last fields support entering short names or full shader names (See 7.1, 11.1).

Note that a specific processing is performed for `TrainGlassWeatherEffects.fx` to make it more convenient to use (see 8.7). Also some automatic adjustments are done depending on the shader used.
Material attributes are initialized in Blender (see 4, 6).

Create animations (See in 5, 8.5)

Setup the behaviour of the exporter (see in 4, 6, 7)

The exporter takes the modifiers into account without applying them to the model. Objects can also have some scaling and rotation (see some restrictions in 8.4 : it could be necessary to use CTRL+A to apply these transformations). Ngons can also be used.

As a default behaviour, several Blender objects are merged into one object when exported, if they all have the same LOD value and distance. This can be cancelled or altered, take a look here 4.2 and here 4.7 for more details.

Main object field (see in 4.4) sets the main object of the model, for example the body of a locomotive.

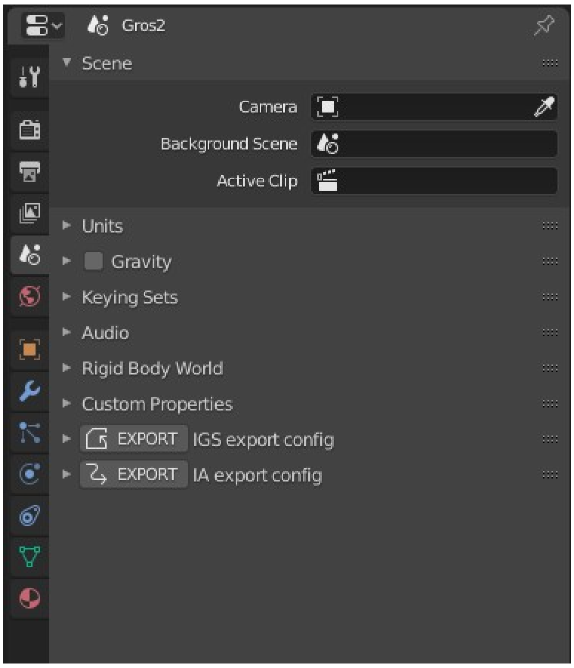
There are 2 modes of export for IGS or IA: the entire model (whether objects are selected or not in Blender views) or selected objects (see 4.2). Main object is set to active object by default.

Annexe 11.4 provides some information about using dynamic representation of character strings (automatic numbering).

Check log or Blender's system console for errors, see 8.4.2 - Reading logs

4 Configuration for IGS Export (IGSE)

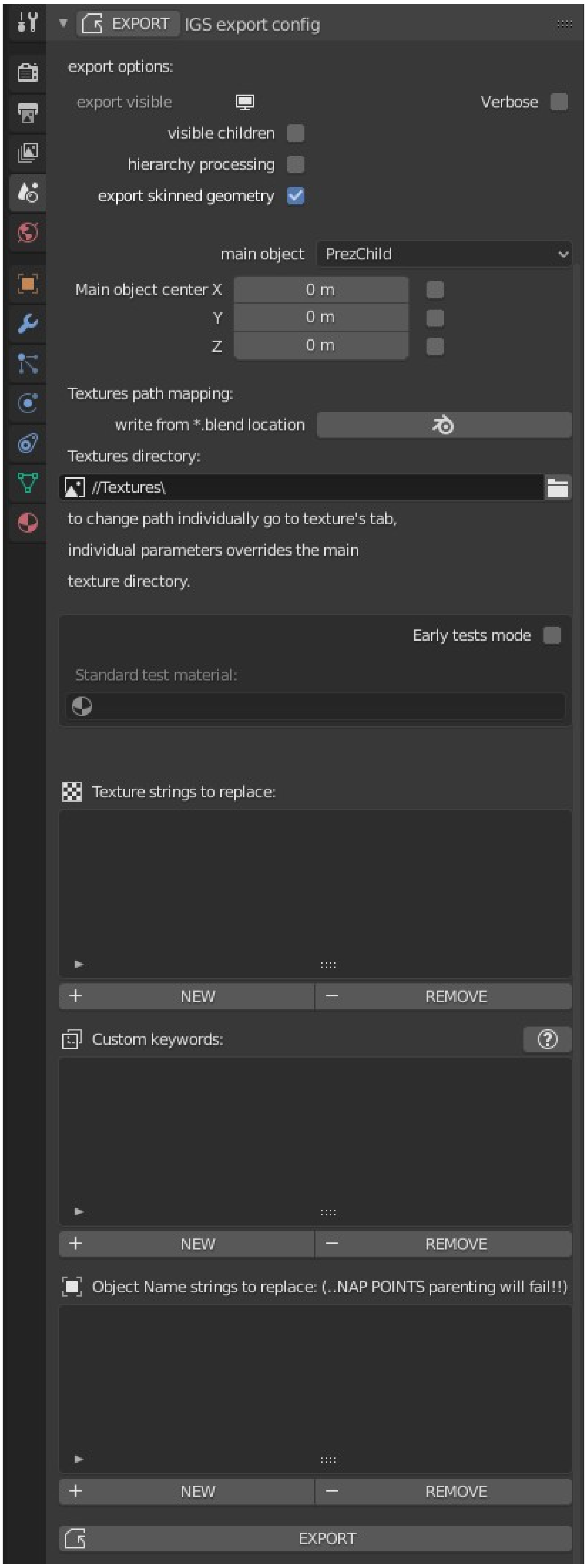
The following paragraphs explain how to configure IGSE (IGS Exporter) main options through the Properties panels:



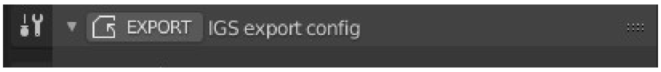
<u>These properties are scene-dependent.</u> Different Blender scene will not share configurations implicitly. This allows the exporter to be more flexible and *.blend files might contain different objects without parameters interfering with each other.

You can store a whole train-set in a blend, sharing materials betweens cars for easy editing, share bogies and doors while keeping each car in its own scene not overlapping with other train parts and each car has its own IGS Export config.

It is ***<u>highly discouraged to use and export object which don't have their scales applied</u>***. Especially if it is a negative scale or a LOD number higher than 1. It is indeed more prone to provoke unexpected results. If used in conjunction with center main object parameters!



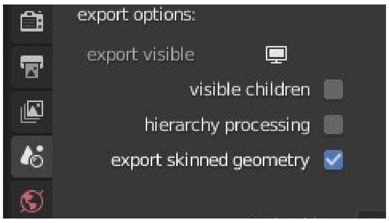
4.1 Main settings



The top Export button allows exporting to IGS format.

Export paths are saved by BRIAGE on export for you to find your way quickly on next export.

4.2 Export mode and other important behaviours



visible children ☐
hierarchy processing ☐
export skinned geometry ☑

The first top left box sets the `Export mode` to export selected or export visible. They are self-explanatory.
“Export selected”: only selected objects are exported with all of their children (selected or not, visible or not).
“Export visible”: only visible objects (selected of not) are exported with all of their children (visible or not).

Visible children modifies children exportation rules. When this setting is enabled, only visible children are exported whatever `Export mode` is selected.
In the *Outliner*, the eye icon in front each object is not greyed out for a visible object.

Hierarchy processing allows the exporter to merge objects together following their LOD's distance and number as well as their matching keywords.
For small models, it is advisable to deactivate this option.

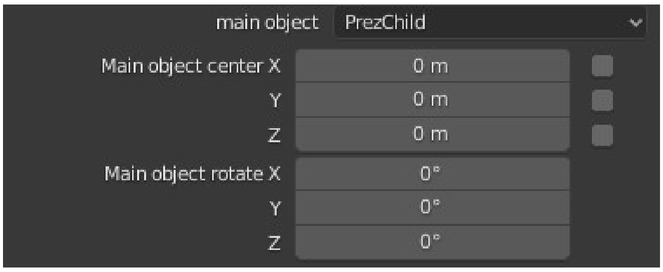
Export skinned geometry enables the export of bones and corresponding Vertex groups weights. See 8.3 - Using skinning and bones for informations.

4.3 Other export toggles

Verbose ☐

Verbose toggle makes the exporter print a full log *.igs.log file alongside your *.igs file instead of writing a short log to Blender's system console. Beware, this option increases export time by a factor of 5. Use it only when something looks wrong and you want to take a look at exported data.

4.4 Main Object settings



main object PrezChild
Main object center X 0 m ☐
Y 0 m ☐
Z 0 m ☐
Main object rotate X 0°
Y 0°
Z 0°

Main Object dynamic enumerator tells to select the pointed object as the one which will be used to merge other objects at 'scene level' hierarchy of the Blender model.
If "active object" is specified, the first mesh object found while scanning list of exportable Blender objects is chosen. Active object is the yellow highlighted object among a selection.

IGSE log indicates used main object:

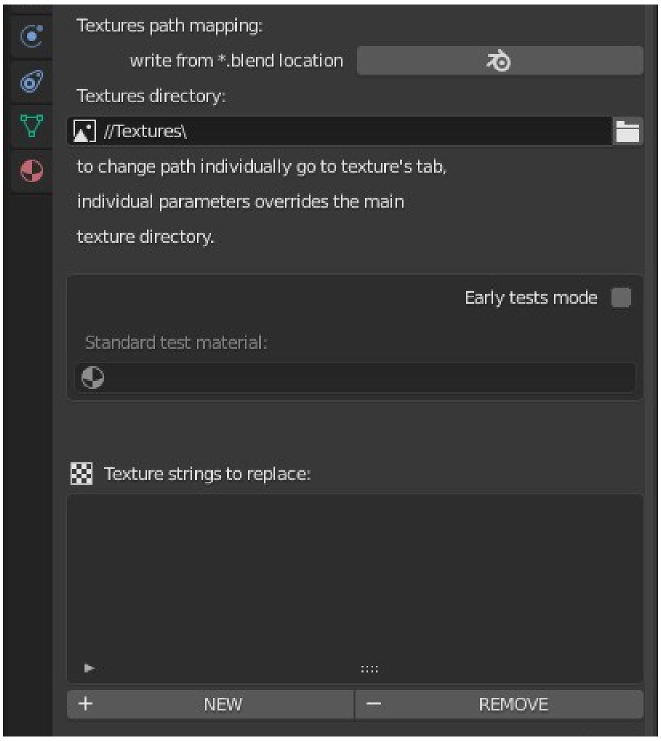
```
BRIAGE:: 'PROPERTY': MainObject to find = 1_1000_BiDuLe
BRIAGE:: 'PROPERTY':  MainObject found = 1_1000_BiDuLe
```

Main object center is a set of 3D coordinates with matching checkboxes. This is used to centre main object origin at these coordinates (and move children accordingly) if the corresponding checkbox is enabled. This avoids moving objects in Blender, for example, to centre a locomotive along the X axis to ensure that it is centred on the rails.
Centring along an axis gives the same result as moving the entire model in Blender so that the pivot of the Main Object object is at coordinate 0 of this axis.

Main object rotate allows you to rotate the whole scene. The scene is first rotated around its center then moved to the place defined by Main object center.

As such a move impacts animations, the IGS file ***settings of "Main object center" , "Main object rotate" and "Main Object" are also read when exporting IA files.***

4.5 Texture manipulation tools



Textures path mapping:
write from *.blend location
Textures directory:
//Textures\
to change path individually go to texture's tab, individual parameters overrides the main texture directory.
Early tests mode ☐
Standard test material:
Texture strings to replace:

Texture path mapping mode sets how the exporter binds IGS file to texture files.

- "write from *.IGS destination" mode: wherever you export your IGS, its texture components will point to the files used/specified in Blender.
This will be referred later as "**IGS texture mode**"
- "write from *.blend location" mode: *the same relative <u>filepath from blender</u> will be applied to IGS blindly*.
For this setting to work properly, work in your source folder sub-directories where you will export your IGS to or make sure your working directories mimic exactly end directories hierarchy.
This will be referred later as "**BLEND texture mode**"

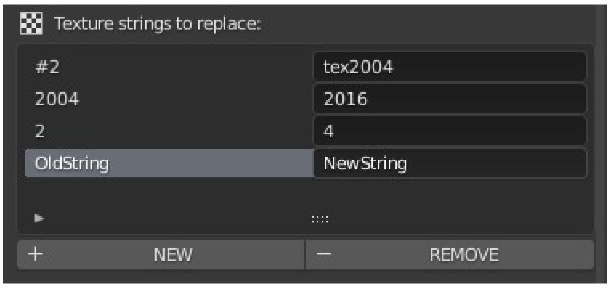
Textures directory field allows you to set a directory where all your model's textures will be stored directory can be child, sibling, uncle or wherever you want. By default every texture filepath is rerouted to this folder, you can fine-tune this behaviour with individual texture settings, see 7.3.

Texture strings to replace is a table where you enter former (double-click to edit) and new strings to be found and replaced by in texture names.

It allows replacing each listed string by another string, <u>for all the texture files</u>, after `"texture name replacement mode"` has been processed. This is "*find all*" mode.

Adding "**<u>#</u>**" at the beginning of **former string field** toggles "**texture name replacement" mode**. Each filename listed as such will be replaced by another name. The file suffix (dds, ace, bmp, etc...) must not be specified.

Right-hand side example :
Texture file `"2"` is changed to `"tex2004"`,
then `"2004"` is replaced by `"2016"` ,
then "2" are translated to "4",
finally it is exported as "`tex4016`".



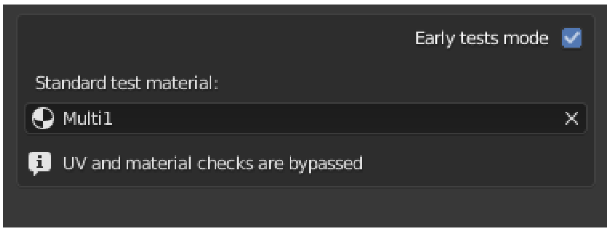
Texture strings to replace:

#2	tex2004
2004	2016
2	4
OldString	NewString

Settings application order:

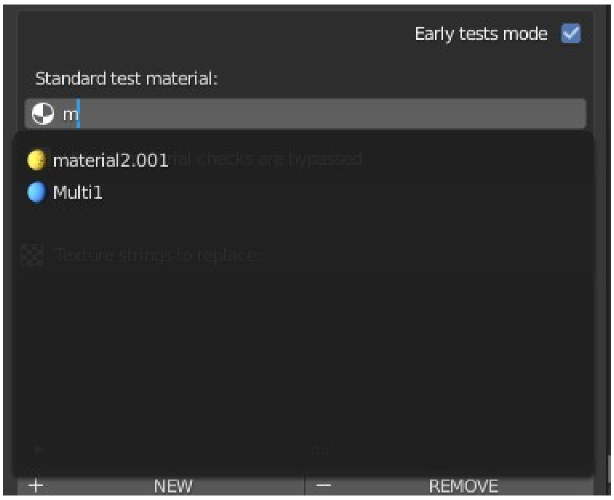
Texture strings to replace ("*replacement mode*" then "*find all mode*") => Cancel 'Textures directory' setting ELSE Export texture path ELSE Texture directory => Texture path mapping

4.6 Early tests mode


Early tests mode ☑

Standard test material: Multi1

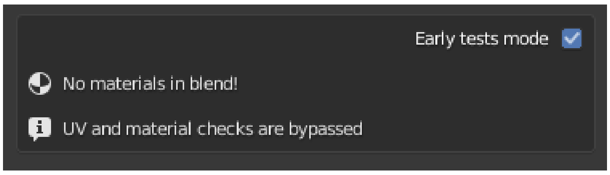
UV and material checks are bypassed


Early tests mode ☑

Standard test material: m

This check box enables BRIAGE to circumvent errors which may arise from exporting your models. Could it be missing or faulty materials or missing UVs , it will try to export everything it can when the checkbox is ticked.

It will nonetheless need a game-ready material in "Standard test material:" field to work. You can type in this field to narrow search results while scrolling the list.


Early tests mode ☑

No materials in blend!

UV and material checks are bypassed

This is displayed when no material is found

4.7 Keywords merging system

As a default behaviour, the Blender model hierarchy is kept only when objects include TS20xx keywords or when a LOD change is found. As much as possible, several Blender objects are merged into one object when exported, if they all have the same LOD value and distance. This is a recommended setting to diminish performance issues linked with a high number of direct3D “draw calls” in TS.

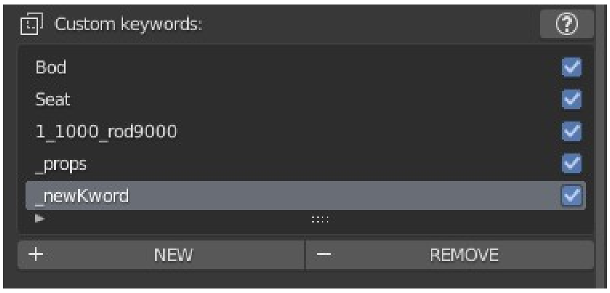
To merge objects, the exporter looks for the following built-in keywords. ('$' represents a digit) :

```
'_day'
'_night'
'_locomotive'
'_tender'
'_coach'
'_vehicle'
'_wagon'
'_carriage'
'_coal'
'_fuel_level_'
'_freight'
'_bulk'
'_lights_fwdhead'
'_lights_revhead'
'_lights_fwdtail'
'_lights_revtail'
'_door$$'
'_step$$'
'_wh$$'
'_bo$$'
'_bo$$wh$$'
'_panto$$'
'_wiper$$'
'_primarydigits_$$'
```

This limits the effect of TS20xx maximum of 256 exported objects and igs format limitation of a maximum of 24 children. However, as explained below, options are available to strictly follow the hierarchy and to ensure that specific sets of objects are merged together or not.

The Custom keywords table allows control of object merging by user-defined keywords when “Hierarchy processing” is activated.

“?” provides the list of built-in keywords which are checked anyway after the custom list.


Custom keywords:

Bod ☑
Seat ☑
1_1000_rod9000 ☑
_props ☑
_newKword ☑

Example of use

Each entry is composed of a *text field* (double-click to modify) for writing in the keyword and a *checkbox* to enable its use on next export.

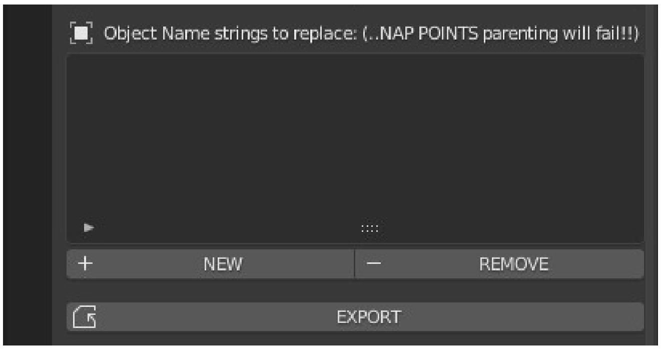
You can specify a full name or a string common to multiple objects. They define different <u>groups of objects to merge separately</u>: those with string `Bod` in the name, those with string `_props` in the name and those with those with string `_newKword` in the name (this is default field's value).

Note that the checkbox in front of `Seat` is unticked, therefore this setting will not be used on next export until enabled again.

If “Hierarchy processing” is enabled and you do not want an animated object to be merged with other objects, unless they move with the same motion, just add its full name to the list. Thus, `1_1000_rod9000` present in Custom Keywords table will be <u>a single object</u>: It will not be merged with any other object.

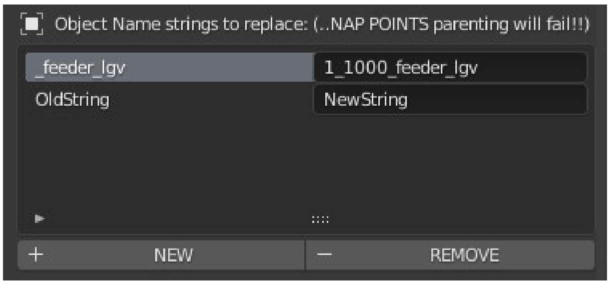
An object named with a TS20XX keyword as listed in above blue box, (for example `1_0100_bo01`) doesn’t need to be in Custom Keywords table.

4.8 Object rename


Object Name strings to replace: (..NAP POINTS parenting will fail!!)

Object Name strings to replace table replaces part of an object name by right field value if left field value is found inside object name.

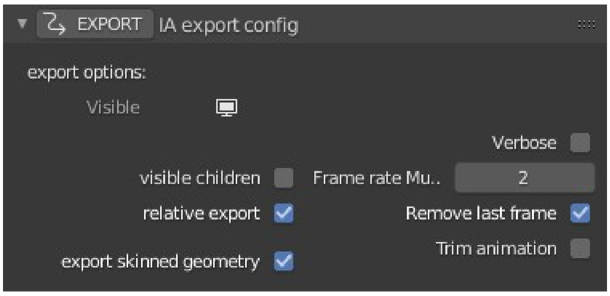
This feature was elaborated to be able to make complete sets of objects inside a blend file which shares alike objects which can't be named the same in blender (Ex: for dynamic gantries wire nodes: allows easier use of Asset Editor later in development process). You can use this to truncate the ".002" produced by Blender on the fly. This assumes theses alike objects do not share the same IGS file where homonymy could cause various or serious problems.


Object Name strings to replace: (..NAP POINTS parenting will fail!!)

_feeder_lgv	1_1000_feeder_lgv
OldString	NewString

Here, every object including “_feeder_lgv” will have this part replaced by “1_1000_feeder_lgv”. This is a very bad idea: it will indeed corrupt your igs names mostly...

5 Configuration for IA Export (IAE)



EXPORT IA export config

export options:

Visible

Verbose ☐

visible children ☐ Frame rate Mu.. 2

relative export ☑ Remove last frame ☑

Trim animation ☐

export skinned geometry ☑

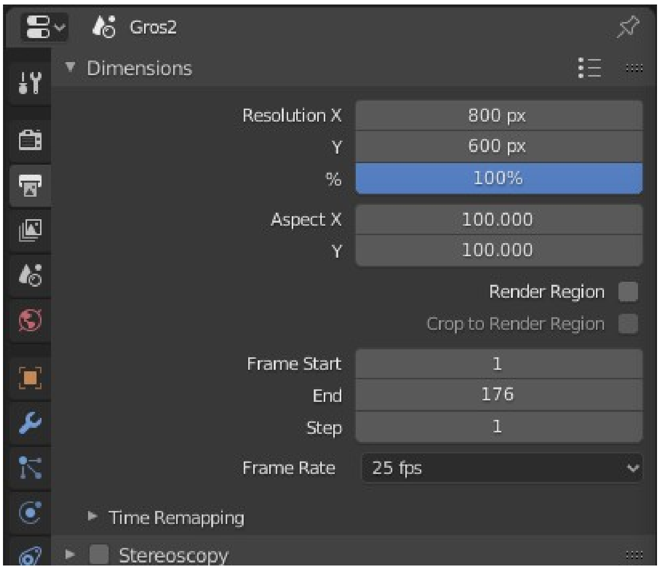
IA export config is located below IGS export config in Scene tab.

In addition to the possible use of Main object center from IGSE configuration (see 4.4), there is an optional configuration panel with two exclusive parameters: Frame rate Multiplier and Remove last frame. Other settings are stolen from IGSE; refer to it for further details: 4.1, 4.2, 4.3 .

See more here: 8.5 - About animation

5.1 Frames per second (fps) setting

The frames per second (fps) setting can be set in the Output properties menu:



Gros2

Dimensions

Resolution X 800 px

Y 600 px

% 100%

Aspect X 100.000

Y 100.000

Render Region ☐

Crop to Render Region ☐

Frame Start 1

End 176

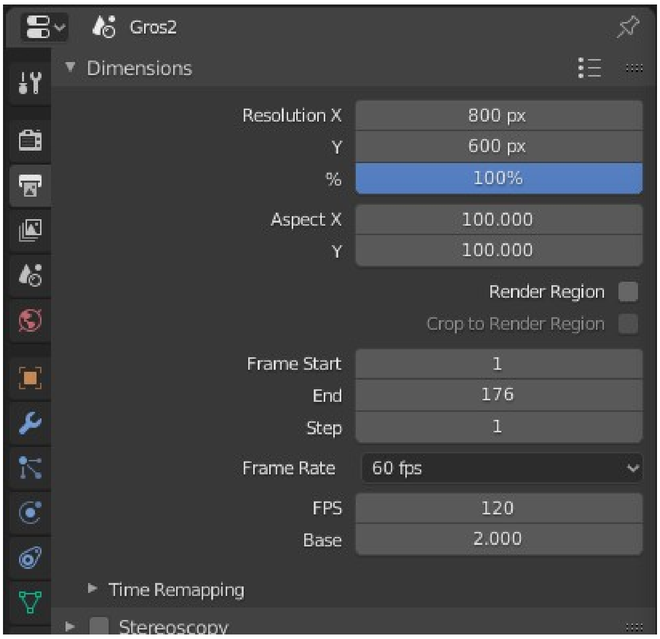
Step 1

Frame Rate 25 fps

Time Remapping

☐ Stereoscopy

You can use the Frame Rate enumerator to select a common fps setting



Gros2

Dimensions

Resolution X 800 px

Y 600 px

% 100%

Aspect X 100.000

Y 100.000

Render Region ☐

Crop to Render Region ☐

Frame Start 1

End 176

Step 1

Frame Rate 60 fps

FPS 120

Base 2.000

Time Remapping

☐ Stereoscopy

You may alternatively use the FPS & Base couple. Here IAE will perform the operation 120 / 2 = 60 fps. The Base can be less than 1

5.2 Frame rate Multiplier

It allows having a frame rate higher that the fps value using parameter Frame rate Multiplier. The frame rate is computed as "Frame rate Multiplier * `fps value`" (see 5.1 for fps setting). The frame rate multiplier is limited to 10. If the computed frame rate is higher than `10`, a message is displayed in the log file and Frame rate Multiplier is reset to `1` for the export.

Setting a correct value for the multiplier might get you out of common animation troubles too. Spinning object are know to be a hassle to export because of jumps after 180 turns. To cure this, use Frame rate Multiplier default value `7` or find another number which does the work for you. But this should now be useless as most of it has been fixed.

Example if the file is set up with:

Frame rate Multiplier = 3

With `fps = 30`, the frame rate for the animation will be `90`. (This value can be found in the log file at line `SampleRate` in section `IAfHeaders`)

5.3 Remove last frame

The IA file describes the positions of the IA elements through time. For a cyclic animation, the last position is identical to the first.
Eventually, removing this last position in the IA file, may make a cyclic animation smoother.

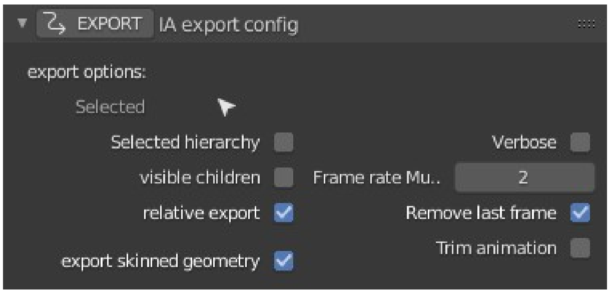
If Remove last frame is ticked in *IA export config*: last position is not exported, else it is written without any change.

5.4 Relative export

This setting is enabled by default. When it is active, only animation relative to Blender object's parent are exported. This is useful for animating buttons mounted on moving parts in cab for example.

Unticking this box exports animations relative to the scene for every object which doesn't have his parent included in exported objects set.

5.5 Selected hierarchy



Selected

Selected hierarchy ☐ Verbose ☐

visible children ☐ Frame rate Mu.. 2

relative export ☑ Remove last frame ☑

Trim animation ☐

export skinned geometry ☑

This option is only visible in export selected mode.
When active, this will export children of selected pose bones and objects recursively.
Else, only selected pose bones and objects are exported.

5.6 Trim animation

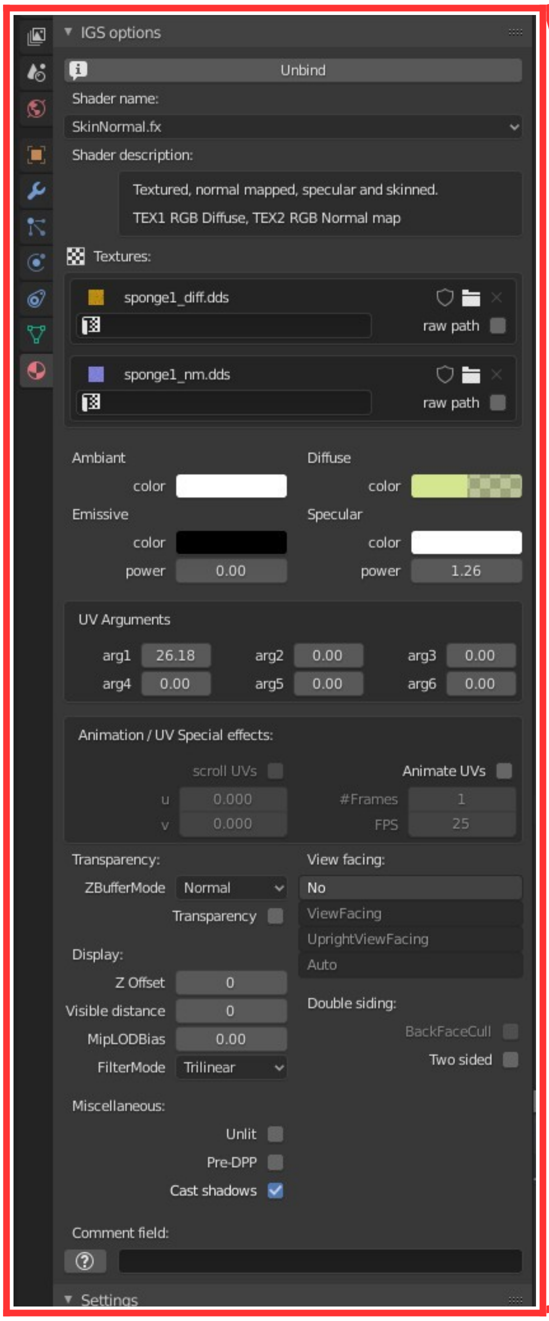
Trim animation option allows you to set a wider time frame than your animation actually makes use of, and have it trimmed down to what is really populated on export.
This may be used to export plenty of cab controls animations in one go without re-framing animations.

6 Material configuration for IGS Export (IGSE)

Material panel IGS options provide powerful tools and UI for setting properties within Blender.

It is located at the bottom of Properties > Material tab but can be moved upward by dragging the dotted surface of the header up.

Left is an overview of its default settings. These will be explained further later in this document.



Shader name: SkinNormal.fx

Shader description:
Textured, normal mapped, specular and skinned.
TEX1 RGB Diffuse, TEX2 RGB Normal map

Textures:
sponge1_diff.dds raw path ☐
sponge1_nm.dds raw path ☐

Ambiant color
Diffuse color
Emissive color power 0.00
Specular color power 1.26

UV Arguments
arg1 26.18 arg2 0.00 arg3 0.00
arg4 0.00 arg5 0.00 arg6 0.00

Animation / UV Special effects:
scroll UVs ☐ Animate UVs ☐
u 0.000 #Frames 1
v 0.000 FPS 25

Transparency:
ZBufferMode Normal
Transparency ☐

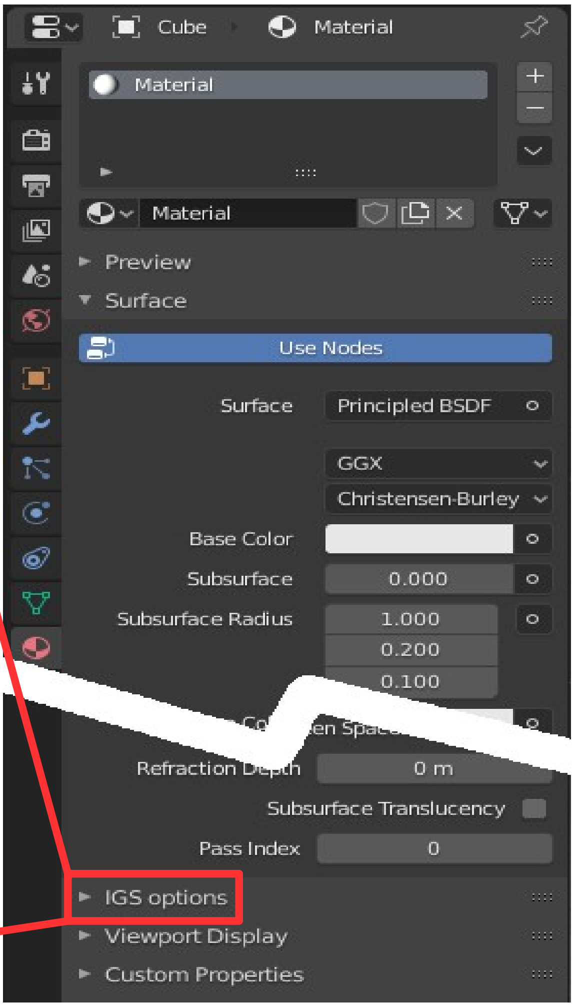
View facing:
No
ViewFacing
UprightViewFacing
Auto

Display:
Z Offset 0
Visible distance 0
MipLODBias 0.00
FilterMode Trilinear

Double siding:
BackFaceCull ☐
Two sided ☐

Miscellaneous:
Unlit ☐
Pre-DPP ☐
Cast shadows ☑

Comment field:

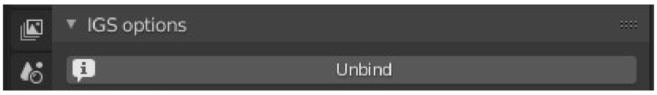


Surface Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius 1.000 0.200 0.100
Refraction Depth 0 m
Subsurface Translucency ☐
Pass Index 0

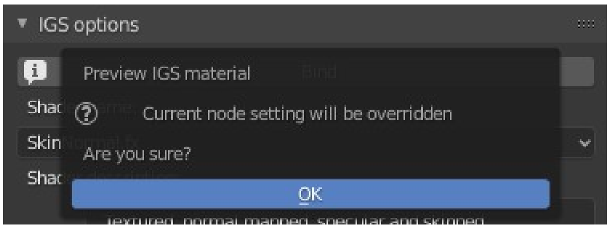
Some options might be greyed out depending on the shader used because all features are not available for all shaders.

Features will be described here from top to bottom.

6.1 Previewing material



IGS materials don't rely on nodes nor PBR to function, hence they are hard to represent in any modern software with true fidelity. Still BRIAGE will try to produce a node arrangement for you to preview your effort by toggling its Bind function. Click on Unbind to free the node setup again.



Preview IGS material

Current node setting will be overridden

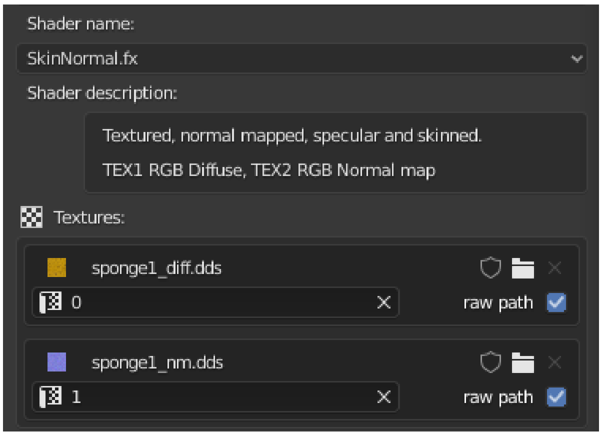
Are you sure?

If some nodes are already present, a prompt will ask you to continue or deny (exit mouse from dialog and click away) to prevent damage to your scene.
A side effect from the displaying methods of blender will be the multiplications of your texture references inside the blend. Indeed some slots of materials only accept Non-Color data to function properly. But since changing colorspaces needs to create a new instance inside the Blend file, your image list my expand a bit.

6.2 Shader assignment

Shaders can be assigned to a material by 2 different manners :

6.2.1 Using UI shader list



Shader name: SkinNormal.fx

Shader description: Textured, normal mapped, specular and skinned. TEX1 RGB Diffuse, TEX2 RGB Normal map

Textures:

sponge1_diff.dds — 0 — raw path ☑

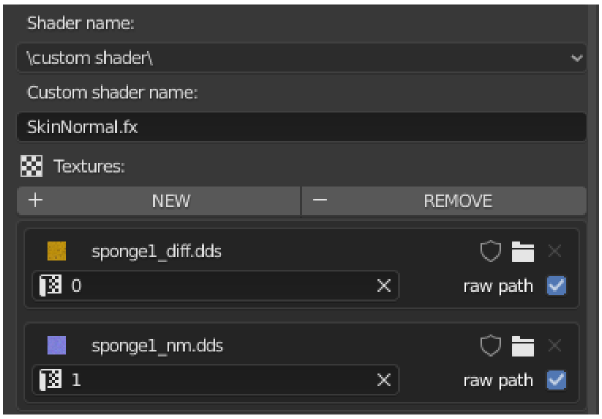
sponge1_nm.dds — 1 — raw path ☑

Below the shader name label is a menu. Clicking on it unfolds a huge pop-up where you can select your shader name. Selecting one will update the shader description box with appropriate informations about use, and shrink or grow the number of texture slots below. Failing to fill all texture slots for a registered shader will generate an error on export.

To learn how to use texture slots, jump to chapter 7.

Pro feature: If you need an unlisted shader constantly and are not afraid to break compatibility with people without your tweak, you can create a new file `Railworks\dev\Shaders\CustomShaders.xml` formatted the same way as DTG's original `Shaders.xml` but with only your own shaders inside.

6.2.2 Using a custom shader



Shader name: \custom shader\

Custom shader name: SkinNormal.fx

Textures:

sponge1_diff.dds — 0 — raw path ☑

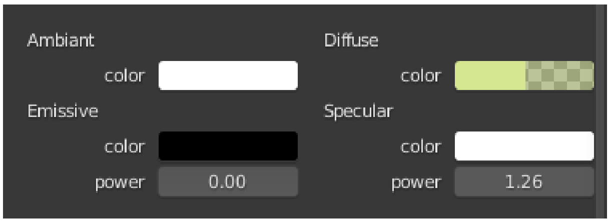
sponge1_nm.dds — 1 — raw path ☑

Selecting "\custom shader\" in shader list brings up a new field where a non-registered shader name can be used. It also supports usual and short shader name but it is not intended to be filled with theses!
You then must make new or remove texture slots as needed for the selected shader to work properly.

Example: `TrainBump.fx` is a non registered yet working 2 slot shader 1rst for diffuse, 2nd for normal map. It can only be set by custom shader name field.

Using a registered shader inside wtih an improper number of slots will fail too. Slot order demonstration on the table below (7.1)

6.3 Ambient, diffuse, specular, emissive



Ambiant
color
Diffuse
color
Emissive
color
power 0.00
Specular
color
power 1.26

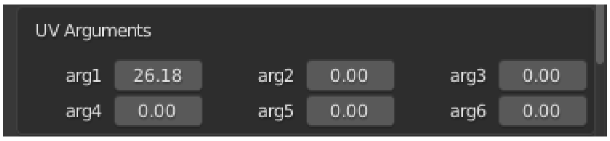
These are available as is. Each field will be processed by TS20XX through the shader.

Specular power (*d3d9 specular exponent*) dictates the intensity of the Phong specular highlight, its default value is 3. Its size come from UV arg1, see below.

6.4 UV Arguments

There are 6 UV args (short for “UV Arguments”) which have various effects depending on the shader used. But the first one is often used for the same setting:

arg1 : *“Phong” size (values between 0<64)*
This is the size of the specular highlight: this is close to blender material's Roughness visual effects but inverted. A higher arg1 values will give you a lower roughness.



UV Arguments
arg1 26.18
arg2 0.00
arg3 0.00
arg4 0.00
arg5 0.00
arg6 0.00

IGSE prints this to logfile for each material:

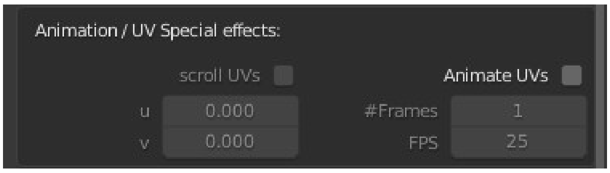
```
BRIAGE:: 'PROPERTY':  UVArguments
for material xxxx: 0.0 0.0 0.0 0.0
0.0 0.0
```

Here are some examples of their functions for different shaders (from http://the-art-of-rws.blogspot.fr:

- `TrainBumpSpec.fx`
  arg1 : “Phong” size (values between 0<64)
- `TrainGlass.fx`
  arg1 : “Phong” size (values between 0<64)
  arg2 : Intensity of the reflection (day-time)
  arg3 : Intensity of the reflection (night-time).
- `WaterCubeMap.fx`
  arg1 :“Phong” size (values between 0<64)
  arg2 : 'Wobble' factor (values between 0<1)
  arg3 : Movement speed (0.02 is a good value)
  arg4 : Oscillation height (0.02 is a good value)
  arg5 : Oscillation scale (6 is a good value)

6.5 UV Special effects

These are special behaviours available for some materials. They allow some sort of texture animation. You can activate them with their associated radio-buttons.



Animation / UV Special effects:
scroll UVs ☐
Animate UVs ☐
u 0.000
v 0.000
#Frames 1
FPS 25

Scroll UVs is available for non-fx shaders. When enabled, U and V coordinates of material's 1rst texture slot are scrolled respectively left and down of “image x**u value*” and “Y *dimension*image y*” each frame.
In other words, UV scroll function scrolls normalized(between 0 and 1) u and v coordinates each render frame.

Animate UVs allows use of “animated” textures (more like a sequence of different images). Bear in mind max value of #frames supported by TS is 256. See this for more informations:

Here is a purpose **modified** version of the article from Railsimilarity:

Train Simulator 20XX supports animated textures. Animated textures refer to a series of identically sized frames or textures which the game will load and display frame-by-frame. Some examples of animated textures would be :

- traffic light sequence
- flashing lights on a police car

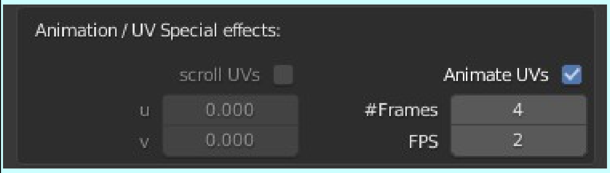
The process is fairly straight-forward. The engine relies of the fact that the frames are named sequentially, and once the material is set up to animate, it will loop through these sequentially named frames at a given speed.

Here, in this example, we will setup a simple 4 frame animation.

1. Initially create the sequence of 4 frames and name then as follows:
XXXXX_anim1.dds, XXXXX_anim2.dds, XXXXX_anim3.dds, XXXXX_anim4.dds etc (where XXXXX is the texture name).

2. In Blender, assign a simple shader such as `TexDiff` (or `AddATex` for glows) to the relevant polygon or mesh. Then in the material, point the texture slot to the first texture XXXXX_anim1.dds. It's important to point to the first texture so the game engine knows where to start the sequence.

3. Still in the material tab, scroll down to find the "UV Special Effects" area, and tick the "Animate UVs" button. Also insert the correct number of frames for your animation in the area marked '#Frames' (in our example this would be 4). Also choose a frame rate in the 'FPS' field. In our example we'll choose 2.

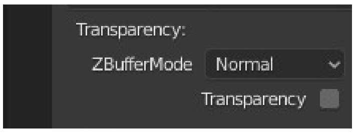

Animation / UV Special effects:
scroll UVs ☐
Animate UVs ☑
u 0.000
v 0.000
#Frames 4
FPS 2

4. Export the shape.

NOTE : The export process will only compile the textures that are referenced in the scene IGS file. As we only reference the starting frame (XXXXX_anim1.dds) in our IGS, only this single frame of the animation will compile. If we view this in the game, we will see the first frame OK but the other frames will be represented as "MISSING TEXTURES". There is a simple solution to this where we can simply force the missing textures to compile from within the blueprint editor. To do this, navigate to find your textures in the blueprint editor and one by one right-click on each one and choose the 'export' option from the roll-out. Once all the textures have been compiled, the shape will be ready to be viewed in-game.

Freely adapted for BRIAGE (2016) from Railsimilarity (2009)

6.6 Transparency

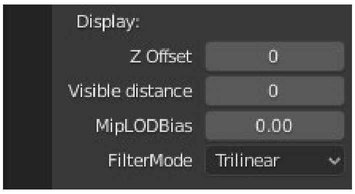

Transparency:
ZBufferMode Normal
Transparency ☐

Transparency settings allow use of transparency related parameters with this material.

ZBufferMode allows modifications of how TS direct3D engine computes Z depth when encountering this material in relation with the shader selected.
To get 8-bit alpha with non-fx shader set this to “test only”

The Transparency checkbox is the former `AlphaTestMode` toggle. It enables transparency for fx shaders, mostly 1-bit alpha.

6.7 Display



Display:

Z Offset 0

Visible distance 0

MipLODBias 0.00

FilterMode Trilinear

These interface material properties with in-game camera distance.

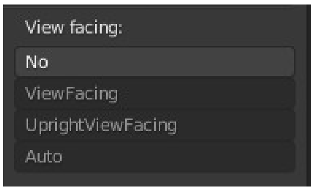
Z Offset helps against Z fighting to set a priority for the material by virtually offsetting his 3D position forwards or backwards.

Visible distance when set at another value than 0 sets maximum distance in meters where this material is viewable. Else, it's considered infinite.

MipLODBias pulls MIPmaps change distance closer(negative values) are away(positive values). This is a particularly effective way of fixing with distance-blurred textures in TS by applying a negative value here.

FilterMode sets the filtering mode used for the texture in-game. Setting this parameter to “5” enables correct MipMap processing for AddATex shader and similar.

6.8 Viewer–facing options



View facing:

Viewfacing enumerator allows you to choose the viewer facing mode you want to apply to your material.

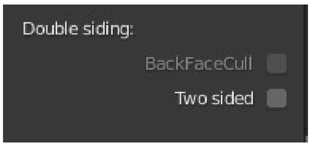
➔ No mode is self-explanatory.

➔ ViewFacing mode enables all-axis view-facing rotations

➔ UprightViewFacing mode enables local Z-constrained view-facing rotations

➔ Auto mode applies the best or most common setting corresponding to the selected shader name.

6.9 Double siding



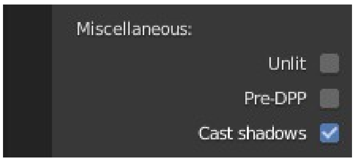
Double siding:

BackFaceCull ☐

Two sided ☐

These two boxes allow culling faces' back for non-fx shader and double-siding for fx shaders.

6.10 Miscellaneous



Miscellaneous:

Unlit ☐

Pre-DPP ☐

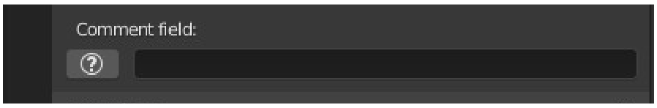
Cast shadows ☑

Unlit activate `LMKeepVertexColors` flag. Fun fact, it doesn't make your material unlit: please use specific shaders for this purpose.

Pre-DPP is a comment field toggle (see 6.11) added on export to the comment field. (DPP means **D**eferred **P**ost **P**rocessing, theses are effects applied after rendering objects like antialiasing or other 2D effects.)

Cast shadows is taken from Blender's interface and activates `LMCastShadows` flag.

6.11 Comment field



This field allows manually entering comment-driven material behaviours. The ? button brings a pop-up with a list of known comment field toggles.

Here is copy of the list:

```
:forceprelit        :fpsfraction        :forceunder
:pointlit           :fadedist           :fullalphasort
:pre-dpp            :noclip             :occlude
:unfog              :startinvisible     :alternode
:animdelay          :startnocollide     :shatter
:animmaxframe       :nocompress         :objectstick
```

6.12 Other properties

One property is still lost in Blender Properties area :

`IGobject_color` : (only set by IGS's main objects) Object tab > Display panel> Object color

7 Materials textures settings

Textures are essential to most of available shaders; don't forget to set them accordingly to what is required by the selected shader.

7.1 Texture slots mapping with TS20xx shaders

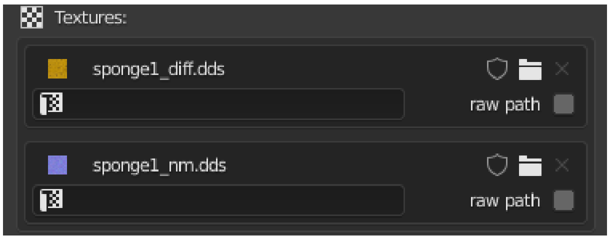
IGSE texture slot available dispositions :

n-th slot	Standard	TrainGlassWeatherEffects.fx
Slot 1	Texture 1	Texture 1
Slot 2	Texture 2	[1] Texture 3
Slot 3	Texture 3	Texture 4
Slot 4	Texture 4	
Slot x	Texture x	

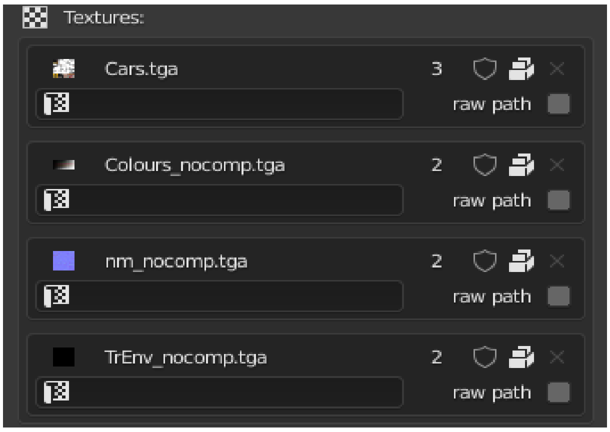
(1) Texture 2 is automatically processed by the exporter.

Some shaders usage examples are given in annex 11.2 and a non-exhaustive shaders list is given in annex 11.1.

Example of slots settings:



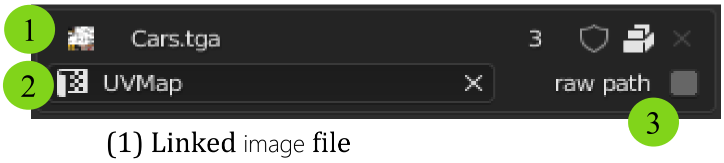
SkinNormal.fx



TrainBasicObjectDiffuse.fx

7.2 Mapping panel for a texture slot associated with an image

Typical setting of the Mapping panel for a texture slot associated with an image:



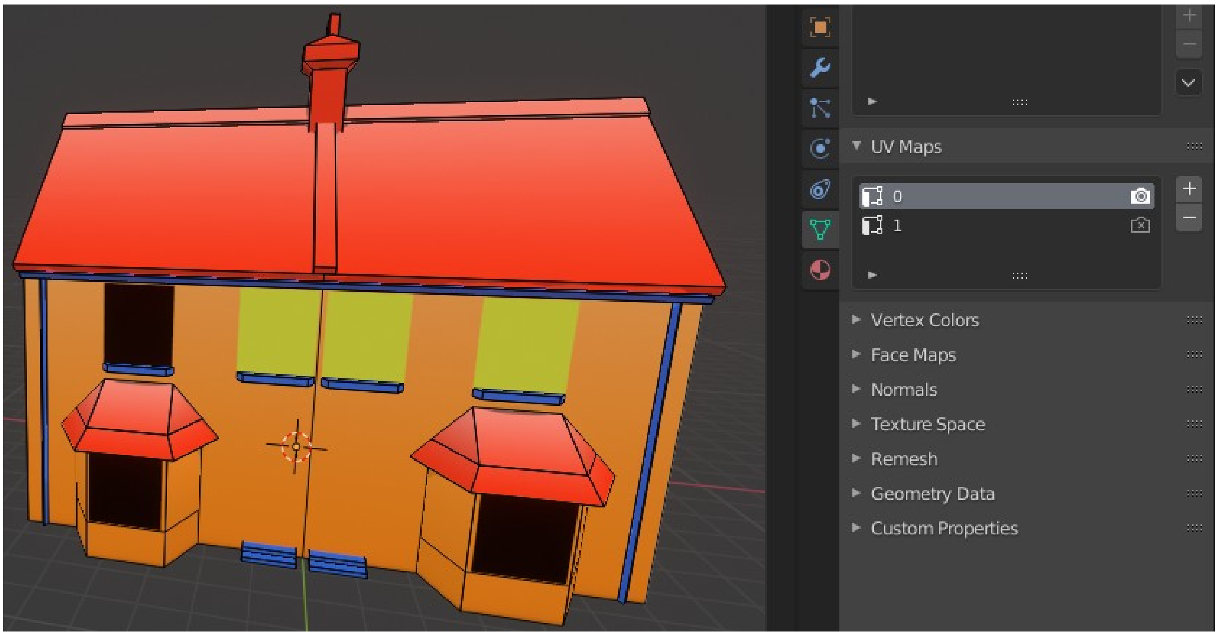
(1) Linked image file
(2) UV Map to use for this slot
(3) Deactivate Textures Directory (see 4.5) setting for this slot's image

The "Map:" field may be left empty when using object active UV channel for mapping this texture slot.

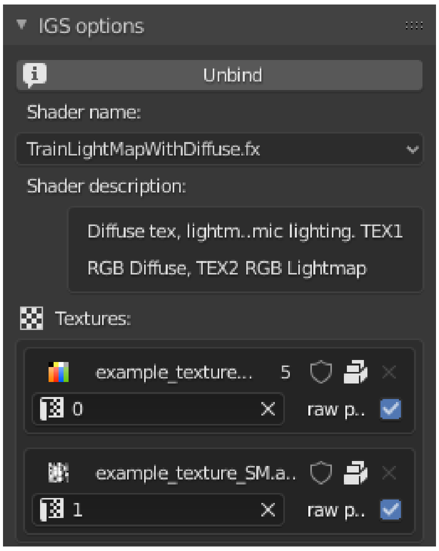
Different mapping informations may be used for different texture slots of the same material.

Here is such an example with shader `TrainLightMapWithDiffuse.fx` :

2 UV maps are defined on this object:

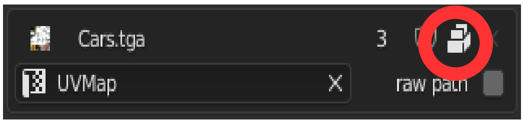


One for diffuse and the other for lightmap:



7.3 Texture path mapping options

There are different ways of binding textures path to materials on export.



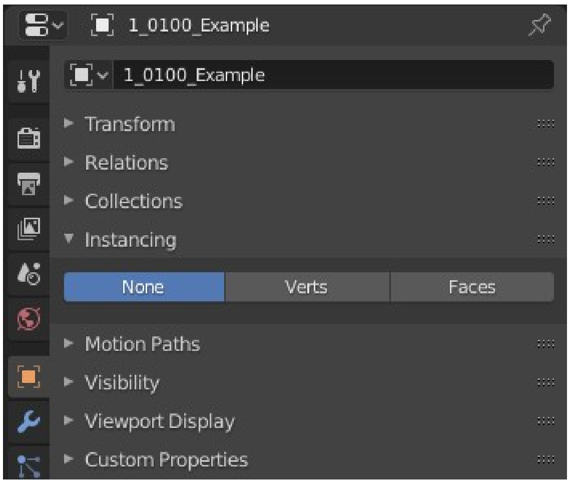
To assign a filepath, just load the image you want to see on your model for this slot. Ensure they are unpacked before exporting or they might have a faulty filepath. (the icon inside the red circle indicates this file is packed)

Raw path setting prevents IGSE from using 'Textures directory' (see 4.5) setting on this texture.
>> If using *"IGS texture mode"*: wherever you export your IGS, its texture component will point to the file used by Blender.
>> If using "*BLEND texture mode*": *the same relative filepath from blend to texture will be applied to IGS*. For this setting to work properly, work in your source folder sub-directories where you will export your IGS to, or make sure your working directories mimic exactly end-directories hierarchy.

8 Notes

Here are various notes you should take into account while using BRIAGE.

8.1 Using Instancing



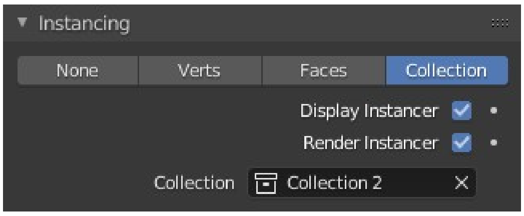
Duplis are objects or groups of objects instanced in Blender by Duplication panel of objects.

These instances are now exported by BRIAGE with the following strategies.

8.1.1 Verts and Faces

These are merged with their respective original object on export.

8.1.2 Collections



The content of the collection is *parented to the duplicator* (mesh or empty) and *renamed with duplicator's name*.

Using an empty without parent as duplicator preserves group's hierarchy; else, group objects with no parent are parented to the duplicator.

The duplicator cannot be a snap point. Duplicator's name may or may not use TS naming conventions, its detected **name is inserted at the beginning** of object's and bone's names of the group instances. You must *be careful not to exceed the overall 24 chars* limit after insertion , else name will be truncated. If a **"*" is present as last** char of duplicator's name, name will be **appended to instance's names** without "*".

Animations inside instanced collections can be exported as well.

8.2 Using Snap Points

Export of Snap Points is now available.

They obey to 3DSmax naming rules; see **'*TS directory*'/dev/Docs/TS2015Tech.pdf**, *chapter 4.*
*Here is a **modified** extract from the above-mentioned documentation:*

1. Place an "arrows" type empty in the Blender scene. The naming convention is important so the blueprint editor can recognise this as a snap point.
   - **#ss**_xxx is a **source** snap point
   - **#st**_xxx is a **target** snap point
   - **#sb**_xxx is both a **source** and **target** snap point

   where xxx is a **unique** name, in this case SnapOut2 (I was using SnapIn for target snap points)

2. Orient the snap point object so that the Z axis is pointing in the direction of the desired alignment when snapped to a target. In general, for all snap point types, this is usually away from the model's surface

3. Setup remaining snap points.

4. All done! Now export as IGS.

5. Add as the model of a scenery blueprint, or any other blueprint that has a container component.

6. Hit export. The snap points will automatically be added to the container component on export, and by default these will be the Kuju\RailSimulatorCore\Editor\Gizmo\DefaultScenerySnapPoint*.xml blueprints.

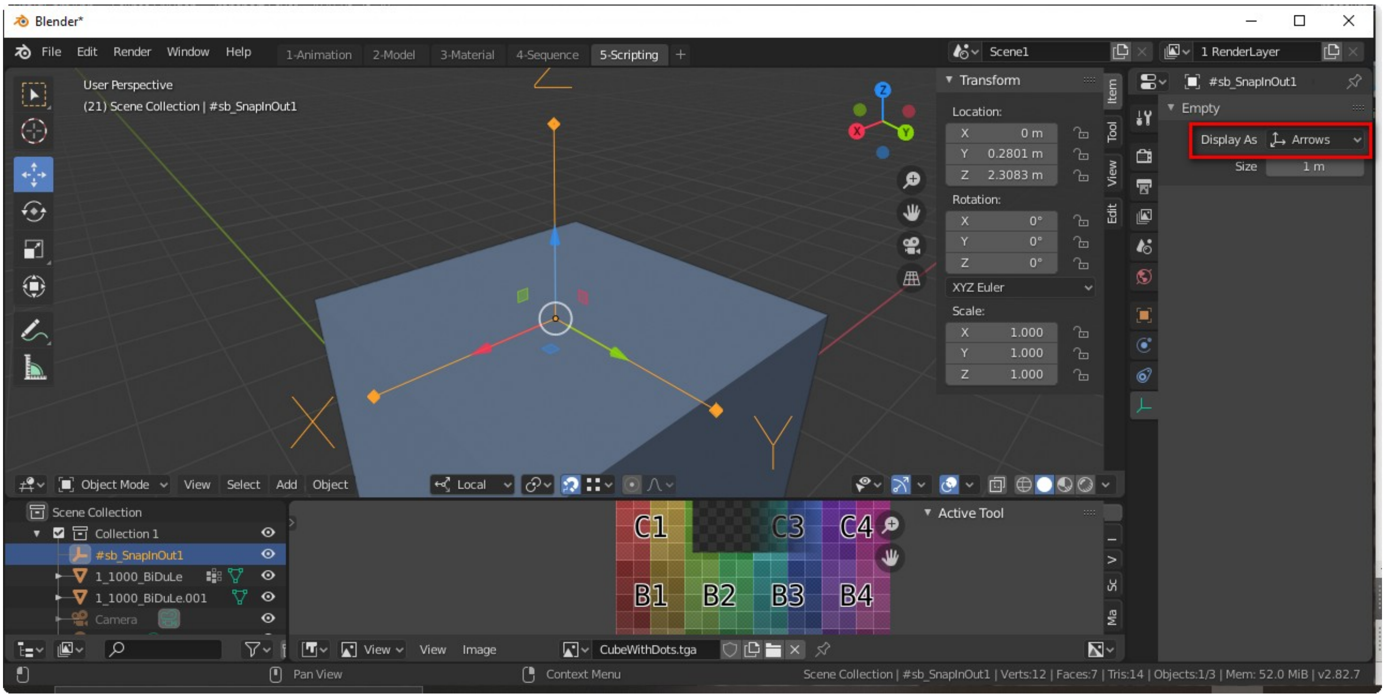
NOTE: You can create your own snap point blueprints and configure various settings for these. To use your own blueprints, you will have to manually add the snap points with the names you gave them in Blender as children of the blueprint, e.g. **#ss_SnapOut2** would be named **SnapOut2**. You assign your custom snap point blueprint as the blueprint ID, and hit export. The blueprint editor will match the names of the snap points in the model to the ones in the blueprint and override their matrices automatically.

Freely adapted for BRIAGE (2016) from TS2015Tech.pdf (2015)

In Blender they follow these additional rules:

- **Place an "`arrows`" type empty (Display type)**
- They can have a parent object
- They mustn't have any children

Also note that they do not count as a child when parented to another object in TS, hence they are immune to the supported max number of children.



Transform
Location:
X 0 m
Y 0.2801 m
Z 2.3083 m
Rotation:
X 0°
Y 0°
Z 0°
XYZ Euler
Scale:
X 1.000
Y 1.000
Z 1.000

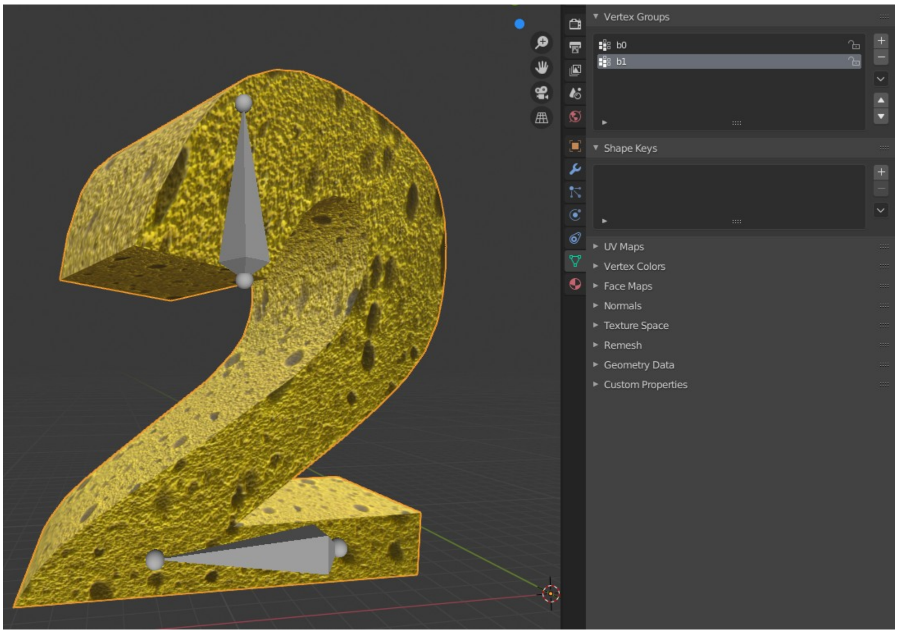
Display As Arrows
Size 1 m

Scene Collection | #sb_SnapInOut1 | Verts:12 | Faces:7 | Tris:14 | Objects:1/3 | Mem: 52.0 MiB | v2.82.7

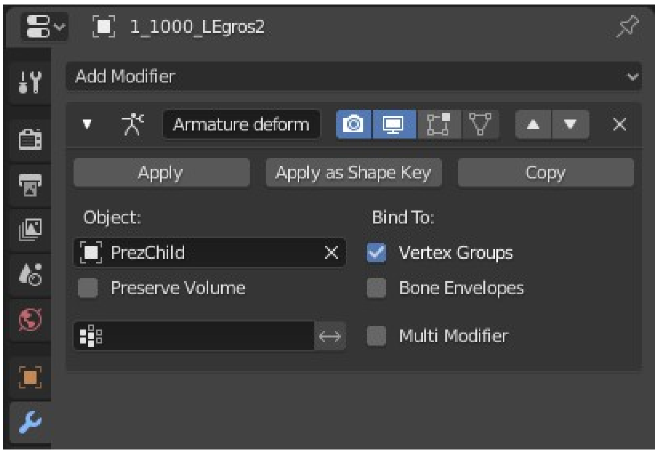
Take a look at “SampleWithDots.blend” example file for a demonstration.

8.3 Using skinning and bones

Export of armatures and skinning is now available.



Armatures are exported when using Export skinned geometry function (see 4.2). Corresponding Vertex groups weights are exported when an armature modifier is present on a skinned object. *Envelope deformations are not supported* if not applied to Vertex groups. Skinned objects must only use “`Skin....fx`” shaders to work properly. Skinned objects doesn't need to be child of an armature they only need an armature modifier set to **deform following vertex groups weights.** Vertex groups names must be the same as the concerned bone name. It is recommended to take a look here: Blender manual “Weight edit”, “Skinning to Shapes” and more generally to “Skinning” chapter. Armatures to export are selected either manually or detected in the armature modifier. When multiple armatures are to be exported, please ensure **bone names are unique among all exported armatures.**



All objects within an IGS doesn't have to be skinned. Used armatures must be exported (it follows rules defined by selected `Export mode`). Make sure all your **bones have unique names** *even across armatures*!

You might want to disable this feature for armature-based animations which doesn't need any deformations on any object to speed up performances.

Take a look at "Prez.blend" example file for a demonstration.

8.4 About model export

8.4.1 General notes

The use of negative scaling requires applying Ctrl A "scale".

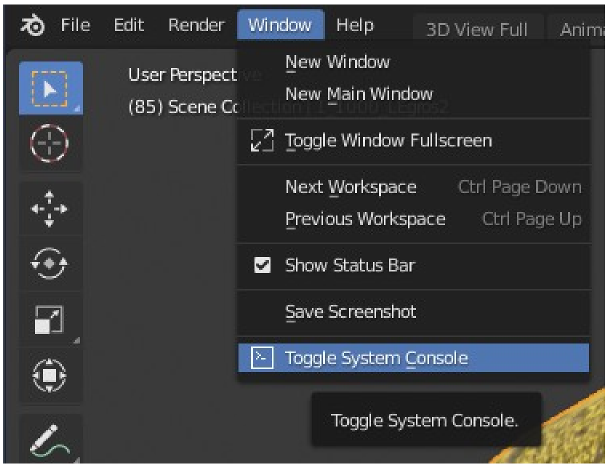
"`current frame`" is set to 0 during the export. The different animated objects must be in a **rest position at "`current frame`" = 0**. At the end of the export, "current frame" is reset to the value before performing the export.

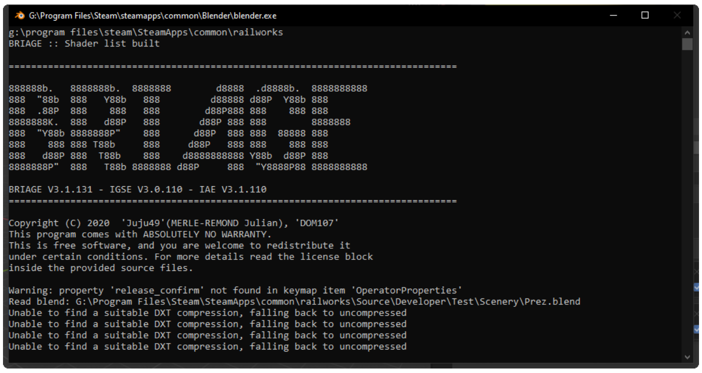
Only 256 objects can be exported within an IGS including LoDs and without snap points.

Max number of faces and vertices is unknown.

8.4.2 Reading logs

A log file is not printed if not especially asked with the "Verbose" toggle. But a shortened log is available after each export in Blender's system console here:



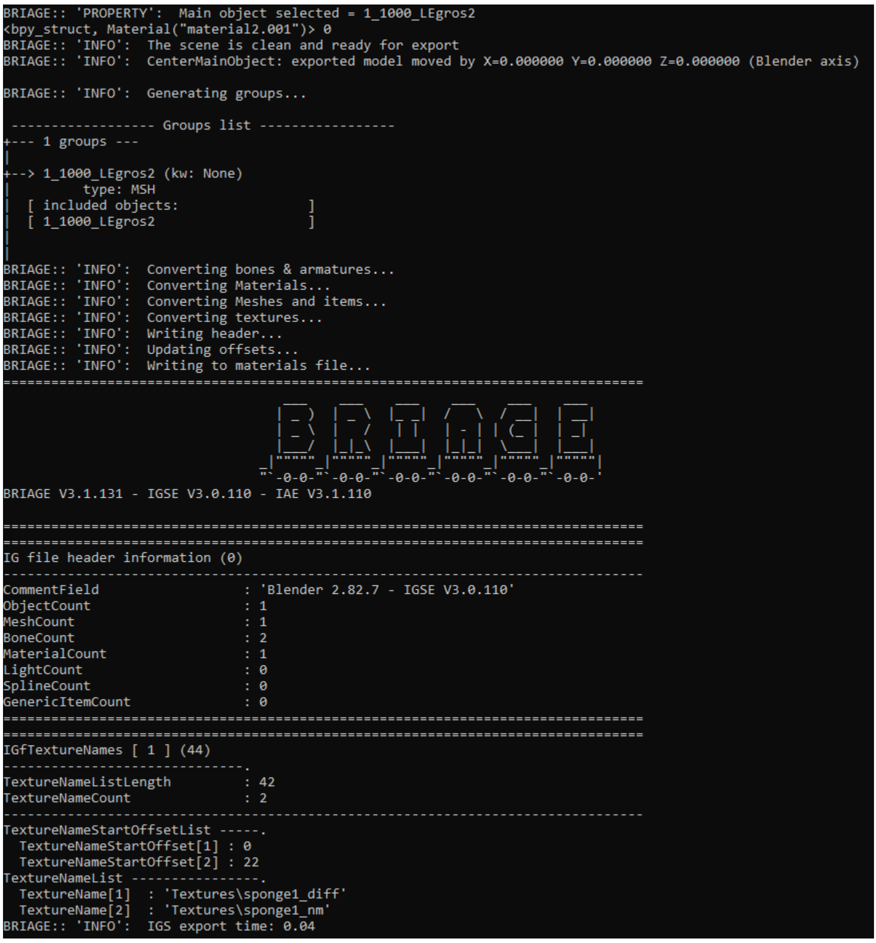


```
g:\program files\steam\SteamApps\common\railworks
BRIAGE :: Shader list built

================================================================================

888888b.   8888888b.  8888888        d8888  .d8888b.  8888888888
888  "88b  888   Y88b   888         d88888 d88P  Y88b 888
888  .88P  888    888   888        d88P888 888    888 888
8888888K.  888   d88P   888       d88P 888 888        8888888
888  "Y88b 8888888P"    888      d88P  888 888  88888 888
888    888 888 T88b     888     d88P   888 888    888 888
888   d88P 888  T88b    888    d8888888888 Y88b  d88P 888
8888888P"  888   T88b 8888888 d88P     888  "Y8888P88 8888888888

BRIAGE V3.1.131 - IGSE V3.0.110 - IAE V3.1.110
================================================================================

Copyright (C) 2020  'Juju49'(MERLE-REMOND Julian), 'DOM107'
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. For more details read the license block
inside the provided source files.

Warning: property 'release_confirm' not found in keymap item 'OperatorProperties'
Read blend: G:\Program Files\Steam\SteamApps\common\railworks\Source\Developer\Test\Scenery\Prez.blend
Unable to find a suitable DXT compression, falling back to uncompressed
Unable to find a suitable DXT compression, falling back to uncompressed
Unable to find a suitable DXT compression, falling back to uncompressed
Unable to find a suitable DXT compression, falling back to uncompressed
```

You will note that when BRIAGE starts properly, it prints this in Blender system console: (right picture)

Paragraph 9 lists typical export errors or problems.

Even without any error, check specifically for:



```
BRIAGE:: 'PROPERTY':  Main object selected = 1_1000_LEgros2
<bpy_struct, Material("material2.001")> 0
BRIAGE:: 'INFO':  The scene is clean and ready for export
BRIAGE:: 'INFO':  CenterMainObject: exported model moved by X=0.000000 Y=0.000000 Z=0.000000 (Blender axis)

BRIAGE:: 'INFO':  Generating groups...

 ------------------ Groups list -----------------
+--- 1 groups ---
|
+--> 1_1000_LEgros2 (kw: None)
|         type: MSH
|  [ included objects:                ]
|  [ 1_1000_LEgros2                   ]
|
|
BRIAGE:: 'INFO':  Converting bones & armatures...
BRIAGE:: 'INFO':  Converting Materials...
BRIAGE:: 'INFO':  Converting Meshes and items...
BRIAGE:: 'INFO':  Converting textures...
BRIAGE:: 'INFO':  Writing header...
BRIAGE:: 'INFO':  Updating offsets...
BRIAGE:: 'INFO':  Writing to materials file...
================================================================================
                                   ___    ___    ___    ___    ___    ___
                                  | _ )  | _ \  |_ _|  /   \  / __|  | __|
                                  | _ \  |   /   | |   | - | | (_ |  | _|
                                  |___/  |_|_\  |___|  |_|_|  \___|  |___|
                                _|"""""_|"""""_|"""""_|"""""_|"""""_|"""""|
                                "`-0-0-"`-0-0-"`-0-0-"`-0-0-"`-0-0-"`-0-0-'
BRIAGE V3.1.131 - IGSE V3.0.110 - IAE V3.1.110

================================================================================
================================================================================
IG file header information (0)
--------------------------------------------------------------------------------
CommentField                  : 'Blender 2.82.7 - IGSE V3.0.110'
ObjectCount                   : 1
MeshCount                     : 1
BoneCount                     : 2
MaterialCount                 : 1
LightCount                    : 0
SplineCount                   : 0
GenericItemCount              : 0
================================================================================
================================================================================
IGfTextureNames [ 1 ] (44)
------------------------------.
TextureNameListLength         : 42
TextureNameCount              : 2
--------------------------------------------------------------------------------
TextureNameStartOffsetList -----.
  TextureNameStartOffset[1] : 0
  TextureNameStartOffset[2] : 22
TextureNameList ----------------.
  TextureName[1]  : 'Textures\sponge1_diff'
  TextureName[2]  : 'Textures\sponge1_nm'
BRIAGE:: 'INFO':  IGS export time: 0.04
```

- Messages about name substitutions, could it be:
  → incorrectly named objects.
  → replaced object names
- The objects groups list can be checked to ensure that objects are grouped as expected.

The short log file allows displaying the following information:

- The **texture files list**

8.5 About animation

8.5.1 Blender 3D view animation

The exporter processes active animations at the time of the export file saving.
Thus, if, for example, a constraint is disabled in the open model, it will not be in the exported animation.
In short, the animation working in the Blender 3D view should be the one played in TS20XX, provided all the objects selection is correct ("Export selection" mode) or all the objects needed for the animation are visible ("Export visible" mode). Some options allow more control over what should come out in TS, see 5.4, 5.5 .

Don't forget to set the frames per second (fps) and the start and end keyframes according to the relevant objects: a switch animation is shorter than a pantograph animation.

Immobile objects are timed away from export.

8.5.2 Scaling

Data in exported IA format are positions and rotations. There is therefore no scale. Remember to apply scale to objects CTRL+A "scale" to avoid strange surprises. Thus the effect of a "copy scale" constraint won't show up in the TS20XX exported animation. Although, a scale animation could be faked using an armature and skinning vertices, one bone for each stretching direction.
A model and tutorial on "*animating the valve-gear of a steam locomotive*" by Mike Adams can be found on UKtrainsim download section. Though this tutorial is being updated for this export tool, it gives a good example of what can handle the exporter: lattices, bones, armatures, constraints of type copy location / copy rotation, limit rotation.

8.5.3 unrelated animations

It is likely that one Blender file will have several unrelated animations (such as for example, wipers and doors). These different animations can all start at keyframe 0. However when animation is played in Blender all the animations in a specified keyframes range will animate. This is no problem for the exporter as, once the keyframes range is set, you'll only select objects to export animations for (in this

example, the wipers) and the other moving objects will not be included.
If you prefer independent animations in the Blender file, you can use different keyframes ranges. Select your object(s) and its(their) related parent objects and set the keyframes range to when they animate and finally export as IA file.
For example, a wiper animation between keyframes 0 and 100, a door animation between 120 and 200, etc.. To export the door animation, select your doors and their related parent objects and set the keyframes range between 120 and 200.

8.5.4 other IA tips

The dope sheet or the graph editor allows to easily move keyframes if necessary.

It is not necessary to select all objects of an animation set and especially an object with an already selected parent if using "selection mode".

Among the selected objects ("selection mode" enabled) or the objects below the scene object ("visible mode" enabled), the criteria of inclusion in the animation export are :

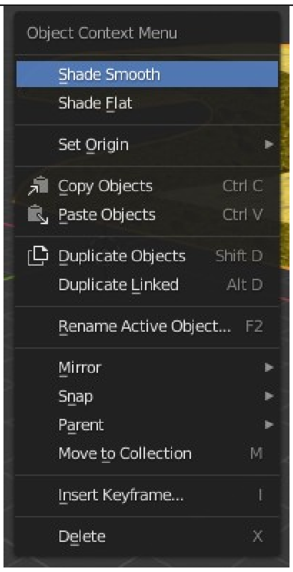
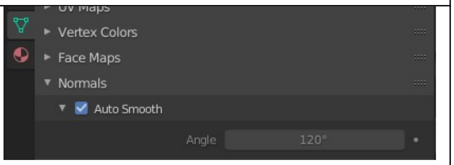
- They have a constraint or an animation.
- They are direct parent of an armature.
- They have a child (parentage created with CTRL+p) with a constraint or an animation.

When enabling "selection mode", **be sure that the selection is consistent with the IGS** structure or bring corrections with Relative export and Selected hierarchy options. For example, if the IGS file was created exporting the entire model, the selected objects for animation should be at the level under the scene object (otherwise, risk of unwanted translation when playing the animation without using previously cited options for adjustments).

By default, the log file lists the selected objects and those taken into account in the export.

8.6 About smooth vertex normals

The exporter takes into account whether to render the object fully smoothed, flat or **auto-smoothed** meshes. It also reads and export **custom normals** and **sharp edges**.
You can read details on smoothing here:
https://docs.blender.org/manual/en/latest/modeling/meshes/editing/normals.html
I use below some information from this link for a few explanations.

The easiest way is to set an entire object as smooth or faceted is by right-clicking a mesh object, and click Smooth in the Object Contecxt Menu. This button does not stay pressed; it forces the assignment of the “smoothing” attribute to each face in the selected mesh. Click the Flat button in the same menu to cancel the smoothing effect.	 Object Context Menu Shade Smooth Shade Flat Set Origin Copy Objects Ctrl C Paste Objects Ctrl V Duplicate Objects Shift D Duplicate Linked Alt D Rename Active Object... F2 Mirror Snap Parent Move to Collection M Insert Keyframe... I Delete X
Auto smoothing can be enabled in the Object data properties panel in the Properties area. Angles on the model that are smaller than the angle specified in the Angle button will be smoothed for the export when that part of the mesh is set to smooth (i.e Smooth was also clicked). Higher values will produce smoother faces, while the lowest setting will look identical to a mesh that has been set completely solid.	 ☑ Auto Smooth Angle 120°
Another way of changing the impact of smoothing is to use the **make edges hard** or **create split normals**.	

If you want to use custom normals for improved lightning effects and cleaner normal maps, have a look at:

- Weighted Normal modifier
- Blender Add-on: Y.A.V.N.E. (Yet Another Vertex Normal Editor)
- Normal Edit modifier

8.7 About materials

8.7.1 Performance notice

It is important to keep the number of different materials low because each material contains a lot of informations which occupies large disk space but also RAM when exporting which slows down the process. The number of quads and tris impact in-game fps far less than a high number of materials.

8.7.2 *TrainBasicObjectDiffuse.fx* 1-bit alpha

`TrainBasicObjectDiffuse.fx` and most *.fx shaders can be used with 1 bit transparency in the alpha channel provided it’s only black (full transparency) or white (no transparency); shades of grey are mostly unsupported (a notable exception are the shaders containing `Glass` in their names). For the transparency to be active, Transparency must be enabled in material's IGS options.

8.7.3 Vertex colours

Active and renderable Vertex colors channel is exported.

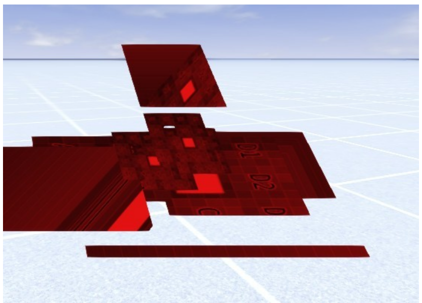
Be careful, if vertices have too different colours for their adjacent faces they will not weld correctly together, causing performance issues in TS.

8.7.4 Viewer facing properties

Viewfacing properties are supported for `AddATex` shader.

Take a look at "UglyToonStreetlight.blend" and "SampleWithDots.blend" example files for a demonstration. Light halo only appears at night.

Viewfacing value is automatically set according to shader names if they contain "Upright" and/or "ViewFacing" (e. g.: `TrainUprightViewFacingFlora.fx`) if you set Viewer-facing options to Auto.

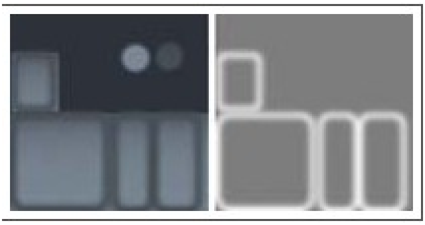
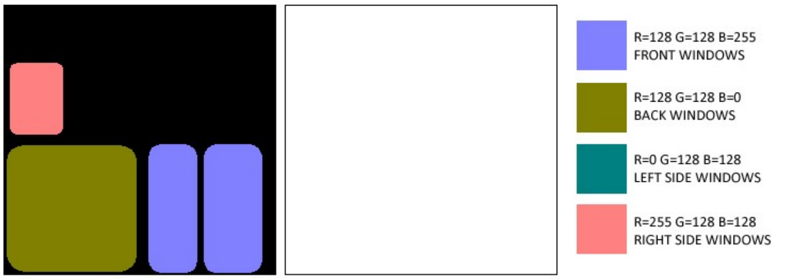
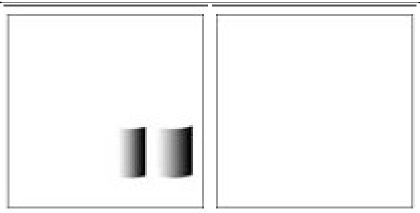


Be careful while using shaders in upright view-facing mode: quads will turn around local Z axis and vertices will use a matrix defined from main object rotations to position themselves. Be sure your main object doesn't have some rotations if you use Upright View-facing quads as they will appear deformed: (screenshot on the left)

An invisible object must be added otherwise the object once added in a scenario won't be pickable. One way is to add a cube with a UV map initialized and use the "`invisible`" shader. The "`invisible`" shader must not be accompanied by a texture file else, it will generate an error.

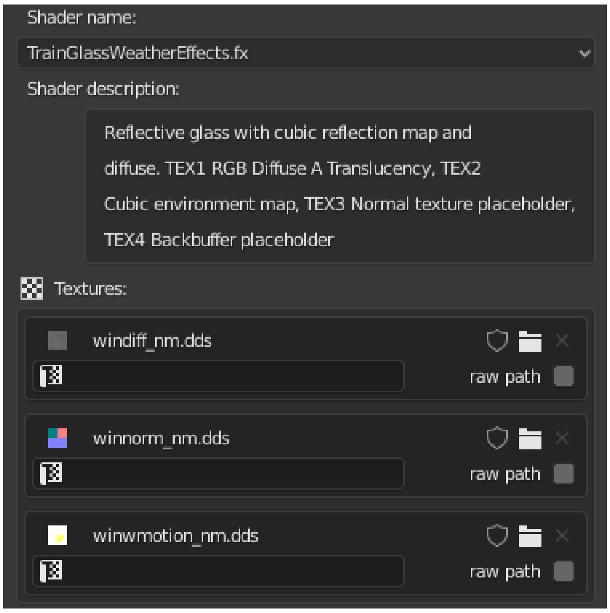
8.7.5 Setting-up *TrainGlassWeatherEffects.fx*

`TrainGlassWeatherEffects.fx` is processed by the exporter. **The material name with such shader must be weatherglass_1 then weatherglass_2**, etc… (According to the number of materials using this shader) you only need to have **3 slots** with the useful texture files. The texture filenames is the user choice. The 3 images below are from https://www.christrains.com/en/ts_faq_weatherglass.html :

windiff_nm.dds : Main texture with alpha channel.	
winnorm_nm.dds : normal map texture to define where the raindrops are.	 R=128 G=128 B=255 FRONT WINDOWS R=128 G=128 B=0 BACK WINDOWS R=0 G=128 B=128 LEFT SIDE WINDOWS R=255 G=128 B=128 RIGHT SIDE WINDOWS Diffuse (left) and alpha (middle) channels for slot 3. RGB Colours (right) for window directions
winwmotion_nm.dds : Texture to define the wipe out pattern of raindrops. May be generated using "*winmotion.exe*" in TS directory	

For `TrainGlassWeatherEffects.fx`, the exporter automatically adds an additional slot for TS20XX. (dummy cubemap slot)

Example in Blender:

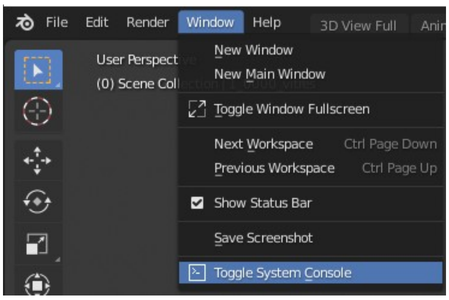


Shader name:
TrainGlassWeatherEffects.fx
Shader description:
Reflective glass with cubic reflection map and diffuse. TEX1 RGB Diffuse A Translucency, TEX2 Cubic environment map, TEX3 Normal texture placeholder, TEX4 Backbuffer placeholder
Textures:
windiff_nm.dds
raw path ☐
winnorm_nm.dds
raw path ☐
winwmotion_nm.dds
raw path ☐

See also some shader usages in annex 11.2.

9 Typical export errors or problems

Most of export errors are caught on export and display pop-ups about what the trouble is. Caution: in Blender, pop-ups disappear when they are no more under the mouse cursor but BRIAGE writes them to Blender's System console(Info area > Window > System console) and/or log file for the user to be able to find them again.



9.1 Warnings

9.1.1 About Asset Editor cache

When exporting the ia animation trying different parameters, it’s better to exit the Asset Editor, sometimes clean the cache and restart the Asset Editor. Otherwise, the Asset Editor might not update the animation properly.

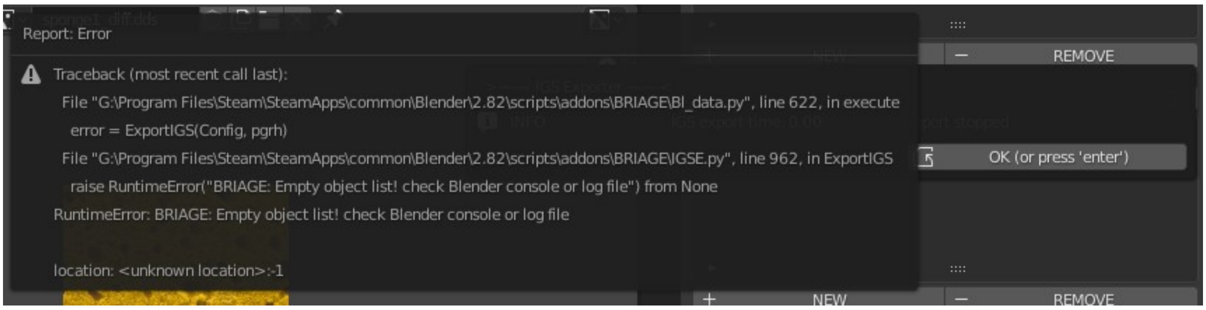
9.1.2 About Asset Editor preview mode

While trying different parameters with lofts, modifications does not update accordingly. Please close and restart the preview.

9.1.3 About TS live object export

Exporting objects from Asset Editor to an enabled provider and product while TS Route Editor is open on your computer and then reload (not F9 but deactivate then reactivate its checkbox) the product can and **will corrupt your route** on save and then **crash TS**. Never do or try this on any valuable route!

9.2 No mesh or empty object found export. Export Stopped

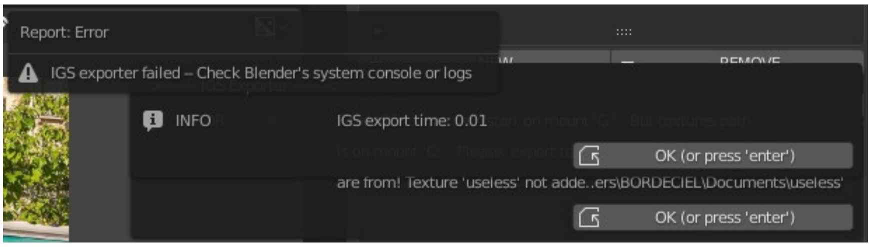


```
Report: Error
Traceback (most recent call last):
  File "G:\Program Files\Steam\SteamApps\common\Blender\2.82\scripts\addons\BRIAGE\Bl_data.py", line 622, in execute
    error = ExportIGS(Config, pgrh)
  File "G:\Program Files\Steam\SteamApps\common\Blender\2.82\scripts\addons\BRIAGE\IGSE.py", line 962, in ExportIGS
    raise RuntimeError("BRIAGE: Empty object list! check Blender console or log file") from None
RuntimeError: BRIAGE: Empty object list! check Blender console or log file

location: <unknown location>:-1
```

This error is generated when nothing is found which can be exported. To solve this :

- if in “Export selected” mode: select objects.
- if in “Export visible” mode: ensure all the objects you want to export are visible in the 3D View.

9.3 Please export to the same drive your data are from! Texture 'xx' not added.

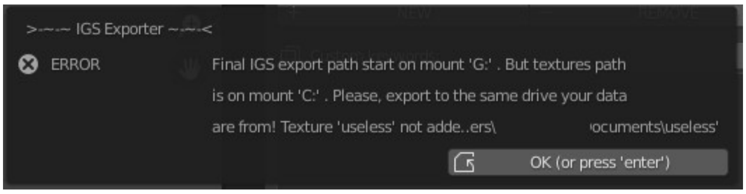


Report: Error

IGS exporter failed – Check Blender's system console or logs

INFO IGS export time: 0.01

are from! Texture 'useless' not adde..ers\BORDECIEL\Documents\useless'



>-~-~ IGS Exporter ~-~-<

ERROR Final IGS export path start on mount 'G:' . But textures path is on mount 'C:' . Please, export to the same drive your data are from! Texture 'useless' not adde..ers\ ocuments\useless'

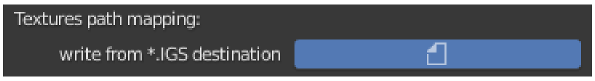
This appears when you export a file to another drive your textures are from, the best way to avoid this is by using TS's source folders as working directories while developing your model.

It also appears when you are exporting an object from a scene never saved before and Blender executable is not on the drive you are exporting to.

Example: Your textures are under **D:**/*myWork/myTrain/myTexures/* and you export to **C:**/*steamLib/common/Railworks/Source/myProvider/myProduct/RailVehicles/myTrain/* the error will pop-up. This will not happen if textures are under **C:**/*myWork/myTrain/myTexures/* but textures paths in the igs will point there and your model will never work on another computer than yours.

To fix this behaviour (picking one is enough):

- If you are accustomed to the old texture processing way, uncheck this and it should work as before:



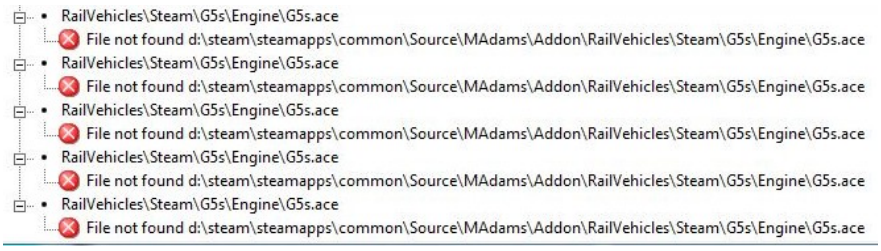
Textures path mapping:

write from *.IGS destination ☑

- Use provided texture path 'on-export' modifier tools (see 4.5 and 7.3)
- Use TS's source directories as working directories

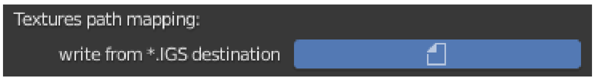
9.4 No texture files found in asset editor or blueprint editor

If no texture files are found when opening the xml model file in the asset editor or the blueprint editor



```
RailVehicles\Steam\G5s\Engine\G5s.ace
    File not found d:\steam\steamapps\common\Source\MAdams\Addon\RailVehicles\Steam\G5s\Engine\G5s.ace
RailVehicles\Steam\G5s\Engine\G5s.ace
    File not found d:\steam\steamapps\common\Source\MAdams\Addon\RailVehicles\Steam\G5s\Engine\G5s.ace
RailVehicles\Steam\G5s\Engine\G5s.ace
    File not found d:\steam\steamapps\common\Source\MAdams\Addon\RailVehicles\Steam\G5s\Engine\G5s.ace
RailVehicles\Steam\G5s\Engine\G5s.ace
    File not found d:\steam\steamapps\common\Source\MAdams\Addon\RailVehicles\Steam\G5s\Engine\G5s.ace
RailVehicles\Steam\G5s\Engine\G5s.ace
    File not found d:\steam\steamapps\common\Source\MAdams\Addon\RailVehicles\Steam\G5s\Engine\G5s.ace
```

Check in the log file that the export used the proper texture mapping informations. If you are not familiar with the new BRIAGE texture workflow, uncheck "write from *.IGS destination":



Textures path mapping:

write from *.IGS destination ☑

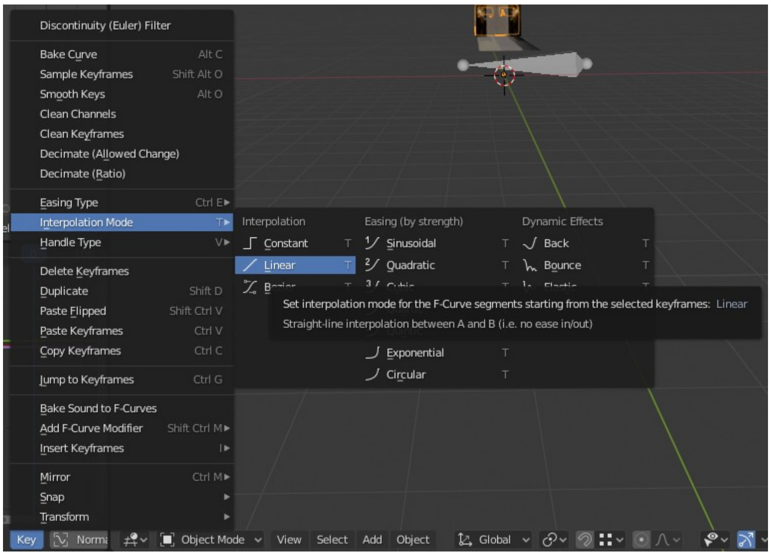
9.5 The animation is working properly in Blender but not in the asset editor

If nothing moves, check in the main blueprint "Anim Set" section that all the animation sets have "Animate in editor" set to "True".

If the parts are not moving as expected:

- **Check the igs / ia export consistency**. For example, selected objects for animation are not under the scene level whereas the igs export was made for the entire model (no selection).
- **Check in the igs export log file ("Groups list" section) that a moving part was not merged with another object**. If this is the case, add keywords in the Custom Keywords table, if the model has not too many objects, disable hierachy processing.
- **Check the FPS value** (number of frames per second) in the igs exporter log file. This value is listed in every description of a material even if option "verbose" is not checked. Setting-up FPS is explained in paragraph 5.1.
- Exit the Asset Editor, clean the cache and restart the Asset Editor.

9.6 The animation is stuttering



Set interpolation mode for the F-Curve segments starting from the selected keyframes: Linear
Straight-line interpolation between A and B (i.e. no ease in/out)

If you can't get a smooth animation, try "Key" menu -> "Interpolation Mode" -> "Linear", either in the dope sheet or in the graph editor.

You can also try to lower the framerate divider (framerate base: see 5.1) or try to **use framerate multiplier option**: see paragraph 5.

9.7 Image filename contains dots '.' which may cause unexpected results in material 'xx', in texture slot 'yy'

This warning appears when the texture filename contains more than 1 dot because the blueprint editor will process these thinking the last part after the first dot he sees is a file extension. Therefore a "`.safe`" is appended at export in this case.

9.8 IGSE configuration parameters are not working as expected

Check the igs log file in Blender's system console or by performing an export with Verbose option activated and check the output:

- Check all your exported objects have their scale applied (active object > CTRL+A > "scale"). This is known to cause untraceable bugs during export.
- Then, check its contents: Are the expected IGS parameters listed as expected (shader names, texture names, etc...)? If not refer back to sections from this very manual about setting correctly the parameters which failed to export.
- For transparency, you may need to change some parameters under Material IGS options panel> Transparency settings according to the shader used (see paragraph 11.2).
- If you still encounter problems, please contact the author or current maintainer(s) of BRIAGE about your issue with a detailed log file and eventually your *.blend file.

9.9 Too many children for bone or object

Maximum number of bone children is 8 and maximum number of object children is 24, snap points excluded.

9.10 An object of an animated set is not in the expected rest position

An object of an animated set may not be in the expected position before running the animation. Still, when running the animation and when the set returns to the rest position, all objects of the animated set are well positioned.

To fix this problem:

- If you still encounter problems, please contact the author or current maintainer(s) of BRIAGE about your issue with a detailed log file and eventually your *.blend file.
- Check the object position with current frame set to 0 (the igs export is done with "`current frame`" = 0).
- Before animating an object, ensure to use Ctrl+A “rotation & scale”.
- If the badly positioned object is a child, clear the parent link (Alt+P Clear and Keep Transformation). Add the link with the parent using the ChildOf constraint + Set Inverse. Don’t forget to export again the igs file. The object must then be included in the selection when exporting the animation.

9.11 "Inconstent LODing" error or some objects are offset

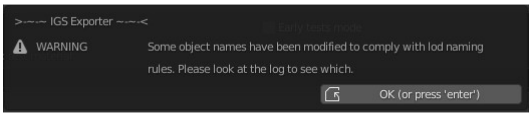
Check that:

- An object of LOD level N has a parent of LOD N-1 (parentage created by Ctrl+p).
- The child name must be the same as the parent name, apart from the LOD level and distance
- The child LOD distance must be greater than the parent LOD distance
- Some objects with different names haven't been truncated to the same 24 chars name to comply with LOD naming conventions

9.12 Could not map shader

Reinstall the default shaders. Early RWE versions messes with them in a way which doesn’t allow one to export all shaders properly.

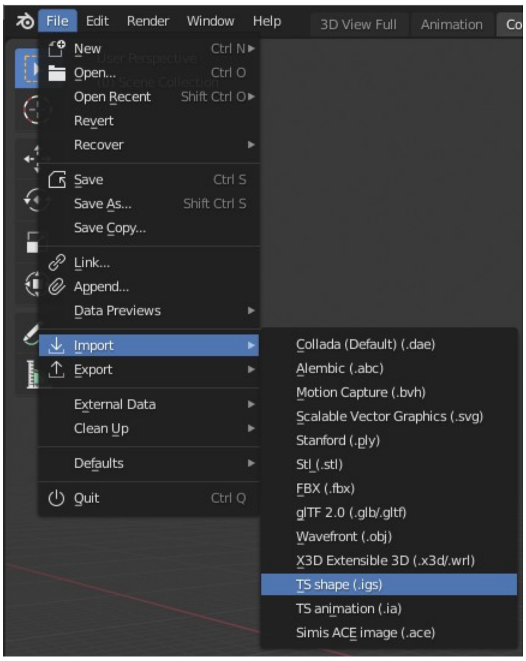
9.13 Some object names have been modified to comply with lod naming rules



```
>-~-~ IGS Exporter ~-~-<
WARNING        Some object names have been modified to comply with lod naming
               rules. Please look at the log to see which.
```

“Some object names have been modified to comply with lod naming rules. Please look at the log to see which.” This message appears when some objects names used as group names have been modified to comply with TS lod naming rules from what is written in Blender’s outliner. They could have been shortened, elongated or replaced altogether to prevent duplicate object names in IGS file: i.e. when two different names when shortened become identical.

10 Importers



The use of the 3 included importers is quite easy, with a few cases to know about the IA importer.

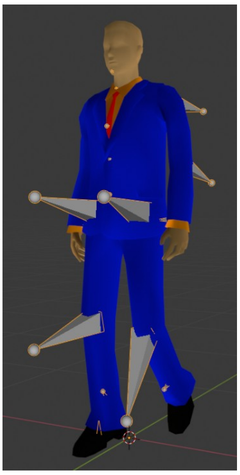
10.1 IGS Importer

Click File > Import > TS shape (.igs) then select a file, confirm and wait.

Your IGS files will load on the active scene collection, you may move them afterwards. Bones will be located in one armature for the whole file and named after it.

10.2 IA Importer

Click File > Import > TS animation (.ia) then select a file, confirm and wait.



If you have loaded the proper IGS file before on the active scene, objects will be animated accordingly.

Missing nodes will be created and animated as empties and bones, named according to what is stated in the ia file.

Beware, if the sample rate is different from your current render fps setting, FPS base will be altered to enable the insertion of all subframes.

10.3 Simis ACE Importer

Click File > Import > Simis ACE image (.ace) then select one or multiple files, then confirm and wait.

If the ace is not DXT compressed, it will create a new image in Blender Image list. Feel free to assign it inside a texture slot or re-export it to png to travel down 1998 memory lane! Only the top mip is loaded in Blender (lack of mip mapping support for custom image opener).

This has been added for compatibility, I strongly advise you *NOT TO USE ACE FILES IN TS*. They are obsolete. Prefer RGBA8 uncompressed DDS instead: they store the exact same info but everybody can read and write them. Do yourself and others a favour and deposit your idea of using it in *Ye Olde Trashe Bine*, near your *Voodoo 3* GFX card.

11 Annexes

11.1 Shaders list

This list is exhaustive: other shaders are used directly by game engine or unfinished, contain bugs or just less useful copies of other listed shaders in the use which can be made of them from a 3D artist point of view.

This table is meant to give informations about which slot each texture type should use. Hence, shaders which don't have their source file(s) included with TS distribution cannot be listed.

Cubemaps can be replaced by dummy textures, TS always uses its default cubemap instead.

All *.fx shaders support at least 1-bit alpha transparency mode when Material IGS option Transparency is ticked. Even if "RGBx" is specified. *I don't know any exception to this rule*

Images contains usually 4 components: RGBA (3 colours [RGB], 1 transparency [A]). Unused components are replaced with an "x". By default, "RGBA" designate here 8-bit per channel textures used as diffuse maps with a transparency channel unless stated otherwise.

"xxxx" represent dummy textures: none of their channels will be used whatever they contain. It is advisable to keep these as small as possible (e.g.: 8×8).

Shaders table colour code	*ConvertToGeo.exe* known supported shaders (App.fbk list)	Blender / 3DsMax UI listed shaders	Shaders I wanted to add
ConvertToGeo.exe known supported shaders (App.fbk list)	*ShaderName*	*ShaderName*	*ShaderName*
Blender / 3DsMax UI listed shaders	*ShaderName*	*ShaderName*	*ShaderName*
Shaders I wanted to add	*ShaderName*	*ShaderName*	*ShaderName*

Some Blue listed shaders might bug *ConvertToGeo.exe* while exporting producing "invalid parameters" errors or provoke an "infinite export" bug.

Each Grey listed shader have been tested and is working properly in TS.

Shader name	Description	Slot 1	Slot 2	Slot 3	Slot 4
AddAlphaDiff	No texture, additive vertex alpha with diffuse colour				
AddATex	Texture mapped, no lighting applied, using additive alpha from texture's alpha channel	RGBA additively drawn			
AddATexAlphaDiff	Texture mapped, with diffuse colour, using additive alpha from texture's alpha channel combined with vertex alpha	RGBA additively drawn			
AddATexDiff	Texture mapped, with diffuse colour, using additive alpha from texture's alpha channel	RGBA additively drawn			
AddDiffuse	No texture, with diffuse colour, using additive alpha				
AddTex	Texture mapped, no lighting applied, using additive alpha	RGBA additively drawn			
AddTexAlphaDiff	Texture mapped, with diffuse colour, with additive vertex alpha	RGBA additively drawn			
AddTexDiff	Texture mapped, with diffuse colour, using additive alpha	RGBA additively drawn			
BlendAlphaDiff	No texture, vertex alpha blending with diffuse colour				
BlendATex	Texture mapped, no lighting applied, using alpha blending from texture's alpha channel	RGBA			
BlendATexAlphaDiff	Texture mapped, with diffuse colour, using alpha blending from texture's alpha channel combined with vertex alpha	RGBA			
BlendATexDiff	Texture mapped, with diffuse colour, using additive alpha from texture's alpha channel	RGBA additively drawn			
BlendATexDiffTrans	Texture mapped, diffuse colour, alpha blending from texture's alpha channel, pixels with alpha=0 are transparent (e.g. alphaed fences).	RGBA			
BlendTexAlphaDiff	Texture mapped, with diffuse colour, with vertex alpha blending	RGBA			
BridgeSplit	Not drawn. Use to define areas where track crosses over itself.				
Clouds.fx	RailWorks specific shader for sky dome cloud layer	RGBA			
Diffuse	No texture, just diffuse colour				
DiffuseBumpmap	Unknown	RGBx	RGBx Normal	RGBx Cubemap	

Shader name	Description	Slot 1	Slot 2	Slot 3	Slot 4
DualAddATexDiffDestBlend	Dual textured, diffuse colour, first pass additive, and second pass blended alpha with the alpha of the first texture (e.g. puddles).	RGBA additively drawn	RGBA		
DualBlendATexDiffAdd	Dual textured, with diffuse colour, using alpha blending for first pass and additive alpha for second pass	RGBA	RGBA additively drawn		
DualTexDiffAdd	Dual textured, with diffuse colour, using additive alpha for second texture	RGBA	RGBA additively drawn		
DualTexDiffAddWithLightIntens	Add second pass to first pass, brightness of second pass affected by lightmaps if used	RGBA	RGBA additively drawn		
DualTexDiffAddWithoutLightIntens	Add second pass to first pass, brightness of second pass not affected by lightmaps if used	RGBA	RGBA additively drawn		
DualTexDiffInvisibleStencilBlend	Dual textured, with diffuse colour, first pass invisible, second pass alphaed using alpha of first pass texture	xxxA	RGBx		
DualTexDiffStencilAdd	Dual textured, with diffuse colour, using additive alpha for second texture only where first texture has solid alpha	RGBA	RGBA A*slot1 A additively drawn		
DualTexDiffStencilBlend	Dual textured, with diffuse colour, using blended alpha for second texture only where first texture has solid alpha	RGBA	RGBA A*slot1 A		
DualTexDiffTAlpha	Dual textured, with diffuse colour, using second texture's alpha channel to blend between textures	RGBx	RGBA		
DualTexDiffTrans	Dual textured, with diffuse colour, using second texture's transparency	RGBx	RGBA		
DualTexDiffVAlpha	Dual textured, with diffuse colour, using vertex alpha to blend between textures	RGBx	RGBx		
EmbossBumpmap	Bumpmap for Train 2 prototype or something like that. Takes 2 OR 3 RGBA textures.	RGBA	RGBx Normal	(RGBA) optional	
InvertATexDiff	Unknown *following naming conventions: Texture mapped, with diffuse colour, using inverted additive alpha from texture's alpha channel*	RGBA			
Invisible	Nothing is drawn; use for invisible collision barriers *or to be able to select viewer facing trees*				

Shader name	Description	Slot 1	Slot 2	Slot 3	Slot 4
LoftBump.fx	RailWorks specific shader for lofts with diffuse and bump channels	RGBx	RGBx Normal		
LoftBumpAlpha.fx	RailWorks specific shader for lofts with diffuse/alpha and bump channels	RGBA	RGBx Normal		
LoftBumpTrans.fx	RailWorks specific shader for lofts with diffuse/ 1-bit alpha and bump channels	RGBA 1-bit alpha	RGBx Normal		
LoftPostDPPTexDiffSpec.fx	RailWorks specific shader for lofts with diffuse channel and fixed specular post DPP stage and shadow mapped *Supports UVarg1 see 6.4*	RGBA 1-bit alpha			
LoftTexDiff.fx	RailWorks specific shader for lofts with diffuse channels	RGBx			
LoftTexDiffAlpha.fx	RailWorks specific shader for lofts with diffuse channels blended between 2 lofts	RGBA			
LoftTexDiffSpec.fx	RailWorks specific shader for lofts with diffuse channel and fixed specular *Supports UVarg1 see 6.4*	RGBx			
LoftTexDiffTrans.fx	RailWorks specific shader for lofts with diffuse channels	RGBA 1-bit alpha			
LoftTunnelBumpSpec.fx	RailWorks specific shader for lofts with diffuse and bump channels generated AO *Supports UVarg1 see 6.4*	RGBx	RGBx Normal		
LoftTunnelTexDiff.fx	RailWorks specific shader for lofts with diffuse channels generated AO	RGBx			
LoftTunnelTexDiffTrans.fx	RailWorks specific shader for lofts with diffuse channels generated AO	RGBA 1-bit alpha			
SkinAmbient.fx	Skinned geometry specific shader Nothing applied, texture mapped looks unfinished	RGBx			
SkinDiffuse.fx	Skinned geometry specific shader with diffuse channel	RGBx			

Shader name	Description	Slot 1	Slot 2	Slot 3	Slot 4
SkinDiffusePallete2.fx	Skinned geometry specific shader with diffuse channel *Palettised, see 11.5.3*	R: palette 1 mask G: palette 2 mask x A: luminosity	RGBA palette 1 256xY image use "_nocomp"	RGBA palette 2 256xY image use "_nocomp"	
SkinGloss.fx	Skinned geometry specific shader with diffuse, bump and specular channels *Supports UVarg1 see 6.4*	RGBx	RGBx Normal	Rxxx Specular	
SkinNormal.fx	Skinned geometry specific shader with diffuse and normal channels	RGBx	RGBx Normal		
SkinNormalPallete2.fx	Skinned geometry specific shader with diffuse channel *Palettised, see 11.5.3*	R: palette 1 mask G: palette 2 mask x A: luminosity	RGBA palette 1 256xY image use "_nocomp"	RGBA palette 2 256xY image use "_nocomp"	RGBx Normal
SkinRESERVED1.fx	Skinned geometry specific shader unfinished	RGBA	xxxx Normal	xxxx Specular	
SkinRESERVED2.fx	Skinned geometry specific shader unfinished	RGBA	xxxx Normal (no forced linear filtering)	xxxx Specular	
SkinRESERVED3.fx	Skinned geometry specific shader unfinished	RGBA	xxxx Normal	xxxx Specular	
SkinSpecular.fx	Skinned geometry specific shader with diffuse channel and specular lighting *Supports UVarg1 see 6.4*	RGBx			
SpecularBumpmap	unknown	RGBx	RGBx Normal	RGBx Cubemap	
Stars.fx	RailWorks specific shader for sky dome stars	RGBA			
StencilShadow.fx	Stencil shadow objects, material must begin with shadow_ to be detected	xxxx			
SubtractATexDiff	Unknown *following naming conventions: Texture mapped, with diffuse colour, using subtractive alpha from texture's alpha channel*	RGBA subtractive draw			
Tex	Unlit mapped texture	RGBA			

Shader name	Description	Slot 1	Slot 2	Slot 3	Slot 4
TexDiff	Texture mapped with single texture, diffuse colour applied	RGBA			
TrainBasicObjectDiffuse.fx	RailWorks specific shader with diffuse channels	RGBx			
TrainBasicObjectUnLit.fx	RailWorks specific shader with diffuse channels. Unlit. *Good replacement for Tex.*	RGBx			
TrainBasicObjectDiffusePallete2.fx	RailWorks specific shader with diffuse channels *Palettised, see 11.5.3*	R: palette 1 mask G: palette 2 mask x A: luminosity	RGBA palette 1 256xY image use "_nocomp"	RGBA palette 2 256xY image use "_nocomp"	
TrainBasicObjectSpecular.fx	RailWorks specific shader with diffuse channel and specular lighting *Supports UVarg1 see 6.4*	RGBx	RGBx Specular		
TrainBump.fx	RailWorks specific shader with diffuse and bump channels *I use it regularly because I don't need Env mapping.*	RGBx	RGBx Normal		
TrainBumpEnv.fx	RailWorks specific shader with diffuse and bump channels plus environment map *Supports UVarg1 see 6.4*	RGBx	RGBx Normal	RGBx Cubemap	
TrainBumpEnvMask.fx	RailWorks specific shader with diffuse and bump channels and a static environment mask *Supports UVarg1 see 6.4*	RGB diffuse A: Envmap	RGBx Normal	RGBx Cubemap	
TrainBumpSpec.fx	*RailWorks specific shader with diffuse and bump channels and specular lighting* *Supports UVarg1 see 6.4*	RGBx	RGBx Normal		
TrainBumpSpecEnv.fx	RailWorks specific shader with diffuse, specular and bump channels plus environment map *Supports UVarg1 see 6.4*	RGB diffuse A: Specular	RGBx Normal	RGBx Cubemap	
TrainBumpSpecEnvMask.fx	RailWorks specific shader with diffuse and bump channels and specular lighting and environment map *Supports UVarg1 see 6.4*	RGB diffuse A: Specular and Envmap	RGBx Normal	RGBx Cubemap	

Shader name	Description	Slot 1	Slot 2	Slot 3	Slot 4
TrainBumpSpecEnvPalette1.fx	RailWorks specific shader with diffuse, specular and bump channels plus environment map *Supports UVarg1 see 6.4* *Palettised, see 11.5.3*	R: palette 1 mask x x A: luminosity and Specular	RGBA palette 1 256xY image use “_nocomp”	RGBx Normal	RGBx Cubemap
TrainBumpSpecMask.fx	RailWorks specific shader with diffuse channel, specular lighting and a static environment mask *Supports UVarg1 see 6.4*	RGB: diffuse A: Specular	RGBx Normal		
TrainDecal.fx	RailWorks specific shader with special case of decals	RGBA			
TrainEmissiveGlow.fx	RailWorks specific shader for glowing emmisive materials	RGBA	Rxxx Emissive		
TrainEnv.fx	RailWorks specific shader with diffuse channel and a static environment *arg2: Reflection intensity*	RGBx	RGBx Cubemap		
TrainEnvMask.fx	RailWorks specific shader with diffuse channel and a static environment mask	RGB: diffuse A: Envmap	RGBx Cubemap		
TrainFlora.fx	RailWorks specific shader with diffuse channels	RGBx			
TrainGlass.fx	Glass shader with env map *Supports UVarg1 see 6.4* *arg2: Reflection intensity* *arg3: Reflection illumination(night)*	RGBA	RGBx Cubemap		
TrainGlassWeatherEffects.fx	Glass shader with weather effects for in cab rain Reflective glass with cubic reflection map and diffuse. *See: 8.7.5 - Setting-up TrainGlassWeatherEffects.fx*	RGBA	~~RGBx Cubemap~~ not in BRIAGE	RGBx Object Normal	xxxA Wipers motion
TrainLightBumpSpecMask.fx	RailWorks specific shader with bump map, diffuse and light map channels, specular mask in bump alpha *Supports UVarg1 see 6.4*	RGBx	RGB: Normal A: Specular	Rxxx LM or AO	
TrainLightMapWithDiffuse.fx	RailWorks specific shader with light map and diffuse channels	RGBx	Rxxx LM or AO		
TrainPostDPPObjectDiffuse.fx	RailWorks specific shader with diffuse channels post DPP stage and shadow mapped	RGBx			
TrainShadowOnly.fx	RailWorks specific empty shader - should never be called, used as a placeholder for shadow only rendering				

Shader name	Description	Slot 1	Slot 2	Slot 3	Slot 4
TrainShadowOnlyDiffuse.fx	RailWorks specific empty shader - should never be called, used as a placeholder for shadow only rendering with alpha	xxxA 1-bit alpha			
TrainSkyDome.fx	Sky dome shader, dynamically vertex lit by game engine.	xxxx	xxxx		
TrainSpecEnv.fx	RailWorks specific shader with diffuse channel, specular and environment map *Supports UVarg1 see 6.4*	RGBx	RGBx Cubemap		
TrainSpecEnvMask.fx	RailWorks specific shader with diffuse channel, specular lighting and a static environment mask *Supports UVarg1 see 6.4*	RGB: diffuse A: Specular and Envmap	RGBx Cubemap		
TrainUberShader.fx	RailWorks specific shader with diffuse and bump map textures plus a 3rd texture with: specular mask, AO mask and emissive mask *Supports UVarg1 see 6.4*	RGBx	RGBx Normal	R: LM or AO G: Emissive B: Envmap A: Specular	
TrainUprightViewFacingFlora.fx	RailWorks specific shader for view facing upright flora	RGBx			
TrainVertexLit.fx	Unknown *Blends mapped texture with vertex colours. Buggy when skydome behind. No transparency at all.*	RGBx			
TrainVertexLitWithDiffuse.fx	Unknown *Mapped texture. Buggy when skydome behind.*	RGBx			
TrainViewFacingFlora.fx	RailWorks specific shader for view facing flora	RGBx			
TrainViewFacingUnlit.fx	RailWorks specific shader for view facing unlit	RGBx			
TripleGlossMap	Triple texture, 2nd pass contains gloss map in alpha channel, 3rd pass (reflection) texture drawn additively	RGBA	xxxA Gloss	RGBA* Gloss reflection to display additively drawn	
TripleGlossMapWithLightIntens	Triple texture, 2nd pass alpha channel gloss map, 3rd pass drawn additively affected by lightmaps if used	RGBA	xxxA Gloss	RGBA* Gloss reflection to display additively drawn	
TripleGlossMapWithoutLightIntens	Triple texture, 2nd pass alpha channel gloss map, 3rd pass drawn additively not affected by lightmaps if used	RGBA	xxxA Gloss	RGBA* Gloss reflection to display additively drawn	

Shader name	Description	Slot 1	Slot 2	Slot 3	Slot 4
TripleTexDiffAddAdd	Triple textured, 2nd and 3rd passes are drawn additively	RGBA	RGBA additively drawn	RGBA additively drawn	
TripleTexDiffTAlpha	Triple textured, with diffuse colour, using each texture's alpha channels to blend between each pair of passes	RGBA	RGBA	RGBA	
TripleTexDiffTAlphaVAlpha	Triple textured, with diffuse colour, pass 2 uses texture alpha for blending, pass 3 uses vertex alpha for blending	RGBA	RGBA	RGBx	
TripleTexDiffVAlpha	Triple textured, with diffuse colour, using same vertex alpha to blend between each pair of passes	RGBA	RGBx	RGBx	
TripleTexDiffVAlphaTAlpha	Triple textured, with diffuse colour, pass 2 uses vertex alpha for blending, pass 3 uses texture alpha for blending	RGBA	RGBx	RGBA	
WaterCubeMap.fx	RailWorks specific shader for reflective water *Supports UVarg1 see 6.4* *arg2: 'Wobble' factor*	RGBx Normal	xxxx	RGBx Cubemap	
WaterFlow.fx	RailWorks specific shader for reflective water *Supports UVarg1 see 6.4* *arg2: 'Wobble' factor*	RGBx Normal	xxxx	RGBx Cubemap	
WaterScenery.fx	RailWorks specific shader for reflective water *Supports UVarg1 see 6.4* *arg2: 'Wobble' factor* *arg3: Movement speed* *arg4: Oscillation height* *arg5: Oscillation scale*	RGBx Normal	RGBA	RGBx Cubemap	

11.2 Shaders usage examples

Use	**Shader name**	**Main texture (slot 1)**	**Bump map**	**Cube map**	**Additional settings**
Light halo around lightsources (e.g. streetlights)	AddATex	name.dds (transparency in alpha channel)			→ IGS options *ZBufferMode* set to *Test only* → *IGS options Viewer facing options* set to *Viewfacing or UprightViewFacing* → *Mesh data tab*: one vertex colors channel active and renderable. See 8.7.4 → If shown at night only, append “`_night`” at the end object name
Headlight or rearlight glass	AddATex	name.dds (transparency in alpha channel)			→ IGS options *ZBufferMode* set to *Test only*
Simple windows only	BlendATexDiff	name.dds (transparency in alpha channel)			→ IGS options *ZBufferMode* set to *Test only*
Animated texture *see 6.5 - UV Special effects*	TexDiff	name_*anim1*.dds which is the first file of the animation.			→ IGS options *AnimateUVs* enabled → *#Frames* and *FPS* according to the animation to implement.
Scrolling text *see 6.5 - UV Special effects*	TexDiff or AddATex	name.dds (transparency in alpha channel)			→ IGS options *Scroll UVs* enabled → *u* and *v* according to the speed and direction of scroll to implement. → IGS options *ZBufferMode* set to *Test only* if transparency needed
Solid texture with holes	TrainBasicObjectDiffuse.fx	name.dds (transparency in alpha channel. Only black or white (1-bit))			→ IGS options *Transparency* enabled
Solid texture	TrainBasicObjectDiffuse.fx	name.dds			
Texture with normal map	TrainBump.fx	name.dds	name_nm.dds		
Texture with specular effects and normal maps	TrainBumpSpecEnvMask.fx	name.dds (specular in alpha channel)	name_nm.dds	env.dds	→ UV arguments suggested values: arg1: 32.0 (all other values = 0.0) *see 6.4 for more infos*

Use	**Shader name**	**Main texture (slot 1)**	**Bump map**	**Cube map**	**Additional settings**
Solid texture with holes for inscriptions (such as rolling stock numbers)	TrainDecal.fx	decal_name.dds (transparence in alpha chanel: any grey value between black and white)			→ IGS options *ZBufferMode* set to *Test only* (needed for the alpha channel to be properly processed as a transparent layer)
Windows with reflections	TrainGlass.fx	name.dds (transparency in alpha channel)		env.dds	→ UV arguments suggested values: arg1: 64 arg2: 0.8 arg3: 0.4 (all other values = 0.0) *see 6.4 for more infos*
Texture with specular effects	TrainSpecEnvMask.fx	name.dds (specular in alpha channel)		env.dds	→ UV arguments suggested values: arg1: 32.0 (all other values = 0.0) *see 6.4 for more infos*
2D Vegetation bushes	TrainUprightViewFacingFlora.fx	name.dds (transparency in alpha channel. 1-bit)			→ IGS options *Transparency* enabled → *Smaller IGS* must be checked.(see 4.3) → *IGS options Viewer facing options* set to *Auto*
2D Vegetation leaves	TrainViewFacingFlora.fx	name.dds (transparency in alpha channel. 1-bit)			→ IGS options *Transparency* enabled → *Smaller IGS* must be checked.(see 4.3) → *IGS options Viewer facing options* set to *Auto*

Report to 11.1 - Shaders list for more precise informations.

11.3 TS20XX naming rule check

> All objects should follow strict naming conventions.
>
> Each name starts with a single digit representing the LOD level, followed by a 4-digit visible distance between underscores. After this a logically chosen object name follows and the whole name is limited to a maximum of 31 characters.
>
> ***From Railworkswiki***

Operations are performed after all name modifications have been applied.

The exporter checks the beginning of the name looking for a digit (LOD level), an underscore, 4 digits (visible distance), an underscore and the name.

If the exporter doesn't find 1 digit, then an underscore, then a number, then an underscore, the name will be automatically converted to LOD = 1, distance = 1000, followed by the original name. For example, a Blender object named “MyObject” or "1-0100_MyObject" will be converted as, respectively, “1_1000_MyObject” or "1_1000_1-0100_MyObject".

Otherwise, a best fit to the number is found. For example:

- 1_02_object is converted as 1_0200_object
- 1_2_object is converted as 1_2000_object
- 2_10000_object is converted as 2_1000_object

Warning messages are written to the log file about misnamed objects which name is converted.

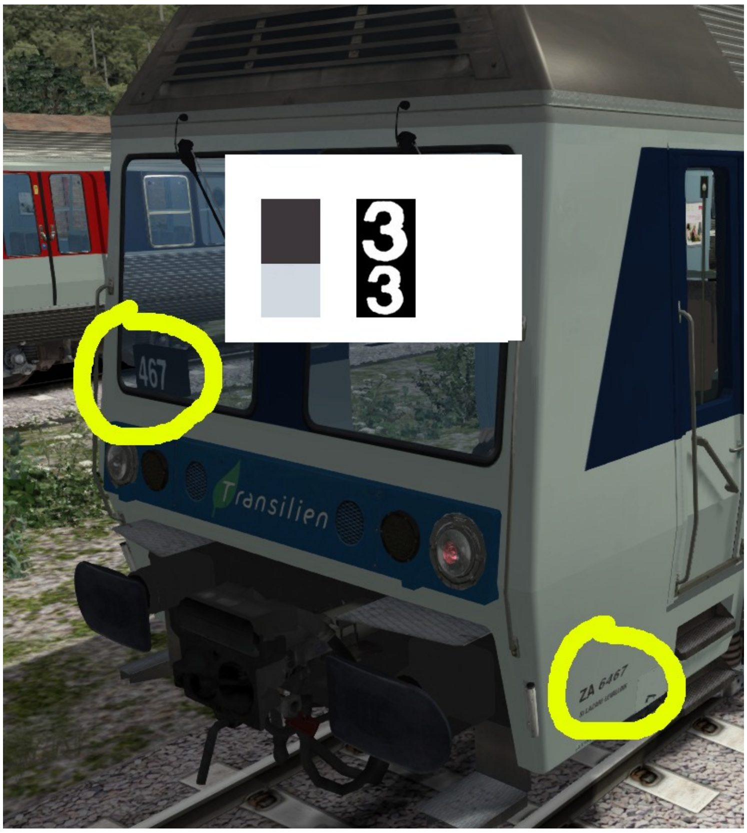
11.4 Dynamic text

Please refer to the excellent tutorial on:
https://www.christrains.com/en/ts_faq_autonumbering.html (The paragraph about "*Multi/Sub-Object* material" is specific to 3DSMax.)

Please note that it may be necessary to import again the equipment in a scenario so that the game takes into account changes in xml files or in the Blender model.

In addition to this tutorial, here is how 3 numbering objects with 3 digits were implemented in Blender:

They are respectively named 1_0050_primarydigits_3 (for the number on the right of the image), 1_0050_primarydigits_3.001 (for the number on the opposite side) and 1_0050_primarydigits_3.002 (for the front number).

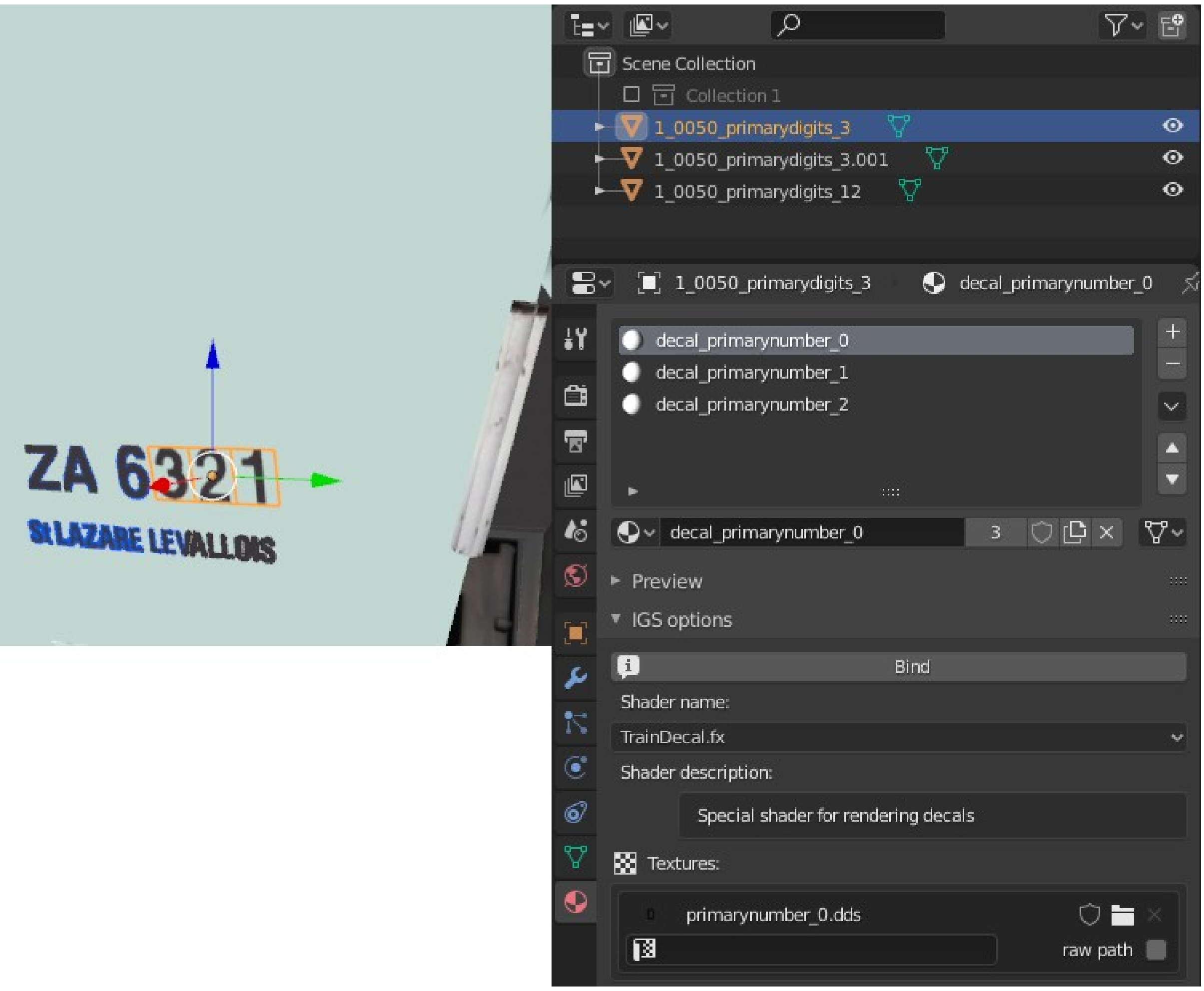


The white rectangle shows the main bicolour texture and the alpha alpha texture for digit 3 included in a 64×128 pixels DDS file. The faces for the side grey number crop the grey area and the faces for the front white number crops the white area.

The text is set here to “467”.

The 1_0050_primarydigits_3 object (1_yyyy_primarydigits_ followed by the number of digits = 3) is a plane divided into 3 faces. The other *1_0050_primarydigits_3.00x* objects are created by duplication.

Each face is assigned a material: primarynumber_0 (units), primarynumber_1 (tens digit) ou primarynumber_2 (hundreds digit).



Shader name: TrainDecal.fx

Shader description: Special shader for rendering decals

In detail, each of these 3 materials is defined by (N is set to 0, 1, 2, ..., 29, ..., 99, and perhaps even higher):

- Material name: primarynumber_N
- Image in UV-editor: decal_primarynumber_N.dds
- Selected shader: `TrainDecal.fx`
- Material first texture slot: texture named decal_primarynumber_N associated in the UV-editor to image decal_primarynumber_N.dds and associated to texture file `decal_primarynumber_N.dds`.

File decal_primarynumber_0.dds is created by copying and renaming *number_1.dds* (thus digit 1 is displayed in the above screenshot. This digit will be replaced in game by the corresponding value read in the csv file).

File *decal_primarynumber_1.dds* is created by copying and renaming *number_2.dds* (thus digit 2 is displayed in the above screenshot).

File *decal_primarynumber_2.dds* is created by copying and renaming *number_3.dds* (thus digit 3 is displayed in the above screenshot).

Nota: Be careful when creating `NamedTextureSets` for your dynamic text. `TextureName` field of each entry must contain *1 ASCII symbol*, else your text won't display at all.

11.5 Texturing lexical notes

11.5.1 3 components maps (RGB)

Diffuse map (*Diff* or *albedo*) contains **colour** informations.

Normal map (or just *Normal*) holds informations about fake **details**. i.e. , details to heavy and too little to be included in final 3D work are printed to a normal map to gain a performance boost. This can be useful for vents, screws, metal panels, hinges, windows, etc...

Cubic map (,*Cubemap* or *Skybox*) is a special texture type used to **fake reflections** on old rendering engines. It contains 6 images mapped inside a virtual cube. They should look like the real mirrored environment around your object for more realism. These can be contained in DDS files; use GIMP or Ps to be able to manipulate them.

11.5.2 1 component maps (greyscale)

Transparency is holding **opacity** informations, it is common to find it nicknamed as Alpha because it is often stored in Alpha channel of a diffuse texture. Its value blends current texture with what stands behind.

Specular map (or *Spec*) specifies **specular intensity** ("specular power" in TS: see 6.3) at given places of the texture . This is used to fake a particular real life material light properties, making it shiny or dull.

Gloss map (or simply *Gloss*) specifies **specular size** ("Phong" size in TS: see 6.4, Hardness in Blender, "tightness") at given places of the texture . This is used to fake a particular real life material light properties, making it glossy or diffuse.

Environment map (or *Envmap*) holds **reflectiveness** of a material making it more or less mirror-like.

Light map (*LM*), which save shadows from all light sources, and *Ambient occlusion* (*AO*) map, which save shadows from bouncing environmental light (in more or less fake way), can be considered the same in TS as they are treated the same way. Both contain fixed **shadows** which are altered by dynamic TS light sources and occlusion volumes. However, you can mix them to your diffuse maps or use them as stencil shadows (e.g. with SubtractATexDiff) to make them unalterable by game engine.

Emissive map specifies which part of a material **emits light** and in which amount. This can be used to make glowing effects or to cut off a light build into a wall texture.

Greyscale maps sometimes need to be cast to RGB components (just copy values to all channels) when TS needs it.

11.5.3 Palettisation

This refers to a semi-random way of generating variations for diffuse channels in TS.
Take a look at this document:
'*TS directory*'/dev/Docs/TS2014Tech_P1.pdf, p19, ***6. Palettisation***

Maintainance and contact infos

Last modification:
> Julian MERLE-REMOND - 19:22, 29/05/23

Redactor coordinates :

Name:	Julian MERLE-REMOND
e-mail :	**julian.merle-remond@devmares.net**

Made with love, dedication and boredom by your servant Juju49.

Total time elapsed on this document: 14:17:26

Documentation - EN - v3.20.835 - Blender 2.8x BRIAGE v315 - EN.odt

If you want to translate this manual or BRIAGE interface in another language: please contact document maintainer for *.odt source files and/or to include your work in the main BRIAGE distribution.