Trainz™ Custom Content Creation Guide

Contents

Introduction

Welcome to the *Trainz™ Custom Content Creation Guide*. The purpose of this document is to assist the production and installation of custom *Trainz™* assets. We are assuming that third party developers have a sound knowledge of 3DS Max™ and therefore only give references to model requirements, rather than a modeling tutorial.

We have included recommended distribution guidelines to ensure end users have correct and easy content installation.

Trainz™ Assets

This section gives brief guidelines for creation of models, texture files, an explanation of naming conventions, config files, and directory structure.

The default location for all custom content is:

C:\Program Files\Auran\Trainz\Custom\

Config.txt

Each custom content folder, from trains to ground textures, contains a file called config.txt. The config.txt file contains the parameters that determine how Trainz™ uses the contents of the folder.

Refer to the Trainz_Dev folder on your Trainz™ cd for source material.
Here you will find Trainz_custom and Source_files folders. The Trainz_custom folder contains all *.pm, config and textures files of trains and bogeys. It also contains sample *.pm and config files of other Trainz™ assets.
In the Source_files folder you will find example 3Dstudio Max and GMax files including their texture maps. The Trainz™ GMax exporter supports the following material properties; diffuse, glossiness (reflection) and opacity. As yet it does not support Mesh Weights for use with progressive meshes. For this reason we would recommend the lower polygon limit for train bodies.

Trains

3D Studio MAX™ Guidelines: Refer to: …\Sample_content\Source_files\Trains

Polygon limits:
Train **body** polygon recommendations (excluding bogies) = 3500-6000 polygons. Less is better ☺

The front end of the train body should be on the LHS when displayed in the right viewport in 3DSMax.

Train body **shadow** polygon recommendations = 1000 polygons or less modeled to the same basic shape and 3D space as the body. No attachments are required within the shadow file.

Attachment points: (MAX: 'Create' tab, 'Helpers', 'Point')
To maintain correct alignment, attachment points should be created in the TOP viewport in 3DSMAX.

These are 'points' in 3D space giving information on various aspects of the train as follows:

a.limfront
- marks the front of the train, used for coupling
- should be roughly the same distance from origin as a.limback
- bogeys can be further forward than a.limfront if desired
- determines the forward headlight position
- height above origin (or Z) = 0.89m (2' 10.8")

a.limback
- marks the rear of the train, used for coupling
- see a.limfront
- height above origin (or Z) = 0.89m (2' 10.8")

a.bog0
- front bogey attachment
- used for positioning the train on the track
- positioned at absolute centre of front bogey

a.bog1
- rear bogey attachment
- used for positioning the train on the track
- positioned at absolute centre of rear bogey

a.bog* (2, 3, etc)
- any other bogey attachments

a.exhaust* (0, 1, etc..)
- smoke generator attachments (where needed)

a.light* (0, 1, etc..)
- light "corona" attachments

a.ditch* (0, 1, etc..)
- ditch light "corona" attachments

a.cabfront
- attachment point for the front cabin of a loco
- located at the centre of cabin

a.pant* (0, 1, etc..)
- attachment point for pantographs (where needed)

Carriage cars need only a.limfront, a.limback, a.bog0, and a.bog1

Train textures:
The materials are of *Multi/Sub-Object* type (one M/SO only per model) and we have used *UVW Map* and *Unwrap UVW* for texture allocation. Textures must be 8x8, 16x16, 32x32, 64x64, 128x128, 256x256, and 512x512 pixels. Maximum ratio = 1:8 e.g. 64x512.

Diffuse Maps: In many cases a single 512x512 16-bit .TGA file is sufficient to texture a locomotive. Occasionally an extra texture (say 128x256) can be added.

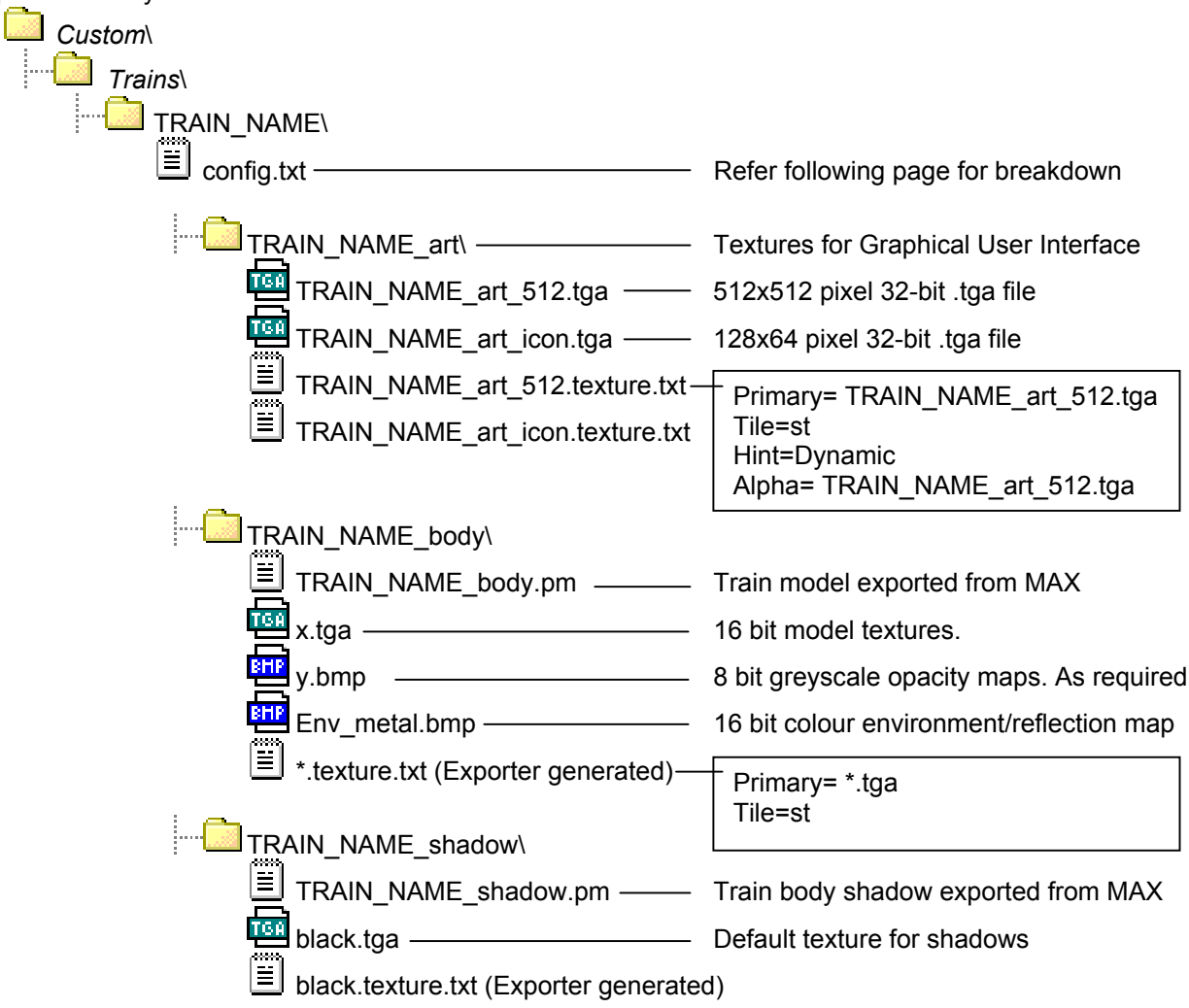
Reflection maps are supported (16 bit colour .bmp). We generally set train body reflection amounts (in MAX) to 10 and windows to 25. Opacity Maps (8 bit greyscale .bmp) are also supported to the same dimensions as the diffuse map. Reflection and Opacity maps must not be used together with-in the same texture.

Note: Dynamic locomotive numbering (alpha_numbers) for custom trains are not currently supported.

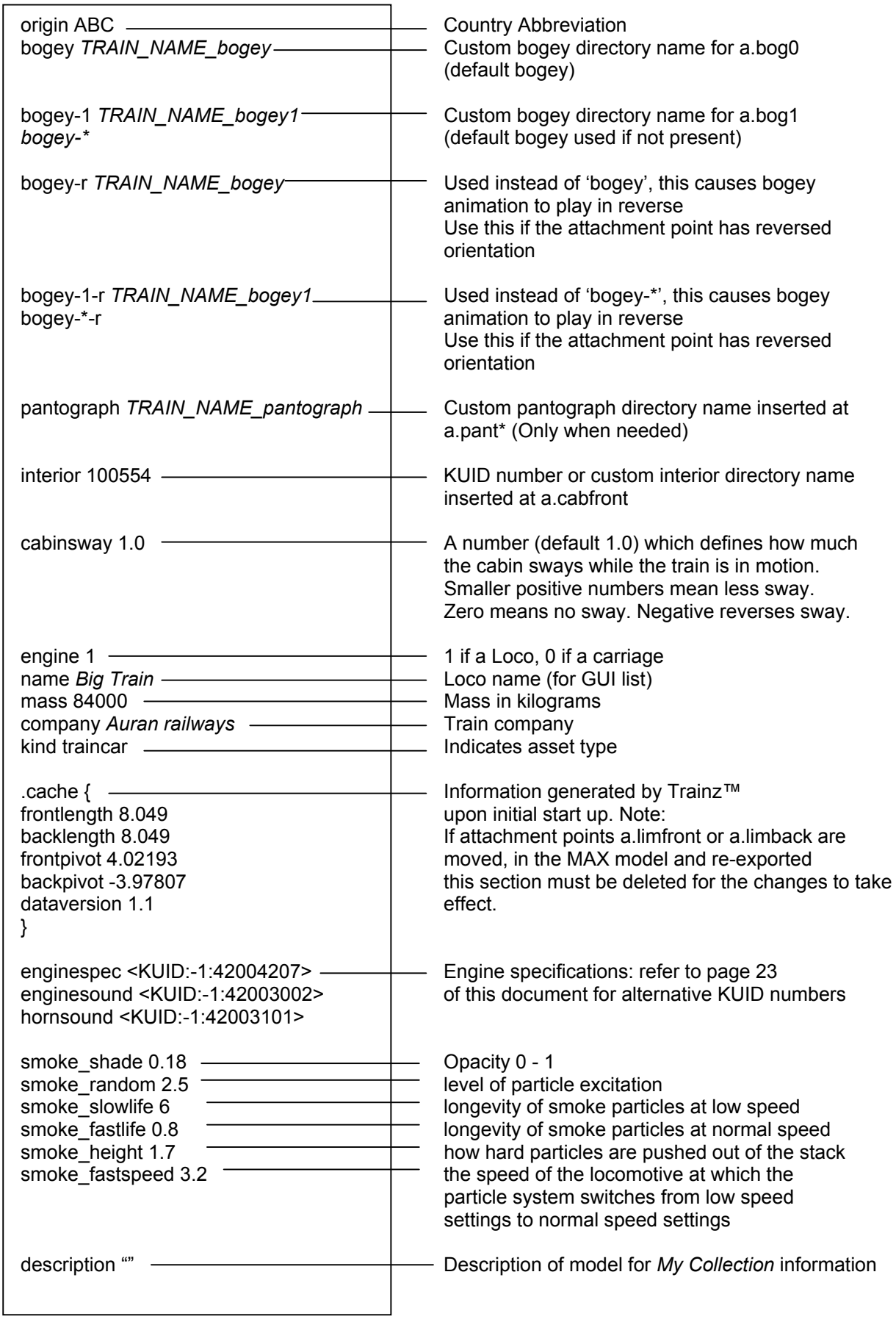
Exporting Models from Max:
Remember to collapse the 'Edit Stack' before exporting.

Export attachment points and model only using the Jet exporter MAX plug in; *Jet format (*.im,*.kin,*.pm)*. Remember naming conventions and to type in the file extension under *file name* (e.g.TRAIN_NAME_body.pm) Note: The Jet exporter also produces .gmw and .imw files (not required)

File Structure & Naming Conventions: Refer to: …\Sample_content\Trainz_custom_files\Trains
Typical directory structure for custom Trains should be:



C:\Program Files\Auran\Trainz\Custom\Trains\TRAIN_NAME\config.txt



Bogeys

3D Studio MAX™ Guidelines: Refer to: …\Sample_content\Source_files\Bogeys

Polygon limits:
Train **bogey** polygon recommendations = <2000 polygons per truck. Less is better ☺

Train bogey **shadow** polygon recommendations = <100 polygons per truck.

Carriage **bogey** polygon recommendations = <300 polygons per truck. Less is better ☺

Carriage bogey **shadow** polygon recommendations = <100 polygons per truck.

The absolute centre of bogeys should be located at World origin point (0,0,0)

Attachment points:
a.ground* (0, 1, etc..)
- slightly offset at the base of each wheel
- determines the wheel spark position

Bogey textures:
The materials are of *Multi/Sub-Object* type (one M/SO only per model) and we have used *UVW Map* and *Unwrap UVW* for texture allocation.

Diffuse Maps: Generally a single 128x128 16-bit .TGA file is sufficient to texture a bogey. Additional maps (e.g. for springs) are also used.

Opacity Maps (8 bit greyscale .bmp) are supported to the same dimensions as the diffuse map. Used regularly for carriage bogey sides. Reflection maps are supported but generally not used on bogey models.

Exporting Models from Max:
As per 'Trains' section. Remember naming conventions and to type in the file extension under *file name* (e.g.TRAIN_NAME_bogey.pm)

File Structure & Naming Conventions:

Typical directory structure for custom Bogeys should be:

- *Custom*\
  - *Bogeys*\
    - TRAIN_NAME_bogey\
      - config.txt - Refer following page for breakdown
      - TRAIN_NAME_bogey.pm - Bogey model exported from MAX
      - x.tga - 16 bit model textures.
      - y.bmp - 8 bit greyscale opacity maps. As required
      - *.texture.txt (Exporter generated)
        - TRAIN_NAME_bogey_shadow\
          - TRAIN_NAME_bogey_shadow.pm - Bogey shadow exported from MAX
          - black.texture.txt (Exporter generated)
          - black.tga - Default texture for shadows

C:\Program Files\Auran\Trainz\Custom\Bogeys\TRAIN_NAME_bogey\config.txt

```
Kind bogey
animdist 2.1
```

Indicates asset type
The distance travelled in meters by the bogeys in 1 second (30 frames) of animation.
Leave this tag out if the bogey is not animated
Bogey animations are called “anim.kin” (rename from exported file)

Interiors

3D Studio MAX™ Guidelines:

Refer to: …\Sample_content\Source_files\Interiors & …\Sample_content\Trainz_custom_files\Interiors

Generally there is a ‘Main’ model that includes various attachment points where other models are inserted. E.g. Speedo needles, levers etc.

Much information can be obtained by reviewing the sample files provided and therefore we shall not bore you with loads of waffle ☺

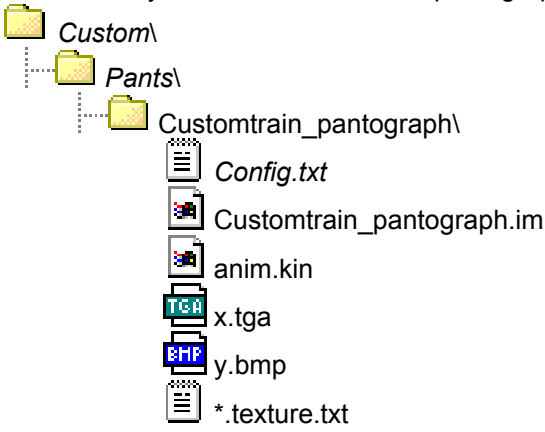
Developers must however follow strict naming conventions that refer to attachment points and models. E.g. samples from bb15000 interior config:

```
kind interior                          Indicates asset type
mesh bb15000_int_cab.pm                ‘Main’ model file
camera 0.8, 0.91, 0.4                  Camera position from a.cabfront

attachment {                           Start of attachments section

pantograph_lever {                     Designated name field
        kind lever                     Inserted model type
        mesh pantograph_lever.pm       Inserted model name
        att a.pantograph_lever         Attachment point
        limits 0, 1                    )
        angles 0, 1.0                  ) Rotational boundaries
        notches 0, 1                   )
        notchheight 3, 3               )
        radius 0.16                    ) Graphical notch boundaries
        }
ampmeter_needle {                      Designated name field
        kind needle
        mesh bar_meter_needle.pm
        att a.ampmeter_motor1
        limits 0, 1500
        angles 0, 0.6
        }
independantbrake_lever {               Designated name field
        kind lever
        mesh lever_handle.pm
        att a.ind_brake_lever
        limits 0, 4
        angles 0, -0.45
        notches 0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1
        notchheight 1,2,2,2,2,2,2,2,2,2,1
        radius 0.15
        }
Etc…
```

Pants (Pantographs)

Refer to: …\Sample_content\Source_files\Pants & …\Sample_content\Trainz_custom_files\Pants

Typical directory structure for a custom pantograph should be:



Config.txt:

```
kind pantograph
```

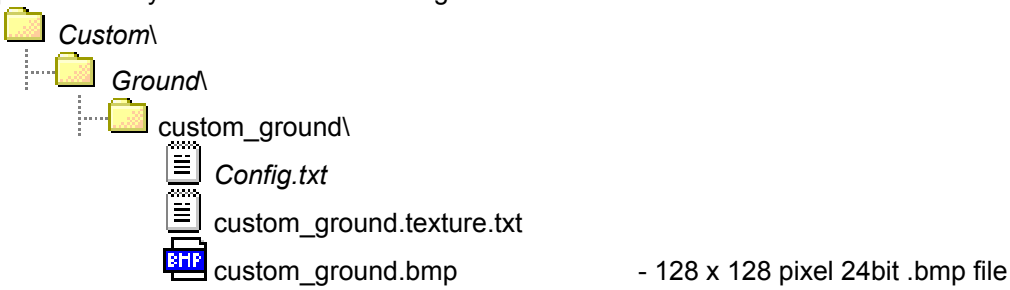
Maps

The maps folder stores terrain and track layout information. This directory contains folders that are generated by Trainz™ for each every layout.

Ground

Refer to: …\Sample_content\Trainz_custom_files\Ground
Custom textures can be used as ground textures.

Typical directory structure for a custom ground texture should be:



Config.txt:

```
kind groundtexture
rgb 0, 0, 0
```

rgb colour for use in minimap

custom_ground.texture.txt

```
Primary= custom_ground.bmp
Tile=st
```

image name for ground texture

Displacements

Refer to: …\Sample_content\Trainz_custom_files\Displacements
Displacement maps can be used to adjust the height and shape of an area of terrain.

Typical directory structure for a custom ground texture should be:

Custom\
  Displacements\
    custom_displacement.bmp - 256 x 256 pixel 24bit .bmp file

Note: Displacement maps do not require a config.txt file.

Environment

Refer to: …\Sample_content\Trainz_custom_files\Environment
The Environment folder accommodates the use of different types of sky and water in Trainz™.

Water

Custom water is based on a two image files, one for the surface texture and the other for the reflection.

Typical directory structure for a custom water type should be:

Custom\
  Environment\
    Custom_Water\
      Config.txt
      water_surface.texture.txt
      water_reflection.texture.txt
      water_surface.tga - 128 x 128 pixel 32bit .tga file
      water_reflection.bmp - 128 x 128 pixel 8bit greyscale .bmp file

Config.txt:

```
kind water
normal water_surface        ——— surface material
reflection water_reflection ——— reflection material
```

water_surface.texture.txt

```
Primary=Water_muddy.tga     image name for surface
Tile=st
```

water_reflection.texture.txt

```
Primary=Water_muddy.tga     ——— image name for reflection
Tile=st
```

Sky

Sky is generated from three source images.

Typical directory structure for a custom sky should be:

Config.txt:

```
kind environment
normal sky_example ——— sky material
storm sky_example_storm ——— storm sky material
night sky_example_night ——— night sky material
```

sky_example.texture.txt

```
Primary= sky_example.tga ——— image name for normal sky
Tile=st
```

sky_example_night.texture.txt

```
Primary= sky_example_night.tga ——— image name for night sky
Tile=st
```

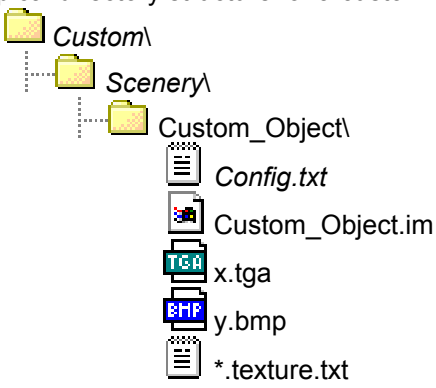
sky_example_storm.texture.txt

```
Primary= sky_example_storm.tga ——— image name for stormy sky
Tile=st
```

Scenery

Refer to: …\Sample_content\Trainz_custom_files\Scenery
Scenery objects can vary greatly in size and appearance. It is recommended to keep the models as simple as is reasonable regarding texture and polygon usage.

Typical directory structure for a custom scenery object should be:

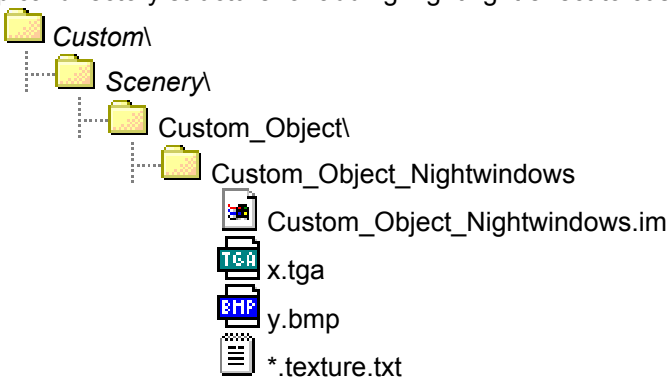


Config.txt:

```
kind scenery
light 1
```

Scenery objects with lights at night
Objects just as buildings or signs can be made to appear to be have lights on at night. A model that contains only the lit areas of the object can be exported into a subdirectory.

Typical directory structure for adding night light effect to custom scenery should be:



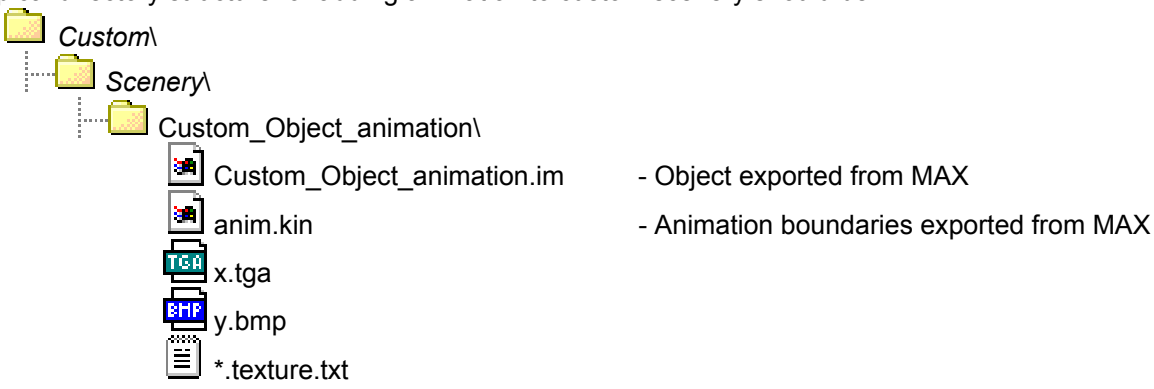
The following lines must be added to the object's config.txt file:

```
kind scenery
NIGHTMODE home
NIGHT Custom_Object_Nightwindows
```

Note: The *nightmode* can also be set as "lamp" if it is to remain lit all night.

Scenery objects with animation

Typical directory structure for adding animation to custom scenery should be:

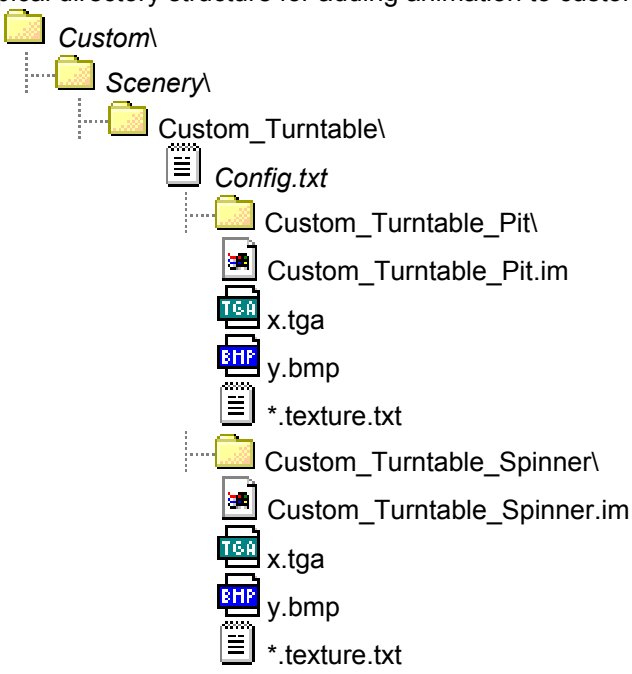


The following lines must be added to the object's config.txt file:

```
kind scenery
autoanimation 1
```

Special Scenery Objects - Turntables

Typical directory structure for adding animation to custom scenery should be:



The following lines must be added to the object's config.txt file:

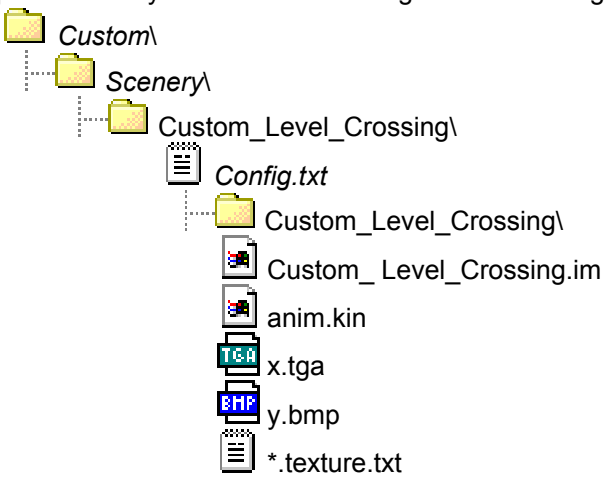
```
kind turntable
light 1
mode0 Custom_Turntable_Pit
mode1 Custom_Turntable_Spinner
type Trackside

angle 0,165,180,345
track 100966

snapmode 2
dighole 3,3
```

- kuid of track for bridge, or name of custom track (ref.p23 for alternative kuid numbers)

- size of hole in ground: 3,3 or 4,4

Special Scenery Objects – Level Crossings

Typical directory structure for creating a level crossing object should be:



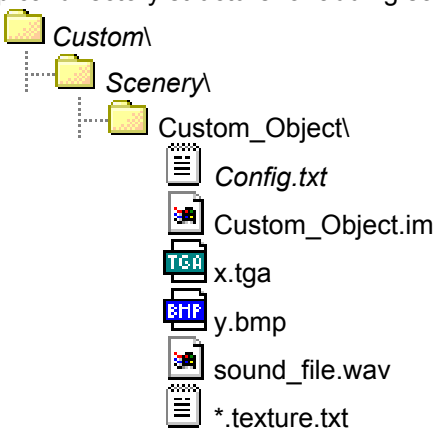
The following lines must be added to the object's config.txt file:

```
kind mocrossing
track 100396
road 100409
mode0 Custom_Level_Crossing
type Trackside
```

- kuid of track for bridge, or name of custom track
- kuid of road, or name of custom road
  (ref.p23 for alternative kuid numbers)

Scenery Objects with sounds

Typical directory structure for adding sounds to custom scenery should be:



Config.txt:

```
kind scenery

soundscript {
 dayloop {
  value-range 1, 0.1

  repeat-delay 10,15
  distance 10, 60
  sound {
   sound_file.wav
  }
 }
}
```

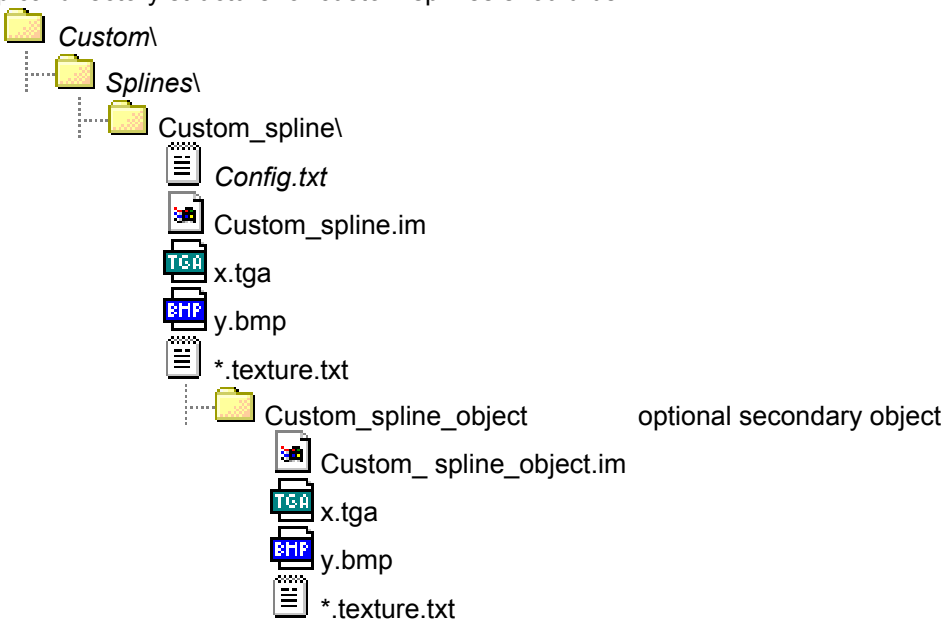
- time of day for full volume, cut off time
  where 0 is midnight, and 1 is midday.(optional)
- minimum and maximum time interval in seconds
- near distance (at which volume is 100%), cut off distance
- list of .wav files, one per line selected randomly

Splines

Refer to: …\Sample_content\Source_files\Splines & …\Sample_content\Trainz_custom_files\Splines

Splines are a useful way of making things like fences and roads in Trainz™.

Typical directory structure for custom splines should be:



Config.txt:

```
KIND track
ISTRACK 0
LENGTH 250
                                  ----------optional parameters
INITIATOR Custom_spline_object        optional secondary object
DIVIDER Custom_spline_object          optional secondary object
TERMINATOR Custom_spline_object       optional secondary object
UPRIGHT 10
WIDTH 1

REPEATS 1

BENDY 1                               0 = off, 1 = on. Disables secondary
GROUNDED 0.4                          height in meters above ground
ISROAD 1                              0 = off, 1 = on.
CARRATE 25                            Amount of cars on road

UNCACHED_ALPHAS 1                     Used for road and track splines.
```

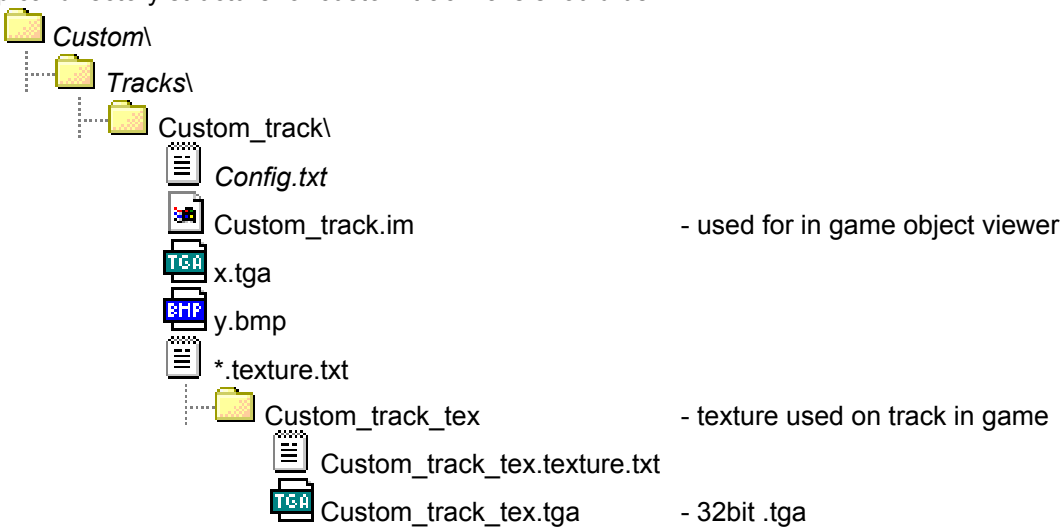
Track

Refer to: …\Sample_content\Source_files\Track & …\Sample_content\Trainz_custom_files\Track

Track folder is used for rails, bridges and tunnels.

Rails

Typical directory structure for custom track rails should be:

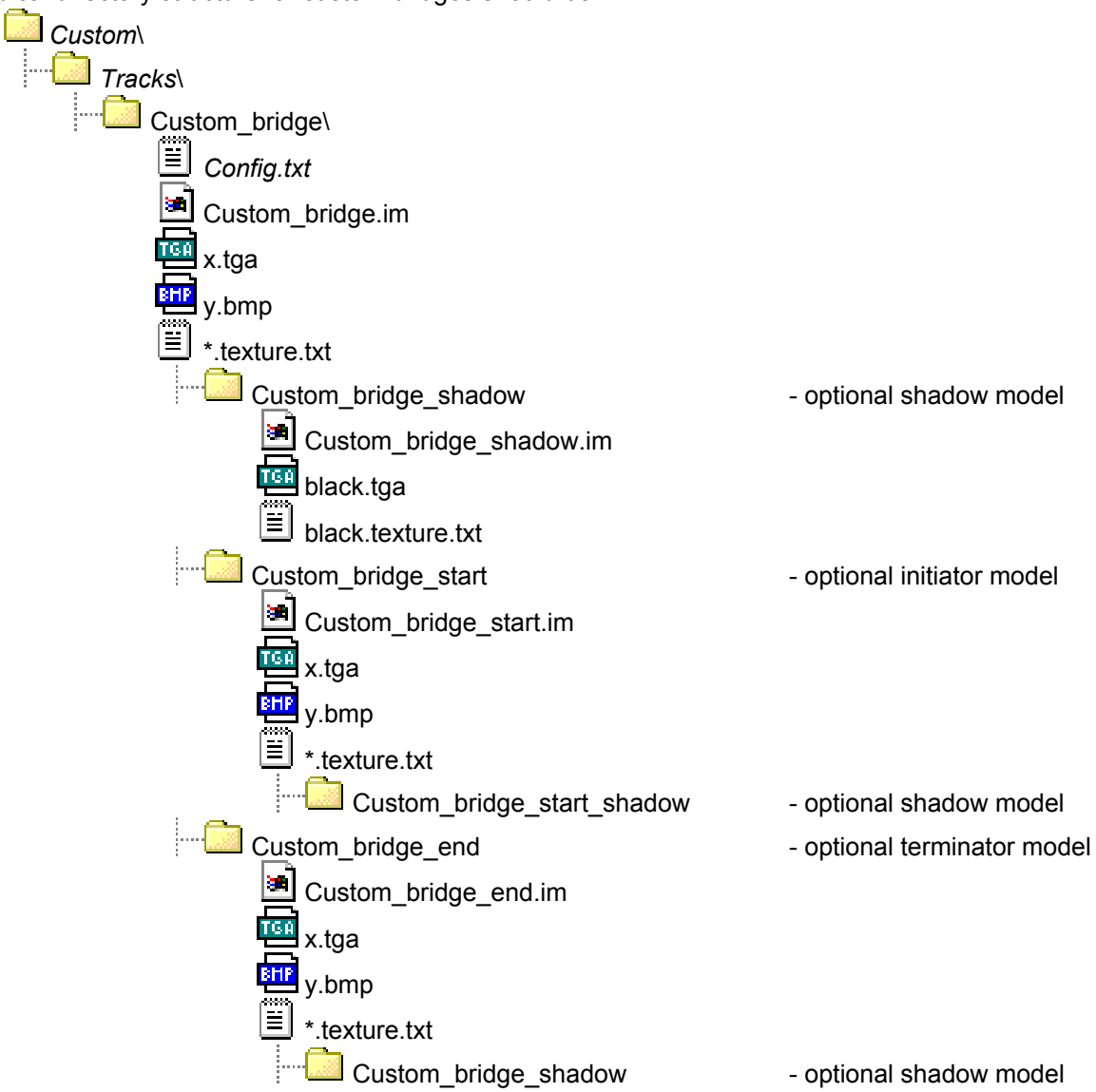


Config.txt for rails:

```
MINI 43
RGB 255,200,0

LENGTH 4
ISTRACK 1
WIDTH 4

CHUNKY_MESH Custom_track_tex
CHUNKY_INFO 0, 2, 1.2, 0.2, 0.85, 0.3, 0.7

KIND track

TYPE Rails
```

Bridges

Typical directory structure for custom bridges should be:

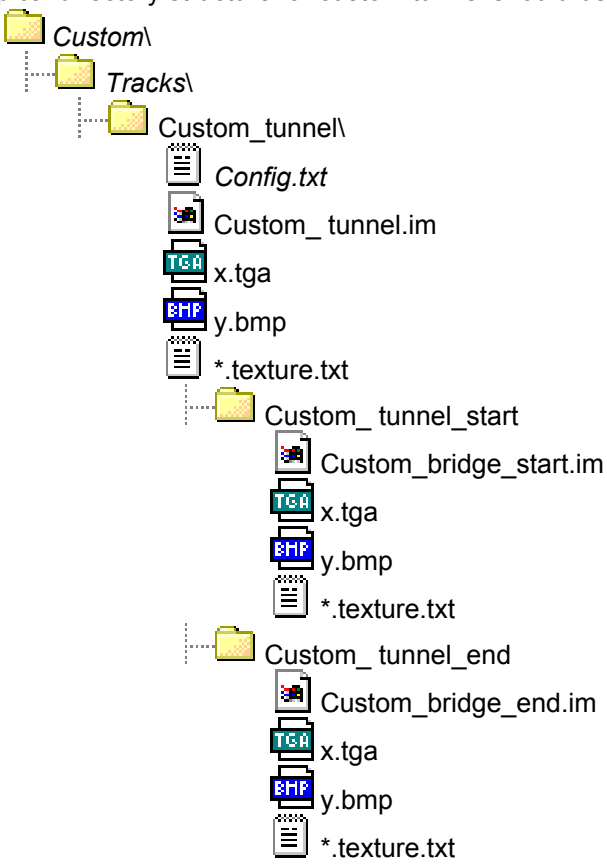


Config.txt for rails:

```
kind bridge
height –15                          - distance from rail to ground, note this is negative
length 20
istrack 1
trackoffsets -2.5, 2.5
bridgetrack 100395                  - kuid of track for bridge, or name of custom track
rgb 200, 100, 0                       (ref.p23 for alternative kuid numbers)
casts_shadows 1                     - using 0 will disable shadow, ie. Shadow model not required.
type Bridges
initiator Custom_bridge_start       - optional starting model
terminator  Custom_bridge_end       - optional ending model
```

Tunnels

Typical directory structure for custom tunnel should be:



Config.txt for rails:

```
kind bridge
height 8
trackoffsets -4.5, 4.5
bridgetrack 100395
rgb 180, 180, 180
length 20
istrack 1
initiator custom_tunnel_start
terminator custom_tunnel_end
endlength 20
```

- distance from rail to ground, note this is positive

- kuid of track for bridge, or name of custom track (ref.p23 for alternative kuid numbers)

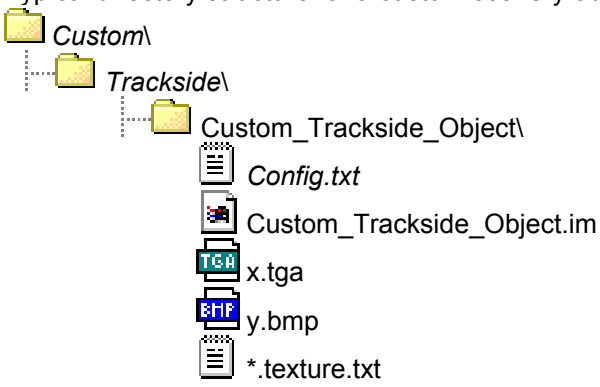
Trackside

Refer to: …\Sample_content\Source_files\Trackside & …\Sample_content\Trainz_custom_files\Trackside

Trackside is used for special scenery objects that can be placed on or near the track, such as signals and speed limit signs.

The config.txt file includes a *trackside* parameter that defines the object position with relation to the center of the track. Negative values appear on the left of the track, positive values on the right, and using 0 (zero) will cause the object to be placed in the center of the rails.

Typical directory structure for a custom scenery object should be:



Config.txt for signals:

```
kind mosignal
trackside –2.5
function TrackSignal
```

Config.txt for speed limit signs:

```
kind mospeedboard
trackside -2.5
speedlimit 5.56
```

- speed limit in meters/second

Engine Config Files

Default location: C:\Program Files\Auran\Trainz\Engines

The \TRAIN_NAME\config.txt tells the game which enginespec to refer to by KUID number.

The engine config files are subdivided into relevant categories as follows:

flowsize	Affects the speed of flow through the pipes, bigger is faster. **Generally leave these settings**
volume	Affects the size of the pipes and cylinders, bigger, is bigger. **Generally leave these settings**
pressure	These figures are in **grams per m cubed**. *****See Note below
brakepipe	The fully charged brake pipe pressure
brakeinitial	The *brakepipe* pressure after smallest reduction on self lapping systems.
brakefull	The *brakepipe* pressure after maximum reduction on self lapping systems, in keeping with equalisation of pressures.
	Then there are some variables for telling the game where you want the brakes to start when you enter the cab, at the moment, they are set up for maximum braking.
	They too are expressed in **grams per m cubed**.
mass	Generally leave these settings
motor	
resistance	The power setting, smaller resistance=more power, a fractional change makes a big difference
adhesion	higher = less wheelslip
maxvoltage	**Generally leave these settings**
maxspeed	In metres per second (divide km/h by 3.6) higher = faster
brakeratio	Brake force per brake pipe pressure reduction. higher = better breaking

*Note:
Converting PSI to Grams /m cubed…

e.g. 90psi...
(90+14.7).0000703
104.7 x .0000703=.00736041

brakepipe is the fully charged brake pipe pressure. brakeinitial is the brakepipe pressure after smallest reduction on self lapping systems. brakefull is the brakepipe pressure after maximum reduction on self lapping systems, it is set to 64 on the 90psi pipes, in keeping with equalisation of pressures.

Equalisation of Pressures

There is a point at which no further brake pipe pressure reduction will result in increased braking effort, this is known as full application or equalisation of pressures.

Imagine you made a 26 psi reduction when operating a loco with a 90psi brake pipe. 90psi in the train pipe minus 26psi reduction equals 64 psi in the pipe. Due to the 2.5 : 1 ratio of auxiliary reservoir volume to brake cylinder volume, the 26 psi reduction puts 64 psi into the brake cylinder. As the pressure in the reservoir and the pressure in the cylinder is now equal, no more air will flow into the brake cylinder; and making a further reduction in brake pipe pressure will have no effect on braking.

Equalisation occurs at different pressures, depending on the train pipe feed pressure.

100 psi pipe (e.g. the UK locos - 7 bar) equalisation at 71 psi.
90 psi pipe (e.g. the US locos) equalisation at 64 psi.
80 psi pipe (e.g. the Australian ALCos) equalisation at 57 psi.
72 psi pipe (e.g. French & Queensland locos) equalisation at 49 psi.

The easiest way to set your custom content to the desired brake pipe feed pressure is to copy the whole pressure section from the config of a loco that uses the pressure you desire.

WARNING: ALTERING THESE FIGURES MAY RESULT IN UNDESIRED EFFECTS IN PERFORMNACE AND BEHAVIOR OF YOUR TRAINS. (MAKE BACK-UP COPIES OF YOUR ENGINE CONFIG FILES!!)

Multiple Instances in the Custom Folder

If you want to create multiple instances of the same thing in the custom folder...

Imagine I've already got a ZRV brew, and I want to re-skin it.

Copy the folder and paste it back in with a new name, i.e. *zrv_brew_2*

Open the folder and rename the three subdirectories.... zrv_brew_2_art
zrv_brew_2_body
zrv_brew_2_shadow

Open the folder **zrv_brew_2_art** and change the name of the two texture.txt files to...

zrv_brew_2_art_icon & *zrv_brew_2_art_512* (don't change the name of the .tga files)

Open the folder **zrv_brew_2_body** and change the name of the .pm file to *zrv_brew_2_body.pm*

Open the folder **zrv_brew_2_shadow** and change the name of the .pm file to *zrv_brew_2_shadow.pm*

Repainting Existing Models

From the Trainz CD, open the Trainz_Dev\Trains folder and copy the folder for the train you want to repaint. E.g. Class43_original

Paste into your …/Custom/Trains folder

Now go back to the CD, open the Trainz_Dev\Bogeys folder and copy the folder for the bogie. E.g. Class43_bogey

Paste into your …/Custom/Bogeys folder

Start something like Photoshop and browse to …/Custom/Trains/Class43_original/Class43_original_body

You'll find the associated .tga files in there, they're read only so turn that off in their properties before you try to save them.

It's a good idea to open up the config.txt file in the …/Custom/Trains/Class43_original/ folder

….and change the name of it so you can tell it apart form the original item in your vehicle menu. Of course you can repaint the icon and even the big render for the Driver consist if you have the energy. Those images are found in the Class43_original_art folder as described before.

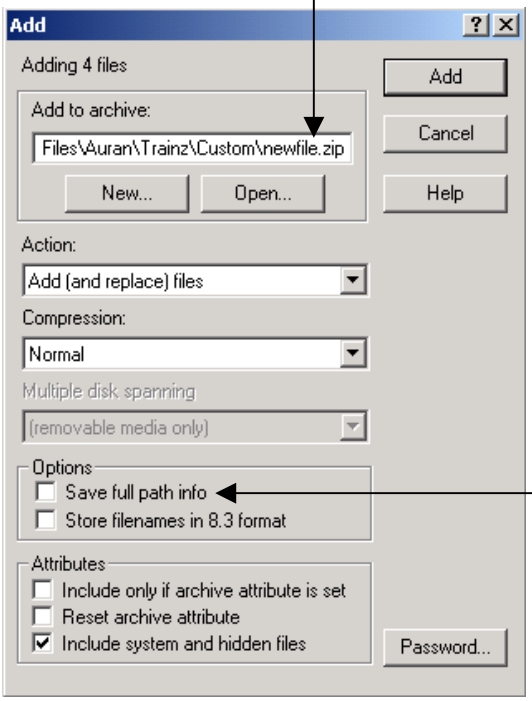
Preparing content for distribution

It is important that all custom content be easily installed to the correct locations with in Trainz™. We have compiled some basic recommendations to help with this process.

Archiving content

Use a Zip program such as WinZip to compress your files. Make sure you **do not** include full path information, as some users may not have installed Trainz™ to the default C:\ location.

You can zip one directory or several directories at once. If you wish to add several directories you will need to type in the name of the new file you wish to create at the *Add to archive:* dialogue box (as shown below).



Tip:
Remember to archive the ***directories*** within the Custom folder along with a read_me.txt for ease of installation. See Read_me.txt information next page

With this option un-selected, only the local file path will be saved, e.g. Maps\NewLayout\

Read_me Template.txt

We have provided a Read_me Template.txt file within this package. It should be included when the Custom directories are zipped. The format is as follows:

READ ME

TRAINZ ® Model railroading on your PC
www.virtualtrainz.com

CONTENT NAME, v 1.0

By:
Email:
Website:

Install these files by unzipping into your ..\Trainz\Custom\ folder. (Default location is: C:\Program Files\Auran\Trainz\Custom)

Please preserve directory structure for correct installment.

Information:

Changes:

v 1.0 - Initial public release

Reference KUID numbers

Engine sounds:

Alco	kuid <KUID:-1:42003001>
Electric	kuid <KUID:-1:42003002>
EMD	kuid <KUID:-1:42003000>

Horn sounds:

Alco	kuid <KUID:-1:42003103>
BR	kuid <KUID:-1:42003102>
Default	kuid <KUID:-1:42003101>

Interiors:

BB15000	kuid 100554
Class37	kuid 100382
Class43	kuid 101168
F7 kph right-hand-drive	kuid 100186
F7 kph left-hand-drive	kuid 101211
F7 mph left-hand-drive	kuid 101212
F7 mph right-hand-drive	kuid 101202

Pantographs:

1044 pantograph	kuid 101255
6E1SAR pantograph	kuid 110002
NS1600 pantograph	kuid 100860
Rc4 pantograph	kuid 101177

Engines:

1044	kuid <KUID:-1:42004207>
1600	kuid <KUID:-1:42004208>
2100	kuid <KUID:-1:42004209>
340	kuid <KUID:-1:42004210>
44	kuid <KUID:-1:42004211>
6E1	kuid <KUID:-1:42004212>
930	kuid <KUID:-1:42004213>
BB15000	kuid <KUID:-1:42004205>
Class37	kuid <KUID:-1:42004204>
Class43	kuid <KUID:-1:42004206>
Default wagon	kuid <KUID:-1:42004201>
Deltic	kuid <KUID:-1:42004215>
DL531	kuid <KUID:-1:42004224>
F7	kuid <KUID:-1:42004202>
FA1	kuid <KUID:-1:42004216>
FA2	kuid <KUID:-1:42004217>
FPA4	kuid <KUID:-1:42004218>
GM	kuid <KUID:-1:42004219>
Rc4	kuid <KUID:-1:42004220>
SD40 2	kuid <KUID:-1:42004221>
V200	kuid <KUID:-1:42004222>

Tracks:

Australia	
4m standard	kuid 100396
4m standard no gravel	kuid 100395
4m standard double	kuid 100736
4m standard solid	kuid 100966

Britain	
4m standard	kuid 15
4m standard double	kuid 13228
USA	
4m standard	kuid 100608
4m standard concrete	kuid 101038
4m standard no gravel	kuid 100609
4m standard double	kuid 110017
4m standard solid	kuid 110016

Roads:

Australia	
Road	kuid 100409
Road one lane	kuid 100410
Road dirt	kuid 100456
Road for bridges	kuid 100413
Britain	
Road	kuid 19
Road narrow	kuid 1017
Road for bridges	kuid 20
USA	
Road	kuid 100598
Road narrow	kuid 100588
Road for bridges	kuid 100614

These KUID’s may be copied and pasted into the relevant config.txt files.