

# CHAPTER 4 - MODELING GUIDELINES AND DIRECTORY STRUCTURE

The purpose of this chapter 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™ or Gmax™ and therefore only give references to model requirements, rather than a modeling tutorial

## DIRECTORY STRUCTURE OVERVIEW

The 'custom content' directory structures in SP3 and UTC are different from those in previous versions of Trainz. This is in order to facilitate a 'clean' changeover from pre SP3 content to post SP3 content. By keeping the directory structures isolated we intend to be able to maintain the content in their distinct areas and allow a future version of Trainz to remove the pre SP3 directory structures, thereby completing the upgrade process.

**In order for this process to work correctly it is therefore very important that no pre-SP3 content is manually moved into the new SP3/UTC directory structures.**

We do however realise that content creators need to test their own masterpieces in SP3 before packaging with Content Dispatcher takes place. Only in this instance should there be any manual transfer.

The default location for all SP3/UTC custom content is:

C:\Program Files\Auran\Trainz\World\custom\



## TRAINS

### 3D Studio MAX™ and Gmax™ Modeling Guidelines:

*Download Source files from the Trainz Website*  
**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.

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 & GMAX: 'Create' tab, 'Helpers', 'Point')

To maintain correct alignment, attachment points should be created in the TOP viewport.

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)

#### a.driver\* (0, 1, etc..)

- attachment point for driver mesh (for future versions of Trainz)

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

### New Attachment points in UTC:

#### a.cabback

- attachment point for the rear cabin of a loco. Use this for dual cab locomotives.
- located at the centre of cabin
- front/back cab toggled using the 'c' key when using the internal camera mode.

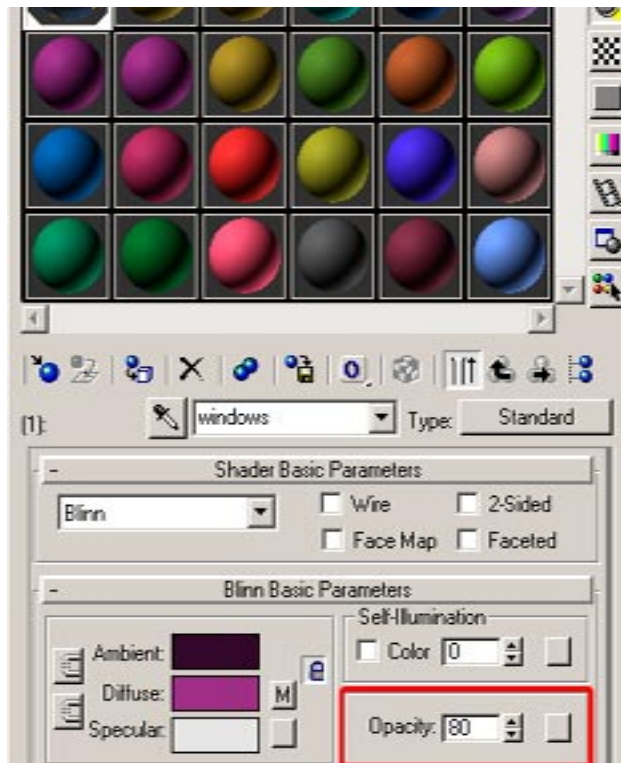
#### a.outsideview\* (0, 1, etc..)

- these are located external of the loco body mesh.
- the camera is positioned to face the *negative Y* direction of the attachment.
- toggled using [ and ] using the internal camera mode after default interior camera view(s).

### 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 of following pixel dimensions: 8, 16, 32, 64, 128, 256, and 512 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.

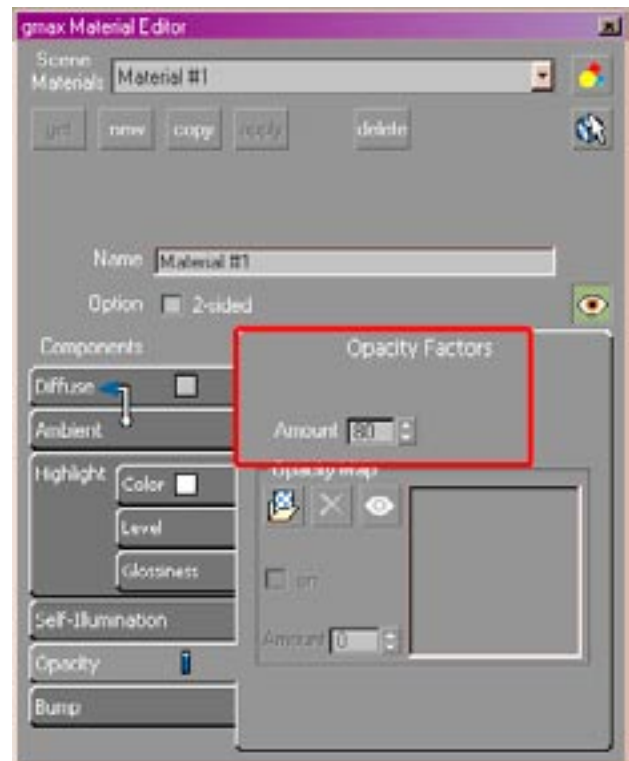


3D Studio MAX

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 pixel dimensions as the diffuse map.

*Reflection and Opacity maps must not be used together with-in the same texture. Reflection and Opacity maps must not be used on digits.*

Window opacity is derived from the material opacity setting. See figures below.



G MAX

### Locomotive numbering:

Dynamic locomotive numbering for custom content (using alpha-numbers) are now supported in SP3.

Digits are modeled as 6 individual rectangular polygons offset from the face of the Loco body (about 5mm). Digit polygons must be texture mapped using the correct texture naming and alpha-number naming conventions as follows:

If *one* font type used:

Digit textures (*digit\_1.tga to digit\_6.tga*) are replaced automatically with alphanumeric textures (*alphanumeric\_0 to alphanumeric\_9*) as numbers are changed in 'My Collection'.

If *two or more* font type used:

Digit textures (*digit\_1a.tga to digit\_6a.tga* and *digit\_1b.tga to digit\_6b.tga* etc) are replaced automatically with alphanumeric textures (*alphanumeric\_0a to alphanumeric\_9a* and *alphanumeric\_0b to alphanumeric\_9b*) as numbers are changed in 'My Collection'.

*Refer to [Source files](#) for configuration of Loco numbering digit's*

**Train Directory Structure & Naming Conventions:**

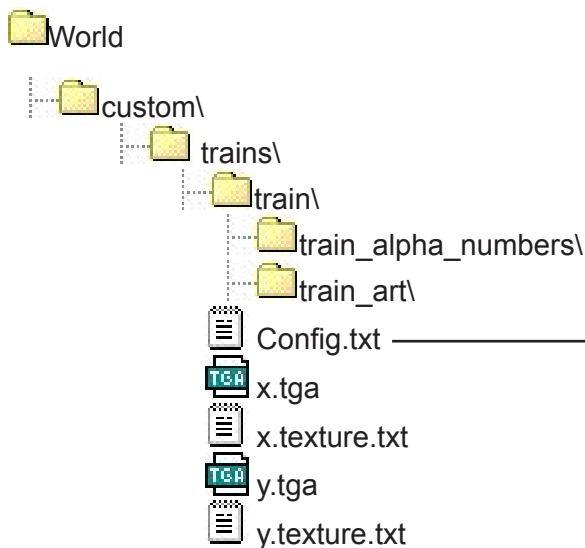
Typical directory structure for custom Trains should be:



## ALIASING TRAINS

Trainz SP3 and UTC locomotives can reference archived locomotive mesh assets for use with custom textures. This process is done by aliasing the KUID of the archived Locomotive.

A typical structure of an aliased loco could be as follows:



The textures must have exactly the same names and have exactly the same quantity and pixel dimensions that the aliased mesh uses. Trainz Paintshed (Version 1.3+) uses mesh aliasing.

The shadow file of the aliased loco will also be read (if present).

## TYPICAL LOCO CONFIG.TXT:

```

kuid <KUID:###:#####>
alias <KUID:-10:183> (the aliased mesh it
uses)
name train
company Auran
origin AU
bogey <KUID:###:#####>
engine 1
interior <KUID:###:#####>
fonts 1
mass 97600
kind traincar
running-numbers {
  rn-0 #0003
  rn-1 #0004
  rn-2 #0005
  rn-3 #0006
}
smoke_shade 0.18
smoke_random 2.5
smoke_slowlife 6
smoke_fastlife 0.8
smoke_height 1.7
smoke_fastspeed 3.2
enginespec <KUID:-1:42004209>
enginesound <KUID:-12:2100>
hornsound <KUID:-1:42003103>
description ""
kuid-table {
  0 <KUID:###:#####>
  1 <KUID:###:#####>
  2 <KUID:###:#####>
}
obsolete-table {
}
username My locomotive
trainz-build 1.5
category-class AC
category-region-0 AT
category-era-0 1980s
  
```



## BOGEYS

### 3D Studio MAX™ and Gmax™ Modeling Guidelines:

*Download Source files from the Trainz Website*

#### 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.

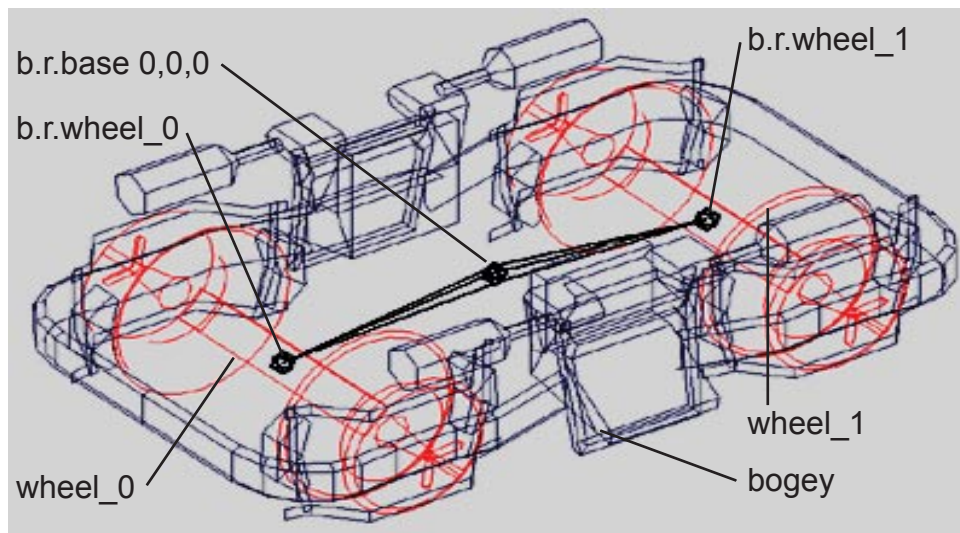
The absolute centre of bogeys should be located at World origin point (0,0,0)

#### Attachment points:

**a.ground\*** (0, 1, etc..)

- Slightly offset at the base of each wheel
- Determines the wheel spark position

#### Animated Bogey Example 1



#### Hierarchal Sub-tree:

```

b.r.base
  b.r.wheel0
    wheel_0
  b.r.wheel1
    wheel_1
  bogey
  
```

*In this example, the bogey will be inserted into the Train model attachment point (e.g. a.bog0) at b.r.base (or 0,0,0). b.r.wheel0, and b.r.wheel1 (bones) were animated to turn 360° over 32 frames.*

*Bones must have the b.r.\* naming convention for Trainz to recognise them.*

#### 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.

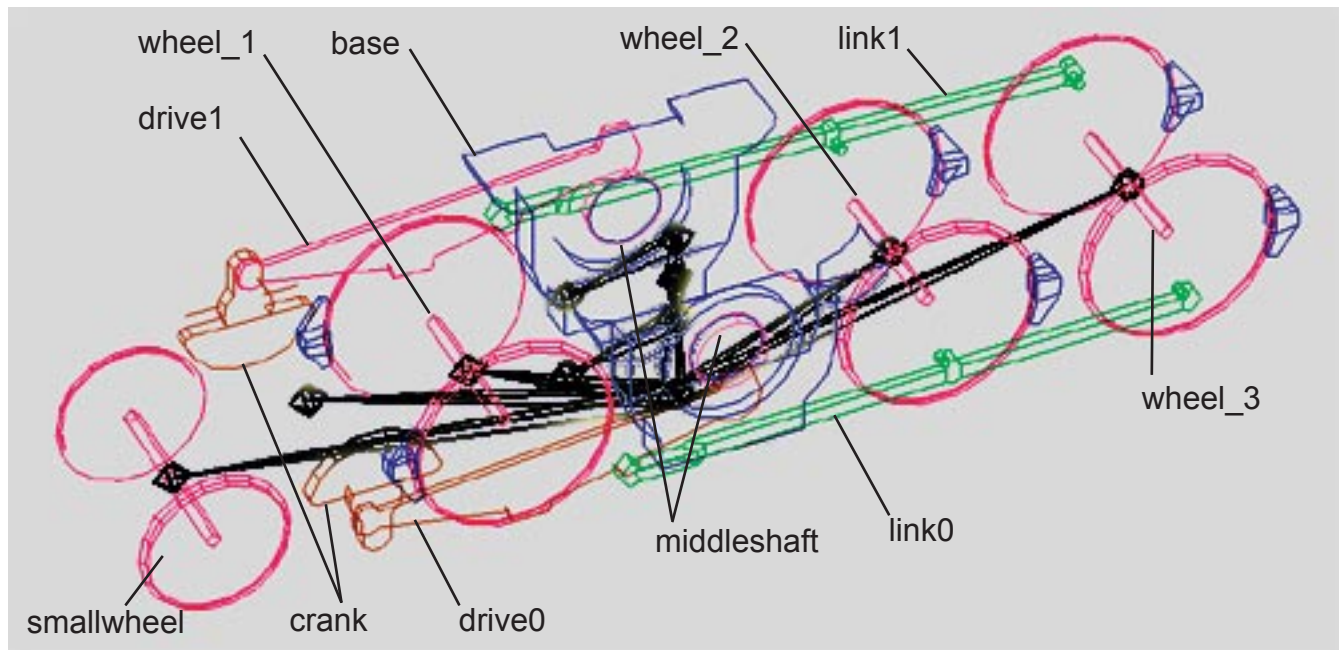
Diffuse Maps: Generally a single 128x128 16-bit .TGA file is sufficient to texture a bogey. Additional maps (e.g. for springs) are also used.

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

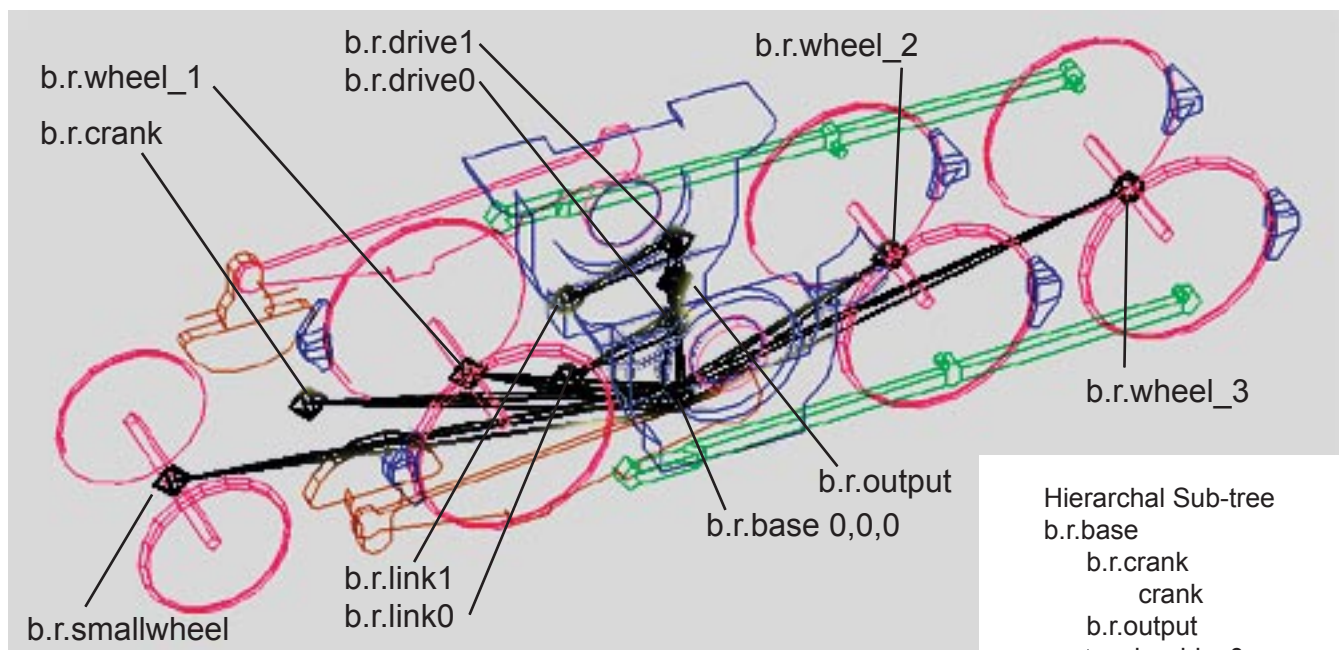
#### Exporting Models:

As per 'Modeling Trains' section. Remember naming conventions and to type in the file extension under *file name* (e.g. TRAIN\_NAME\_bogey.pm)

## Animated Bogey Example 2a - Objects



## Animated Bogey Example 2b - Bones



## Hierarchal Sub-tree

```

b.r.base
  b.r.crank
    crank
  b.r.output
    b.r.drive0
      b.r.link0
        link0
      drive0
    b.r.drive1
      b.r.link1
        link1
      drive1
    middleshaft
  b.r.smallwheel
    smallwheel
  b.r.wheel_1
    wheel_1
  b.r.wheel_2
    wheel_2
  b.r.wheel_3
    wheel_3
  base
  
```

## Animated Bogey Example 2c - Side view after a few frames



This example is much more complex than the previous example. Animation frames = 120

**Wheel\_1, 2 & 3** circumference = 4.2m

- Animated to turn  $720^\circ$  over 120 frames:  
Distance traveled: 4.2m x 2 revs = 8.4m

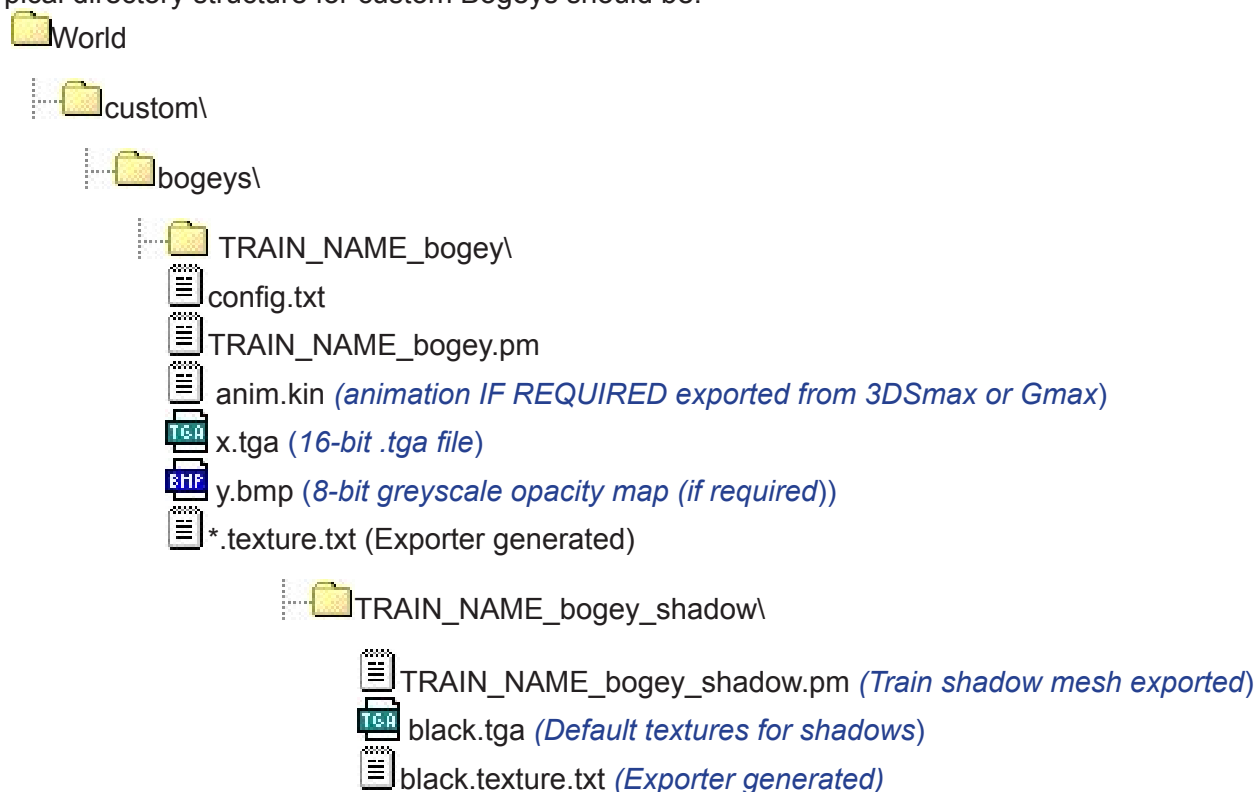
**smallwheel** circumference = 2.8m

- Animated to turn  $1080^\circ$  over 120 frames:  
Distance traveled: 2.8m x 3 revs = 8.4m

- Animdist: (worked out from distance travelled in 30 frames:  $8.4 / 4 = 2.1$ ) therefor **animdist 2.1**

**Bogey Directory Structure & Naming Conventions:**

Typical directory structure for custom Bogeys should be:





## PANTOGRAPHS

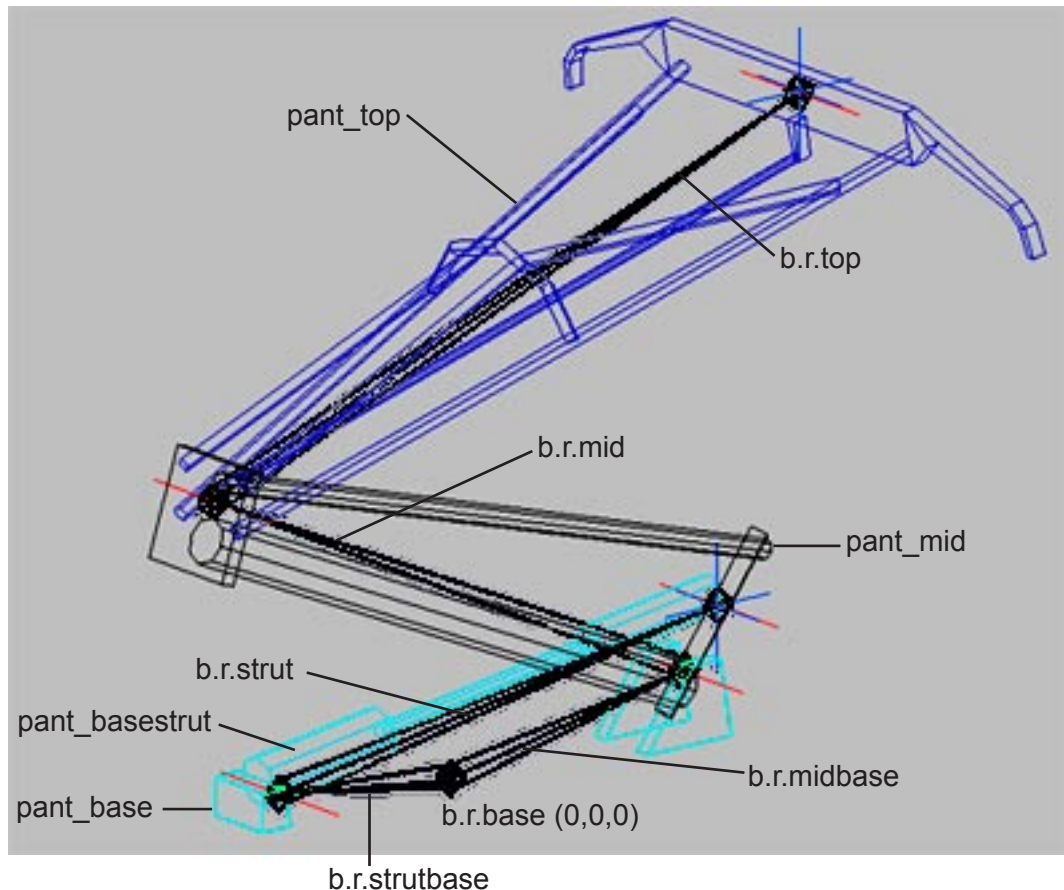
(Pants)

*Download Source files from the Trainz Website*

Pantographs are the animated mechanisms on the roof of electric locomotives that conduct to an electric catenary (wires) above.

### Model configuration:

Typical model configuration: (based on the bb15000 pantograph)



*In this example, the Pantograph will be inserted into the Train model attachment point (a.pant0) at b.r.base (or 0,0,0).*

### Typical Hierarchal Sub-tree

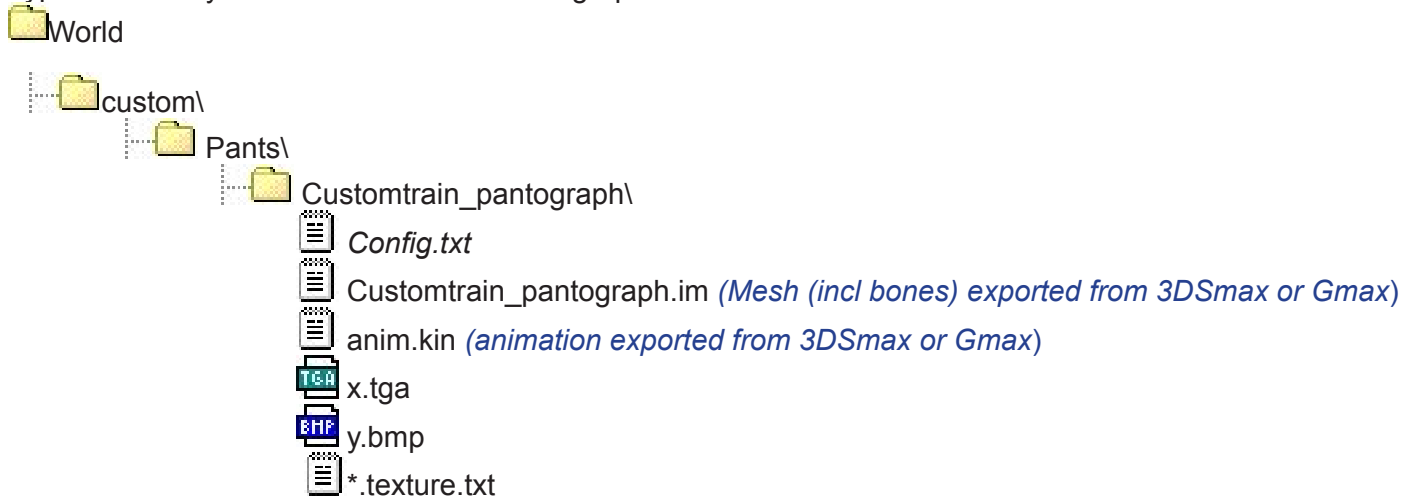
```

b.r.base
  b.r.midbase
    b.r.mid
      b.r.top
        pant_top
      pant_mid
    b.r.strutbase
      b.r.strut
        pant_basestrut
      pant_base
  
```

Generally Pantograph animations should take place over 16 frames only. Bones must have the b.r.\* naming convention for Trainz to recognise them. Refer to the [Source files](#) from the Trainz Website for a working example.

## Pantograph Directory Structure & Naming Conventions:

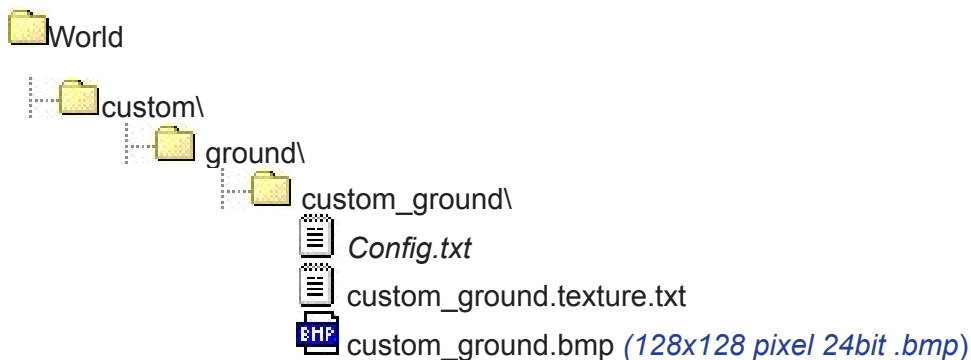
Typical directory structure for custom Pantographs should be:



## GROUND

*Download Source files from the Trainz Website*

Typical directory structure for a custom ground texture should be:

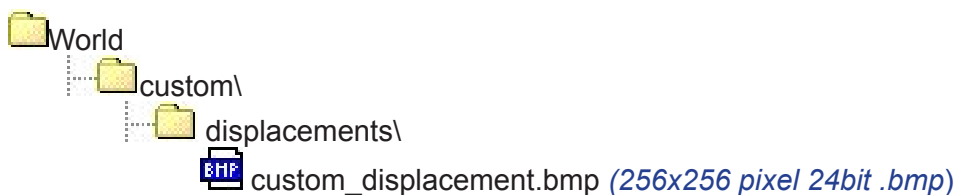


## DISPLACEMENTS

*Download Source files from the Trainz Website*

Displacement maps are used to adjust the height and shape of an area of terrain.

Typical directory structure for a custom ground texture should be:



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

## ENVIRONMENT

*Download Source files from the Trainz Website*

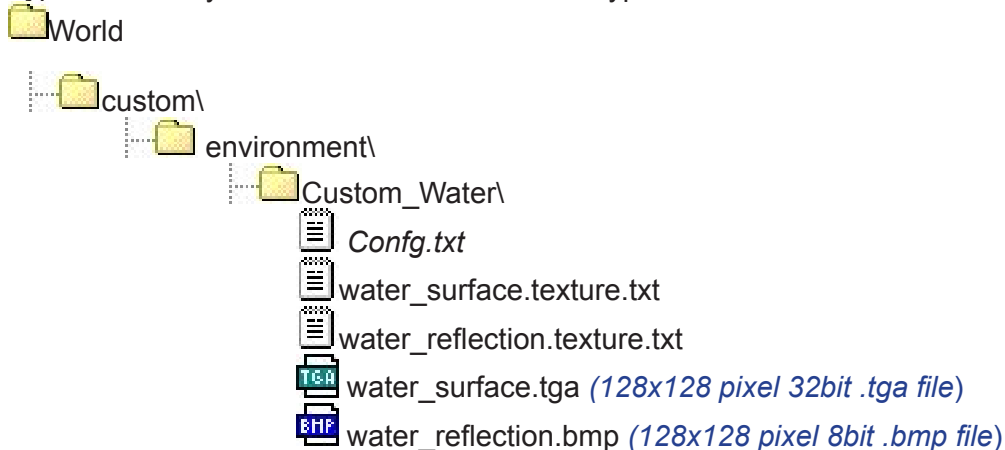
The Environment folder accommodates the use of different types of sky and water in Trainz™.

## WATER

*Download Source files from the Trainz Website*

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:

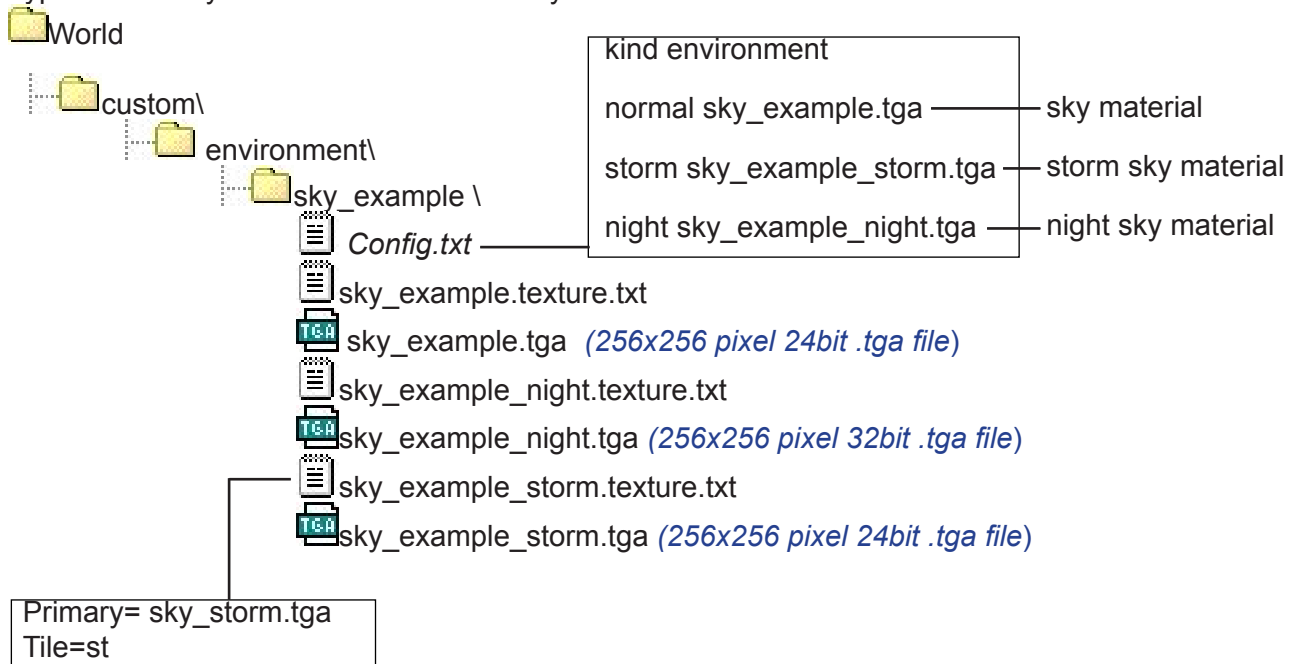


## SKY

*Download Source files from the Trainz Website*

Sky is generated from three source images.

Typical directory structure for a custom sky should be:

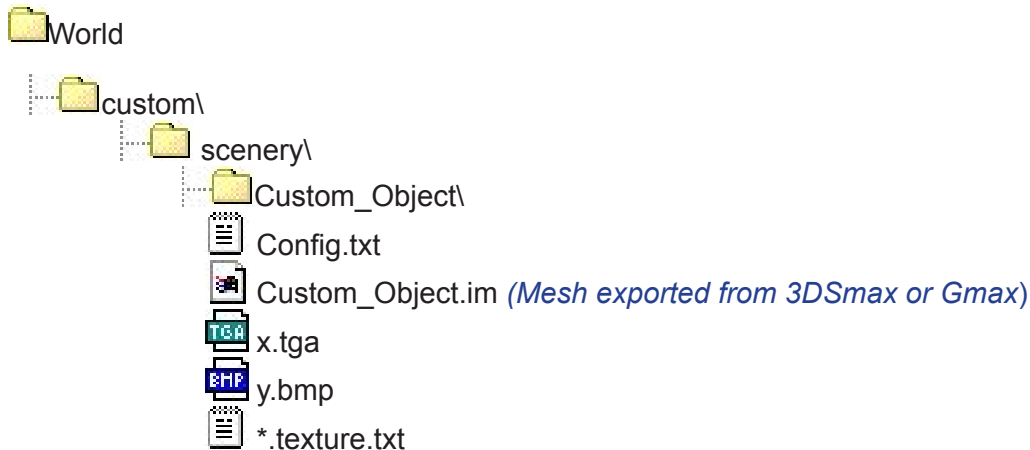


## SCENERY

### Download Source files from the Trainz Website

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.

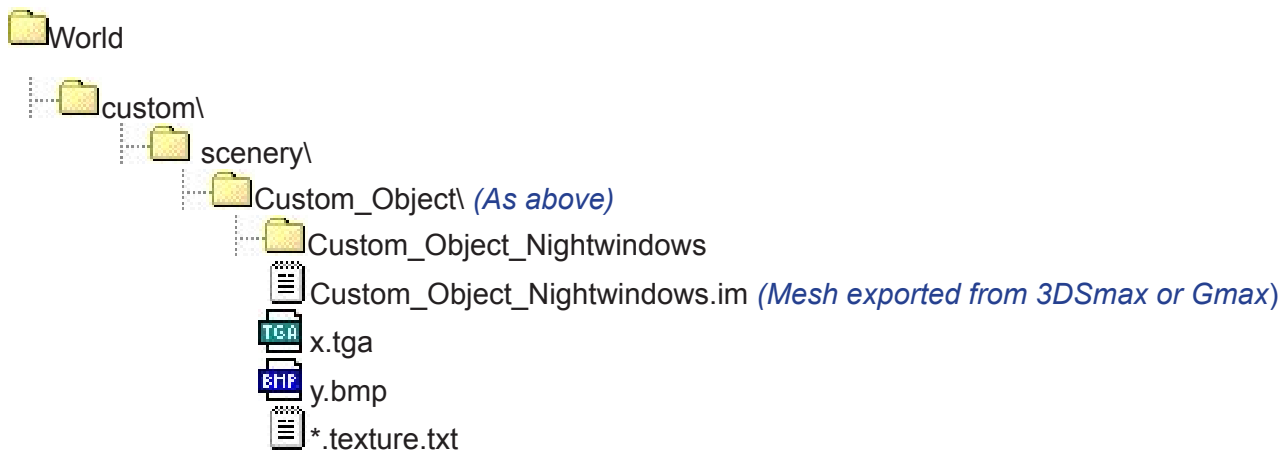
Typical directory structure for a custom scenery object should be:



## SCENERY OBJECTS WITH LIGHTS AT NIGHT

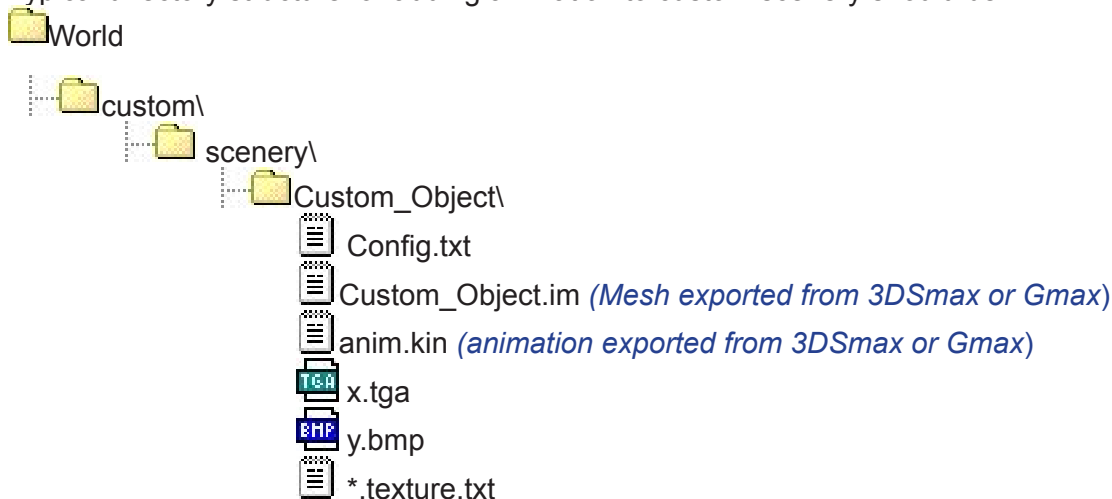
Objects just as buildings or signs can be made to appear to be have lights on at night. A model that contains only the lit areas of the object can be exported into a subdirectory.

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



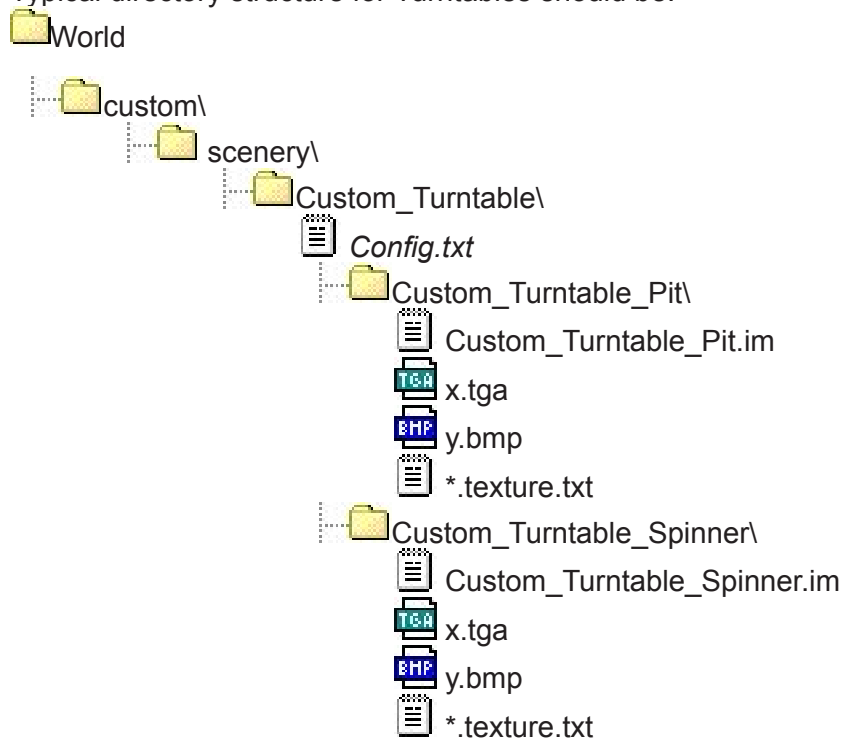
## SCENERY OBJECTS WITH ANIMATION

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

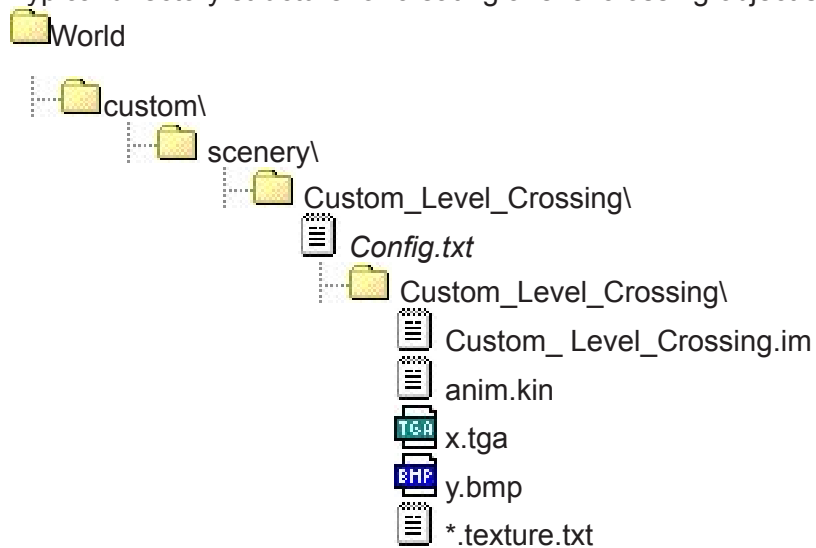


**SPECIAL SCENERY OBJECTS - TURNTABLES**

Typical directory structure for Turntables should be:

**SPECIAL SCENERY OBJECTS – LEVEL CROSSINGS**

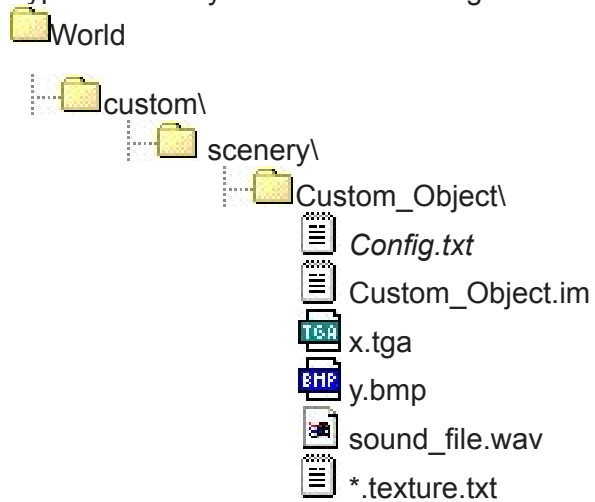
Typical directory structure for creating a level crossing object should be:





## SCENERY OBJECTS WITH SOUNDS

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

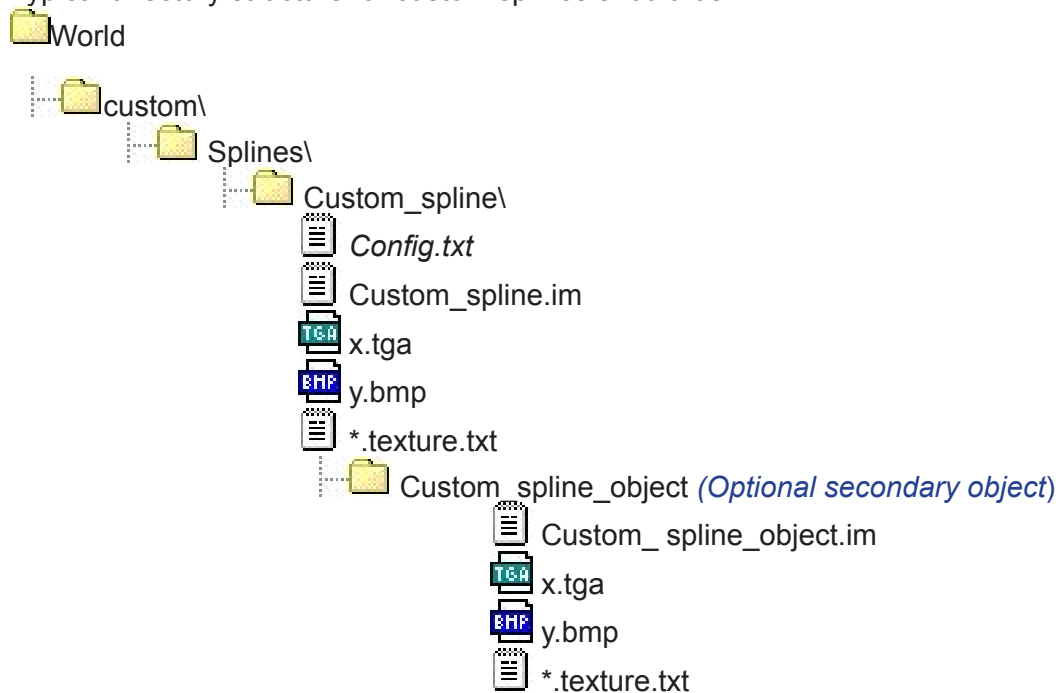


## SPLINES

### *Download Source files from the Trainz Website*

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

Typical directory structure for custom splines should be:



## TRACK

### *Download Source files from the Trainz Website*

Track folder is used for rails, bridges and tunnels.

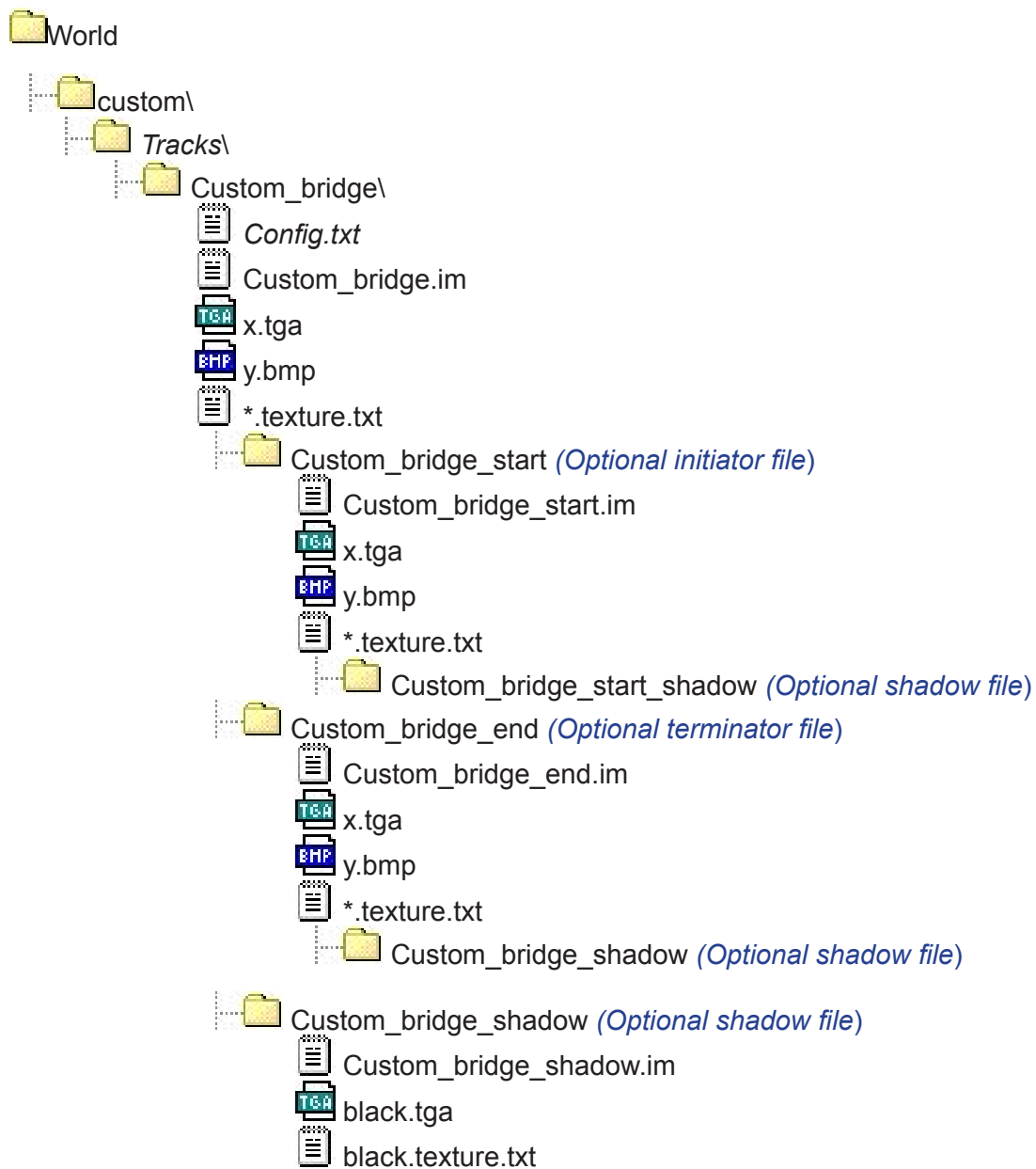
## Rails

Typical directory structure for custom track rails should be:



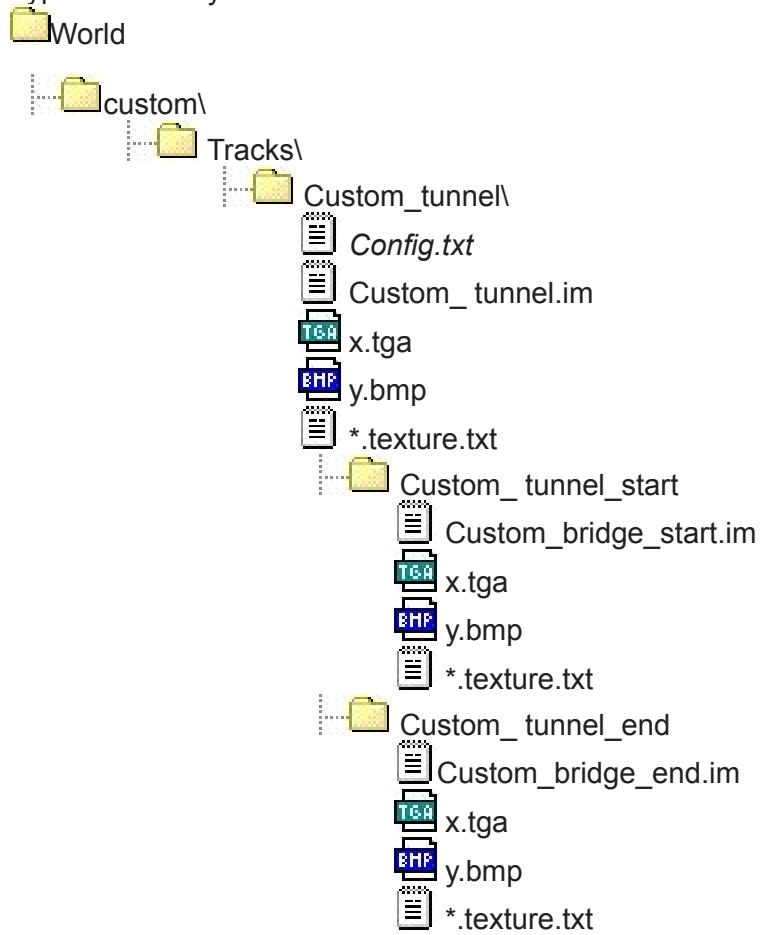
## BRIDGES

Typical directory structure for custom bridges should be:



## TUNNELS

Typical directory structure for custom tunnel should be:





## TRACKSIDE

Download [Source files](#) from the Trainz Website

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

Typical directory structure for a custom scenery object should be:

