



Ultimate Trainz Collection

Content Creation Procedures

© Auran Games Pty. Ltd.
January 2003

ULTIMATE TRAINZ COLLECTION

CONTENT CREATION PROCEDURES

CONTENTS

CHAPTER 1 - THE BASICS 1

INTRODUCTION	1
OVERVIEW	2
WHAT IS A KUID?	4
CONTENT ID	4
THE DO'S AND DON'TS OF CONTENT CREATION	5

CHAPTER 2 - CLASSES AND CODES 6

CONTENT ID RANGES	6
CATEGORY CLASS	7
A MOTIVE POWER	7
B BUILDINGS & STRUCTURES	7
C CABEES	7
D DEFENCE	7
E ENVIRONMENT	7
F FOLIAGE	7
G GROUND	7
L LIGHT RAIL & MONORAIL	7
M MAINTENANCE OF WAY	8
O ORGANISM	8
P PASSENGER & MAIL CARS	8
R RAILCARS & MULTIPLE UNIT SETS	8
S SPLINES	8
T TRACK	8
V VEHICLES	8
W WAYSIDE	8
X FREIGHT CARS	8
Y MAPS & SCENARIOS	9
Z TRAIN PARTS	9
REGION CODES	10
ERA CODES	12

CHAPTER 3 - CONFIG AND KIND 13

CONFIG.TXT	13
KIND: ENGINE	14
KIND: BOGEY	19
KIND: TRAINCAR	20
KIND: ENGINESOUND	22
KIND: HORNSOUND	23
KIND: INTERIOR	24
KIND: PANTOGRAPH	26
KIND: WATER	26
KIND: ENVIRONMENT	26
KIND: MAP	27

KIND: GROUNDTEXTURE	27
KIND: SCENERY	28
KIND: TRACK – RAILS	29
KIND: TRACK – ROAD	30
KIND: BRIDGE – BRIDGE	31
KIND: BRIDGE – TUNNEL	32
KIND: BRIDGE – DOUBLE TRACK	33
KIND: MOSPEEDBOARD	34
KIND: MOSIGNAL	35
SIGNALS	35
LIGHTS	36
KIND: MOJUNCTION	37
KIND: MOTURNTABLE	38
KIND: MOCROSSING	39
KIND: ACTIVITY	40

CHAPTER 4 - MODELING GUIDELINES AND DIRECTORY STRUCTURE 41

DIRECTORY STRUCTURE OVERVIEW	41
TRAINS	42
ALIASING TRAINS	45
BOGEYS	46
PANTOGRAPHS	49
GROUND	50
DISPLACEMENTS	50
ENVIRONMENT	51
WATER	51
SKY	51
SCENERY	52
SPLINES	55
TRACK	56
BRIDGES	57
TUNNELS	58
TRACKSIDE	59

CHAPTER 5 - PARTICLE EFFECTS AND SCRIPTS 60

SMOKE EFFECTS	60
ADDING SMOKE TAGS	61
EXAMPLE 1	63
EXAMPLE 2	64
SOUND SCRIPTS	65
TRAINZSCRIPT TUTORIAL	67

CHAPTER 6 - UTILITIES AND KUIDS 70

TRAINZ DOWNLOAD STATION	70
TRAINZ CONTENT FOUNDRY	71
AURAN KUIDS	72
TRAINS	72
BOGEYS	74
PANTOGRAPHS	75
HORNSOUND	75
ENGINESOUND	75
INTERIORS	75
ENGINESPEC	76
SCENERY	77
TRACKS	86
SPLINES	88
TRACKSIDE	90
ENVIROMENT	95
PAINTSHED	96

CHAPTER 1 - *THE BASICS*

INTRODUCTION

Welcome to the Ultimate Trainz Collection Content Creation Procedures document.

This document is designed to be read by persons intending to create 3rd party content for Trainz or those persons who are interested in a 'behind the scenes' view of how content is created.

The purpose of this document is to detail the way in which 3rd party content should be designed and built to be compatible with UTC and future versions of Trainz. We ask all content creators to make a concerted effort to follow the procedures in this document as much as possible. Doing so will reduce the work needed in the future and will help us make sure that future versions of Trainz have a much greater chance of working with your content.

Finally, after much consideration, we have decided to make this document a formal procedures document as opposed to a guidelines document. Given the importance of creating content that is standardized we feel that the best way to accomplish this is for all content to be created to a common set of criteria. We ask that all content creators adopt the procedures outlined in this document without delay. For its part, Auran will make every effort to support content that is created to the procedures outlined in this document in future versions of Trainz.

You will find some sample config files attached within the zip file where this document was extracted. These are referred to in the 'interiors' and the 'mosignal' sections of this document

OVERVIEW

Creating new content for Trainz is a five-step process. Not all steps are required, but a minimum of 3 are usually required.

1: Research (Mandatory)

The research step involves finding out all the relevant information that you can about the item you wish to create. Research usually covers the accumulation of data about the content in question. It may be performance figures, taking photos or even a visit in person to see the object you wish to create.

2: Create a .PM or .IM file (Optional)

A .PM file is a Progressive Mesh file. PM is an acronym that stands for Progressive Mesh. An .IM file is an Indexed Mesh. These files are created by '3D Studio Max' or 'Gmax' using an Auran Jet plug in. Any file that has the .PM extension is a Progressive Mesh file, similarly any file with the .IM extension is a Indexed Mesh file.

Gmax is a program created by Discreet. It's a game-specific version of their very popular '3D Studio Max' program. Gmax ships with the retail version of Trainz and is available for free download from the Discreet site. In order to use Gmax with Trainz you will also need to download the Trainz 'Content Creation Pack' from the Auran website. This pack installs into Gmax and will enable you to export content directly into the .PM or IM file formats that Trainz uses.

So what is Gmax? Gmax is a 3D creation program that enables you to make things such as locomotives, items of rolling stock or scenery and trackside accessories. It is quite a complex program, and you can expect quite a steep learning curve should you decide to dive in and learn it. However, on the plus side, the benefits are well worth it, and if you take the time to learn it well, you will certainly be able to create some masterpieces.

3: Create textures (Optional)

Creating textures for your creations is a very important part of the content creation process. Making good textures is one of the hardest things to do; but they can be the difference between a good-looking model and a great looking model.

Textures are created for Trainz using any 3rd party program that supports the creation of 2D art, like Adobe Photoshop or Paint Shop Pro. If you're on a budget I'd suggest Paint Shop Pro. The latest version retails for just US\$109 boxed and it is a great program. Photoshop is arguably a more powerful program, but it's equally more expensive.

4: Create a configuration file (Mandatory)

Each and every item of content for Trainz has what's called a configuration file (config.txt). This file is a human-readable text file that resides in the directory along with its corresponding item of content. Depending on what the item of content is will determine the necessary contents of the text file, but it will always contain a description and a KUID (A KUID is defined a little later in this document).

Items of content created for Trainz are always assigned to a group of content called a KIND. A KIND is a type of content that has particular properties that Trainz recognizes. For example one type of KIND is a TRACK. Trainz understands that items of content that belong to a group of this KIND are able to be used to run locomotives and rolling stock over it. Other KINDs are listed further below in this document.

5: Upload your new content to the Auran Download Station (Mandatory)

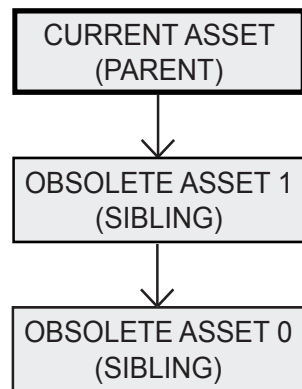
Auran has produced a product called 'Content Dispatcher' (default location: `C:\Program Files\Auran\Trainz\`). This is a stand alone application that content creators use to package their creations prior to uploading them to Auran.

Content Dispatcher is a very useful application in that it automatically performs error checking and simplifies the preparing your content for upload.

Put simply, Content Dispatcher understands the way Trainz works and as such it is able to watch for a number of common mistakes that you may make in preparing your content for distribution. It can also spot certain types of errors within the `config.txt` file; and hence we strongly recommend its use in preparing your content.

Since the Auran Download Station expects the uploads that it receives to have been processed by the Content Dispatcher, using any other utility may mean that the contents of your upload pack may be incorrectly identified and processed. The Content Dispatcher program embeds information into the upload package that is required by the Auran Download Station. Not using Content Dispatcher will almost certainly cause this process to fail and your content will not be easy to locate or may be missing vital 'parental' information.

The term 'parental' is a general term used to describe the asset's heirarchy - much like a family tree. See the following example:



Side note:

'Parental' information is a concept new to Trainz but we're sure you'll see just how useful it can be. As a content creator you may, from time to time, wish to make changes to content that you've already created and uploaded. Perhaps you made a loco shed and you've improved its look with a better 3D model and textures. Either way, you want to upload the new shed to Auran. The problem is that hundreds, maybe thousands

of people have already downloaded your older shed and many hundreds of users have made layouts that included your old shed as well. How then do users know that you have a new shed available and how do users downloading the layouts using your shed know to get the new shed and not the old one?...

The answer lies in Parental information and the 'obsoletes' field in the contents `config.txt` file is the key to this. The obsoletes field is seen by the Auran Download Station when it receives an upload package from you. When it sees this entry, it causes your package to be processed in a special way. The obsoletes field tells the Download Station the 'old' KUID that your 'new' loco shed is replacing. The relevant entries in the `config.txt` file might look like this.

```
kuid <KUID:1234:5678>
obsolete-table {
    0 <KUID:1234:5677>
}
```

This entry would tell the DS that your new loco shed (KUID 1234:5678) replaces your older shed (KUID1234:5677). Upon seeing this the DS does the following. It deletes your old loco shed but retains the index that pointed to it. It then updates that index such that it now points to your new loco shed. It then files your new shed and creates an index for it. So what has this accomplished? Well for one, when someone now comes to the DS and searches for KUID 1234:5677 (Your old shed) the DS will respond 'KUID 1234:5677 has been superseded by KUID 1234:5678' and then display the new shed for download. If they search for the new shed directly then it will be displayed. This parental information can go

on indefinitely. In other words you can replace your older items over and over with new ones; and if you use the obsoletes fields, all will be well.

Perhaps the best example of the use of the obsoletes field is with layouts. Continuing the above example, a user downloads a map in which your old loco shed was used. Because they don't have your old shed they will receive the 'Update Content' report listing the old shed as missing content and providing a URL for them to locate it. Of course when they click on the link to search for your shed, the DS sees that the old shed has been replaced and promptly displays the new one for download, all without the user ever knowing anything about your content or how many revisions of it you've made!

Side note:

You may be wondering why it is that you only need to perform steps 1, 4 and 5 to make a new item of content. Surely you would need to either make a new model or re-texture an old one, right? Well you can in fact make a new item of content just by creating a new configuration file and then using a 3D model and a set of textures that have already been created. An example of this would be making a new locomotive with slightly different performance characteristics. In this case all that's required is to do the research, make a new configuration file that references the already created 3D model and textures and then upload it.

By using the Content Dispatcher you package the content you've created into a parcel and then uploaded it to Auran's Download Station.

WHAT IS A KUID?

A KUID is a serial number that is allocated to all content created for Trainz and can be thought of much like a bar code. A KUID takes the form of two numbers each separated by a colon; for example, 98765:43210 would be a KUID. As you can see, there are two numbers, one to the left and the other to the right of the colon.

USER ID

The number to the left of the colon in a KUID is the USER ID of the content creator. This is the same number you will have entered into Trainz when you told it your USER ID and the same number that you got from the Planet

Auran 'YOUR USER ID' selection. In fact, every member of the Trainz community who is a member of Planet Auran gets a USER ID. Now, you may be wondering why you need a USER ID if you don't intend to create any content for Trainz (like a new locomotive for example). Well, if you intend to make a layout at some point in time and you'd like to share that layout with your friends or other community members, then you are in fact a content creator.

CONTENT ID

The number to the right of the colon in a KUID is the CONTENT ID. This is a number that the content creator assigns to each of their creations to uniquely identify them. The only circumstances where you would not assign a KUID to one of your creations in the case of a saved layout, or when exporting from Trainz Paintshed. In these circumstances, number allocation will be done for you automatically with a number starting at 100,000 and going up from there. Numbers below 100,000 therefore are designed to be used by those members of the community who make other items of content (not made by Surveyor or Trainz Paintshed). In that case they can specify the number they use and there are guidelines governing the use and allocation of these numbers later in this document.

Side note:

In case you're wondering, two individual content creators can use the same CONTENT ID for their creations and Trainz will still be able to uniquely identify them since each CONTENT ID is preceded (To the left of the colon) by a USER ID. In other words 1:1 is seen a differently from 2:1 by Trainz.

THE DO'S AND DON'TS OF CONTENT CREATION

DO!

...make use of Auran Content Dispatcher (CD) to package and upload your content to Auran's Download Station. The CD forms an integral part of future versions of Trainz and bypassing it's use may render your content unusable once Trainz is able to natively interrogate the Download Station (DS).

...make sure that each and every version of your content is packaged and uploaded to the DS. In other words if you make refinements to your creation, send each new version to the DS.

...make sure that you use a new KUID for each version of your content. Even if you make minor refinements, every time an item of content is uploaded to the DS you must give it a new KUID.

...make sure that for each new version of content you create that you include a reference to the older version by using the 'obsoletes' keyword in the content config.txt file. This is required as people with older versions of your content need to be able to locate the new version of it. This keyword is used by the DS to index content so that the 'auto download' systems of Trainz will work. If you omit this keyword you will inhibit this function and make it a lot harder, or even impossible, for people to locate a new version of your content.

DON'T!

...extract data from Auran's .JA files. This process is totally unsupported and content that references extracted files will almost certainly not work in future version of Trainz.

...create or use 3rd party utilities that alter any content files directly. If you are doing so in order to make a small change to a file to fix a problem, your best approach is to pack and upload a new version to the DS.

CHAPTER 2 - CLASSES AND CODES

CONTENT ID RANGES

KUID Syntax: kuid <KUID:xxxxx:yyyyyy>

The xxxxx is your USER ID.

The yyyyyy is the CONTENT ID.

1 to 9999	Locomotives (We would suggest using the locomotives road number if possible)	37000-37999	Splines (including roads, powerlines etc)
10000-14999	Passenger cars	38000-38999	Rails
15000-19999	Freight cars	39000-39999	Buildings civil (Police, Fire, Hospital, Schools, Library)
20000-20999	Routes (Only use these numbers if for some reason you wish to over-ride the Trainz assigned number (for distribution for example). Trainz automatically assigns a content ID above 100000 for routes)	40000-40999	Buildings military
21000-21999	Textures	41000-49999	Reserved (Do not use this range, it is reserved for future use)
22000-22999	Foliage	50000-50999	Bogies/Trucks
23000-23999	Signposts (Other than train signalling)	51000-52999	Enginespec
24000-24999	Signalling	53000-53999	Enginesound
25000-25999	Buildings residential	54000-54999	Hornsound
26000-26999	Buildings commercial	55000-56999	Interior
27000-27999	Buildings industrial	57000-57999	Pants (Pantographs)
28000-28999	Buildings railroad	58000-58999	Public Fun Places (Stadiums, Racetracks, Amusement Parks, Public Monuments, Circus)
29000-29999	Transportation land (This includes only those objects that actually move)	60000-99999	Anything else (Use at your discretion for objects that do not fit into the other categories)
30000-30999	Transportation sea (Same as above)	100000+	Numbers in this range are auto allocated by Trainz. Do not manually allocate any numbers in this range.
31000-31999	Transportation air (Same as above)		
32000-32999	Bridges		
33000-33999	Tunnels		
35000-35999	People		
36000-36999	Animals		

Sidenote:

Always use a new Content ID for each new item of content. For each new version of content make sure you use a new Content ID and make use of the obsoletes field of the config.txt file.

CATEGORY CLASS

Trainz Loco, Car Classifications are codes that are added to the config.txt file of locos and rolling stock only. They represent a standardized system for referring to the various types of locos and rolling stock. Future version of Trainz will make use of the codes so adding them to any locos and rolling stock you create is advisable.

The Category Classes are:

Class "A"	Motive Power
Class "B"	Buildings and Structures
Class "C"	Cabeese
Class "D"	Defence
Class "E"	Environment
Class "F"	Foliage
Class "G"	Ground
Class "L"	Light Rail & Monorail
Class "M"	Maintenance Of Way
Class "O"	Organism
Class "P"	Passenger & Mail Cars
Class "R"	Railcars & Multiple Unit Sets
Class "S"	Splines
Class "T"	Track
Class "V"	Vehicles
Class "W"	Wayside
Class "X"	Freight Cars
Class "Y"	Maps and Scenarios
Class "Z"	Train Parts

Each Category Class may have many subcategories as listed below. Please choose the most appropriate Category Class for your item.

The Category Class is listed in the config.txt file of each item of content as follows:

category-class xxx (Where xxx is the Category Class)

Selecting a correct Category Class is important since future versions of Trainz will allow users to use the Category Class as a sort and selection criteria inside My Collection and in Drivers Consist selection screen.

A MOTIVE POWER

AA	Electric Multi-current
AC	AC Electric
AD	DC Electric
AE	Experimental or Special
AG	Gas Turbine
AH	Diesel Hydraulic
AL	Diesel & Diesel Electric
AM	Mammal
AS	Steam Loco & Tender
AT	Steam Tank

B BUILDINGS & STRUCTURES

BC	Commercial
BI	Industrial
BH	Home & Residential
BR	Railway
BS	Special (I.e. military)
BT	Traffic & Streetscape
BU	Utility (incl. Civil buildings)

C CABEESE

CB	Brake van
CC	Caboose

D DEFENCE

DA	Military motive power
DE	Military experimental & special vehicles
DP	Military equipment - lab & personnel vehicles
DX	Military equipment - freight

E ENVIRONMENT

ES	Sky
EW	Water

F FOLIAGE

FC	Cactii
FF	Flowers
FO	Orchards & Crops
FS	Shrub
FT	Trees

G GROUND

GA	Arid
GL	Lush
GS	Seasonal

L LIGHT RAIL & MONORAIL

LS	articulated train sets
LT	trolleys, trams & streetcars

LM monorail vehicles

M MAINTENANCE OF WAY

MA Camp vehicles
 MB Ballast cars
 MC Cranes/lifting
 MD Diagnostic vehicles (e.g. dynamometer)
 ME Instructional vehicles
 MF Fire vehicles
 MI Inspection vehicles
 MT Track vehicles (e.g. tamper)
 MP Snow ploughs
 MS Section cars (e.g. fairmont)
 MX Freight equipment (for MoW traffic)
 MW Weed spray

O ORGANISM

OA Animal Kingdom
 OH Human

P PASSENGER & MAIL CARS

PA Suburban/short haul (no W.C.)
 PB Baggage cars
 PC Coach/chair cars
 PD Dome cars
 PH Bar/cafe/restaurant cars
 PL Lounge cars
 PM Mail cars
 PO Observation cars
 PP Power cars
 PR Buffet/dining/restaurant cars
 PS Sleeping cars
 PU Special cars (e.g. Gaming Cars)
 PV Private cars
 PX Composite passenger cars

R RAILCARS & MULTIPLE UNIT SETS

RA AC electric
 RC DC electric
 RD Diesel & diesel electric
 RH Diesel hydraulic
 RP Petrol
 RS Steam

S SPLINES

SF Fences
 SR Roads
 SP Platforms
 SS Structure
 SV Vegetation

T TRACK

TB Bridge
 TR Rails
 TT Tunnel

V VEHICLES

VA Air
 VL Land
 VS Sea

W WAYSIDE

WA Signalling
 WS Trackside signage
 WX Accessories

X FREIGHT CARS

XA Auto transporter

XAA Open sides
 XAB Auto box car

XB Box car/covered van

XBD Dangerous goods
 XBG General service
 XBI Insulated

XF Flat

XFA articulated
 XFC Intermodal
 XFD depressed center
 XFH heavy duty
 XFM general service

XG Gondola/open wagon

XGB Bottom dumping
 XGC Combination bottom/end/side dumping
 XGE End dumping
 XGR Rotary dumping
 XGS Side dumping
 XGT Covered

XH Hopper

XHB Bottom dumping
 XHC Combination bottom/end/side dumping
 XHE End dumping
 XHR Rotary dumping
 XHS Side dumping
 XHT Covered

XI Foundry

XIB Bottle/torpedo cars

XIT Tipper/slag cars

XL Livestock

XLA Single deck

XLC Multiple deck and convertible

XLH Horse box

XR Refrigerated

XRI Ice chilled

XRM Mechanically chilled

XS Special

XSN Novelty

XSU Unclassified

XT Tanker

XTA Domeless

XTS Single dome

XTM Multiple dome

XV Ventilated car/louvred van

XVG General service

XVP Produce service

Y MAPS & SCENARIOS

YM Map

YS Scenario

Z TRAIN PARTS

ZB Bogie/Truck

ZE Enginespec

ZH Hornsound

ZI Interior

ZP Pantographs

ZS Enginesound

REGION CODES

Region codes are a single or multiple line code that is included in the config.txt file. The codes added here will be used in future versions of Trainz as a sort and selection criteria. For content that exists in multiple areas, list each area on a separate line.

For example, a locomotive that was available in the United States and Canada would be specified as follows:

```
category-region-0 US
category-region-1 CA
```

The first region code specified must be category-region-0. Subsequent entries increase this code by one for each entry, so the next entry would be category-region-1 and so on.

Side note: Do not skip a code! For example specifying your regions as category-region-0 US and then category-region-2 CA would result in the second region code being missed or unread!

The region codes that are recognized by Trainz are as follows:

AD Andorra
 AE United Arab Emirates
 AF Afghanistan
 AG Antigua and Barbuda
 AI Anguilla
 AL Albania
 AM Armenia
 AN Netherland Antilles
 AO Angola
 AQ Antarctica
 AR Argentina
 AS American Samoa
 AT Austria
 AU Australia
 AW Aruba
 AZ Azerbaidjan
 BA Bosnia-Herzegovina
 BB Barbados
 BD Bangladesh
 BE Belgium
 BF Burkina Faso
 BG Bulgaria
 BH Bahrain
 BI Burundi
 BJ Benin
 BM Bermuda
 BN Brunei Darussalam

BO Bolivia
 BR Brazil
 BS Bahamas
 BT Buthan
 BV Bouvet Island
 BW Botswana
 BY Belarus
 BZ Belize
 CA Canada
 CCocos (Keeling) Isl.
 CF Central African Rep.
 CG Congo
 CH Switzerland
 CI Ivory Coast
 CK Cook Islands
 CL Chile
 CM Cameroon
 CN China
 CO Colombia
 CR Costa Rica
 CS Czechoslovakia
 CU Cuba
 CV Christmas Island
 CY Cyprus
 CZ Czech Republic
 DE Germany
 DJ Djibouti
 DK Denmark
 DM Dominica
 DO Dominican Republic
 DZ Algeria
 EC Ecuador
 EE Estonia
 EG Egypt
 EH Western Sahara
 ES Spain
 ET Ethiopia
 FI Finland
 FJ Fiji
 FK Falkland Isl.(Malvinas)
 FM Micronesia
 FO Faroe Islands
 FR France
 GA Gabon
 GB Great Britain
 GD Grenada
 GE Georgia
 GH Ghana
 GI Gibraltar
 GL Greenland
 GP Guadeloupe (Fr.)
 GQ Equatorial Guinea
 GF Guyana (Fr.)
 GM Gambia
 GN Guinea
 GR Greece
 GT Guatemala
 GU Guam (US)
 GW Guinea Bissau
 GY Guyana
 HK Hong Kong
 HM Heard & McDonald Isl.

HN	Honduras	NL	Netherlands
HR	Croatia	NO	Norway
HT	Haiti	NP	Nepal
HU	Hungary	NR	Nauru
ID	Indonesia	NT	Neutral Zone
IE	Ireland	NU	Niue
IL	Israel	NZ	New Zealand
IN	India	OM	Oman
IO	British Indian O. Terr.	PA	Panama
IQ	Iraq	PE	Peru
IR	Iran	PF	Polynesia (Fr.)
IS	Iceland	PG	Papua New Guinea
IT	Italy	PH	Philippines
JM	Jamaica	PK	Pakistan
JO	Jordan	PL	Poland
JP	Japan	PM	St. Pierre & Miquelon
KE	Kenya	PN	Pitcairn
KG	Kirgistan Ex-USSR	PT	Portugal
KH	Cambodia	PR	Puerto Rico (US)
KI	Kiribati	PW	Palau
KM	Comoros	PY	Paraguay
KN	St. Kitts Nevis Anguilla	QA	Qatar
KP	Korea (North)	RE	Reunion (Fr.)
KR	Korea (South)	RO	Romania
KW	Kuwait	RU	Russian Federation Ex-USSR
KY	Cayman Islands	RW	Rwanda
KZ	Kazachstan	SA	Saudi Arabia
LA	Laos	SB	Solomon Islands
LB	Lebanon	SC	Seychelles
LC	Saint Lucia	SD	Sudan
LI	Liechtenstein	SE	Sweden
LK	Sri Lanka	SG	Singapore
LR	Liberia	SH	St. Helena
LS	Lesotho	SI	Slovenia
LT	Lithuania	SJ	Svalbard & Jan Mayen Is
LU	Luxembourg	SK	Slovak Republic
LV	Latvia	SL	Sierra Leone
LY	Libya	SM	San Marino
MA	Morocco	SN	Senegal
MC	Monaco	SO	Somalia
MD	Moldavia Ex-USSR	SR	Suriname
MG	Madagascar	ST	St. Tome and Principe
MH	Marshall Islands	SU	Soviet Union
ML	Mali	SV	El Salvador
MM	Myanmar	SY	Syria
MN	Mongolia	SZ	Swaziland
MO	Martinique (Fr.)	TC	Turks & Caicos Islands
MR	Mauritania	TD	Chad
MS	Montserrat	TF	French Southern Terr.
MT	Malta	TG	Togo
MU	Mauritius	TH	Thailand
MV	Maldives	TJ	Tadjikistan Ex-USSR
MW	Malawi	TK	Tokelau
MX	Mexico	TL	East Timor
MY	Malaysia	TM	Turkmenistan Ex-USSR
MZ	Mozambique	TN	Tunisia
NA	Namibia	TO	Tonga
NC	New Caledonia (Fr.)	TR	Turkey
NE	Niger	TT	Trinidad & Tobago
NF	Norfolk Island	TV	Tuvalu
NG	Nigeria	TW	Taiwan
NI	Nicaragua	TZ	Tanzania

UA	Ukraine
UG	Uganda
UK	United Kingdom
UM	US Minor outlying Isl.
US	United States
UY	Uruguay
UZ	Uzbekistan Ex-USSR
VA	Vatican City State
VC	St.Vincent & Grenadines
VE	Venezuela
VG	Virgin Islands (British)
VI	Virgin Islands (US)
VN	Vietnam
VU	Vanuatu
WF	Wallis & Futuna Islands
WS	Samoa
YE	Yemen
YU	Yugoslavia
ZA	South Africa
ZM	Zambia
ZR	Zaire
ZW	Zimbabwe

The era codes that are recognized by Trainz are as follows:

1800s
1810s
1820s
1830s
1840s
1850s
1860s
1870s
1880s
1890s
1900s
1910s
1920s
1930s
1940s
1950s
1960s
1970s
1980s
1990s
2000s
2010s

ERA CODES

Era codes are a single or multiple line code that is included in the config.txt file. The codes added here will be used in future versions of Trainz as a sort and selection criteria. For content that exists in multiple eras list each era on a separate line.

For example, a locomotive that was available in the 1960s and 1970s would be specified as follows:

```
category-era-0 1960s
category-era-1 1970s
```

The first era code specified must be category-era-0. Subsequent entries increase this code by one for each entry, so the next entry would be category-era-1 and so on.

Side note: Do not skip a code! For example specifying your eras as category-era-0 1960s and then category-era-2 1970s would result in the second era code being missed or unread!

CHAPTER 3 - CONFIG AND KIND

CONFIG.TXT

Each item of content that you create is required to have a config.txt file. This file is a simple text file that is used to describe the item of content to Trainz. The contents of a typical config file may look like this:

Blue text indicates Required tags
Green text indicates Optional tags.

```
kuid <KUID:1234:5678>
kuid-table {
    0 <KUID:1234:6000>
    1 <KUID:1234:6001>
}
obsolete-table {
    0 <KUID:1234:5676>
    1 <KUID:1234:5677>
}
username My Locomotive
description "This is an example Asset.
Note that you can have multiple lines but
no double quote characters in here. Trainz
automatically wraps this text."
region Australia
trainz-build 1.5
kind engine
category-class AD
category-region-0 US
category-region-1 UK
category-region-2 AU
category-era-0 1960s
category-era-1 1970s
author Greg Lane
organisation Trainz Thingz
contact-email helpdesk@auran.com
contact-website http://www.auran.com
```

*Side-note:
(kuid-table must be
included where the
config references
additional KUIDs,
such as a bogey, or
a pantograph. The
Download Station
performs a search, and
those found are added
to the download pack)*

kuid-table

A list of KUIDs required for this asset to function correctly.

obsolete-table

A history of this asset. Each KUID listed is a previous version of this asset. If there are no previous versions display as follows:

```
obsolete-table {
}
```

username

The human-readable English name of this asset.

description

The human-readable multi-line English description of this asset.

region

The country region to which this asset belongs. This should be one of the Auran-supplied region names

trainz-build

The Trainz build number for which this asset was created.

kind

The asset kind. Must be one of the Auran-supplied asset kinds.

category-class

The class code for this asset. Classes are unique per kind of asset but may not be across all assets

category-region-0

A list of REGION codes or REGION GROUP codes, starting at category-region-0 and progressing category-region-1, etc. with one code per tag

category-era-0

A list of ERA codes, starting at category-era-0 and progressing category-era-1, etc. with one code per tag.

Breakdown of Config.txt

kuid

Unique ID of this asset. Any subsequent versions of this asset MUST have a different KUID. The KUID contains basic creator information.

KIND (Config variables)**KIND: ENGINE**

File location - \Trainz\Engines\

Locomotive performance parameters...

Config.txt :

```

kuid <KUID:-1:42004202>
kind engine
rem F7 Generic Engine
flowsize {
trainbrakepipe 170000
epreservoirpipe 0.1
no3pipe 0.1
no4pipe 0.1
auxreservoirvent 0.1
auxreservoir_no3 0.1
auxreservoir_trainbrakepipe 0.1
autobrakecylindervent 0.1
auxreservoir_autobrakecylinder 0.1
equaliser_mainreservoir 0.06
equaliservent 0.06
equaliserventhandleoff 0.1
equaliserventemergency 0.1
no3pipevent 1.5
no3pipe_mainreservoir 0.1
compressor 10
trainbrakepipe_reservoir 1
trainbrakepipevent 0.06
no3pipe_autobrakecylinder 0.1
epreservoirpipe_autobrakecylinder 0.1
mainreservoir_ep 0.1
vacuumbrakepipe 0.1
vacuumbrakepipereleasevent 0.1
vacuumbrakepipevent 0.1
vacuumbrakereservoir_vacuumbrakepipe
0.1
vacuumbrakecylinder_vacuumbrakepipe
0.1
highspeedexhauster_vacuumbrakepipe 0.1
}
volume {
scale 1
trainbrakepipe 0.2
epreservoirpipe 0.2
no3pipe 0.2
no4pipe 0.2
auxreservoir 0.0384678
autobrakecylinder 0.00969387
vacuumbrakepipe 0
vacuumbrakereservoir 0
vacuumbrakecylinder 0
mainreservoir 0.9
equaliser 0.5
independantbrakecylinder 0.0103239

```

```

}
pressure

{
scale 1
compressor
0.00946941
mainreservoir
0.00946941
highspeedexhauster 0
brakepipe 0.00665741
brakeinitial 0.00609501
brakefull 0.00504051
indbrakefull 0.00504051
trainbrakepipe_start 0.00504051
epreservoirpipe_start 0
no3pipe_start 0
no4pipe_start 0
auxreservoir_start 0.00504051
autobrakecylinder_start 0.00507566
vacuumbrakepipe_start 0
vacuumbrakereservoir_start 0
vacuumbrakecylinder_start 0
mainreservoir_start 0.00806341
equaliser_start 0.00504051
independantbrakecylinder_start
0.00507566
}
mass {
scale 1
fuel 6.2156e+006
}
motor {
resistance 1.7
adhesion 2.5
maxvoltage 600
maxspeed 40
brakeratio 55000
max-accel
3500
max-decel
9000
axle-count 4
surface-area 80
moving-friction-coefficient .03
air-drag-coefficient .00017
}

throttle-power
{
0 {
0 0
}
1 {
0 30
5 25
}

```

```

    10    15
    12    0
  }
2 {
    0    55
    5    48
    10   40
    15   30
    30   10
  }
3 {
    0    100
    5    70
    10   55
    15   50
    30   10
  }
4 {
    2    150
    5    80
    10   60
    15   55
    30   10
  }
5 {
    0    200
    5    150
    10   80
    15   65
    30   10
  }
6 {
    0    231
    5    180
    10   120
    15   90
    30   20
  }
7 {
    0    260
    5    220
    10   200
    15   110
    30   20
  }
8 {
    0    300
    3.5  250
    5    190
    10   160

```

```

    20    120
    30    110
    35    140
    44    20
  }
}

dynamic-brake
{
    0 {
        0    0
    }
    1 {
        1.333  0
        2      30
        5      25
        10     15
        12     0
    }
    2 {
        1.333  0
        3      50
        10     35
        14     20
        15     0
    }
    3 {
        1.333  0
        3      60
        10     40
        17     20
        22     0
    }
    4 {
        1.333  0
        4      80
        10     60
        20     20
        25     0
    }
    5 {
        1.333  0
        5      90
        10     70
        25     25
        29     0
    }
    6 {
        1.333  0
        5      150

```

```

        10      80
        29      70
        32      0
    }
    7 {
        1.333  0
        5      200
        10     100
        32     60
        36     0
    }
    8 {
        1.33  0
        5     200
        10    150
        36    50
        40    0
    }
}

region Australia
kuid-table {
    0 <KUID:###:#####>
    1 <KUID:###:#####>
}
obsolete-table {
}
username " "
description " "
trainz-build 1.5
category-class
category-region-0
category-era-0

```

Breakdown of engine

kind – asset type

rem – comment used to display engine name

flowsize

rate of flow through pipes, generally leave these settings:

```

trainbrakepipe 170000
epreservoirpipe 0.1
no3pipe 0.1
no4pipe 0.1
auxreservoirvent 0.1
auxreservoir_no3 0.1
auxreservoir_trainbrakepipe 0.1
autobrakecylindervent 0.1
auxreservoir_autobrakecylinder 0.1
equaliser_mainreservoir 0.06
equaliservent 0.06

```

```

equaliserventhandleoff 0.1
equaliserventemergency 0.1
no3pipevent 1.5
no3pipe_mainreservoir 0.1
compressor 10
trainbrakepipe_reservoir 1
trainbrakepipevent 0.06
no3pipe_autobrakecylinder 0.1
epreservoirpipe_autobrakecylinder 0.1
mainreservoir_ep 0.1
vacuumbrakepipe 0.1
vacuumbrakepipereleasevent 0.1
vacuumbrakepipevent 0.1
vacuumbrakereservoir_vacuumbrakepipe 0.1
vacuumbrakecylinder_vacuumbrakepipe 0.1
highspeedexhauster_vacuumbrakepipe 0.1

```

volume – size of pipes and appliances

scale 1

trainbrakepipe 0.2
brake pipe volume

epreservoirpipe0.2
For electro pneumatic braking - not currently in use, generally leave this setting

no3pipe 0.2
Independent brake pipe

no4pipe0.2
Bail pipe - not currently in use, generally leave this setting

Auxreservoir0.0384678
Auxiliary reservoir volume.

Autobrakecylinder 0.00969387
Brake cylinder volume.

vacuumbrakepipe0
vacuumbrakereservoir 0
vacuumbrakecylinder 0
For vacuum braking - not currently in use, generally leave this setting

mainreservoir0.9
Main reservoir volume.

equaliser0.5
Equalising reservoir volume

independantbrakecylinder0.0103239
Loco brake cylinder volume

pressure
brake system pressures.

scale 1

multiplies pressure by given value, generally leave this setting.

Compressor.....0.00946941
(120psi expressed in grams/m³)
compressor maximum pressure.

mainreservoir.....0.00946941
main reservoir maximum pressure

highspeedexhauster0
For vacuum braking - not currently in use,
generally leave this setting

brakepipe0.00665741
(80psi expressed in grams/m³)
brake pipe pressure when fully charged

brakeinitial.....0.00609501
(72psi expressed in grams/m³)
brake pipe pressure after initial service
reduction (for self lapping brakes)

brakefull0.00504051
(57psi expressed in grams/m³)
Brake pipe pressure after full service reduction
(for self lapping brakes)

indbrakefull0.00504051
Brake cylinder pressure for independant brake
service.

trainbrakepipe_start0.00504051
Brake pipe pressure on loading the game.

epreservoirpipe_start0
For electro pneumatic braking - not currently in
use, generally leave this setting

no3pipe_start0
no4pipe_start0
Generally leave these settings.

auxreservoir_start.....0.00504051
Auxiliary reservoir pressure on loading the
game.

autobrakecylinder_start0.00504051
Train brake cylinder pressure on loading the
game.

vacuumbrakepipe_start0
vacuumbrakereservoir_start0

vacuumbrakecylinder_start0
For vacuum braking - not currently in use,
generally leave this setting

mainreservoir_start0.00806341
(100psi expressed in grams/m³)
Main Reservoir pressure on loading the game.

equaliser_start0.00504051
Equalising Reservoir pressure on loading the
game.

independantbrakecylinder_start0.00504051
Locomotive brake cylinder pressure on loading
the game.

mass**scale 1**

multiplies fuel mass by given value, not
currently in use, generally leave this setting.

fuel6.2156e+006
fuel level, not currently in use, generally leave
this setting.

motor

resistance.....1.7
power figure for DCC, higher resistance
value=less power

adhesion2.5
adhesion parameter, higher value=greater
adhesion

maxvoltage600
generally leave this setting

maxspeed40
maximum speed for DCC, expressed in
metres per second.

Brakeratio..... 55000
brake force for pressure reduction

max-accel.....3500
max-decel.....9000
parameters for DCC acceleration &
deceleration.

axle-count.....4
Resistance – axle count

surface-area80
Resistance – surface area

moving-friction-coefficient.....0.03

Resistance – moving friction

air-drag-coefficient.....0.00017

Resistance – air drag

throttle-power

Acceleration variables in cabin mode

```
1 {      = notch number (1)
    0      30
    5      25      = At speed 5
    10     15      acceleration = 25
    12     0
}
```

dynamic-brake

Deceleration variables while dynamic braking in cabin mode

```
1 {      = notch number (1)
    1.333  0
    2      30
    5      25      = At speed 5
    10     15      deceleration = 25
    12     0
}
```

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.

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 entire **pressure** section from the config of a loco that uses the pressure you desire.

*Note: Converting PSI to Grams /m cubed...

e.g. 90psi... (90+14.7).0000703

104.7 x .0000703=.00736041

KIND: BOGEY

File location - \World\custom\bogeys\

This is a bogey.

Referenced by the *bogey* tag in a traincar config.txt

Config.txt:

```
kuid <KUID:###:#####>
kind bogey
obsolete-table {
}
username
description
trainz-build 1.5
category-class AC
category-region-0 AT
category-era-0 1980s
animdist 2.1
```

Breakdown:**animdist**

Leave this tag out if the bogey is not animated.

The distance travelled in meters by the bogeys in 1 second (30 frames) of animation. Bogey animations (exported from Gmax or 3ds Max) are called “anim.kin”.

Refer to Chapter 4 - Bogeys for modeling guidelines

KIND: TRAINCAR

File location - \World\custom\trains\

This is a locomotive or car item.

Config.txt:

```
kuid <KUID:###:#####>
kind traincar
origin AUT
name OBB 1044
company OBB
mass 84000
bogey <KUID:###:#####>
pantograph <KUID:###:#####>
interior <KUID:###:#####>
engine 1
fonts 2
running-numbers {
  rn-0 #104420
  rn-1 #104427
  rn-2 #104430
  rn-3 #104452
}
enginespec <KUID:-1:42004207>
enginesound <KUID:-1:42003002>
hornsound <KUID:-1:42003101>
smoke0
{
  attachment      a.steam.l
  mode             anim
  color            255,255,255,150

  start           0
  period          0.4
  rate            2
  velocity        1
  lifetime        2
  minsize         0.05
  maxsize         1
}
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
light_color 255,255,255
ditch_color 255,200,200
```

Breakdown of Traincar Config.txt:**origin**

The Country Abbreviation

company

The Locomotive or car owner

mass

Mass in kilograms

bogey

The bogey kuid number (default for a.bog0 and a.bog1)

bogey-1

The bogey kuid number for a.bog1 (Used only if different to a.bog0)

bogey-r and bogey-1-r

Used instead of 'bogey' and bogey-1. The bogey will have reversed orientation. Note: This will cause bogey animation to play in reverse.

pantograph

The pantograph kuid number inserted at a.pant0, a.pant1, etc. Use this tag only when needed.

interior

Kuid number of the required interior. Inserted at a.cabfront. Use this tag only when needed eg Locomotive's.

engine

States type of traincar.

0 = Car

1 = Locomotive

fonts

Indicates how many types of numbering fonts used.

E.g. **0** = no fonts used

1 = one font

Digit textures (*digit_1.tga to digit_6.tga*) replaced automatically with alphanumeric textures (*alphanumeric_0 to alphanumeric_9*) as numbers are changed in 'My Collection'.

2 = two fonts

Digit textures (*digit_1a.tga to digit_6a.tga and digit_1b.tga to digit_6b.tga*) replaced automatically with alphanumeric textures (*alphanumeric_0a to alphanumeric_9a and alphanumeric_0b to alphanumeric_9b*) as numbers are changed in 'My Collection'.

running-numbers

The default numbers viewed before being changed in 'My Collection'. Generally Auran uses 4 variations of numbers (up to 6 digits each). Note: This tag not required if *fonts 0* is used.

enginespec

The engine kuid number. This specifies the driver physics boundaries for the traincar.

Located in \Trainz\Engines\

Refer Chapter 3, KIND: TRAINCAR

WARNING:

ALTERING *ENGINESPEC* FIGURES MAY RESULT IN UNDESIRE EFFECTS IN PERFORMNACE AND BEHAVIOR OF YOUR TRAINS. (MAKE BACK-UP COPIES OF YOUR ENGINE CONFIG FILES!!)

enginesound

The kuid number for the traincar's sound.

Located in \world\custom\enginesound\

Refer Chapter 3, KIND: ENGINESOUND

hornsound

The kuid number for the traincar horn sound.

Located in \world\custom\hornsound\

Refer Chapter 3, KIND: HORNSOUND

smoke0

Sets boundaries for smoke, steam, vapor and similar effects. Refer: [Smoke Effects](#)

description “ “

Description of model for 'My Collection' information

light_color

RGB headlight colour. Eg. 255,255,255

ditch_color

RGB ditch light colour. Eg. 255,255,255

KIND: ENGINESOUND

File location - \World\custom\enginesound\

This is the traincar sound.

Referenced by the *enginesound* tag in a traincar config.txt

Config.txt:

```
kuid <KUID:###:#####>
kind enginesound
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class AC
category-region-0 AT
category-era-0 1980s
```

Sound .wav files to be located in the same subfolder as the enginesound config.txt.

down 2 - 1.wav	up 1 - 2.wav	idle 1.wav	start 1-1.wav
down 3 - 2.wav	up 2 - 3.wav	idle 2.wav	stop 1-1.wav
down 4 - 3.wav	up 3 - 4.wav	idle 3.wav	
down 5 - 4.wav	up 4 - 5.wav	idle 4.wav	
down 6 - 5.wav	up 5 - 6.wav	idle 5.wav	
down 7 - 6.wav	up 6 - 7.wav	idle 6.wav	
down 8 - 7.wav	up 7 - 8.wav	idle 7.wav	
		idle 8.wav	

KIND: HORNSOUND

File location - \World\custom*hornsound*

This is the traincar horn sound.

Referenced by the *hornsound* tag in a traincar config.txt

Config.txt:

```
kuid <KUID:###:#####>
kind hornsound
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class
category-region-0
category-era-0
```

Sound .wav files to be located in the same subfolder as the hornsound config.txt.

horn.wav

idle.wav

KIND: INTERIOR

File location - world\custom\Interiors\

This is the traincar interior.

Referenced by the *interior* tag in a traincar config.txt

Green text = New to UTC

Config.txt:

```
kuid <KUID:####:#####>
kind interior
mesh interiormesh.pm (main .pm file)
camera 1.254, -0.053, 0.713
cameralist
{
camera0 -0.779, 0.035, 0.665, -1.16, -0.142
camera1 0.688, 0.1, 0.666, 0.057, -0.006
camera2 1.254, -0.053, 0.713
camera3 -1.169, -0.111, 0.725, 0.83, -6.413
camera4 -0.56, -0.031, 0.6, -13.732, -6.389
}
cameradefault 2
attachment {

    (Refer to additional interior config
    files provided in .zip file as
    different loco types have different
    requirements)

}
obsolete-table {
}
username
description ""
trainz-build 1.5
category-class AC
category-region-0 AT
category-era-0 1980s
```

Breakdown:**mesh**

.pm model file

camera

SP3 camera position relative to a.cabfront (0,0,0 = left/right, front/back, up/down)

cameralist

Multiple in-cab camera positions relative to a.cabfront. This section ignored by SP3.

0,0,0,0,0 =left/right, front/back, up/down, yaw, pitch

To determine these variables add *-freeintcam* to the *trainzoptions.txt*. Pan around the interior using arrow keys and mouse. Co-ordinates at bot./left.

cameradefault

The in-cab camera view Trainz defaults to when entering the cab. (Generally should be the same as that of *camera*). This section ignored by SP3.

attachment

Start of attachments section. Specifies additional meshes and types inserted at specified attachment points within the main .pm model.

INTERIOR ATTACHMENT TYPES:**pantograph_lever**

Pantograph lever/switch. For raising and lowering pantographs on electric locos.

horn

Locomotive's Horn

independantbrake_lever

Independent (Loco) brake lever

reverser_lever

Reverser lever (Forward/Neutral/Reverse)

throttle_lever

Throttle / power handle

trainbrake_lever

Train brake lever - self lapping

trainbrakelap_lever

Train brake lever with lap position.

dynamicbrake_lever

For selecting dynamic brake

bplocomain_needle

Main reservoir pressure needle

bploco_equalizer

Equalising reservoir pressure needle

bptrainbrakepipe_needle

Brake pipe pressure needle

bptrainbrakecylinder_needle

Brake cylinder pressure needle

speedo_needle

Speedometer needle

ampmeter_needle

Power meter needle

flow_needle

Flow gauge needle

windows

Textured mesh with low opacity (semi-transparent) to give impression of reflection. This mesh has the same 3D origin point as the main .pm model, therefore does not require an attachment point

wheelslip_light

A warning light mesh that is only visible when the locomotive loses traction. This mesh has the same 3D origin point as the main .pm model, therefore does not require an attachment point

switch0, switch1 etc

Switches

light_switch

Headlight switch

You can also attach miscellaneous meshes to attachment points. They have no current function in Trainz but they look pretty groovy ☺

E.g. a swivel chair

```
swivel_chair {
  kind lever
  mesh chair.pm
  att a.chair1
  limits 0, 8
  angles 6.8, -6.8
}
```

INTERIOR ATTACHMENT VARIABLES:

Kinds: **lever**

Lever's, switches, dials etc

needle

Guage needles (I.e. Speedo, brake pres.)

pullrope

Pullrope horn as in the F7

light

Wheelslip light

mesh

.pm file inserted

att

Attachment point where mesh is inserted. If no attachment point is specified the mesh will be inserted at a.cabfront (the same insertion point as main mesh)

limits

Mathematical boundaries Trainz uses determine the objects function. These values vary as different objects use different mathematical units. Generally use the default values used in the config files provided.

angles

Rotational boundaries in *radians* relative to its attachment point.

notches

The position of notches within the angle boundaries. These are represented as decimal points between and including 0 and 1.

notchheight

The size of the notches specified.

radius

The notch position relative to the attachment point.

mousespeed

This controls the use of the mouse on screen. Use this to adjust the push/pull functioning of levers and dials for example.

opacity

Used for the window mesh to give transparency (and the impression of reflection).

KIND: PANTOGRAPH

File location - \World\custom\pants\

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

Referenced by the *pantograph* tag in a traincar config.txt

Config.txt:

```
kuid <KUID:###:#####>
kind pantograph
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class AL
category-region-0 US
category-era-0 1960s
```

KIND: WATER

File location - world\custom\environment\

Config.txt

```
kuid <KUID:###:#####>
kind water
region Britain
normal Water1
reflection Water1_R
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class EW
category-region-0 UK
category-era-0
```

Breakdown for Water Config.txt:**region**

surveyor region.

normal

name of image for water texture, file should be 128 x 128 x 24bit tga.

reflection

name of image file for reflection, file should be 128 x 128 pixel x 256 color bitmap.

KIND: ENVIRONMENT

File location - world\custom\environment\

Config.txt:

```
kuid <KUID:###:#####>
kind environment
region Britain
normal mediumclouds
storm mediumClouds_Storm
night mediumClouds_Night
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class ES
category-region-0 UK
category-era-0
```

Breakdown for Environment Config.txt:

region – surveyor region.

normal – name of image file for normal sky, file should be 256 x 256 pixel 24bit tga file.

storm – name of image file for stormy sky, file should be 256 x 256 pixel 24bit tga file.

night – name of image file for night sky, file should be 256 x 256 pixel 24bit tga file.

KIND: MAP

File location - world\custom\maps\

The config.txt for maps are automatically generated by Trainz Surveyor.

You can add a soundscript to the config if desired such as the example below. Refer Soundscripts

Config.txt :

```
kind map
kuid <KUID:###:#####>
username Britain
workingscale 0
workingunits 0
water <KUID:-1:8009>
region Britain
soundscript {
    morning {
        ambient 1
        value-range 1, 0.1
        volume 0.3
        sound {
            ctry_day_1.wav
        }
    }
    night {
        ambient 1
        value-range 0, 0.9
        volume 0.3
        sound {
            night_loop.wav
        }
    }
}
```

KIND: GROUNDTEXTURE

File location - world\custom\ground\

Config.txt :

```
kuid <KUID:###:#####>
kind groundtexture
region Britain
rgb 112, 115, 59
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class GL
category-region-0 UK
category-era-0
```

Breakdown for Groundtexture Config.txt:**region**

surveyor region.

rgb

color to be used in minimaps

KIND: SCENERY

File location - World\custom\scenery\

Config.txt :

```
kuid <KUID:###:#####>
kind scenery
region Britain
type Foliage
light 1
nightmode home
night Custom_Object_Nightwindows
autoanimation 1
kuid-table {
    0 <KUID:###:#####>
    1 <KUID:###:#####>
}
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class FS
category-region-0 UK
category-era-0
```

region

Surveyor region.

type

Surveyor type.

light

Sets lighting to be used for object to be ambient or directional. 0 sets ambient lighting and object is light by general light value, 1 sets directional light which is affected by the position of the sun.

nightmode

Optional *home*, *lamp* or *constant*. *Home* switches on night effect at dusk and off sometime during the night. *Lamp* switches the night effect on from dusk to dawn. *Constant* lights are on day and night.

night

name of object to be shown for night effect, stored in subfolder. Refer Chapter 4, SCENERY OBJECTS WITH LIGHTS AT NIGHT

autoanimation

This tag not required if no animation is present.

1 = contains animation

KIND: TRACK – RAILS

File location - World\custom\track\

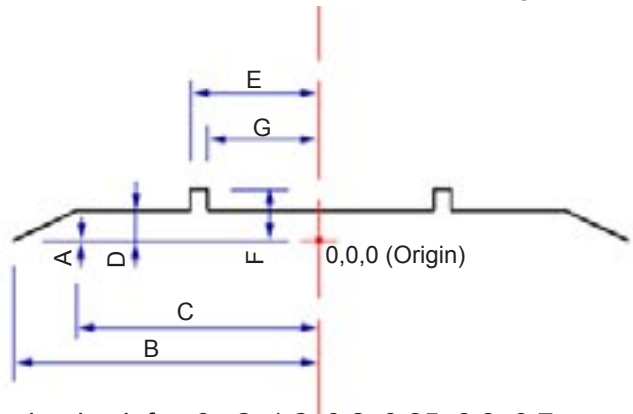
This is used for creating rails.

Config.txt :

```
kuid <KUID:###:#####>
kind track
region Britain
type Rails
rgb 255,200,0
length 4
istrack 1
width 4
chunky_mesh mstand_tex
chunky_info 0, 2, 1.2, 0.2, 0.85, 0.3, 0.7
kuid-table {
    0 <KUID:###:#####>
    1 <KUID:###:#####>
}
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class TR
category-region-0 UK
category-era-0
```

chunky_info

These values (in metres) define the shape of the mesh created for the track. See drawing below:



```
chunky_info 0, 2, 1.2, 0.2, 0.85, 0.3, 0.7
chunky_info A, B, C, D, E, F, G
```

type

Surveyor type.

region

Surveyor region.

rgb

Color used for display in mini map

length

Length of track piece

istrack

Sets whether the track is a rail for trains or not.

1 = This is a rail track

width

Width of track in meters

chunky_mesh

Name of texture to apply to rail

KIND: TRACK – ROAD

File location - world\custom\track\

This is used for creating roads.

config.txt:

```
kuid <KUID:###:####>
kind track
region Australia
length 5
grounded 0.4
istrack 0
width 7.9
bendy 1
isroad 1
carrate 55
uncached_alphas 1
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class SR
category-region-0 AU
category-era-0
```

uncached_alphas

This is used in certain situations to improve alpha sorting. This should only be set to 1 for tracks that use an alpha texture and are always placed flat near the ground (and are not used on bridges or turntables).

Breakdown:**length**

Length of track segment in meters

grounded

Height in meters for the road to be offset from terrain

istrack

0 = This is not rail tracks

width

Width of track mesh in meters.

bendy

Switches how track is bent on corners, set as 1 allows the mesh to be deformed as the spline is bend around corners.

isroad

Specifies track is a road with cars, set to 1 for cars to appear on road.

carrate

Defines traffic density on road (minimum seconds between each car generated).

0 = No traffic.

Number must be greater than 3.

KIND: BRIDGE – BRIDGE

File location - world\custom\track\ or
world\custom\splines\

This kind is used for creating road and rail bridge.

Config.txt :

```
kuid <KUID:###:#####>
kind bridge
type Bridges
region Britain
length 20
bridgetrack <KUID:-1:100395>
trackoffsets -2.5,2.5
height -8
rgb 200,100,0
casts_shadows 1
istrack 1
initiator dark_stone_arch_2t_start
terminator dark_stone_arch_2t_end
endlength 40
kuid-table {
    0 <KUID:###:#####>
    1 <KUID:###:#####>
}
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class TB
category-region-0 UK
category-era-0
```

Breakdown:**type**

Surveyor type – eg. bridge, tunnel or rail.

region

Surveyor region.

length

Length in meters of each bridge piece

bridgetrack

Kuid for the type of rail or road used on bridge.

trackoffsets

Distance in meters the rail/s are attached to the center of the bridge spline. Any number of tracks can be attached to the spline, only splines with the same track offsets can be connected together.

height

Height from the track level to the base of the bridge supports, should be negative for bridges.

rgb

Color used for display in mini map

casts_shadows

Defines whether or not the shadows are cast.

0 = shadows off

1 = shadows on

If shadows are on there needs to be a *bridge_shadow.im* model in a subfolder for the bridge and the initiator and terminator segments (if they are used).

istrack

0 = This is a road bridge

1 = This is a rail bridge

Initiator

Name of model to use at start of bridge, placed in subfolder with same name.

terminator

Name of model to use at end of bridge, placed in subfolder with same name.

endlength

Length in meters of the initiator and terminator models.

KIND: BRIDGE – TUNNEL

File location - world\custom\track\ or
world\custom\splines\

This kind is used for creating road and rail tunnels.

Config.txt :

```
kuid <KUID:###:#####>
kind bridge
type Tunnels
region Australia
length 20
bridgetrack <KUID:###:#####>
trackoffsets -4.5, 4.5
height 8
rgb 180, 180, 180
istrack 1
initiator oz_tunnel_start
terminator oz_tunnel_end
endlength 20
kuid-table {
    0 <KUID:###:#####>
    1 <KUID:###:#####>
}
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class TB
category-region-0 AU
category-era-0
```

rgb

Color used for display in mini map

istrack

0 = This is a road bridge

1 = This is a rail bridge

Initiator

Name of model to use at start of bridge, placed in subfolder with same name.

terminator

Name of model to use at end of bridge, placed in subfolder with same name.

endlength

Length in meters of the initiator and terminator models.

Breakdown:**type**

Surveyor type – eg. bridge, tunnel or rail.

region

Surveyor region.

length

Length in meters of each bridge piece

bridgetrack

Kuid for the type of rail or road used on bridge.

trackoffsets

Distance in meters the rail/s are attached to the center of the bridge spline. Any number of tracks can be attached to the spline, only splines with the same track offsets can be connected together.

height

The height value for tunnels should be positive and greater than the height of the ceiling of the tunnel, but less than the height of the tunnel entrance structure.

KIND: BRIDGE – DOUBLE TRACK

File location - world\custom\track\

This kind can also be configured to create splines that can be used for placing two or more tracks using the *trackoffsets* tag.

Config.txt :

```
kuid <KUID:###:#####>
kind bridge
type Rails
region Australia
length 20
bridgetrack <KUID:-1:100396>
trackoffsets -2.5,2.5
height 0
rgb 255,200,0
istrack 1
kuid-table {
    0 <KUID:###:#####>
    1 <KUID:###:#####>
}
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class TB
category-region-0 UK
category-era-0
```

istrack

0 = This is a road bridge

1 = This is a rail bridge

Breakdown:**type**

Surveyor type – eg. bridge, tunnel or rail.

region

Surveyor region.

length

Length in meters of each bridge piece

bridgetrack

Kuid for the type of rail used on bridge.

trackoffsets

Distance in meters the rail/s are attached to the center of the bridge spline. Any number of tracks can be attached to the spline, only splines with the same track offsets can be connected together.

height

0 is used for double tracks.

rgb

Color used for display in mini map

KIND: MOSPEEDBOARD

File location - World\custom\trackside\

This is a speed limit sign.

Config.txt:

```
kuid <KUID:###:#####>
kind mospeedboard
trackside -2.5
speedlimit 5.56
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class WS
category-region-0 UK
category-era-0
```

Breakdown:**trackside**

This is a value that is the distance in meters the object is placed relative to the center of the track. Negative values will put the object on the left side of the track, and positive values will appear on the right.

Speedlimit

This value is the maximum speed allowed in meters per seconds.

To convert miles per hour to meters per second multiply by a conversion factor of 0.447.

For example 10mph is 4.47 m/s.

To convert kilometers per hour to meters per second multiply by a conversion factor of 0.278.

For example 10kph is 2.78m/s.

KIND: MOSIGNAL

File location - world\custom\trackside\

Refer to config files attached within ZIP file.

Config.txt

```
kuid <KUID:####:#####>
kind mosignal
light 1
trackside -2.7
function TrackSignal
region Britain
name "02"
fontsize 0.07
fontcolor 255,255,255
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class WA
category-region-0 UK
category-era-0
```

Breakdown:**region**

Surveyor region.

light

Sets lighting to be used for object to be ambient or directional. 0 sets ambient lighting and object is light by general light value, 1 sets directional light which is affected by the position of the sun.

trackside

Origin offset from track center expressed in metres.

function

Must be set to TrackSignal

name

Default number board entry

fontsize 0.07

Font size for number boards

fontcolor 255,255,255

R.G.B. font colour for number boards

SIGNALS

The next section of the config explains which aspects the signal is capable of displaying, and also which light points are activated when each state is displayed. The supported aspects are configured by reference number as follows...

0	STOP
1	STOP THEN PROCEED
2	CAUTION AND LEFT DIVERGE
3	CAUTION AND RIGHT DIVERGE
4	CAUTION
5	PROCEED AND LEFT DIVERGE
6	PROCEED AND RIGHT DIVERGE
7	ADVANCED CAUTION
8	PROCEED

The following two aspects are only used for scenarios....

9	SLOW
10	MEDIUM SPEED

The aspect section of the config.txt is arranged as follows.....

```
signals
{
0 {
    light 7
}
2 {
    light 6,0,1,2,3,4
}
4 {
    light 6
}
}
```

(dont forget the 'space' between the number and bracket)

Looking at the example above, under the heading 'signals' we see the states the signal is capable of displaying in the left column. From this extract we can see that this signal is capable of displaying aspects 0, 2 & 4.

When aspect 0 (stop) is displayed, light point 7 is activated.

When aspect 2 (caution left) is displayed, light points 6,0,1,2,3,4 are activated

When aspect 4 (caution) is displayed, light point 6 is activated.

LIGHTS

Each light point needs to have a corona associated with it. Coronas are stored in each signal object's directory alongside its textures. Examples have been packaged within the zip file this document was located.

```
lights
{
0 {
corona corona_white.tga
}
1 {
corona corona_white.tga
}
2 {
corona corona_white.tga
}
3 {
corona corona_white.tga
}
4 {
corona corona_white.tga
}
5 {
corona corona_green.tga
}
6 {
corona corona_yellow.tga
}
7 {
corona corona_red.tga
}
8 {
corona corona_white.tga
}
9 {
corona corona_white.tga
}
}
```

(dont forget the 'space' between the number and bracket)

Looking at the example above, under the heading 'lights' we see the light points that are attached to the 3D model. This model has 10 of them, they are named a.light0 to a.light9.

From the signals section we know that when aspect 0 (stop) is displayed, light point 7 is activated.

Looking at the extract left...

When light point 7 is activated, it displays corona red.

When aspect 2 (caution left) is displayed, light points 6,0,1,2,3,4 are activated

When light point 6 is activated, it displays corona_yellow.

When light points 0 – 4 are activated, each displays corona_white.

Signal placement is very important for correct operation of the system. There are some rules to consider while signalling your map which if not observed may cause problems with getting the correct aspects to display.

There are also various departures from prototypical operation which should be considered when designing new signalling, and also when installing it into a map.

Please refer to the SP3 Signalling document for further details. Default location: \Trainz\Docs

KIND: MOJUNCTION

File location - World\custom\trackside\

This is used for creating junction control levers.

Config.txt :

```
kuid <KUID:####:#####>
kind mojunction
region Australia
trackside 2
light 1
mode0 lever1
mode1 lever2
soundscript
{
    toggle
    {
        trigger toggle
        distance 5, 100
        nostartdelay 1
        repeat-delay 1
        sound
        {
            points.wav
        }
    }
}
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class WX
category-region-0 AU
category-era-0
```

Breakdown:**region**

Surveyor region.

trackside

This is a value that is the distance in meters the object is placed relative to the center of the track. Negative values will put the object on the left side of the track, and positive values will appear on the right.

light

Sets lighting to be used for object to be ambient or directional. 0 sets ambient lighting and object is light by general light value, 1 sets directional light which is affected by the position of the sun.

mode0

The model name, located in subfolder, of the initial junction. Example refers to a file lever1\lever1.im

mode1

The model name, located in subfolder, of the switched junction. Example refers to a file lever2\lever2.im

soundscript

Soundscripts for mojunction objects can be activated with toggle trigger message as in example. Refer to Soundscripts section (page 46)

KIND: MOTURNTABLE

File location - world\custom\scenery\

This is a turntable object.

Config.txt :

```
kuid <KUID:###:#####>
kind turntable
region Australia
type Trackside
light 1
mode0 oz_turntable2
mode1 oz_turntable2_spinner
angle 0,165,180,345
track <KUID:-1:100966>
snapmode 2
dighole 3,3
kuid-table {
    0 <KUID:###:#####>
    1 <KUID:###:#####>
}
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class TR
category-region-0 AU
category-era-0
```

Breakdown:**type**

Surveyor type.

region

Surveyor region.

light

Sets lighting to be used for object to be ambient or directional. 0 sets ambient lighting and object is light by general light value, 1 sets directional light which is affected by the position of the sun.

mode0

The name of the main turntable object, model located in subfolder. Example refers to oz_turntable2\ oz_turntable2.im

mode1

Name of the rotating turntable part, model located in subfolder. Example refers to oz_turntable2_spinner \ oz_turntable2_spinner.im

angle

Specifies the angles at which the turntable stops.

track

Kuid for track to be attached to turntable

snapmode

Specifies the alignment of the turntable to the surveyor grid. 1 = origin snaps to grid (use for removing even dighole values), 2 = origin snaps to the center of a grid square (use for odd dighole values)

dighole

Specifies the number of grid segments (length, width) to be removed from the surveyor grid to accommodate the turntable pit.

KIND: MOCROSSING

File location - world\custom\scenery\

This is a level crossing.

Config.txt :

```
kuid <KUID:###:#####>
kind mocrossing
region Australia
type Trackside
track <KUID:-1:100396>
road <KUID:-1:100409>
mode0 level_crossing_1track
kuid-table {
    0 <KUID:###:#####>
    1 <KUID:###:#####>
}
obsolete-table {
}
username
description " "
trainz-build 1.5
category-class TR
category-region-0 AU
category-era-0
```

Breakdown:**type**

Surveyor type.

region

Surveyor region.

track

Kuid for track to be used on crossing.

road

Kuid for road to be used on crossing.

mode0

Name of the model for the crossing object, stored in subfolder of same name. Example refers to level_crossing_1track\ level_crossing_1track.im

KIND: ACTIVITY

File location - world\custom\scenarios\

Config.txt :

```

kind activity
kuid <KUID:-14:160>
username Highland Valley (DCC)

scriptlibrary SP3S1DCC
scriptclass SP3S1DCC

driver-settings
{
  autopilotmode 0
  startingtime 0.4
  timerate 1
  deraillevel 0
  showhelp 0
  controlmethod 0
  weather 3
  changeability 1
}
kuid-table
{
  highland_valley      <KUID:-12:132>

  f7_sfred              <KUID:-1:1>
  atsf_chair            <KUID:-1:100160>
  atsf_pullman_pine     <KUID:-1:100163>
  atsf_baggage          <KUID:-1:100159>
  cflow_fert            <KUID:-1:100012>
  prr_fm_tuscan         <KUID:-1:100017>
  40ft_boxcar           <KUID:-1:100085>
  pdhc_babyruth         <KUID:-1:100066>
  4bhopper_il           <KUID:-1:100929>
  50ft_boxcar           <KUID:-1:100086>
  gatx_pennsalt         <KUID:-1:100092>
  60ft_boxcar           <KUID:-1:100087>
  sd40_2_santafe        <KUID:-1:100871>
  4bhopper_il_coal_full <KUID:-1:101224>
  foundry_car           <KUID:-1:101220>
}
description "Take contol of the morning
passenger service to Highland Valley
stopping at all stations and return to
Greenwood. Bad weather is forecast so
drive with care.

Service : Highland Valley Passenger
Train No. : 7528
Consist : F7A + 5 cars
Weight in Tow : 300t
Total Length : 490"
```

Breakdown:**username**

Name of scenario displayed in trainz

scriptlibrary

The name of the .gsl (compiled script) library on disk, without the ".gsl" extension.

scriptclass

The name of the scenario class within the script file.

driver-settings{}

Specify the settings of this scenario, similar to Driver's 'settings' screen. Allows you to set the weather, control method (0 – dcc, 1 – cabin controlled) etc.

autopilotmode	0=off 1=on
startingtime	0..1 (0.5=midday)
timerate	1=real-time
deraillevel	0=none 1=arcadE 2=realistic
showhelp	0=off 1=on
controlmethod	0=dcc 1=cabin
weather	0=clear 1=cloudy 2=drizzle 3=rain 4=stormy 5=light snow 6=medium snow 7=heavy snow
changeability	0=none 1=periodic 2=extreme

kuid-table{}

A list of named assets used in the scenario. Scripts refer to assets (eg trains) by the names in this table.

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.

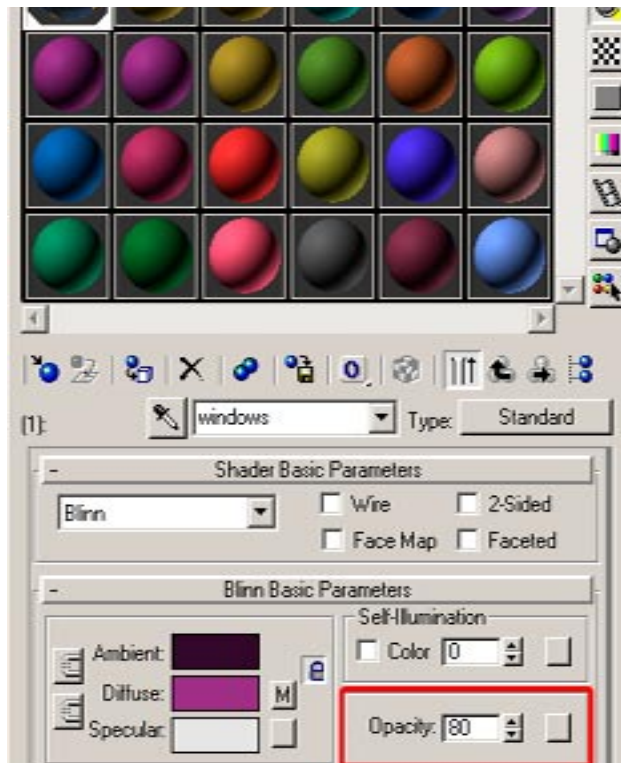
a.outsideview* (0, 1, etc..)

- 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.

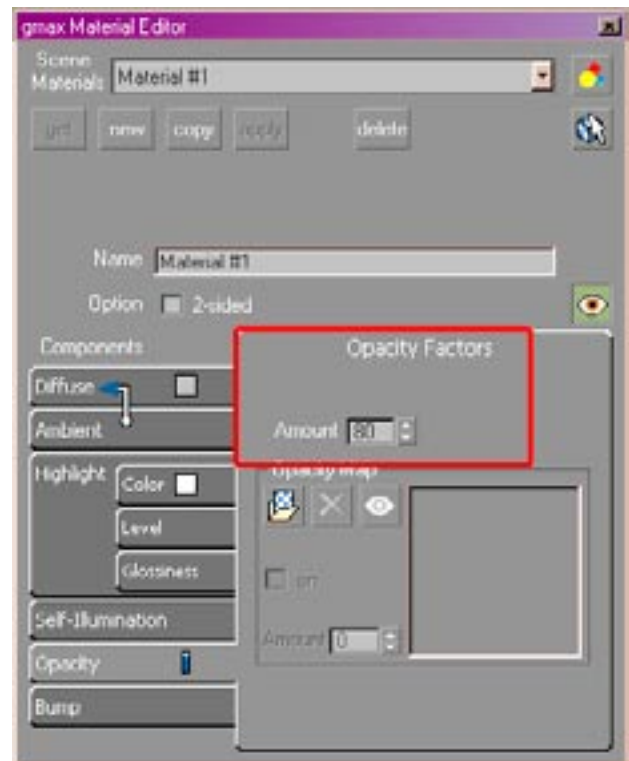


3D Studio MAX

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'.

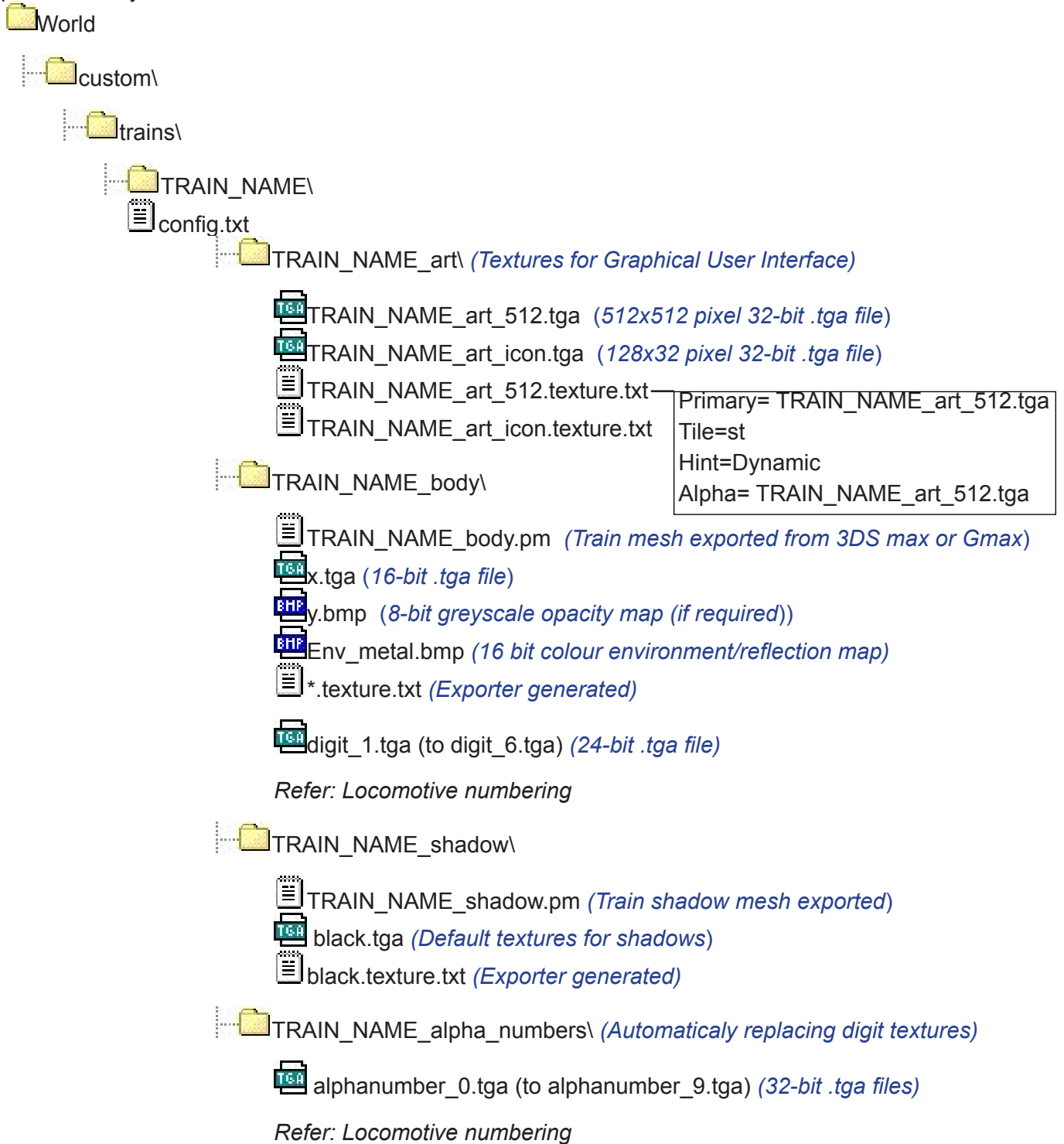
If *two or more* font type used:

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:

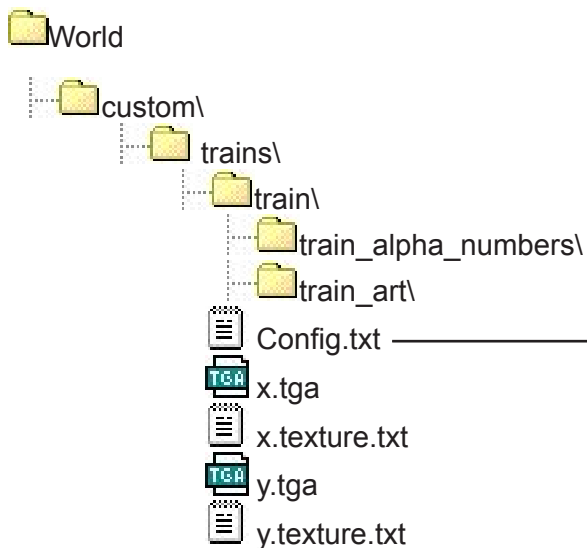
Typical directory structure for custom Trains should be:



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.

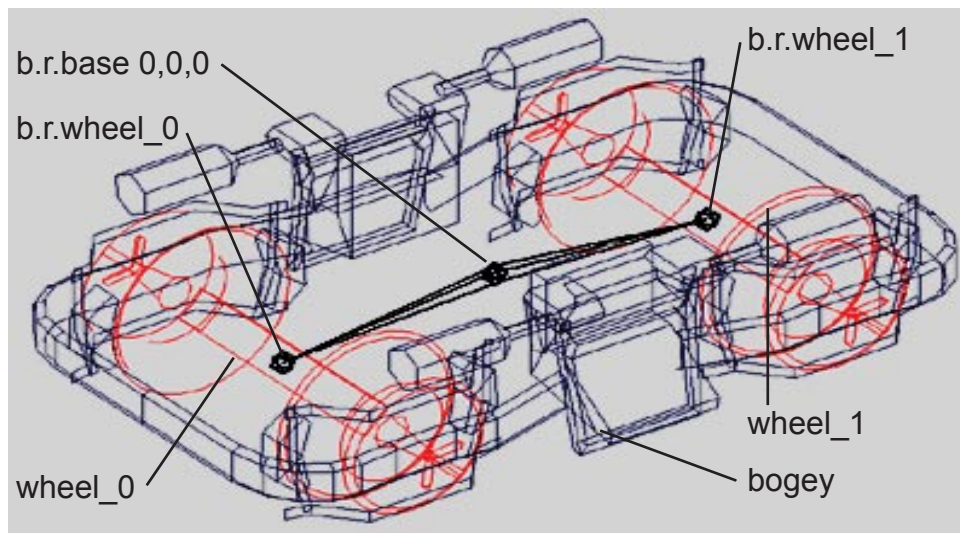
The absolute centre of bogeys should be located at World origin point (0,0,0)

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.

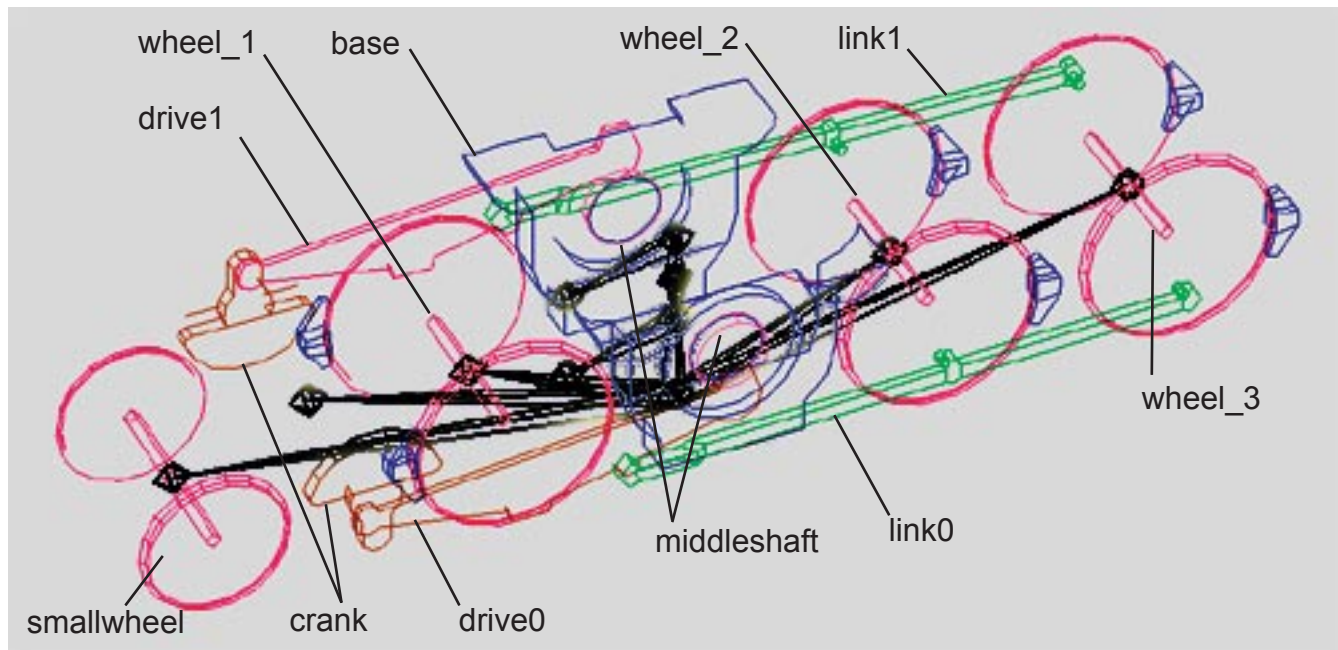
Diffuse Maps: Generally a single 128x128 16-bit .TGA file is sufficient to texture a bogey. Additional maps (e.g. for springs) are also used.

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.

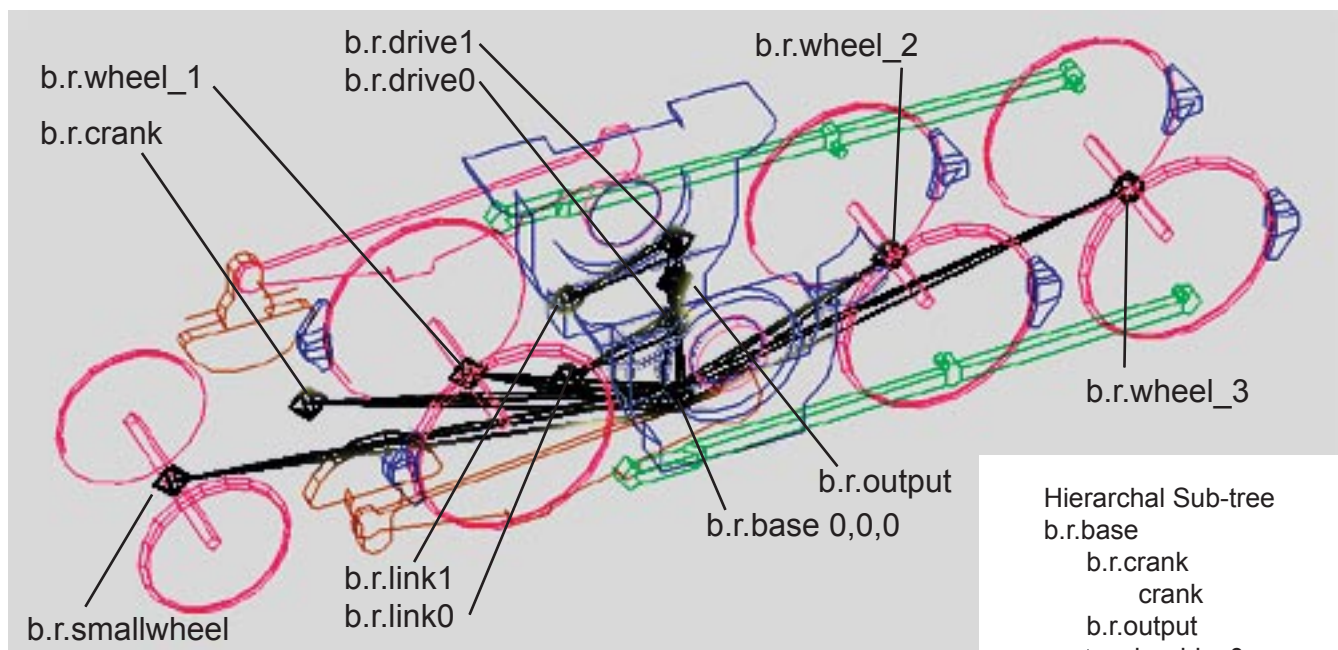
Exporting Models:

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

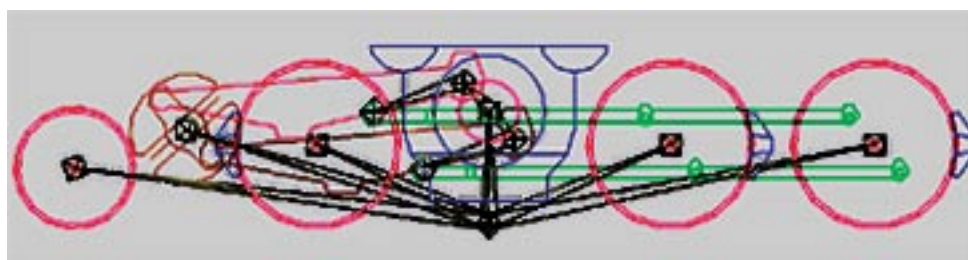


Hierarchal Sub-tree

```

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° over 120 frames:
Distance traveled: 4.2m x 2 revs = 8.4m

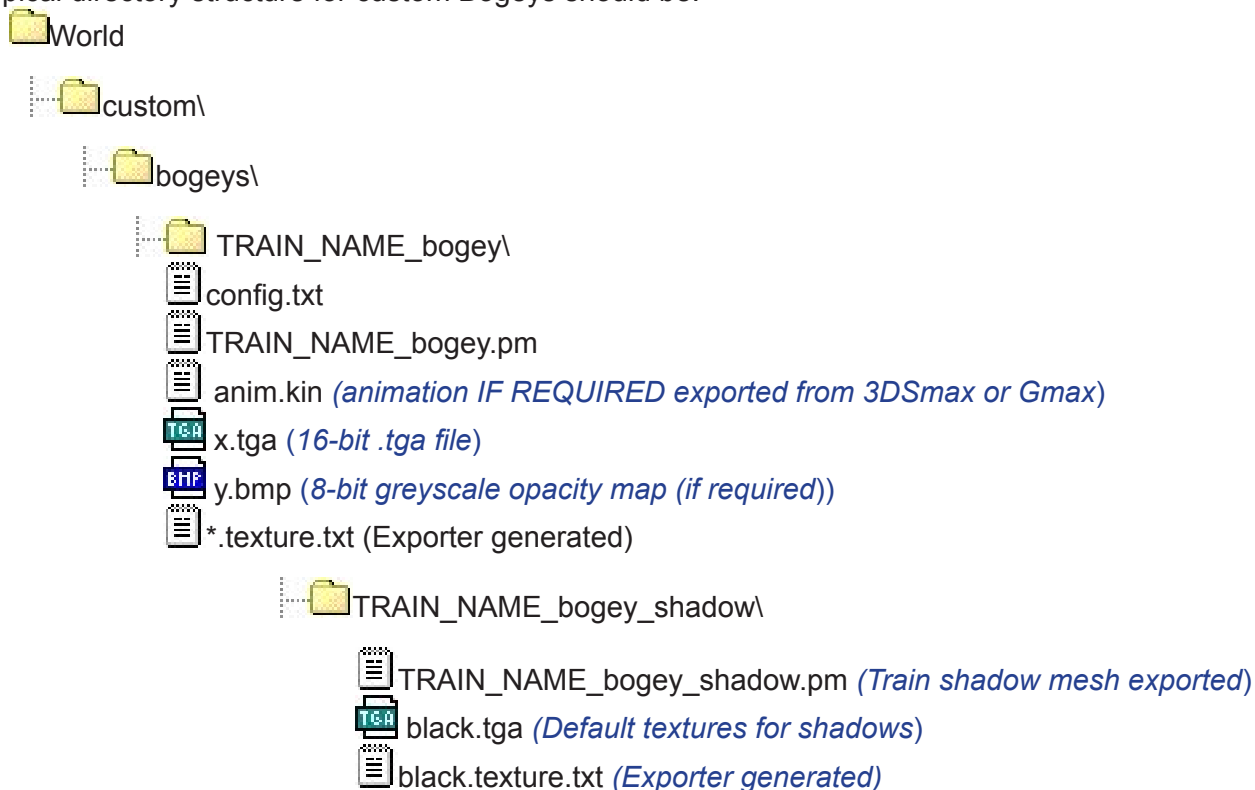
smallwheel circumference = 2.8m

- Animated to turn 1080° 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

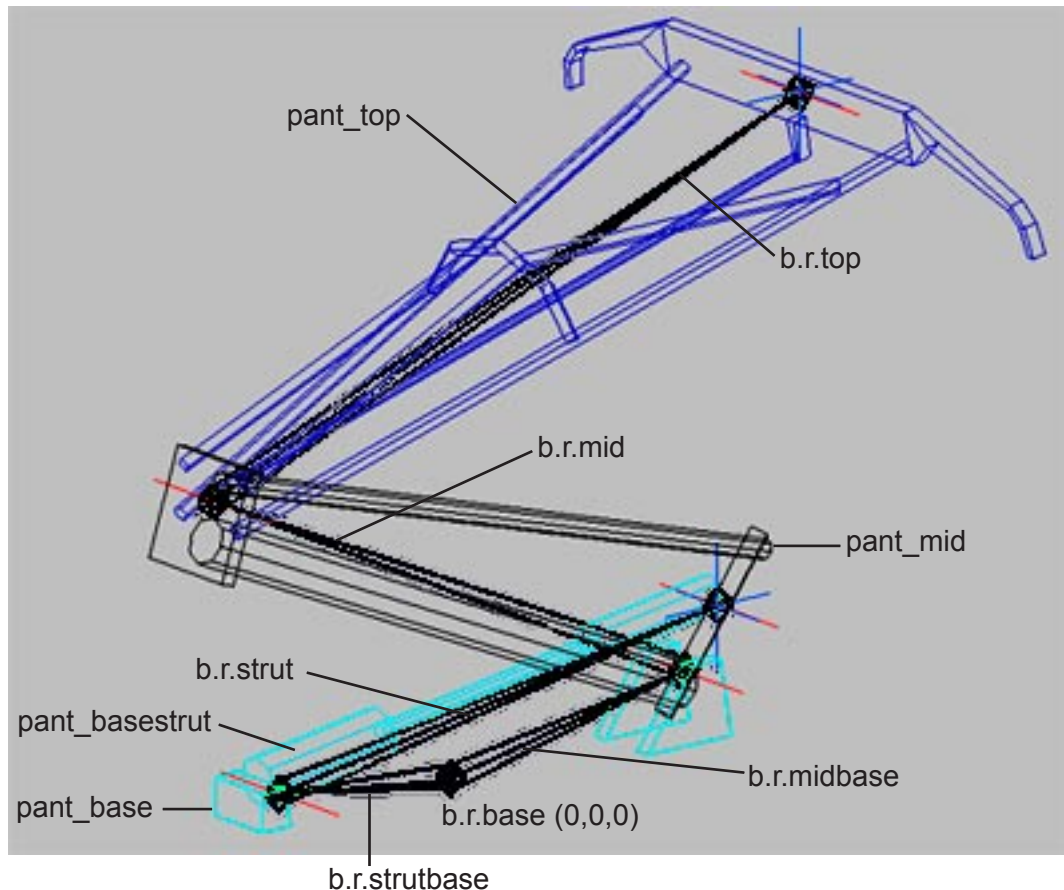
(Pants)

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

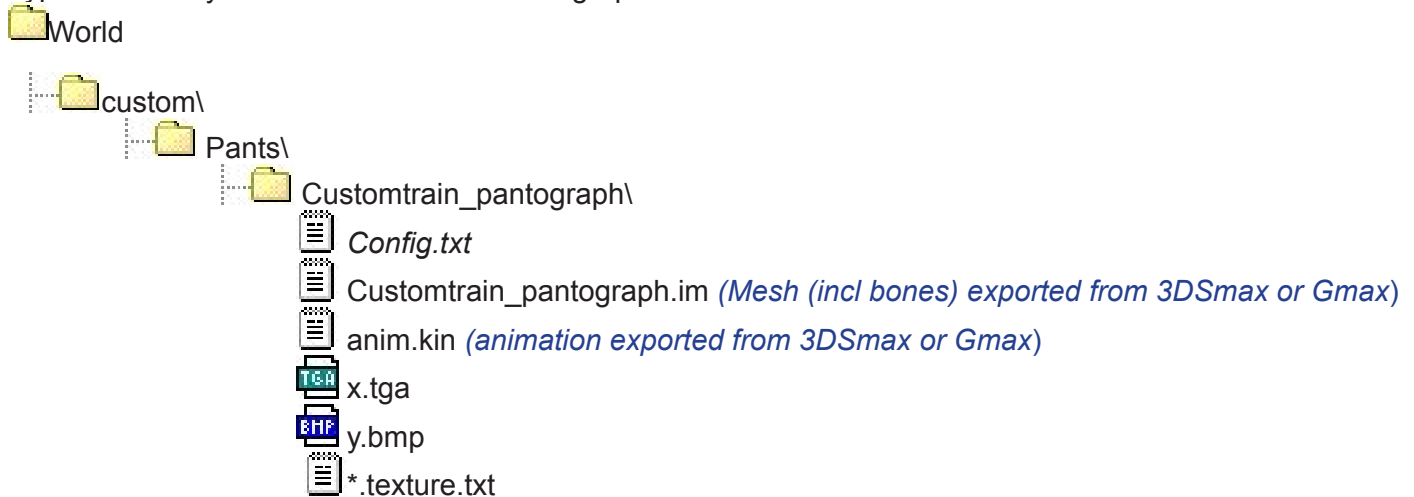
```

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:

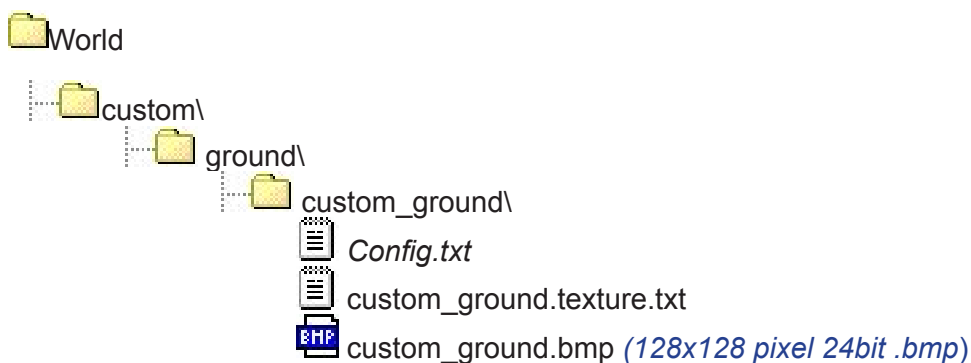
Typical directory structure for custom Pantographs should be:



GROUND

Download Source files from the Trainz Website

Typical directory structure for a custom ground texture should be:

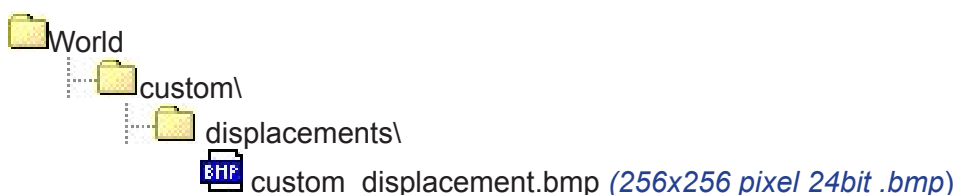


DISPLACEMENTS

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

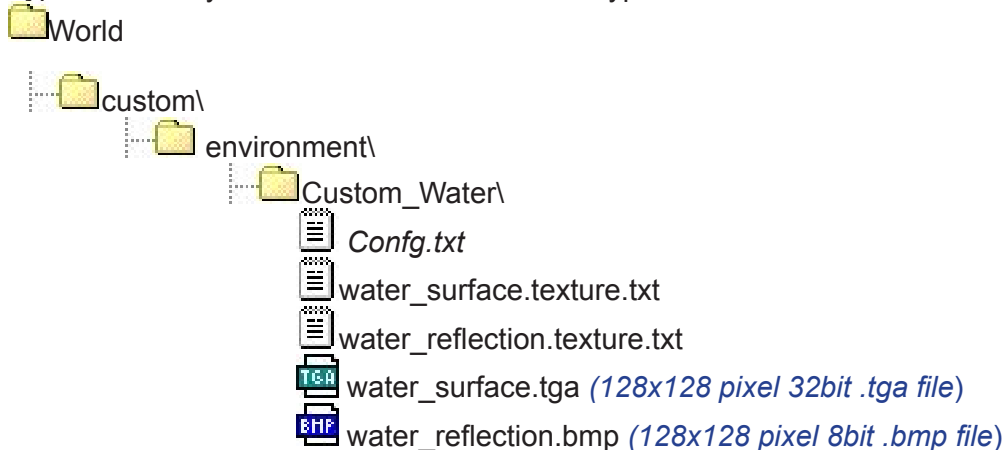
The Environment folder accommodates the use of different types of sky and water in Trainz™.

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:

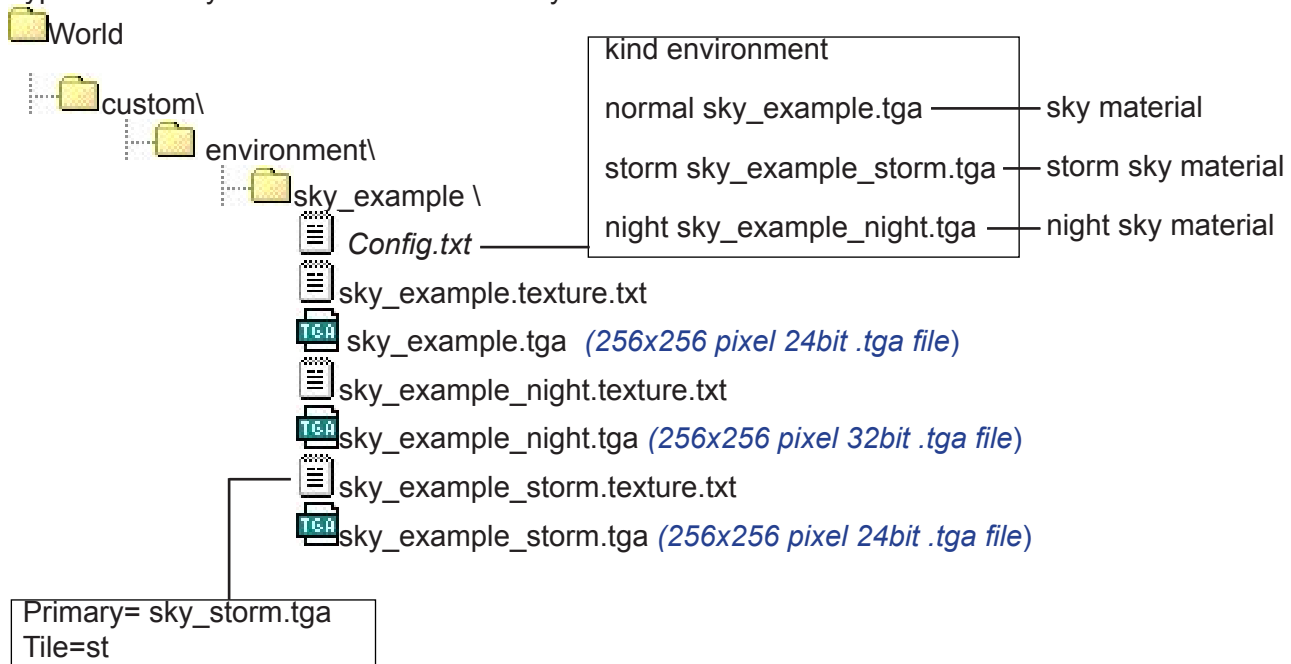


SKY

Download Source files from the Trainz Website

Sky is generated from three source images.

Typical directory structure for a custom sky should be:

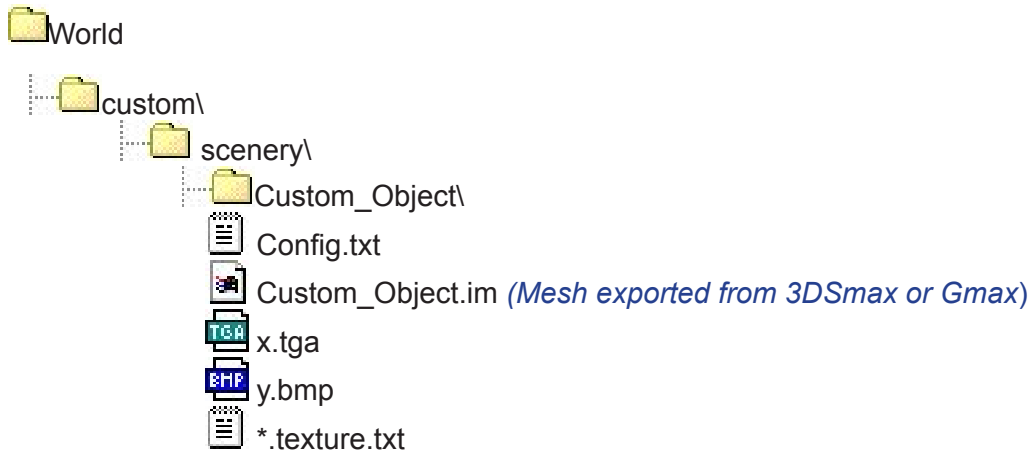


SCENERY

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.

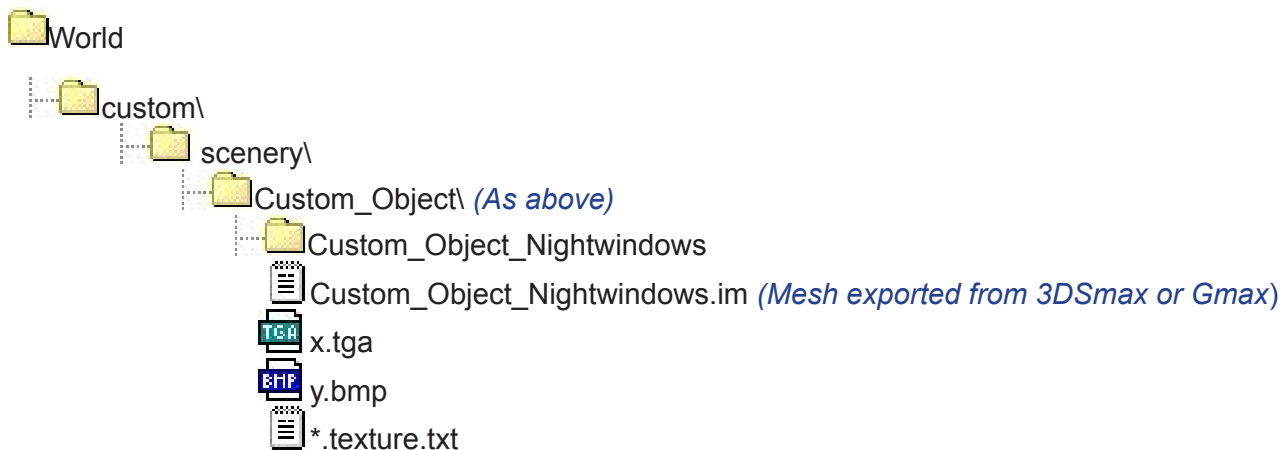
Typical directory structure for a custom scenery object should be:



SCENERY OBJECTS WITH LIGHTS AT NIGHT

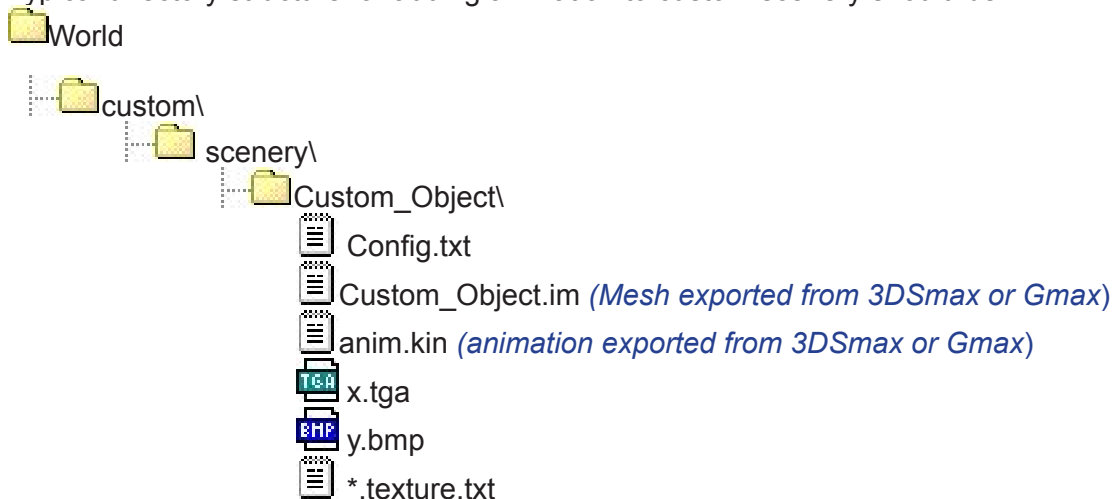
Objects just as buildings or signs can be made to appear to be have lights on at night. A model that contains only the lit areas of the object can be exported into a subdirectory.

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



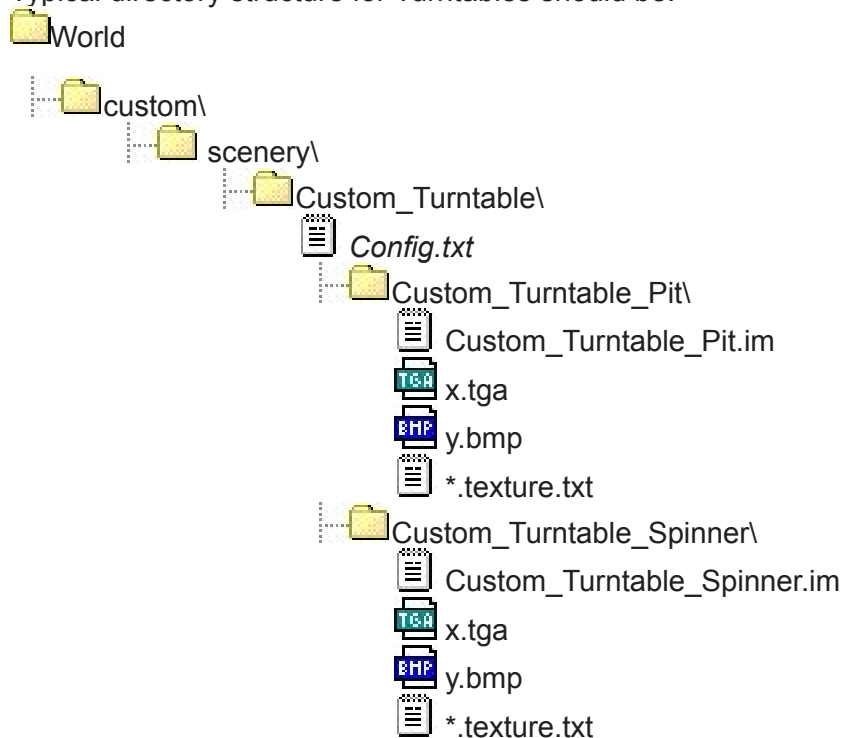
SCENERY OBJECTS WITH ANIMATION

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

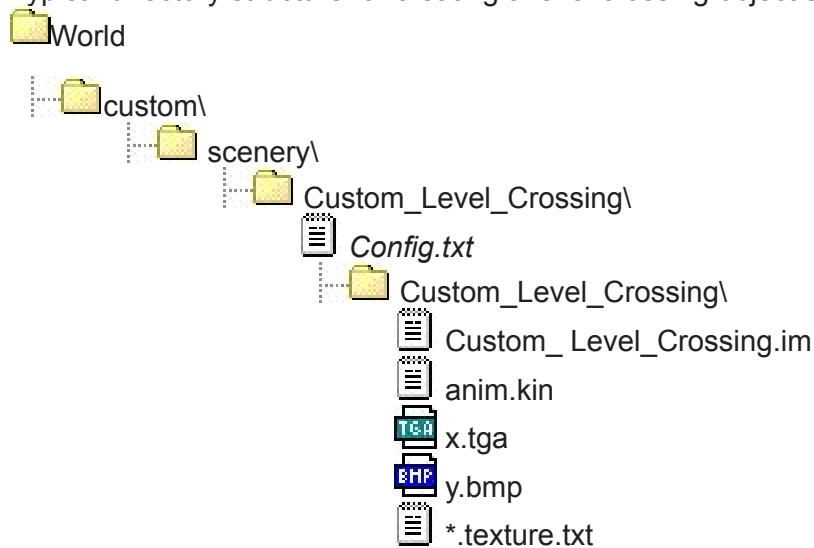


SPECIAL SCENERY OBJECTS - TURNTABLES

Typical directory structure for Turntables should be:

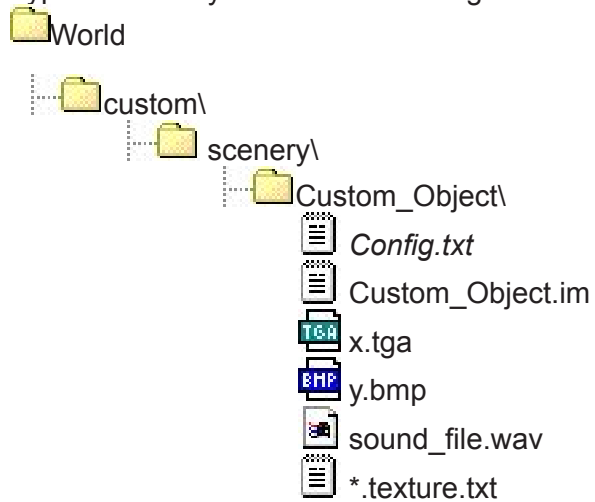
**SPECIAL SCENERY OBJECTS – LEVEL CROSSINGS**

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:

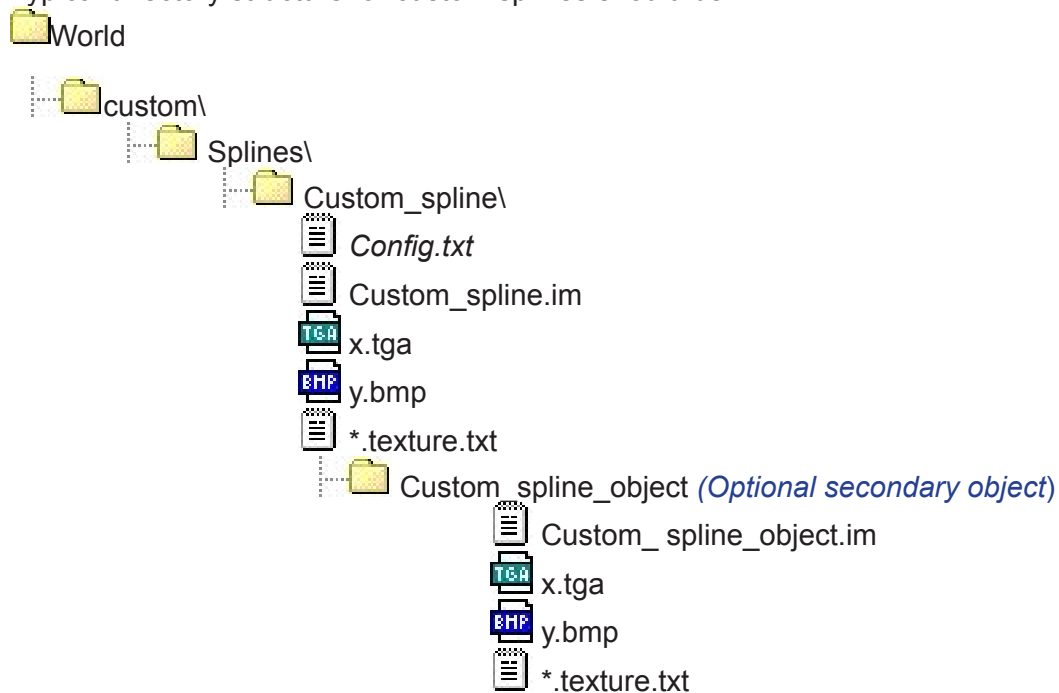


SPLINES

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.

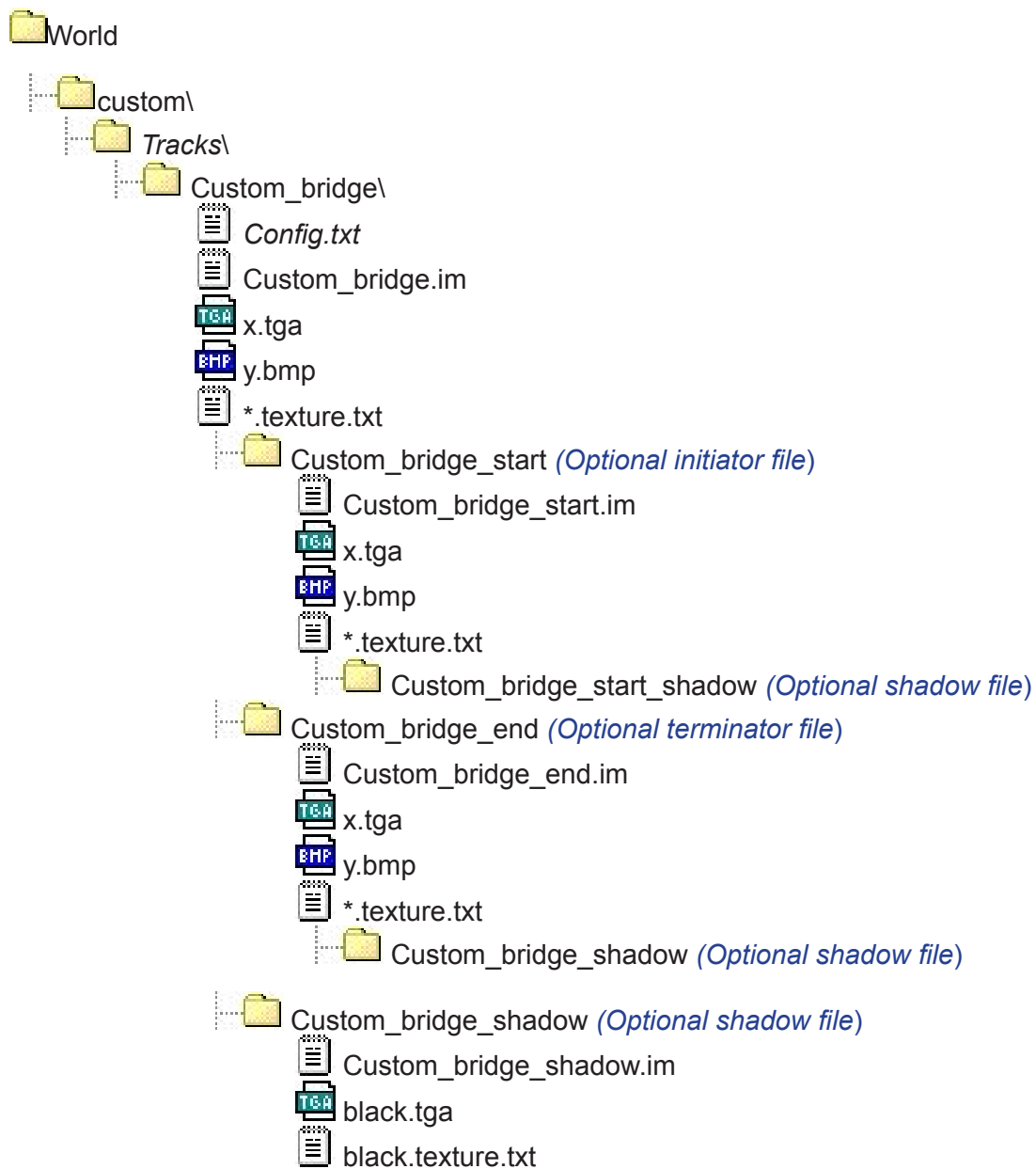
Rails

Typical directory structure for custom track rails should be:



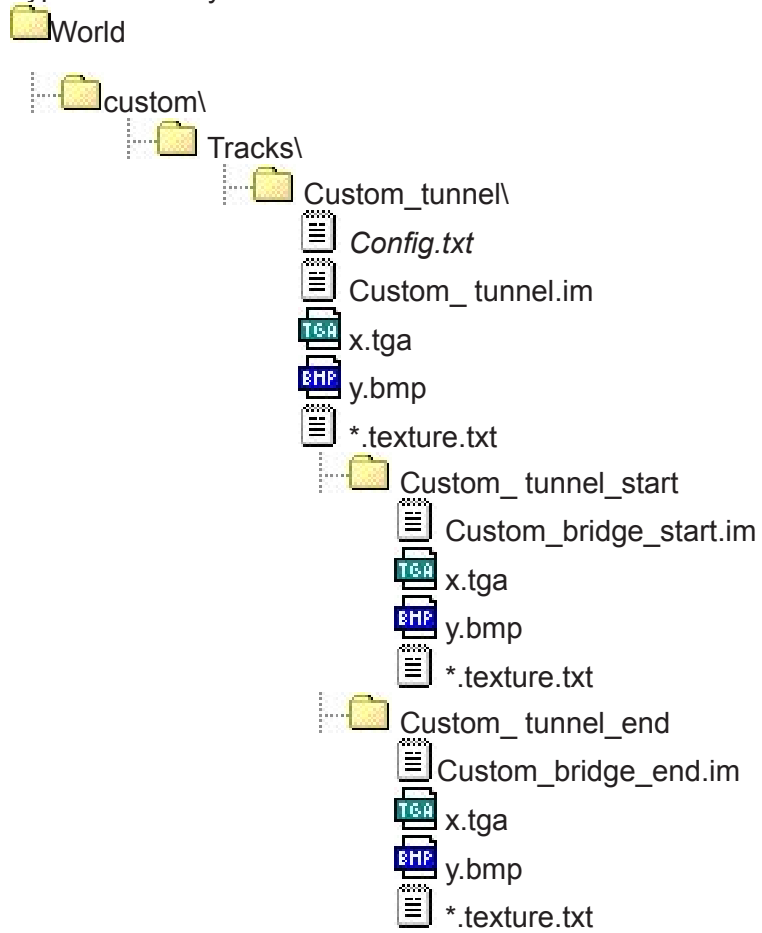
BRIDGES

Typical directory structure for custom bridges should be:



TUNNELS

Typical directory structure for custom tunnel should be:

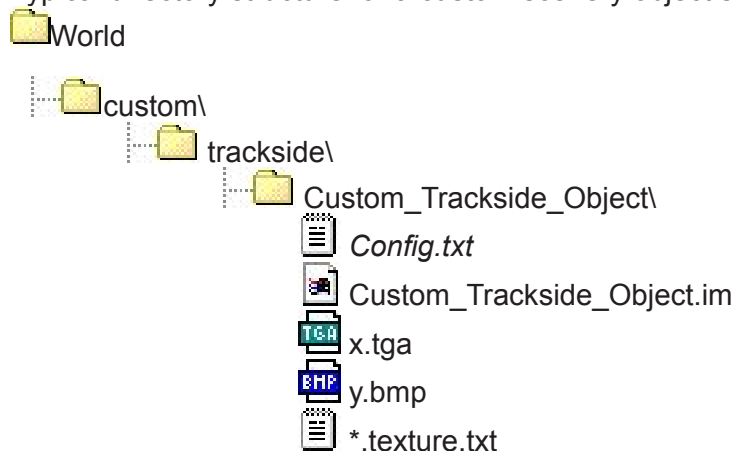


TRACKSIDE

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:



CHAPTER 5 - PARTICLE EFFECTS AND SCRIPTS

SMOKE EFFECTS

Introduction

Trainz version 1.3 (Service Pack 3) gives you the ability to add customizable smoke, steam, vapor and similar effects to your custom trains and scenery objects. For simplicity, this document will refer to this set of effects as simply smoke effects.

It is assumed the reader is already familiar with creating and exporting models from either 3D Studio Max or GMax.

Method

Smoke effects are added to custom trains and scenery objects in two steps:

1. *Add attachment points to the original model.*
2. *Add smoke tags to the object's config.txt file.*

Adding Attachment Points

Attachment points are added to the original model using 3D Studio Max or GMax wherever a smoke effect is desired. See figures 1 and 2 below to locate the Insert Point tool. After a point is inserted, it must be given a name with a prefix of 'a.' to identify it as an attachment point, e.g. a.smoke, a.steam, a.safety, a.mist, etc. The attachment point should also be rotated so that its Y axis is pointing in the direction that smoke particles will be emitted. (Ensure Axis Tripod is checked to see the point's orientation.) When finished, save and export the model as per normal.



3DS Max insert point



Gmax insert point

ADDING SMOKE TAGS

Smoke blocks are added to an object's config.txt file to describe each smoke effect that will be created on the object. Smoke blocks are named smoke# (where # is a number) and are sequentially numbered starting at 0. See Example 2 for an example.

Smoke blocks have two sections: main and sequence properties. Main properties describe the attributes that do not change based on the mode's key. Sequence properties describe a set of one or more phases/periods in the smoke emission sequence.

A smoke block has the following format:

```
smoke#
{
  mode          time | speed | anim | timeofday
  attachment    <name of attachment point>
  color         <red>, <green>, <blue>,
               <opacity>
  accel        <x>, <y>, <z>
  loop         <n>

  start        <n> [, <n>] ...
  period       <n> [, <n>] ...
  rate         <n> [, <n>] ...
  velocity     <n> [, <n>] ...
  lifetime     <n> [, <n>] ...
  minsize     <n> [, <n>] ...
  maxsize     <n> [, <n>] ...
}
```

Notation:

- # Is a number, starting with 0
- [] Means optional,
- ... Indicates a variable number of parameters,
- | Means or.

Breakdown:

<name of attachment point>

Is the name of an attachment point in the model. eg a.smoke, a.steam, a.chimney etc

<red>, <green>, <blue>

Are numbers from 0 to 255 describing the intensity of that color component.

<opacity>

Is a number from 0 to 255 describing the effect's initial opacity / transparency.

<x>, <y>, <z>

Are vector components pointing in the direction of the sum of all forces affecting this smoke effect. Essentially, <z> describes gravity, and <x>, <y> describe the force of wind.

<n>

Is a decimal number.

MAIN PROPERTIES:

mode

Describes the mode or type of this smoke effect. This affects how start and period are interpreted. Default is time. In all modes, period can be set to -1 (default) to imply the phase is active until the next phase begins.

If set to time, start is a set of time values in seconds after the creation of this effect's parent object when this phase of the effect will start. period is the duration of time this effect will remain active. Scenery objects currently only support time mode.

If set to speed, start is a speed in meters per second (m/s) and period is not used. (Note: 1 m/s = 3.6 km/hr.) All other sequence attributes (rate, velocity, lifetime, minsize, maxsize) are interpolated so there are smooth transitions between phases. See smoke3 in Example 2 for an example.

If set to anim, start is a value from 0.0 to 1.0 which describes the start time into the object's animation cycle. period is a value from 0.0 to 1.0 that describes the duration over which the effect is active. start + period must not exceed 1.0.

If set to timeofday, start is a value from 0.0 to 1.0 which describes the time of day when this effect will start. Values range as follow:

0 - midnight, 0.25 - 6am, 0.5 - midday, 0.75 6pm, 1.0 - midnight.

color

The color of the smoke effect. eg '150,150,150,255' for dark smoke; '255,255,255,150' for steam; '150,150,255,255' for water. Default is '255,255,255,255'.

accel

Acceleration. A vector pointing in the direction of the sum of all forces affecting this smoke effect. Essentially, <z> describes gravity, and <x>, <y> describe the force of wind. Default is 0,0,0.

loop

Time in seconds to loop the smoke sequence. Only valid if mode is set to time.

SEQUENCE PROPERTIES:

The following properties can be set to a single value or a set of values for multiple phases of the smoke effect. Please note that phases must not overlap as only one phase can be active at any one time. If a property has a set of values, it must be the same length as **start**. If a single value is given then it will be used for all phases of the effect. See Example 1 for an example of using multiple phases.

start, period

See mode.

rate

The rate of emission in particles per second for modes time, speed, and timeofday, or the number of particles to emit over the animation period for anim mode. Default is 4.

velocity

The initial speed of emitted smoke particles. Default is 1.

lifetime

Time in seconds that smoke particles exist for. Default is 3.

minsize

Start size of smoke particles. Default is 0.

maxsize

End size of smoke particles. Default is 3.

In general, it is better to use a low emission rate with large particles (ie min/max size) than using a high emission rate with small particles to reduce the impact on frame rate. Smoke effects can be quite stunning but are best used in moderation.

Try experimenting with the different values to get a feel of how they affect the smoke effects. Many different types of effects other than smoke are possible with only a little imagination, e.g. waterfalls, mist, toxic green clouds, fire by using a few effects at the same position to simulate the smoke and flames etc.

EXAMPLE 1

- SMOKE FROM A FACTORY'S CHIMNEY

Using a model of a factory with a chimney, an attachment point called 'a.smoke' is placed at the top of the chimney with it's Y axis pointing up. The factory is then exported as an indexed mesh (.im file type) to the Trainz\world\custom\scenery\factory folder and the model's art assets are copied to the same location. The following config.txt file will cause smoke to come out of the factory's chimney between 6am and midday and 3pm and 6pm. Please note the given KUID is invalid and should not be used in your own custom context.

Config.txt

```
kuid <KUID:###:#####>
region Custom
kind scenery
type Industrial
light 1

smoke0
{
  attachment a.smoke
  mode timeofday
  color 150,150,150,250
  accel 1,0.3,0

  start 0.25, 0.5
  period 0.25, 0.125
  rate 8
  velocity 3
  lifetime 5
  minsize 0.5
  maxsize 2
}
```

EXAMPLE 2

- STEAM TRAIN

An animated steam train model that requires four smoke points may be set up as follow:

- Dark smoke from the main chimney stack that is dependant on the trains velocity (a.smoke, Y pointing up),
- A constant steam trail from a small safety pipe on top (a.steam.safety, Y pointing up),
- 2 steam trails on each side of the train that alternately expel steam keyed to the animation of the trains wheels (a.steam.l, a.steam.r, Y pointing outwards).

The model is exported as a progressive mesh (.pm file type) to 'Trainz\world\custom\trains\steam_train\steam_train_body' folder and the model's art assets are copied to the same location. Please see the custom content creation guide for more information on creating your own custom trains. The following config.txt file in the parent folder will generate the desired smoke effects. Please note the given KUID is also invalid and should not be used in your own context.

For example purposes, the settings of an F7 train have been used.

Config.txt

```
kuid <KUID:###:#####>
kind traincar
bogey 0
engine 1
name Steam Train
mass 100000

enginespec <KUID:###:#####>
enginesound <KUID:###:#####>
hornsound <KUID:###:#####>
interior <KUID:###:#####>

smoke0
{
    attachment a.steam.l
    mode anim
    color 255,255,255,150

    start 0
```

```
    period 0.4
    rate 2
    velocity 1
    lifetime 2
    minsize 0.05
    maxsize 1
}

smoke1
{
    attachment a.steam.r
    mode anim
    color 255,255,255,150

    start 0.5
    period 0.4
    rate 2
    velocity 1
    lifetime 2
    minsize 0.05
    maxsize 1
}

smoke2
{
    attachment a.steam.safety
    mode time
    color 255,255,255,150

    rate 2
    velocity 1
    lifetime 2
    minsize 0.05
    maxsize 1
}

smoke3
{
    attachment a.smoke0
    mode speed
    color 100,100,100,200

    start 0,10,20,30
    rate 3,5,7,9
    velocity 3,4,5,5
    lifetime 4,3,2.5,2
    minsize 0.3
    maxsize 2
}
```

SOUND SCRIPTS

Soundscripts give ambient or directional sounds to objects. They cannot be used on track, bridge or spline objects. Wav files should be located within the same directory as the config.txt file. Examples as follows:

MOJUNCTION

Config.txt

```
kuid <KUID:####:#####>
kind mojunction
region Australia
trackside 2
light 1
mode0 lever1
mode1 lever2
soundscript
{
    toggle{
        trigger toggle
        distance 5, 100
        nostartdelay 1
        repeat-delay 1
        sound
        {
            points.wav
        }
    }
}
etc.
```

PEOPLE CROWD

Config.txt :

```
kind scenery
region Australia
kuid <KUID:####:#####>
type People

soundscript
{
    daysingle {
        repeat-delay 0
        distance 3,150
        sound
        {
            crowd_1.wav
        }
    }
}
etc.
```

MAP

Config.txt

```
kind map
kuid <KUID:####:#####>
soundscript
{
    morning {
        ambient 1
        value-range 1, 0.1
        volume 0.3
        sound {
            ctry_day_1.wav
        }
    }
    night {
        ambient 1
        value-range 0, 0.9
        volume 0.3
        sound {
            night_loop.wav
        }
    }
}
username Britain
workingscale 0
workingunits 0
water <KUID:-1:8009>
region Britain
etc.
```

THUNDERBOX

Config.txt

```
kuid <KUID:####:#####>
region Australia
light 1
kind scenery
type Residential

soundscript
{
    dayloop {
        repeat-delay 15,50
        distance 5, 50
        sound
        {
            strain_1.wav
        }
    }
}
etc.
```


Breakdown of Soundscripts:**repeat-delay**

1 or 2 numbers (min, max, in sec) time to delay between the end of the sound playing, and playing it again randomised between (min .. max)

default min = 0 default max = min

attachment

attachment point on the object to attach the sound to.

default: origin of parent object
(not used for ambient sound)

distance

2 numbers (meters)

1st number = the distance at which the sound is played at 100%

2nd number = the cut-off distance.
doesn't affect the volume of the sound
default: 50m, 150m

sound

list of .wav files to play (randomly picked)

volume

gain of the sound
default 1.0 = 100%

ambient

0 or 1, default 0

Ambient sounds have no 3d "position" and may be stereo non-ambient (positional) sounds are positioned on the object and must be mono

value-range

2 numbers, currently used only for day/night sound effects.

midnight is 0.5, midday = 0.0 or 1.0

Where the numbers are not the same, this sets the start and end times for the sound to play.

default 0,0 (off)

trigger

Currently used only for levers. the sound doesn't play until the trigger message happens

nostartdelay

0 or 1, default 0

If not set, the sound will have a short delay before playing, this stops flanging.
(flanging = really nasty sound caused when several copies of the same sound are played at once)

dayloop, daysingle, morning, night, toggle

These have no function in Trainz and have only been put in for user reference.

Note: Single word only. Do not use a space.

TRAINZSCRIPT TUTORIAL

Introduction

Welcome to the first TrainzScript tutorial.

TrainzScript is the scripting language developed for Auran Trainz. This document will teach you how to create a very simple scenario - it does not aim to teach you how to program, or teach you programming concepts. TrainzScript may be used from Version 1 Service Pack 3 onward to create scenario content.

If you do not understand programming concepts, you may need to read further tutorials before trying to create Scenarios for Trainz. We will be releasing a user-friendly interface for compiling powerful scripts in a later version.

Please take the time to follow the steps of this document from start to finish. The tutorial is presented in an informal (non text book) manner. More help can be found by visiting the Scenarios forum at <http://www.auran.com/trainz/forum/default.htm>

Where do I find TrainzScript

TrainzScript is a scripting language used to drive the scenarios. Each scenario will have one or more TrainzScript files (.gs) located in its directory in the **World\Custom\Scenarios** folder. The supporting TrainzScript files are found in the **\Scripts** folder. The script files in this folder give you all of the supported functions required to control Trainz. Over time, you will learn most of the functions provided in these files. The TrainzScript compiler (gsc.exe) is located in the **\Bin** folder.

Go to the bin folder in your dos prompt, and run the compiler with the -d flag as follows.

```
gsc -d > reference.txt
```

This will copy the TrainzScript documentation to the file reference.txt. Consult this document as a reference manual for the TrainzScript language.

Creating A Scenario

Lets create our very first scenario. This will teach you how to use the TrainzScript compiler and guide you through the process required to make a simple scenario that just loads a map.

1. Launch Trainz, and create a new map in Surveyor. Lay a small track loop, and place a track mark named "START" somewhere on your track. Make sure you name the track mark using UPPERCASE, as all named object in TrainzScript are case-sensitive. Place a lever somewhere on the track so we have something for the camera to focus on. Save your map and call it "Tutorial1".
2. Before exiting out of Surveyor, select the "Export Scenario TSO" from the Trainz Main Menu, and type in "Tutorial1".
3. Quit out of Trainz go to your **Trainz\World** folder. Locate the config.txt file in the folder **World\Custom\Maps\Tutorial1**.
4. Open the config.txt file of your Tutorial1map. It should look something like the following.

```
kuid <KUID:-2:2023211879>
kind map
username Tutorial1
workingscale 0
workingunits 0
water <KUID:-1:110015>
region Australia
```

Note the KUID for your new layout. In this case, it is -2:2023211879.

- Now go to the **World\Custom\Scenarios\Tutorial1** folder and open the config.txt file of your Tutorial1 Scenario. It should look something like the following.

```
kind activity
username Tutorial1
scriptlibrary Tutorial1
scriptclass MyTutorial1
kuid <KUID:-2:2023211880>
kuid-table {
tutorial1 <KUID:-2:2023211879>
}
description "Tutorial1"
```

A few notes on this. The kuid-table is a name-KUID translation table used by the scenario. All objects loaded by the scenario must be entered in this table. The scenario will reference the KUID by name, which is case-sensitive. For example, when the script loads the map, it will reference it by the name "Tutorial1", which will be looked up in the kuid-table, and found as KUID -2:2023211879. Trains and rolling stock etc are referenced in the same manner. You will also notice that the scenario has its own unique KUID. The description text is displayed in the scenario selection screen.

- Next, we will begin editing the TrainzScript file. Open the Tutorial1.gs file with your desired programming editor. This file is created from the template.gst in the **scripts** folder when you export the TSO. The syntax of the following script is explained in the compiler documentation, but if you have a read through it, it is pretty self-explanatory. The file looks like this.

```
include "trainz.gs"

//
// class MyTutorial1
// brief This is the scenario class. Modify this
// class with
// your own gameplay.
//
```

```
game class MyTutorial1 isclass Scenario
{
    Train myConsist;
    bool scenarioDone = false;

    //
    // Load will be called by Trainz to load the
    // scenario map, and when the user presses Ctrl-L
    // param data is the save game data if loading a
    // saved game.
    //
    bool Load(string data)
    {
        if(data and data.size())
        {
            Interface.Load(data);
        }

        // load the map
        if(!World.LoadMap(World.FindKUID("Tutorial1")))
        {
            Interface.Log("Error loading scenario map");
            return false;
        }

        return true;
    }

    //
    // Save will be called by Trainz when the user
    // presses Ctrl-S.
    // return the save game string, such that load
    // will be able to restore the save game
    // from the last save check point.
    //
    string Save()
    {
        return Interface.Save();
    }

    //
    // TrainDerailed will be called by Trainz when a
    // train derails
    //
    void TrainDerailed(int trainId)
    {
        if(!scenarioDone)
        {
            World.EndScenario(10);
            scenarioDone = true;
        }
    }
}
```

```

//
// TrainCollided will be called by Trainz when a
train collides
//
void TrainCollided(int trainId)
{
    if(!scenarioDone)
    {
        World.EndScenario(10);
        scenarioDone = true;
    }
}

//
// TrainSpeedingFine() is called by Trainz every
second your trains speed exceeds the floating limit
//
void TrainSpeedingFine()
{
    //Interface.AdjustScore(-10);
}

//
// TrainBadCouple() is called by Trainz when
vehicles couple greater than 8KPH.
//
void TrainBadCouple(int vehicleId)
{
    //Interface.AdjustScore(-200);
}

//
//
// main thread
// brief main is executed automatically after
Load() is called. edit
// main to contain your scenarios gameplay.
//
//

thread void main(void)
{
    // Start the monitor thread to monitor speeding,
derailing etc.
    Monitor();

    //
    // create consist specs
    //

    //
    // create consists

```

```

//
//
// gameplay
//
scenarioDone = true;
}
};

```

7. The final step is to compile the scenario. Copy the ***makescript.bat*** file from ***Scripts\Docs*** folder into your ***World\Custom\Scenarios\Tutorial1*** folder. Run this batch file and make sure no errors are reported (consult the forum for assistance). Notice that this batch file uses the TrainzScript compiler (gsc) to create the Tutorial1.gsl file.
8. You are now ready to launch your very first scenario. The scenario should load your Tutorial1 map. You will notice that the camera is focused on the junction we placed on the map. Had we not placed that junction, the map would not be visible, as Trainz has nowhere to focus the camera.

You have now successfully created your very first scenario. Study the Highland Valley scenarios and the script .gs files for further TrainzScript information. Scripting questions may also be asked on the forums.

Good luck, and we hope you will enjoy your scripting efforts

CHAPTER 6 - UTILITIES AND KUIDS

TRAINZ DOWNLOAD STATION

The Trainz Dowload Station has undertaken a complete overhaul following the release of Trainz SP3. All content upoaded to the download Station must first be packaged using 'Trainz Content Dispatcher' available on the Trainz Website www.virtualtrainz.com.

Upload Checks:

Below is the list of checks made during the approval of Custom Content uploaded to the Upload Station via a user's Planet Auran Profile.

Step 1. Upload Process (User)

To access the upload page, a user is required to be registered with Planet Auran and be logged in. When logged in, a user can access the upload page via the 'Your Content' option found on the left hand side menu. On this page, the user will find an 'Upload New Content' button and a list of any other approved custom content the user has uploaded. Clicking on the Upload button will take the user to an upload screen, where they will select the file for upload, check the disclaimer and submit the form.

Checks:

- Username
The user is required to be logged in to upload any Custom Content. Once uploaded, the user's User ID is then associated with that upload.
- File extension
The uploaded file is checked for the '.cdp' extention.
- File size
The maximum file size that can be uploaded is 5mb. This may be revised in the future.

Step 2. Extraction Process (Automated)

This process is run nightly. It extracts all new uploads, that have been submitted for processing.

Checks:

- Valid Archive file

Step 3. Process Content (Automated)

This process is run nightly after Step 2.

This process, individually examines each content item that has been extracted in Step 2. If an error is found in an item, an error report is created, sent to the user and the file is removed. The whole pack is examined for errors prior to the error report being sent, therefore, the report will detail all errors in the pack.

Checks (If a user's content fails any of these checks, the content will be removed and the user notified)

- Checks file/s were extracted successfully
- Compares the number of files extracted in Step 2 to the number processed in this step.
- Ensures that the User ID of the content belongs to the user uploading the content
- If content was uploaded via a Group member, it checks that the User ID belongs to that Group.
- Checks that the KUID is a valid
- Checks that the KUID of custom content doesn't already exist in the Download Station
- Checks name for: Minimum length of (4) and swear words
- Checks Description for: swear words
- If updating content, it checks that the content being updated is the latest version and not a previous version.

Other Checks:

- Checks the Region / Country codes are valid
- Checks the Eras are valid
- Checks Description for: Maximum length of (255), if over 255 description is concatenated.
- Checks name for: Maximum length of (64), if over 64 name is concatenated.

If the file a user uploads contains multi-files, each file will be added to the Download Station separately, and a Pick List will be created for the pack of content.

Step 4. Approval Process (Auran Staff)

This is a visual confirmation of the content. The user will receive email to indicate if the content has been approved or declined. If it is approved, the content will be available approx. 1 hour after the email is sent.

Checks:

Failing these checks will not always fail the upload

- Name and Description: If it is appropriate, it will be approved
- Image visually acceptable
- Category Selected

TRAINZ CONTENT FOUNDRY

Trainz Content Foundry is included in the UTC install. Users of Trainz V1.3 can download the 'Trainz Content Update Pack' from the Trainz website (www.virtualtrainz.com).

This program generally has two functions:

1 Trainz Content Foundry can be used to set up the config.txt file and the default directory structure for any type of scenery object. Template meshes and default textures are put in place - ready for substitution with custom meshes and textures.

2 It can also be used to set up aliasing to *Trainz Paintshed V1.3* meshes. It sets up the config.txt file, extracts the required textures from the Paintshed archive. You can then edit the textures at will using a 2d graphics program (such as Photoshop or Paintshop Pro).

Note: You need to purchase Trainz Paintshed 1.3 or UTC (or upgrade to Trainz Paintshed V1.3 if you have version 1) to be able to use this feature.

AURAN KUIDS**TRAINS**

VERSION 1

<kuid:-1:100752>	OBB 1044	<kuid:-1:100043>	B&O Cupola Caboose
<kuid:-1:100754>	QR 2100 Class	<kuid:-1:100046>	NYC Cupola Caboose
<kuid:-1:100085>	PRR 40'Boxcar	<kuid:-1:101253>	DB DDm915
<kuid:-1:100929>	IC 4 Bay Hopper	<kuid:-1:101248>	IP DF
<kuid:-1:101224>	IC 4 bay Hopper Loaded	<kuid:-1:100861>	NSW SRA 44 Class
<kuid:-1:100086>	PRR 50'Boxcar	<kuid:-1:100825>	DL500
<kuid:-1:100087>	PRR 60'Boxcar	<kuid:-1:100778>	SAR 930 Class
<kuid:-1:100023>	SAR/SAS 6E1	<kuid:-1:100820>	SAR 930 Class single
<kuid:-1:100024>	SAR/SAS 6E1	<kuid:-1:100812>	NSWGR 44 Class
<kuid:-1:100143>	ER FA1	<kuid:-1:101032>	DB Erz IIIId
<kuid:-1:100144>	LV FA1	<kuid:-1:101254>	DB Erz IIIId Loaded
<kuid:-1:100145>	Rock Island FA1	<kuid:-1:100117>	B&O F7A
<kuid:-1:100150>	CN FPA4	<kuid:-1:100118>	B&O F7B
<kuid:-1:100153>	CP FPA2	<kuid:-1:100121>	CP FP7A
<kuid:-1:101242>	IP ARL	<kuid:-1:100122>	CP F7B
<kuid:-1:100159>	ATSF Baggage Car	<kuid:-1:100131>	NYC F7A
<kuid:-1:100160>	ATSF Chair Car	<kuid:-1:100132>	NYC F7B
<kuid:-1:100161>	ATSF Diner	<kuid:-1:1>	ATSF F7A
<kuid:-1:100162>	ATSF Pleasure Dome	<kuid:-1:100135>	ATSF F7B
<kuid:-1:100163>	ATSF Pullman - Pine series	<kuid:-1:100136>	Southern F7A
<kuid:-1:100019>	SNCF 15000 BB	<kuid:-1:100137>	Southern F7B
<kuid:-1:100004>	NS 1600	<kuid:-1:101220>	SS Foundry Car
<kuid:-1:100878>	SNCF Container Wagon	<kuid:-1:100049>	Freightliner Container Flat
<kuid:-1:101246>	IP BRJ	<kuid:-1:100092>	Penn Salt GATX Tanker
<kuid:-1:101425>	SAR Centenary Car	<kuid:-1:101292>	SAR Guard Van
<kuid:-1:100012>	CO-OP Bulk Hopper	<kuid:-1:100093>	SJ Hbbins881
<kuid:-1:100090>	BR Class 37 yellow	<kuid:-1:101204>	SJ Habbins941
<kuid:-1:100039>	BR HST Class 43	<kuid:-1:101249>	IP HGM
<kuid:-1:100029>	BR Deltic blue	<kuid:-1:101251>	IP HM
<kuid:-1:100030>	BR Deltic green	<kuid:-1:100064>	IF Refrigerated Van
<kuid:-1:100042>	CN Cupola Caboose	<kuid:-1:100072>	DB Sssy - 716
<kuid:-1:100048>	NSWGR NQO Flatcar	<kuid:-1:100106>	BR MK2 BFK
		<kuid:-1:100107>	BR MK2 FK
		<kuid:-1:100108>	BR MK2 TSO
		<kuid:-1:100058>	BR HST MK3 TGS
		<kuid:-1:100059>	BR HST MK3 TRBS

<kuid:-1:100060>	BR MK3 Sleeper	<kuid:-12:500>	QR MAS 1488
<kuid:-1:100062>	BR HST MK3 TS	<kuid:-12:501>	QR MAS 1491
<kuid:-1:100770>	NSWGR NLKY louvered van	<kuid:-12:502>	QR MBC
<kuid:-1:100828>	NSWGR NOD	<kuid:-12:503>	QR MCC
<kuid:-1:101222>	NSWGR NOD Loaded	<kuid:-12:504>	QR MPC
<kuid:-1:101048>	OBB ABmz	<kuid:-12:505>	QR PWA
<kuid:-1:101047>	OBB Wrmz	<kuid:-12:506>	QR PWA Loaded
<kuid:-1:101044>	OBB Bmz	<kuid:-12:3537>	BN SD40-2
<kuid:-1:101051>	OBB Amz	<kuid:-12:3534>	CN SD40-2
<kuid:-1:101216>	SJ Os Flat Car		
<kuid:-1:100814>	V & S.A.R Overland Club Car	VERSION 1.5 - ULTIMATE TRAINZ COLLECTION	
<kuid:-1:100815>	V & S.A.R Overland First Class	<kuid:-1:101457>	DB Class 218 Ozeanblau
<kuid:-1:100816>	V & S.A.R Overland Second Class	<kuid:-1:101458>	DB Class 218 Rot & Beige
<kuid:-1:100817>	V & S.A.R Overland Sleeper Car	<kuid:-1:100089>	SAR/Spoornet Class 34 EMD
<kuid:-1:100066>	Baby Ruth Hi-Cube Boxcar	<kuid:-1:100737>	AN 830 Class
<kuid:-1:100017>	PRR FM flat car	<kuid:-1:100734>	DL531 Silverton
<kuid:-1:101153>	QR QLX Bronco	<kuid:-1:100725>	NSWGR 48 Class
<kuid:-1:101154>	QR QLX	<kuid:-1:101393>	EF81
<kuid:-1:100094>	SJ Rc4	<kuid:-1:100841>	PRR GG1
<kuid:-1:100871>	ATSF SD40-2	<kuid:-1:100013>	AN GM Class
<kuid:-1:101060>	SS Tipper Car	<kuid:-1:100167>	GP38-2 - Canadian National
<kuid:-1:100822>	UC Silos	<kuid:-1:100176>	GP38-2 - Milwaukee Road
<kuid:-1:100068>	DB V200 Ozeanblau	<kuid:-1:100182>	GP38-2 - Seaboard System
<kuid:-1:100070>	RENFE Class 340	<kuid:-1:101412>	SNCF TGV SE Unit 1
<kuid:-1:100018>	VR S blue	<kuid:-1:101416>	SNCF TGV SE Unit 2
<kuid:-1:100007>	V/Line S	<kuid:-1:101417>	SNCF TGV SE Unit 3
<kuid:-1:100763>	BR ZHV Mineral Wagon	<kuid:-1:101420>	SNCF TGV SE Unit 4
<kuid:-1:101229>	BR ZHV Mineral Wagon Loaded		

UTC SCENARIO

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:47092:20>	Robe Alco 636	<kuid:-10:223>	drivertrain
<kuid:2512:1372>	BR Class 37 Yellow-C	<kuid:-10:224>	inv_train
<kuid:2512:1371>	BR Class 37 Coal-C	<kuid:3801:1001001>	Pilbara Ore Hopper
<kuid:-1:100127>	MR F7A	<kuid:3801:1001002>	Pilbara Ore Hopper Loaded
<kuid:-1:100128>	MR F7B	<kuid:-18:1100>	SW7 Burlington
<kuid:-12:460>	AN Guard Van	<kuid:523:101060>	SS Tipper Car Loaded

BOGEYS

VERSION 1

<kuid:-1:100749>	1044_obb_bogey
<kuid:-1:100750>	2100_bogey
<kuid:-1:100073>	40ft_Boxcar_bogey
<kuid:-1:100074>	50ft_Boxcar_bogey
<kuid:-1:100075>	60ft_Boxcar_bogey
<kuid:-1:100022>	6e1_bb_bogey
<kuid:-1:100138>	AlcoFA1_bogey
<kuid:-1:101241>	arl_bogey
<kuid:-1:100139>	atsf_car_bogey
<kuid:-1:100005>	bb_bogey
<kuid:-1:101424>	centenary_bogey
<kuid:-1:100008>	cflow_fert_bogey
<kuid:-1:100866>	class103_redbeige_bogey
<kuid:-1:100078>	Class37_bogey
<kuid:-1:100033>	Class43_bogey
<kuid:-1:100028>	class_55_bogey
<kuid:-1:100047>	Container_flat_bogey
<kuid:-1:100041>	cupola_caboose_bogey
<kuid:-1:101252>	DDm915_bogey
<kuid:-1:100777>	dl500_sar_bogey
<kuid:-1:100811>	dl500_tuscan_bogey
<kuid:-1:101394>	EF81_Red_bogey
<kuid:-1:101392>	EF81_Red_midbogey
<kuid:-1:0>	f7_bogey
<kuid:-1:101219>	foundry_car_bogey
<kuid:-1:100080>	GATX_pennsalt_bogey
<kuid:-1:101291>	guard_van_bogey
<kuid:-1:100081>	Hbbins881_freightcar_bogey
<kuid:-1:101203>	Hbbins941_freightcar_bogey
<kuid:-1:100052>	hst_mk3_bogey
<kuid:-1:100063>	interfrigo_van_bogey
<kuid:-1:100071>	lwagon_bogey
<kuid:-1:100099>	mk2_bogey

<kuid:-1:100769>	nlky_bogey
<kuid:-1:100827>	nod_bogey
<kuid:-1:101043>	obb_eurofima_bogey
<kuid:-1:101215>	Os_freightcar_bogey
<kuid:-1:100813>	overland_bogey
<kuid:-1:100065>	pdhc_bogey
<kuid:-1:100011>	prf_fm_tuscan_bogey
<kuid:-1:100082>	c4_bogey
<kuid:-1:100870>	sd40_2_santafe_bogey
<kuid:-1:101059>	tipper_bogey
<kuid:-1:100821>	uc_silos_bogey
<kuid:-1:100067>	v200_bogey
<kuid:-1:100006>	vrs_bogey

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:2512:33371>	class37_bogey
<kuid:-10:149>	invisible_bogey
<kuid:-12:153>	QR_car_bogey
<kuid:-12:3535>	sd40_2_santafe_dk_bogey
<kuid:-12:2536>	tuscan_bogey

VERSION 1.5 - ULTIMATE TRAINZ COLLECTION

<kuid:-1:100076>	Class218_Bogey
<kuid:-1:100077>	Class34-200Red_bogey
<kuid:-1:100691>	dl531_SAR_bogey
<kuid:-1:100724>	dl531_tuscan_bogey
<kuid:-1:100839>	gg1_bogey
<kuid:-1:100840>	gg1_bogey_big
<kuid:-1:100009>	gm_bogey
<kuid:-1:100141>	gp38_2_dark_bogey
<kuid:-1:100142>	gp38_2_light_bogey
<kuid:-1:101414>	invis_bogey
<kuid:-1:101418>	TGV_end_unit_bogey
<kuid:-1:101410>	TGV_SE_loco_bogey
<kuid:-1:101415>	TGV_unit_bogey

PANTOGRAPHS

VERSION 1

<kuid:-1:101255>	1044_panto
<kuid:-1:110002>	6e1_bb_pantograph
<kuid:-1:100860>	bb_pantograph
<kuid:-1:101395>	EF81_Red_pantograph
<kuid:-1:101177>	Rc4_pantograph

VERSION 1.5 - ULTIMATE TRAINZ COLLECTION

<kuid:-1:100838>	gg1_pantograph
<kuid:-1:101411>	TGV_pantograph

HORNSOUND

VERSION 1

<kuid:-1:42003103>	ALCo
<kuid:-1:42003102>	br
<kuid:-1:42003101>	default
<kuid:523:100433>	redhen

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:-12:37>	br2
<kuid:-12:38>	br3

ENGINESOUND

VERSION 1

<kuid:-1:42003001>	alco
<kuid:-1:42003002>	electric
<kuid:-1:42003000>	emd
<kuid:523:100432>	redhen

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:-12:2100>	QR2100
-----------------	--------

INTERIORS

VERSION 1

<kuid:-1:100554>	bb15000
<kuid:-1:100382>	class37_int_cab
<kuid:-1:101168>	class43_interior
<kuid:-1:100186>	f7interior
<kuid:-1:101211>	f7interior_kph_lhd
<kuid:-1:101212>	f7interior_mph_lhd
<kuid:-1:101202>	f7interior_mph_rhd

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:-10:172>	Gen1960_op1
<kuid:-10:167>	gen_1044
<kuid:-10:168>	Gen_6e1
<kuid:-10:169>	Gen_deltic
<kuid:-1:101472>	Gen_DL500
<kuid:-10:170>	Gen_DL500_mph_rhd
<kuid:-10:173>	Gen_RC4
<kuid:-1:101475>	Gen_sd
<kuid:-10:166>	v200

VERSION 1.5 - ULTIMATE TRAINZ COLLECTION

<kuid:-10:221>	Gen_218
<kuid:-1:101473>	Gen_dl
<kuid:-1:101474>	gen_gg
<kuid:-10:216>	gen_tgv

UTC SCENARIO

<kuid:-18:1101>	sw
-----------------	----

ENGINESPEC

VERSION 1

<kuid:-1:42004203> default_loco
<kuid:2512:337> EE Type 3
<kuid:-1:42004250> F7_hotrod
<kuid:523:402> RedHen
<kuid:-1:42004222> V200
<kuid:-1:42004221> SD40 2
<kuid:-1:42004220> RC4
<kuid:-1:42004219> GM
<kuid:-1:42004218> FPA4
<kuid:-1:42004217> FA2
<kuid:-1:42004216> FA1
<kuid:-1:42004202> F7
<kuid:-1:42004215> Deltic
<kuid:-1:42004201> default_wagon
<kuid:-1:42004206> Class 43
<kuid:-1:42004204> Class 37
<kuid:-1:42004205> BB 15000
<kuid:-1:42004213> 930
<kuid:-1:42004212> 6E1
<kuid:-1:42004211> 44
<kuid:-1:42004210> 340
<kuid:-1:42004209> 2100
<kuid:-1:42004208> 1600
<kuid:-1:42004207> 1044

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:2512:308> Class 08
<kuid:-1:42004242> default_car
<kuid:-1:44482> Fscotsman
<kuid:-1:42004243> gantrycrane

VERSION 1.5 - ULTIMATE TRAINZ COLLECTION

<kuid:-1:42004224> dl531
<kuid:-1:42004231> GP38_2
<kuid:-1:42004229> TGV

UTC SCENARIO

<kuid:-18:1103> SW7

SCENERY

VERSION 1

Australia

<kuid:-1:100263> Building_Corner_Shop

<kuid:-1:100468> Misc_Hills_Hoist

<kuid:-1:100251> Animal_Horse

<kuid:-1:100252> Animal_Horse2

<kuid:-1:100253> Animal_Horse3

<kuid:-1:100256> Building_Tilnowle_Pub

<kuid:-1:100257> House_1

<kuid:-1:100202> House_1a

<kuid:-1:100258> House_2

<kuid:-1:100259> House_2b

<kuid:-1:100203> House_2a

<kuid:-1:100260> House_3

<kuid:-1:100204> House_3a

<kuid:-1:100261> House_Hiset_1

<kuid:-1:100205> House_Hiset_2

<kuid:-1:100206> House_Hiset_2a

<kuid:-1:100213> House_Hiset_2b

<kuid:-1:100323> House_Hiset_1a

<kuid:-1:100823> Tside_2t_Level_Crossing

<kuid:-1:100824> Tside_1t_Level_Crossing

<kuid:-1:100254> Building_Cane_Mill

<kuid:-1:100551> People_Crowd

<kuid:-1:100540> People_Group

<kuid:-1:100541> People_Group2

<kuid:-1:100542> People_Lineup

<kuid:-1:100543> People_Lineup2

<kuid:-1:100544> People_person1

<kuid:-1:100545> People_person2

<kuid:-1:100546> People_person3

<kuid:-1:100547> People_person4

<kuid:-1:100548> People_person5

<kuid:-1:100549> People_person6

<kuid:-1:100550> People_person7

<kuid:-1:100255> Building_School_Hall

<kuid:-1:100262> Building_Shearing_Shed

<kuid:-1:100274> Animal_Sheep_Flock

<kuid:-1:100276> Tree_Large_shrub

<kuid:-1:101434> Sign_Station

<kuid:-1:100264> Tside_Signal_Box1

<kuid:-1:100265> Tside_Signal_Box2

<kuid:-1:100266> Tside_Signal_Box3

<kuid:-1:100267> Tside_Signal_Box4

<kuid:-1:100419> Sign_Road_Giveway

<kuid:-1:101435> Sign_Tall

<kuid:-1:100001> Building_Grain_Silos

<kuid:-1:100268> Building_Small_Shops

<kuid:-1:100269> Tside_Small_Station

<kuid:-1:100384> Tside_Station

<kuid:-1:100270> House_Thunderbox_1

<kuid:-1:100271> House_Thunderbox_1b

<kuid:-1:100272> House_Thunderbox_2

<kuid:-1:100273> House_Thunderbox_2b

<kuid:-1:100388> Vehicle_Tracktor_Blue

<kuid:-1:100389> Vehicle_Tracktor_R

<kuid:-1:100390> Vehicle_Tracktor

<kuid:-1:100218> Sign_Road_Lights_1

<kuid:-1:100219> Sign_Road_Lights_2

<kuid:-1:100391> Tree_Bunya_Group

<kuid:-1:100275> Tree_Gum_Group

<kuid:-1:100461> Tree_Gum_Group_2

<kuid:-1:100277> Tree_Bunya_Pine

<kuid:-1:100278> Tree_Bunya_Pine_2

<kuid:-1:100279> Tree_Bunya_Pine_3

<kuid:-1:100280> Tree_Flame

<kuid:-1:100281> Tree_Flame_I

<kuid:-1:100282> Tree_Gum_1

<kuid:-1:100283> Tree_Gum_2

<kuid:-1:100407> Tree_Gum_3

<kuid:-1:100408> Tree_Gum_4

<kuid:-1:100392>	Tree_Palm_1a	<kuid:-1:32>	Vehicle_Hilux_Blue
<kuid:-1:100393>	Tree_Palm_1b	<kuid:-1:33>	Vehicle_Hilux_Grey
<kuid:-1:100394>	Tree_Palm_2	<kuid:-1:570>	House_Brick_Red
<kuid:-1:100284>	Tree_Small	<kuid:-1:571>	House_Small
<kuid:-1:100285>	Tree_Stickish	<kuid:-1:572>	House_Stone
<kuid:-1:100286>	Tree_Stickish_Group	<kuid:-1:569>	House_Brick_White
<kuid:-1:100287>	Tree_Umbrella	<kuid:-1:34>	Vehicle_Jag_Green
<kuid:-1:100385>	Tside_Turntable	<kuid:-1:35>	Vehicle_Jag_Black
<kuid:-1:100506>	Tside_Turntable_2	<kuid:-1:36>	Vehicle_Jag_Red
<kuid:-1:100288>	Tside_2t_Walkover	<kuid:-1:37>	Vehicle_Jetta_Blue
<kuid:-1:100415>	Building_watertank	<kuid:-1:38>	Vehicle_Jetta_Green
<kuid:-1:100289>	Building_Wharf	<kuid:-1:1009>	Misc_Jetty
<kuid:-1:100290>	Building_Wharf_Shed	<kuid:-1:3577>	Tside_Crossing2
<kuid:-1:110018>	Tside_Sign_Wistle	<kuid:-1:100735>	Tside_Crossing1
<kuid:-1:100386>	Misc_Windmill	<kuid:-1:1022>	Building_PowerTower
<kuid:-1:100387>	Building_Workshed	<kuid:-1:574>	Tside_Relaybox
English Countryside		<kuid:-1:594>	Building_Shed
	<kuid:-1:521>	<kuid:-1:101438>	Tside_Sign_Station
<kuid:-1:568>	House_Brick_Unit	<kuid:-1:101439>	Tside_Sign_Station_Tall
<kuid:-1:1012>	Building_Factory	<kuid:-1:586>	Tside_Station
<kuid:-1:524>	Tside_Junction_Building	<kuid:-1:100220>	Misc_Traffic_Light1
<kuid:-1:525>	Tside_Junction_Building2	<kuid:-1:100221>	Misc_Traffic_Light2
<kuid:-1:1021>	Building_Silo	<kuid:-1:564>	Tree_VNS
<kuid:-1:1008>	Building_Tankshed	<kuid:-1:565>	Tree_VWF
<kuid:-1:520>	Building_Church	<kuid:-1:527>	Tree4
<kuid:-1:573>	House_Cottage	<kuid:-1:528>	Tree9
<kuid:-1:587>	Flower_Yellow2	<kuid:-1:529>	Tree9_Large
<kuid:-1:591>	Flower_Lavender	<kuid:-1:593>	Tree9_Side
<kuid:-1:588>	Flower_Yellow	<kuid:-1:560>	Tree26
<kuid:-1:589>	Flower_Yellow_Spots	<kuid:-1:561>	Tree28
<kuid:-1:590>	Flowers_Lavender2	<kuid:-1:562>	Tree52
<kuid:-1:6>	Tree_Forrest9	<kuid:-1:563>	Tree52M
<kuid:-1:16>	Tree_Forrest_VWF	<kuid:-1:27>	Tree_Group1
<kuid:-1:526>	Building_Church_Headstones	<kuid:-1:1015>	Tree_Group2
<kuid:-1:31>	Vehicle_Hilux_Red	<kuid:-1:1016>	Tree_Group3
		<kuid:-1:566>	Tree10

<kuid:-1:30>	Vehicle_Van_White	<kuid:-1:100500>	House_4
<kuid:-1:28>	Vehicle_Van_ATeam	<kuid:-1:100501>	House_5
<kuid:-1:567>	Tside_Walkover	<kuid:-1:100507>	House_6
<kuid:-1:581>	Wall_Rock	<kuid:-1:100508>	House_7
<kuid:-1:582>	Wall_Stone	<kuid:-1:100509>	House_8
<kuid:-1:41>	Vehicle_X4_Brown	<kuid:-1:100510>	House_9
<kuid:-1:42>	Vehicle_X4_Grey	<kuid:-1:100563>	Animal_Horse_1
<kuid:-1:530>	Tree_Forest_X9	<kuid:-1:100564>	Animal_Horse_2
<kuid:-1:5>	Tree_Forest_XVNS	<kuid:-1:100601>	Animal_Horse_3
<kuid:-1:577>	Tside_Sign_Xing	<kuid:-1:100565>	House_10
		<kuid:-1:100566>	House_11a
USA		<kuid:-1:100602>	Building_Interlocking_Tower
<kuid:-1:100599>	Vehicle_4x4_Green	<kuid:-1:100593>	Vehicle_Jag_Red
<kuid:-1:100558>	Vehicle_4x4_Police	<kuid:-1:100567>	Vehicle_Jetta_Blue
<kuid:-1:100600>	Vehicle_4x4_Red	<kuid:-1:100958>	Shrub_Lichen_1
<kuid:-1:100559>	Building_Bakery	<kuid:-1:100959>	Shrub_Lichen_2
<kuid:-1:100585>	Building_Shed_Big	<kuid:-1:100960>	Shrub_Lichen_3
<kuid:-1:100635>	Building_Blast_Furnace	<kuid:-1:100961>	Shrub_Lichen_4
<kuid:-1:100638>	Vehicle_Bug	<kuid:-1:100594>	House_Log_Cabin
<kuid:-1:100915>	Tree_Cactus_1	<kuid:-1:100502>	Station_Metro
<kuid:-1:100932>	Tree_Cactus_1b	<kuid:-1:100503>	Station_Metro_Single
<kuid:-1:100916>	Tree_Cactus_2	<kuid:-1:100433>	Building_Mining_Co
<kuid:-1:100917>	Tree_Cactus_3	<kuid:-1:100504>	Building_Motel
<kuid:-1:100918>	Tree_Cactus_4	<kuid:-1:100560>	Building_Music_Shop
<kuid:-1:100496>	Building_Cafe	<kuid:-1:100526>	Vehicle_Mustang_Blue
<kuid:-1:100872>	Building_Coke_Oven	<kuid:-1:100623>	Vehicle_Mustang_Blue_FB
<kuid:-1:100431>	Sign_East_West	<kuid:-1:100624>	Vehicle_Mustang_Yellow_FB
<kuid:-1:100561>	Building_Factory	<kuid:-1:100527>	Vehicle_Mustang_Red
<kuid:-1:100622>	Vehicle_Falcon	<kuid:-1:100742>	Building_Open_Hearth
<kuid:-1:100432>	Building_Freight_Depot	<kuid:-1:100639>	People_Crowd
<kuid:-1:100592>	Building_Gas_Station	<kuid:-1:100640>	People_Group_1
<kuid:-1:100533>	Building_Grain_Elevator	<kuid:-1:100641>	People_Group_2
<kuid:-1:100562>	Vehicle_Hilux_Grey	<kuid:-1:100642>	People_Lineup_1
<kuid:-1:100497>	House_1	<kuid:-1:100643>	People_Lineup_2
<kuid:-1:100498>	House_2	<kuid:-1:100644>	People_1
<kuid:-1:100499>	House_3	<kuid:-1:100645>	People_2

<kuid:-1:100646>	People_3	<kuid:-1:100920>	Shrub_2
<kuid:-1:100647>	People_4	<kuid:-1:100930>	Shrub_3
<kuid:-1:100648>	People_5	<kuid:-1:100931>	Shrub_4
<kuid:-1:100649>	People_6	<kuid:-1:101440>	Sign_Station
<kuid:-1:100650>	People_7	<kuid:-1:101441>	Sign_Station_Tall
<kuid:-1:100655>	People_Pink	<kuid:-1:100873>	Building_Sintering_Plant
<kuid:-1:100656>	People_Suit_1	<kuid:-1:100436>	Building_Stone_1
<kuid:-1:100657>	People_Suit_2	<kuid:-1:100437>	Building_Stone_2
<kuid:-1:100658>	People_White	<kuid:-1:100438>	Building_Stone_3
<kuid:-1:100570>	Tree_Pine_High	<kuid:-1:100439>	Building_Stone_4
<kuid:-1:100935>	People_Platform_Group_1	<kuid:-1:100440>	Building_Stone_5
<kuid:-1:100936>	People_Platform_Group_2	<kuid:-1:100441>	Building_Stone_6
<kuid:-1:100942>	People_Platform_Group_3	<kuid:-1:100733>	Building_Stone_7
<kuid:-1:101257>	People_Platform_Group_4	<kuid:-1:100663>	Misc_Street_Lamp
<kuid:-1:100924>	People_Platform_1	<kuid:-1:100571>	Building_Supermarket
<kuid:-1:100943>	People_Platform_10	<kuid:-1:100595>	Building_Surveyor
<kuid:-1:100948>	People_Platform_11	<kuid:-1:100442>	Building_Terminus
<kuid:-1:100949>	People_Platform_12	<kuid:-1:100443>	Building_Tower_1
<kuid:-1:100950>	People_Platform_13	<kuid:-1:100444>	Building_Tower_2
<kuid:-1:100925>	People_Platform_2	<kuid:-1:100445>	Building_Tower_3
<kuid:-1:100926>	People_Platform_3	<kuid:-1:100446>	Building_Tower_4
<kuid:-1:100937>	People_Platform_4	<kuid:-1:100572>	Building_Toy_Shop
<kuid:-1:100938>	People_Platform_5	<kuid:-1:100573>	Vehicle_Tractor
<kuid:-1:100939>	People_Platform_6	<kuid:-1:100665>	Misc_Traffic_Lights_USA_1
<kuid:-1:100940>	People_Platform_7	<kuid:-1:100666>	Misc_Traffic_Lights_USA_2
<kuid:-1:100941>	People_Platform_8	<kuid:-1:100667>	Misc_Traffic_Lights_USA_3
<kuid:-1:100944>	People_Platform_9	<kuid:-1:100574>	Tree_Pine_Group
<kuid:-1:100636>	Building_Rolling_Mill	<kuid:-1:100802>	Tree_Autumn_9
<kuid:-1:100434>	Sign_Route_66	<kuid:-1:100505>	Tree_1
<kuid:-1:100435>	Sign_RX	<kuid:-1:100803>	Tree_Autumn_1
<kuid:-1:100596>	Building_School	<kuid:-1:100740>	Tree_2
<kuid:-1:100625>	Vehicle_Semi_Grey	<kuid:-1:100575>	Tree_26
<kuid:-1:100626>	Vehicle_Semi_Red	<kuid:-1:100804>	Tree_Autumn_2
<kuid:-1:100962>	Shrub_Group_1	<kuid:-1:100844>	Tree_3
<kuid:-1:100963>	Shrub_Group_2	<kuid:-1:100805>	Tree_Autumn_3
<kuid:-1:100919>	Shrub_1	<kuid:-1:100845>	Tree_5

<kuid:-1:100576>	Tree_52	<kuid:-1:101266>	Sign_Xing_USA_Boom
<kuid:-1:100806>	Tree_Autumn_5	<kuid:-1:100637>	Building_Yard_Office
<kuid:-1:100603>	Tree_Group_1		
<kuid:-1:100807>	Tree_Autumn_Group_1	VERSION 1.3 - SERVICE PACK 3 (SP3)	
<kuid:-1:100604>	Tree_Group_2	<kuid:9:24006>	10m_concrete_hid
<kuid:-1:100808>	Tree_Autumn_Group_2	<kuid:-12:162>	Sign_Curve_20_R
<kuid:-1:100577>	Tree_Group_3	<kuid:-12:404>	Sign_Curve_20_L
<kuid:-1:100578>	Tree_Gum_Group_1	<kuid:-12:490>	Sign_Curve_60_R
<kuid:-1:100605>	Tree_Gum_Group_2	<kuid:-12:403>	Sign_Curve_60_L
<kuid:-1:100606>	Tree_Gum_1	<kuid:9:24010>	6m_concrete_twin45_hps
<kuid:-1:100581>	Tree_Gum_2	<kuid:9:24009>	6m_concrete_twin_hid
<kuid:-1:100582>	Tree_Gum_4	<kuid:44482:26008>	Misc_Bench_Platform
<kuid:-1:100583>	Tree_Palm_1a	<kuid:44482:26009>	Misc_Bench
<kuid:-1:100607>	Tree_Palm_1b	<kuid:-10:156>	Animal_Buffalo_1
<kuid:-1:100579>	Tree_Pine_1	<kuid:-10:157>	Animal_Buffalo_2
<kuid:-1:100580>	Tree_Pine_2	<kuid:-10:158>	Animal_Buffalo_3
<kuid:-1:100568>	Tside_Turntable	<kuid:-10:159>	Animal_Buffalo_Herd
<kuid:-1:100809>	Building_USA_Brick_1	<kuid:-10:160>	Animal_Buffalo_Herd_L
<kuid:-1:100875>	Building_USA_Brick_2	<kuid:1942:26014>	Building_Bank_Co-Op
<kuid:-1:100951>	Building_USA_Brick_3	<kuid:44797:28013>	Ind_Coal_Dock
<kuid:-1:101156>	Building_USA_Brick_4	<kuid:44797:28014>	Ind_Coal_Dock_Powerhouse
<kuid:-1:101135>	Building_USA_Brick_5	<kuid:1942:25006>	House_Cottage
<kuid:-1:101137>	Building_USA_Brick_6	<kuid:55712:26001>	Building_Diner
<kuid:-1:100863>	Building_USA_Radio_Antenna	<kuid:3801:27000>	dumper
<kuid:-1:100651>	Building_USA_Silo_Complex	<kuid:47092:50221>	Building_Warehouse_EDC
<kuid:-1:100688>	Station_USA_1	<kuid:1942:28000>	Building_Shed_Engine
<kuid:-1:100689>	Station_USA_1_Platform	<kuid:1942:27004>	Building_Factory_1
<kuid:-1:100921>	Tree_USA_1	<kuid:1942:27005>	Building_Factory_5_1942
<kuid:-1:100922>	Tree_USA_2	<kuid:-10:154>	People_Flagman_Green
<kuid:-1:100584>	Tside_Walkover	<kuid:-10:155>	People_Flagman_Yellow
<kuid:-1:100151>	Wall_1	<kuid:3801:20001>	Bridge_Fort
<kuid:-1:100569>	Building_WaterTank	<kuid:44797:28002>	Tside_Platform_Freight
<kuid:-1:100512>	Sign_Whistle	<kuid:1942:27006>	Misc_Gasometer
<kuid:-1:100964>	Sign_Xing_Rural	<kuid:55712:25001>	House_1
<kuid:-1:100448>	Sign_Xing_USA	<kuid:3801:80001>	Animal_Kangaroo
<kuid:-1:101265>	Sign_Xing_USA_1	<kuid:44482:26010>	Tside_Platform_Lamp

<kuid:44482:26006>	Building_Library	<kuid:-12:166>	Tside_Overhead_QR
<kuid:1942:34007>	Misc_Memorial_1942	<kuid:-12:791>	Tside_Payphone_QR
<kuid:26001:23030>	Tside_MPM	<kuid:-12:680>	Tside_Rubbish_Bin_QR
<kuid:26001:23031>	Tside_MPW_Frac1	<kuid:-12:681>	Tside_Seat_QR
<kuid:26001:23032>	Tside_MPW_Frac2	<kuid:-12:790>	Tside_Phone_Security_QR
<kuid:26001:23033>	Tside_MPW_Frac3	<kuid:-12:123455>	Tside_Shelter2_QR
<kuid:44797:28001>	Tside_Watertower_NYC	<kuid:-12:123454>	Tside_Shelter_QR
<kuid:30204:23001>	Sign_Station_Oz_1a	<kuid:-12:796>	Station_QR2
<kuid:30204:23002>	Sign_Station_Oz_1b	<kuid:-12:679>	Station_QR
<kuid:30204:23003>	Sign_Station_Oz_2a	<kuid:-12:792>	Tside_Ticketmachine_QR
<kuid:30204:23004>	Sign_Station_Oz_2b	<kuid:3801:23037>	Sign_RMines2
<kuid:30204:23005>	Sign_Station_Oz_3a	<kuid:44797:28011>	Building_Sandhouse
<kuid:30204:23006>	Sign_Station_Oz_3b	<kuid:-12:123456>	Building_Servo
<kuid:9:23007>	Misc_Phone	<kuid:-12:798>	Building_Shed_Big
<kuid:2512:41>	Station_SVR_Highley	<kuid:-12:799>	Building_Shed_Goods
<kuid:1942:34006>	Building_Police_Station	<kuid:-12:800>	Building_Shed_Tan
<kuid:1942:34004>	Misc_Postbox	<kuid:50435:26001>	Building_Cafe_Shore
<kuid:1942:34003>	Building_Post_Office	<kuid:-12:163>	Building_Shed_Small
<kuid:1942:26016>	Building_Pub_Carvery	<kuid:44482:26005>	Station_St_Anton
<kuid:1942:26005>	Building_Pub_White_Horse	<kuid:1942:28003>	Station_1942_2
<kuid:-12:795>	Building_QFS_Depot	<kuid:1942:28007>	Station_1942_3
<kuid:46415:23510>	Tside_Cable_Marker_QR	<kuid:1942:28009>	Station_1942_4
<kuid:46415:28060>	Tside_Cable_Pit_QR	<kuid:-12:650>	Station_1track
<kuid:46415:28070>	Tside_JunctionBox_QR	<kuid:-12:323>	Station_2track
<kuid:46415:23530>	Sign_QR_Keepout	<kuid:-12:324>	Station_3track
<kuid:46415:23520>	Sign_QR_Ohead_Wires_1	<kuid:9:23001>	Station_CP_Medium
<kuid:46415:23525>	Sign_QR_Ohead_Wires_2	<kuid:9:23002>	Station_CP_Medium_Lite
<kuid:46415:28000>	Tside_Phonebox_QR	<kuid:9:23003>	Station_CP_Medium_Old
<kuid:46415:28005>	Tside_Phonebox_QR_Call	<kuid:1942:28006>	Station_Forecourt_2
<kuid:46415:28200>	Tside_Relay_Hut_QR	<kuid:1942:28008>	Station_Forecourt_3
<kuid:46415:28050>	Tside_TC_Box_1_QR	<kuid:1942:25014>	House_5
<kuid:46415:28055>	Tside_TC_Box_2_QR	<kuid:1942:26006>	Building_Shop
<kuid:-12:123453>	Fence_4Platform_QR_2	<kuid:44797:28017>	Tside_Switchtower_LPC
<kuid:-12:123452>	Fence_4Platform_QR	<kuid:1942:34005>	Misc_Phone_Box
<kuid:-12:405>	Tside_Light_QR	<kuid:-12:44444>	Sign_Australia_Zoo
<kuid:-12:402>	Tside_Name1_QR	<kuid:-10:150>	People_Track_Gang

<kuid:-12:3545>	Tree_Gum	<kuid:-1:101287>	Building_Chicken_coop
<kuid:-12:314>	Station_Union	<kuid:-1:100668>	Building_Church
<kuid:-12:427>	Misc_Water_Column	<kuid:-1:101300>	Building_Church2
<kuid:-12:498>	Misc_Wooden_Crates	<kuid:-1:101198>	Misc_Coaster
<kuid:-12:499>	Misc_Wooden_Crates_2	<kuid:-1:101230>	Misc_Coastercar_front
<kuid:-12:161>	Building_Woombye	<kuid:-1:101231>	Misc_Coastercar_middle
<kuid:44482:26007>	Misc_WTower_Euro	<kuid:-1:101061>	Building_Pub_corner
<kuid:-12:164>	Sign_Xing_Rural_Oz	<kuid:-1:101150>	Tside_Cover_station
<kuid:3801:28000>	Building_Shed_Yellow	<kuid:-1:100869>	Ind_Crane
<kuid:44482:23084>	trackside Gantry_3Line	<kuid:-1:101085>	Building_Dieseltanks
<kuid:44482:23085>	trackside Gantry_4Line	<kuid:-1:101171>	Tside_2t_Walkover
<kuid:44482:23086>	trackside Gantry_5Line	<kuid:-1:101335>	Misc_Duckshoot
<kuid:44482:23087>	trackside Gantry_6Line	<kuid:-1:101181>	Building_Factory

VERSION 1.5 - ULTIMATE TRAINZ COLLECTION

Germany

<kuid:-1:101191>	House_Alps	<kuid:-1:101195>	House_Farm2
<kuid:-1:101192>	House_Alps2	<kuid:-1:100739>	Building_Farmsilo
<kuid:-1:101333>	Vehicle_Truck_1	<kuid:-1:101295>	Vehicle_Farmtruck
<kuid:-1:101334>	Vehicle_Truck_2	<kuid:-1:101049>	Misc_Ferris_Wheel
<kuid:-1:101319>	Misc_Balloon	<kuid:-1:101331>	Building_Fueltanks
<kuid:-1:101320>	Misc_Balloon2	<kuid:-1:101223>	Building_Gas_Station
<kuid:-1:101084>	Building_Barn	<kuid:-1:101088>	Building_Gateway
<kuid:-1:101286>	Misc_Bench	<kuid:-1:101089>	Building_Haybarn
<kuid:-1:100732>	Building_Big	<kuid:-1:101268>	Animal_Horse
<kuid:-1:100690>	Building_Castle_big	<kuid:-1:101288>	Animal_Horses
<kuid:-1:100738>	House_Block1	<kuid:-1:101301>	Building_Lookout
<kuid:-1:100741>	House_Block2	<kuid:-1:101062>	House_Log_Cabin
<kuid:-1:100743>	House_Block3	<kuid:-1:101106>	Building_Lumber_Mill
<kuid:-1:101090>	Vehicle_Mini	<kuid:-1:101151>	Tside_Station_med
<kuid:-1:101314>	Vehicle_Mini_red	<kuid:-1:101250>	Misc_Merrygoround
<kuid:-1:101205>	Vehicle_Saloon	<kuid:-1:101201>	Building_Mill
<kuid:-1:101315>	Vehicle_Saloon_black	<kuid:-1:101172>	Building_Mine
<kuid:-1:101316>	Vehicle_Saloon_purple	<kuid:-1:101050>	Tside_Overhead
<kuid:-1:101217>	Vehicle_Starsky	<kuid:-1:101226>	People_1
<kuid:-1:100672>	Building_Castle	<kuid:-1:101227>	People_2
		<kuid:-1:101262>	People_3
		<kuid:-1:101228>	People_4

<kuid:-1:101318>	People_Group	<kuid:-1:101076>	Tree_Medium2_hi
<kuid:-1:101210>	Animal_Pig	<kuid:-1:101077>	Tree_Medium1_lo
<kuid:-1:101234>	Animal_Piggies	<kuid:-1:101078>	Tree_Medium2_lo
<kuid:-1:101235>	Animal_Pigs	<kuid:-1:101091>	Tree_Pine_low1_hi
<kuid:-1:101285>	Tside_Platform_seat	<kuid:-1:101092>	Tree_Pine_low2_hi
<kuid:-1:101018>	People_Platform_group	<kuid:-1:101093>	Tree_Pine_low3_hi
<kuid:-1:101099>	Tside_Pointstation	<kuid:-1:101094>	Tree_Pine_low4_hi
<kuid:-1:101100>	Building_Railshed	<kuid:-1:101095>	Tree_Pine_low1_lo
<kuid:-1:101144>	Sign_Road1	<kuid:-1:101096>	Tree_Pine_low2_lo
<kuid:-1:101175>	Sign_Road2	<kuid:-1:101097>	Tree_Pine_low3_lo
<kuid:-1:101188>	Sign_Road3	<kuid:-1:101098>	Tree_Pine_low4_lo
<kuid:-1:101189>	Sign_Road4	<kuid:-1:101108>	Tree_Pine_tall1_hi
<kuid:-1:101190>	Sign_Road5	<kuid:-1:101109>	Tree_Pine_tall2_hi
<kuid:-1:101104>	Sign_Road6	<kuid:-1:101110>	Tree_Pine_tall1_lo
<kuid:-1:101232>	Sign_Road7	<kuid:-1:101111>	Tree_Pine_tall2_lo
<kuid:-1:101263>	Sign_Road8	<kuid:-1:101079>	Tree_Small_hi
<kuid:-1:101302>	Misc_Rock_Fall	<kuid:-1:101080>	Tree_Small_lo
<kuid:-1:10132>	Building_Round_tower	<kuid:-1:101081>	Tree_Tall2_hi
<kuid:-1:101101>	Building_Shop1	<kuid:-1:101082>	Tree_Tall2_lo
<kuid:-1:101187>	Tree_Shrub1	<kuid:-1:101239>	Tside_Turntable
<kuid:-1:101197>	Tree_Shrub2	<kuid:-1:101173>	Tside_1t_Walkover
<kuid:-1:101112>	Tree_Snow_Pine1	<kuid:-1:101103>	Building_Warehouse
<kuid:-1:101113>	Tree_Snow_Pine2	<kuid:-1:101259>	Building_Warehouse2
<kuid:-1:101114>	Tree_Snow_Pine3	<kuid:-1:100830>	Building_Water_Tower
<kuid:-1:101102>	Building_Stables	<kuid:-1:101105>	Building_Water_Wheel
<kuid:-1:101083>	Tside_Station	<kuid:-1:100659>	Building_Windmill
<kuid:-1:100843>	Building_Tower_Tall	<kuid:-1:101186>	Tside_crossing_Sign
<kuid:-1:101258>	Building_Townhall	<kuid:-1:101336>	People_Yeti
<kuid:-1:100768>	Building_Town_Clock		
<kuid:-1:101208>	Vehicle_Tracktor1	Swiss	
<kuid:-1:101209>	Vehicle_Tracktor2	<kuid:-1:101349>	Vehicle_4WD_1
<kuid:-1:101086>	Tree_Fruit	<kuid:-1:101405>	Vehicle_4WD_2
<kuid:-1:101087>	Tree_Fruit_lo	<kuid:-1:101406>	Vehicle_4WD_3
<kuid:-1:101073>	Tree_Large_hi	<kuid:-1:100653>	Alps_cabin1
<kuid:-1:101074>	Tree_Large_lo	<kuid:-1:100654>	Alps_church1
<kuid:-1:101075>	Tree_Medium1_hi	<kuid:-1:100716>	cabin

<kuid:-1:101350>	Vehicle_Car_1	<kuid:-1:100705>	Tree_Snowpine4_big_I
<kuid:-1:101407>	Vehicle_Car_2	<kuid:-1:100706>	Tree_Snowpine4
<kuid:-1:101408>	Vehicle_Car_3	<kuid:-1:100707>	Tree_Snowpine4_I
<kuid:-1:101409>	Vehicle_Car_4	<kuid:-1:100708>	Tree_Snowpine4_sml
<kuid:-1:101351>	Tside_Crossing_Sign	<kuid:-1:100709>	Tree_Snowpine4_sml_I
<kuid:-1:101388>	Tside_Freight_Depot	<kuid:-1:100710>	Tree_Snowpine5_big
<kuid:-1:101384>	Tside_Rural_Depot	<kuid:-1:100711>	Tree_Snowpine5_big_I
<kuid:-1:101390>	Tside_Engine_Shed	<kuid:-1:100712>	Tree_Snowpine5
<kuid:-1:101347>	Building_Swiss_Hotel	<kuid:-1:100713>	Tree_Snowpine5_I
<kuid:-1:101398>	House_1	<kuid:-1:100714>	Tree_Snowpine5_sml
<kuid:-1:101399>	House_2	<kuid:-1:100715>	Tree_Snowpine5_sml_I
<kuid:-1:101400>	House_3	<kuid:-1:100717>	Tree_Snowpine_big_I
<kuid:-1:101401>	House_4	<kuid:-1:100718>	Tree_Snowpine
<kuid:-1:101385>	Building_Lumber_Yard	<kuid:-1:100719>	Tree_Snowpine_I
<kuid:-1:101352>	Sign_No_Entry	<kuid:-1:100720>	Tree_Snowpine_sml
<kuid:-1:101433>	Tside_Relay_Box	<kuid:-1:100721>	Tree_Snowpine_sml_I
<kuid:-1:101404>	Building_Shop	<kuid:-1:101344>	Tside_Station_Metro
<kuid:-1:101353>	Vehicle_Ski_Doo	<kuid:-1:101317>	Tside_Station_Rural
<kuid:-1:101354>	People_Skier_1	<kuid:-1:101362>	Sign_Road_Stop
<kuid:-1:101355>	People_Skier_2	<kuid:-1:101387>	Building_Storage
<kuid:-1:101356>	People_Skier_3	<kuid:-1:101363>	People_Swiss_Skier_1
<kuid:-1:101357>	People_Skier_4	<kuid:-1:101364>	People_Swiss_Skier_2
<kuid:-1:101358>	People_Snowboarder	<kuid:-1:101365>	People_Swiss_Skier_3
<kuid:-1:100670>	Tree_Snowpine_big2	<kuid:-1:100726>	House_Townhouse_1
<kuid:-1:100693>	Tree_Snowpine_big2_I	<kuid:-1:100727>	House_Townhouse_2
<kuid:-1:100694>	Tree_Snowpine2	<kuid:-1:100728>	House_Townhouse_3
<kuid:-1:100695>	Tree_Snowpine2_I	<kuid:-1:101391>	House_Townhouse_4
<kuid:-1:100696>	Tree_Snowpine2_sml		
<kuid:-1:100697>	Tree_Snowpine2_sml_I	UTC SCENARIO	
<kuid:-1:100698>	Tree_Snowpine3_big	<kuid:-10:226>	inv_slag
<kuid:-1:100699>	Tree_Snowpine3_big_I	<kuid:-13:1004330>	mine2
<kuid:-1:100722>	Tree_Snowpine3		
<kuid:-1:100701>	Tree_Snowpine3_I		
<kuid:-1:100702>	Tree_Snowpine3_sml		
<kuid:-1:100703>	Tree_Snowpine3_sml_I		
<kuid:-1:100704>	Tree_Snowpine4_big		

TRACKS

VERSION 1

Australia

<kuid:-1:100395>	Track_1t_No_gravel
<kuid:-1:100396>	1_track_dusty
<kuid:-1:100736>	2_track_dusty
<kuid:-1:100966>	4mstandard_solid
<kuid:-1:100292>	Bridge_1t_Brick
<kuid:-1:101159>	Bridge_2t_Brick
<kuid:-1:100293>	Bridge_1t_Kuranda
<kuid:-1:101160>	Bridge_1t_Kuranda
<kuid:-1:100294>	Bridge_1t_Log
<kuid:-1:101161>	Bridge_2t_Log
<kuid:-1:100398>	Tunnel_2track
<kuid:-1:100399>	Tunnel_1track
<kuid:-1:101163>	Bridge_1t_Concrete
<kuid:-1:100295>	Bridge_2t_Concrete
<kuid:-1:100397>	Bridge_1t_Steel
<kuid:-1:101162>	Bridge_2t_Steel

English Countryside

<kuid:-1:15>	1_track_wood
<kuid:-1:13228>	2_track_wood
<kuid:-1:1013>	Bridge_Brick_Steel_1t
<kuid:-1:1023>	Bridge_Brick_Steel_2t
<kuid:-1:110000>	Bridge_Stone_1t
<kuid:-1:1028>	Bridge_Stone_2t
<kuid:-1:1014>	Track_Rock_Wall_1t
<kuid:-1:101165>	Track_Rock_Wall_2t
<kuid:-1:101166>	Bridge_Stone_Pole_1t
<kuid:-1:1024>	Bridge_Stone_Pole_2t
<kuid:-1:1029>	Tunnel_Darkstone

USA

<kuid:-1:100608>	1_track_wood_US
<kuid:-1:101038>	1_track_concrete

<kuid:-1:100609>	Track_No_Gravel
<kuid:-1:110017>	2_track_concrete
<kuid:-1:110016>	Track_USA_Single
<kuid:-1:101218>	Bridge_Steel_1t
<kuid:-1:101213>	Bridge_Steel_2t
<kuid:-1:101214>	Bridge_Stone_1t
<kuid:-1:100611>	Bridge_Stone_2t
<kuid:-1:100516>	Tunnel_Stone
<kuid:-1:100610>	Bridge_Steel_USA

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:44482:22013>	1_track_greasy
<kuid:9:50001>	1_track_as49
<kuid:9:50003>	1_track_as49_aggregate
<kuid:9:50002>	2_track_as49
<kuid:9:50004>	2_track_as49_aggregate
<kuid:-10:135>	Track_Barge_Route
<kuid:-10:136>	Track_Barge_Tracks
<kuid:44482:22015>	2_track_greasy
<kuid:-10:137>	Track_Invisible
<kuid:-10:128>	Track_Platform
<kuid:-10:139>	Road_Scripted
<kuid:44482:22014>	Bridge_Steel
<kuid:44482:22012>	Tunnel_Euro
<kuid:44482:22011>	Bridge_Steel_2
<kuid:-12:368>	Tunnel_Stone_Low_Cat
<kuid:-12:367>	Tunnel_Stone_Low
<kuid:47092:32001>	Bridge_Trestlepiet1_a
<kuid:47092:32004>	Bridge_Trestlepiet1_b
<kuid:47092:32003>	Bridge_Trestlepiet1_c
<kuid:47092:32005>	Bridge_Truss_Spline
<kuid:-12:325>	Tunnel_QR_Cat
<kuid:-12:369>	Tunnel_QR
<kuid:44482:22010>	Tunnel_Euro
<kuid:11:32001>	1_track_fabricated

VERSION 1.5 - ULTIMATE TRAINZ COLLECTION

Germany

<kuid:-1:101179>	2_track_wood_damp
<kuid:-1:100673>	1_track_wood_damp
<kuid:-1:101115>	Bridge_2t_Steel_Arch
<kuid:-1:101116>	Bridge_1t_Steel_Arch
<kuid:-1:101117>	Bridge_1t_Brick
<kuid:-1:101206>	Bridge_2t_Brick
<kuid:-1:101119>	Misc_Roller_Coaster

Swiss

<kuid:-1:101340>	1_track_snow_a
<kuid:-1:100836>	1_track_snow_b
<kuid:-1:100837>	2_track_snow
<kuid:-1:101372>	Bridge_1t_Stone_Low
<kuid:-1:101342>	Bridge_1t
<kuid:-1:101370>	Bridge_1t_Stone
<kuid:-1:101374>	Bridge_2t_Stone
<kuid:-1:101379>	Tunnel_1t_Swiss
<kuid:-1:101371>	Bridge_1t_Steel_Arch
<kuid:-1:101337>	Bridge_2t_Steel_Arch
<kuid:-1:101421>	Tunnel_Swiss_Narrow

SPLINES

VERSION 1

Australa

<kuid:-1:100451>	Fence_Armco
<kuid:-1:100452>	Misc_Crop
<kuid:-1:100296>	Fence_Arcmesh
<kuid:-1:100297>	Fence_Picket
<kuid:-1:100298>	Fen_Tall_Blue
<kuid:-1:100418>	Fence_Barbed_Wire
<kuid:-1:100299>	Fence_Shearer_Yard
<kuid:-1:100457>	Power_Wire
<kuid:-1:100409>	Road_2l
<kuid:-1:100410>	Road_1l
<kuid:-1:100456>	Road_Dirt
<kuid:-1:100413>	road_oz_forbridge
<kuid:-1:100300>	Tside_2t_Platform
<kuid:-1:100301>	Tside_2t_Platform2
<kuid:-1:100412>	Bdge_Road_2l
<kuid:-1:100411>	Misc_Crop_Sugar

English Countryside

<kuid:-1:100834>	Catenary_16m
<kuid:-1:110001>	Catenary_24m_2t
<kuid:-1:110005>	Catenary_Double
<kuid:-1:110004>	Catenary_Single
<kuid:-1:1018>	Fence_BWire_3
<kuid:-1:1019>	Fence_CMesh_6
<kuid:-1:17>	Fence_Wood_4
<kuid:-1:18>	Fence_Hedge_4MR
<kuid:-1:25>	Power_Wire
<kuid:-1:1025>	Power_Wire_Big
<kuid:-1:19>	Road
<kuid:-1:20>	Road_Bridge
<kuid:-1:1017>	Road_Narrow
<kuid:-1:1058>	Road_SteelBridge
<kuid:-1:1026>	Tree_Spline

<kuid:-1:1027>

Trees_Spline_980

USA

<kuid:-12:304>	Catenary_Double
<kuid:-12:305>	Catenary_Double_3
<kuid:-12:306>	Catenary_Double_3_40m
<kuid:-12:307>	Catenary_Double_40m
<kuid:-12:308>	Catenary_Double_Centre
<kuid:-12:309>	Catenary_Double_Centre_40m
<kuid:-12:320>	Catenary_Single_16m
<kuid:-12:321>	Catenary_Single_40m
<kuid:-12:311>	Catenary_Single_Wide
<kuid:-12:312>	Catenary_Single_Wide_40m
<kuid:-12:313>	Building_Union_Shelter
<kuid:-12:220>	Misc_Wire
<kuid:-1:100586>	Fence_BWire_3
<kuid:-1:100597>	Tree_Spline_Mix_High
<kuid:-1:100612>	Tree_Spline_Mix_Low
<kuid:-1:100923>	Road_No_Traffic
<kuid:-1:100613>	Powerline
<kuid:-1:100587>	Powerline_B
<kuid:-1:100598>	Road
<kuid:-1:100614>	Road_Bridge
<kuid:-1:100588>	Road_Narrow
<kuid:-1:100589>	Road_Steel_Bridge

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:50435:34001>	Path_Boardwalk_12
<kuid:50435:34003>	Path_Boardwalk_Nosides_12
<kuid:50435:34002>	Path_Boardwalk_Sup_12
<kuid:44482:20000>	Catenary_Double_RO
<kuid:44482:20001>	Catenary_Single_RO
<kuid:3801:34001>	Path_Concrete
<kuid:3801:34002>	Path_Dirt_Track
<kuid:44482:20002>	Tside_Platform_Euro
<kuid:9:53014>	Catenary_PRR_Hard_2Level
<kuid:9:53011>	Catenary_PRR_Soft_1Level

<kuid:9:53012> Catenary_PRR_Soft_2Level
 <kuid:9:53013> Catenary_PRR_Wire
 <kuid:44482:20003> Tside_Platform_Euro
 <kuid:3801:34003> Path_Sidewalk
 <kuid:-12:40> Fence_QR_4Platform_2
 <kuid:-12:401> Fence_QR_4Platform
 <kuid:-12:678> Tside_Platform_QR
 <kuid:-12:59> Tside_Platform_QR_Ramp

VERSION 1.5 - ULTIMATE TRAINZ COLLECTION

Germany

<kuid:-1:101183> Fence_Barbed_Wire
 <kuid:-1:101184> Fence_Brickwall
 <kuid:-1:101196> Road_Carpark
 <kuid:-1:101207> Fence_Coaster
 <kuid:-1:101120> Fence_Crash_Barrier
 <kuid:-1:101121> Trees_Mixed
 <kuid:-1:103122> Trees_Mixed2
 <kuid:-1:101123> Fence_Fancy_wall
 <kuid:-1:101145> Road_Farm_Track
 <kuid:-1:101124> Fence_Farmyard
 <kuid:-1:101146> Road_Grass_Track
 <kuid:-1:101125> Fence_Hedge
 <kuid:-1:101129> Trees_Pines_low
 <kuid:-1:101130> Trees_Pines_low2
 <kuid:-1:101126> Trees_Pines
 <kuid:-1:101131> Trees_Pines1
 <kuid:-1:101132> Trees_Pines2
 <kuid:-1:101133> Trees_Pines3
 <kuid:-1:101164> Tside_Platform
 <kuid:-1:101127> Power_Wire
 <kuid:-1:101299> Bridge_Road
 <kuid:-1:101147> Road_Unmarked
 <kuid:-1:100511> Trees_Shrubs
 <kuid:-1:101148> Road_Small
 <kuid:-1:101134> Trees_Snowpines_low1

<kuid:-1:101136> Trees_Snowpines_low2
 <kuid:-1:101280> Road_Snow_Track
 <kuid:-1:101284> Fence_Stone_wall
 <kuid:-1:101185> Fence_Tall_Wood
 <kuid:-1:101128> Fence_Wire
 Swiss
 <kuid:-1:101380> Fence_Barbed_Wire
 <kuid:-1:101381> Fence_Wood_Panel
 <kuid:-1:101383> Power_Wire
 <kuid:-1:101376> Road
 <kuid:-1:101382> Fence_Ski
 <kuid:-1:101377> Road_Dirt_Track
 <kuid:-1:101378> Road_Grass_Track

TRACKSIDE

VERSION 1

Australia

<kuid:-1:110011>	dwarf
<kuid:-1:100633>	lever
<kuid:-1:101152>	rail_end
<kuid:-1:100899>	speed_100
<kuid:-1:100900>	speed_20
<kuid:-1:100901>	speed_30
<kuid:-1:100902>	speed_40
<kuid:-1:100903>	speed_50
<kuid:-1:100904>	speed_60
<kuid:-1:100905>	speed_70
<kuid:-1:100906>	speed_80
<kuid:-1:100907>	speed_90
<kuid:-1:101142>	track_ends

English Countryside

<kuid:-1:100968>	Tside_Speed_10
<kuid:-1:100997>	Tside_Speed_100
<kuid:-1:100998>	Tside_Speed_110
<kuid:-1:100999>	Tside_Speed_120
<kuid:-1:101000>	Tside_Speed_130
<kuid:-1:101006>	Tside_Speed_140
<kuid:-1:101001>	Tside_Speed_15
<kuid:-1:101007>	Tside_Speed_150
<kuid:-1:101008>	Tside_Speed_160
<kuid:-1:101009>	Tside_Speed_170
<kuid:-1:101010>	Tside_Speed_180
<kuid:-1:101002>	Tside_Speed_20
<kuid:-1:101003>	Tside_Speed_25
<kuid:-1:101004>	Tside_Speed_35
<kuid:-1:101011>	Tside_Speed_40
<kuid:-1:101012>	Tside_Speed_45
<kuid:-1:101013>	Tside_Speed_50
<kuid:-1:100893>	Tside_Speed_60
<kuid:-1:101015>	Tside_Speed_65

<kuid:-1:101016>	Tside_Speed_70
<kuid:-1:101019>	Tside_Speed_90
<kuid:-1:100553>	lever
<kuid:-1:110006>	Tside_Rail_End
<kuid:-1:101140>	Tside_Sign_Track_Ends

USA

<kuid:-1:100874>	Sign_Speed_10
<kuid:-1:100882>	Sign_Speed_100
<kuid:-1:100883>	Sign_Speed_110
<kuid:-1:100881>	Sign_Speed_120
<kuid:-1:100884>	Sign_Speed_130
<kuid:-1:101026>	Sign_Speed_140
<kuid:-1:100885>	Sign_Speed_150
<kuid:-1:101028>	Sign_Speed_160
<kuid:-1:101029>	Sign_Speed_170
<kuid:-1:101030>	Sign_Speed_180
<kuid:-1:100886>	Sign_Speed_20
<kuid:-1:100887>	Sign_Speed_25
<kuid:-1:100888>	Sign_Speed_30
<kuid:-1:100889>	Sign_Speed_35
<kuid:-1:100890>	Sign_Speed_40
<kuid:-1:100891>	Sign_Speed_45
<kuid:-1:100892>	Sign_Speed_50
<kuid:-1:101045>	Sign_Speed_55
<kuid:-1:100894>	Sign_Speed_60
<kuid:-1:100895>	Sign_Speed_65
<kuid:-1:100896>	Sign_Speed_70
<kuid:-1:100897>	Sign_Speed_79
<kuid:-1:101031>	Sign_Speed_80
<kuid:-1:100898>	Sign_Speed_90
<kuid:-1:100913>	Signal_Dwarf
<kuid:-1:100634>	Tside_Lever
<kuid:-1:110007>	Signal_Rail_End_USA
<kuid:-1:101143>	Signal_Track_Ends
<kuid:-1:100854>	Sign_Yard_Limit

VERSION 1.3 - SERVICE PACK 3 (SP3)

		<kuid:-12:15660>	Signal_BR_8
<kuid:3801:23001>	Sign_Bboard_10	<kuid:-12:15661>	Signal_BR_8RH
<kuid:3801:23019>	Sign_Bboard_100	<kuid:-12:300>	Misc_Dwarf_LHS
<kuid:3801:23020>	Sign_Bboard_105	<kuid:-12:301>	Misc_Dwarf_RHS
<kuid:3801:23021>	Sign_Bboard_110	<kuid:-12:292>	Gantry_L
<kuid:3801:23022>	Sign_Bboard_115	<kuid:-12:293>	Gantry_R
<kuid:3801:23023>	Sign_Bboard_120	<kuid:9:55001>	Signal_PKP_Lightpost
<kuid:3801:23024>	Sign_Bboard_125	<kuid:9:55002>	Signal_PKP_Lightpost_Long
<kuid:3801:23025>	Sign_Bboard_135	<kuid:9:55003>	Signal_PKP_Lightpost_Side
<kuid:3801:23027>	Sign_Bboard_138	<kuid:9:55004>	Signal_PKP_Lightpost_S/L
<kuid:3801:23002>	Sign_Bboard_15	<kuid:46415:28100>	Sign_Beacons_ATC_QR_2
<kuid:3801:23003>	Sign_Bboard_35	<kuid:46415:28105>	Sign_Beacons_ATC_QR_3
<kuid:3801:23007>	Sign_Bboard_40	<kuid:46415:23502>	Sign_KM_Post_QR
<kuid:3801:23008>	Sign_Bboard_45	<kuid:46415:23500>	Sign_Limit_Of_Shunt_QR
<kuid:3801:23000>	Sign_Bboard_5	<kuid:46415:28150>	Tside_Points_Clip_QR
<kuid:3801:23009>	Sign_Bboard_50	<kuid:46415:28155>	Tside_Point_Motor_QR
<kuid:3801:23010>	Sign_Bboard_55	<kuid:46415:23185>	Sign_Speed_RHS_120_50L_QR
<kuid:3801:23011>	Sign_Bboard_60	<kuid:46415:23186>	Sign_Speed_RHS_120_50R_QR
<kuid:3801:23012>	Sign_Bboard_70	<kuid:46415:23110>	Sign_Speed_RHS_50_25L_QR
<kuid:3801:23014>	Sign_Bboard_75	<kuid:46415:23111>	Sign_Speed_RHS_50_25R_QR
<kuid:3801:23015>	Sign_Bboard_80	<kuid:46415:23120>	Sign_Speed_RHS_60_25L_QR
<kuid:3801:23016>	Sign_Bboard_85	<kuid:46415:23121>	Sign_Speed_RHS_60_25R_QR
<kuid:3801:23017>	Sign_Bboard_90	<kuid:46415:23130>	Sign_Speed_RHS_70_25L_QR
<kuid:3801:23018>	Sign_Bboard_95	<kuid:46415:23131>	Sign_Speed_RHS_70_25R_QR
<kuid:-12:5601>	Signal_BR_1	<kuid:46415:23140>	Sign_Speed_RHS_80_25L_QR
<kuid:-12:5801>	Signal_BR_1RH	<kuid:46415:23142>	Sign_Speed_RHS_80_25LR_QR
<kuid:-12:5603>	Signal_BR_2	<kuid:46415:23141>	Sign_Speed_RHS_80_25R_QR
<kuid:-12:5803>	Signal_BR_2RH	<kuid:46415:23145>	Sign_Speed_RHS_80_50L_QR
<kuid:-12:5656>	Signal_BR_3	<kuid:46415:23146>	Sign_Speed_RHS_80_50R_QR
<kuid:-12:5856>	Signal_BR_3RH	<kuid:46415:23010>	Sign_Speed_RHS_07k_QR
<kuid:-1:575>	Signal_BR_4	<kuid:46415:23065>	Sign_Speed_RHS_100k_QR
<kuid:-1:100519>	Signal_BR_4RH	<kuid:46415:23015>	Sign_Speed_RHS_10k_QR
<kuid:-12:5658>	Signal_BR_5	<kuid:46415:23070>	Sign_Speed_RHS_110k_QR
<kuid:-12:5858>	Signal_BR_5RH	<kuid:46415:23075>	Sign_Speed_RHS_120k_QR
<kuid:-12:5660>	Signal_BR_7	<kuid:46415:23020>	Sign_Speed_RHS_15k_QR
<kuid:-12:5860>	Signal_BR_7RH	<kuid:46415:23025>	Sign_Speed_RHS_25k_QR

<kuid:46415:23030>	Sign_Speed_RHS_30k_QR	<kuid:46415:23346>	Sign_Speed_LHS_80_50R_QR
<kuid:46415:23035>	Sign_Speed_RHS_40k_QR	<kuid:46415:23210>	Sign_Speed_LHS_07k_QR
<kuid:46415:23040>	Sign_Speed_RHS_50k_QR	<kuid:46415:23365>	Sign_Speed_LHS_100k_QR
<kuid:46415:23045>	Sign_Speed_RHS_60k_QR	<kuid:46415:23315>	Sign_Speed_LHS_10k_QR
<kuid:46415:23050>	Sign_Speed_RHS_70k_QR	<kuid:46415:23370>	Sign_Speed_LHS_110k_QR
<kuid:46415:23055>	Sign_Speed_RHS_80k_QR	<kuid:46415:23375>	Sign_Speed_LHS_120k_QR
<kuid:46415:23060>	Sign_Speed_RHS_90k_QR	<kuid:46415:23220>	Sign_Speed_LHS_15k_QR
<kuid:46415:23504>	Tside_Whistle_Board_QR	<kuid:46415:23225>	Sign_Speed_LHS_25k_QR
<kuid:-12:1201>	Signal_QR_01	<kuid:46415:23230>	Sign_Speed_LHS_30k_QR
<kuid:-12:1401>	Signal_QR_01RH	<kuid:46415:23235>	Sign_Speed_LHS_40k_QR
<kuid:-12:1303>	Signal_QR_02	<kuid:46415:23240>	Sign_Speed_LHS_50k_QR
<kuid:-12:1403>	Signal_QR_02RH	<kuid:46415:23245>	Sign_Speed_LHS_60k_QR
<kuid:-12:3456>	Signal_QR_03	<kuid:46415:23250>	Sign_Speed_LHS_70k_QR
<kuid:-12:3457>	Signal_QR_03RH	<kuid:46415:23255>	Sign_Speed_LHS_80k_QR
<kuid:-1:100630>	Signal_QR_04	<kuid:46415:23260>	Sign_Speed_LHS_90k_QR
<kuid:-12:100630>	Signal_QR_04RH	<kuid:-12:21821>	Signal_2_001
<kuid:-12:3458>	Signal_QR_05	<kuid:-12:21802>	Signal_2_002
<kuid:-12:3459>	Signal_QR_05RH	<kuid:-12:21803>	Signal_2_003
<kuid:-12:210>	Signal_QR_06	<kuid:-12:207>	Signal_2_004
<kuid:-12:209>	Signal_QR_06b	<kuid:-12:208>	Signal_2_005
<kuid:-12:3460>	Signal_QR_07	<kuid:-12:21861>	Signal_2_L001
<kuid:-12:3461>	Signal_QR_07RH	<kuid:-12:21862>	Signal_2_L002
<kuid:-12:13461>	Signal_QR_08	<kuid:-12:21863>	Signal_2_L003
<kuid:-12:13460>	Signal_QR_08RH	<kuid:-12:21701>	Signal_001
<kuid:46415:23385>	Sign_Speed_LHS_120_50L_QR	<kuid:-12:21702>	Signal_002
<kuid:46415:23386>	Sign_Speed_LHS_120_50R_QR	<kuid:-12:21703>	Signal_003
<kuid:46415:23310>	Sign_Speed_LHS_50_25L_QR	<kuid:-1:110012>	Signal_004
<kuid:46415:23311>	Sign_Speed_LHS_50_25R_QR	<kuid:-12:315>	Signal_005
<kuid:46415:23320>	Sign_Speed_LHS_60_25L_QR	<kuid:-12:291>	Signal_G_005
<kuid:46415:23321>	Sign_Speed_LHS_60_25R_QR	<kuid:-12:21201>	Signal_L001
<kuid:46415:23330>	Sign_Speed_LHS_70_25L_QR	<kuid:-12:22702>	Signal_L002
<kuid:46415:23331>	Sign_Speed_LHS_70_25R_QR	<kuid:-12:25703>	Signal_L003
<kuid:46415:23340>	ign_Speed_LHS_80_25L_QR	<kuid:44482:23006>	Signal_3_Gantry
<kuid:46415:23342>	Sign_Speed_LHS_80_25LR_QR	<kuid:44482:23004>	Signal_3_Rolft
<kuid:46415:23341>	Sign_Speed_LHS_80_25R_QR	<kuid:44482:23005>	Signal_3_Rorgt
<kuid:46415:23345>	Sign_Speed_LHS_80_50L_QR	<kuid:44482:23061>	Sign_Speed_100

<kuid:44482:23008>	Sign_Speed_100_Advance	<kuid:44482:23066>	Sign_Speed_190
<kuid:44482:23062>	Sign_Speed_105	<kuid:44482:23027>	Sign_Speed_190_Advance
<kuid:44482:23009>	Sign_Speed_105_Advance	<kuid:44482:23065>	Sign_Speed_195
<kuid:44482:23083>	Sign_Speed_110	<kuid:44482:23028>	Sign_Speed_195_Advance
<kuid:44482:23010>	Sign_Speed_110_Advance	<kuid:44482:23064>	Sign_Speed_20
<kuid:44482:23082>	Sign_Speed_115	<kuid:44482:23063>	Sign_Speed_200
<kuid:44482:23011>	Sign_Speed_115_Advance	<kuid:44482:23045>	Sign_Speed_200_Advance
<kuid:44482:23081>	Sign_Speed_120	<kuid:44482:23029>	Sign_Speed_20_Advance
<kuid:44482:23012>	Sign_Speed_120_Advance	<kuid:44482:23060>	Sign_Speed_25
<kuid:44482:23080>	Sign_Speed_125	<kuid:44482:23030>	Sign_Speed_25_Advance
<kuid:44482:23013>	Sign_Speed_125_Advance	<kuid:44482:23059>	Sign_Speed_30
<kuid:44482:23079>	Sign_Speed_130	<kuid:44482:23031>	Sign_Speed_30_Advance
<kuid:44482:23014>	Sign_Speed_130_Advance	<kuid:44482:23058>	Sign_Speed_35
<kuid:44482:23078>	Sign_Speed_135	<kuid:44482:23032>	Sign_Speed_35_Advance
<kuid:44482:23015>	Sign_Speed_135_Advance	<kuid:44482:23057>	Sign_Speed_40
<kuid:44482:23077>	Sign_Speed_140	<kuid:44482:23033>	Sign_Speed_40_Advance
<kuid:44482:23016>	Sign_Speed_140_Advance	<kuid:44482:23056>	Sign_Speed_45
<kuid:44482:23076>	Sign_Speed_145	<kuid:44482:23034>	Sign_Speed_45_Advance
<kuid:44482:23017>	Sign_Speed_145_Advance	<kuid:44482:23055>	Sign_Speed_50
<kuid:44482:23075>	Sign_Speed_15	<kuid:44482:23035>	Sign_Speed_50_Advance
<kuid:44482:23074>	Sign_Speed_150	<kuid:44482:23054>	Sign_Speed_55
<kuid:44482:23018>	Sign_Speed_150_Advance	<kuid:44482:23036>	Sign_Speed_55_Advance
<kuid:44482:23073>	Sign_Speed_155	<kuid:44482:23053>	Sign_Speed_60
<kuid:44482:23019>	Sign_Speed_155_Advance	<kuid:44482:23037>	Sign_Speed_60_Advance
<kuid:44482:23020>	Sign_Speed_15_Advance	<kuid:44482:23052>	Sign_Speed_65
<kuid:44482:23072>	Sign_Speed_160	<kuid:44482:23038>	Sign_Speed_65_Advance
<kuid:44482:23021>	Sign_Speed_160_Advance	<kuid:44482:23051>	Sign_Speed_70
<kuid:44482:23071>	Sign_Speed_165	<kuid:44482:23039>	Sign_Speed_70_Advance
<kuid:44482:23022>	Sign_Speed_165_Advance	<kuid:44482:23050>	Sign_Speed_75
<kuid:44482:23070>	Sign_Speed_170_Advance	<kuid:44482:23040>	Sign_Speed_75_Advance
<kuid:44482:23069>	Sign_Speed_175	<kuid:44482:23049>	Sign_Speed_80
<kuid:44482:23024>	Sign_Speed_175_Advance	<kuid:44482:23041>	Sign_Speed_80_Advance
<kuid:44482:23068>	Sign_Speed_180	<kuid:44482:23048>	Sign_Speed_85
<kuid:44482:23025>	Sign_Speed_180_Advance	<kuid:44482:23042>	Sign_Speed_85_Advance
<kuid:44482:23067>	Sign_Speed_185	<kuid:44482:23047>	Sign_Speed_90
<kuid:44482:23026>	Sign_Speed_185_Advance	<kuid:44482:23043>	Sign_Speed_90_Advance

<kuid:44482:23046>	Sign_Speed_95	UTC scenarios	
<kuid:44482:23044>	Sign_Speed_95_Advance	<kuid:523:100883>	speed_110
<kuid:-12:22901>	Signal_USA_2_01	<kuid:523:100881>	speed_120
<kuid:-12:22902>	Signal_USA_2_02	<kuid:523:100884>	speed_130
<kuid:-12:22904>	Signal_USA_2_03	<kuid:523:101026>	speed_140
<kuid:-12:303>	Signal_USA_2_04	<kuid:523:100885>	speed_15
<kuid:-12:302>	Signal_USA_2_05	<kuid:523:101027>	speed_150
<kuid:-12:22961>	Signal_USA_2_L01	<kuid:523:101028>	speed_160
<kuid:-12:22962>	Signal_USA_2_L02	<kuid:523:100886>	speed_20
<kuid:-12:22964>	Signal_USA_2_L03	<kuid:523:100887>	speed_25
<kuid:-12:21901>	Signal_USA_01	<kuid:523:100888>	speed_30
<kuid:-12:21902>	Signal_USA_02	<kuid:523:100889>	speed_35
<kuid:-12:21913>	Signal_USA_03	<kuid:523:100890>	speed_40
<kuid:-1:100880>	Signal_USA_04	<kuid:523:100891>	speed_45
<kuid:-12:214>	Signal_USA_05	<kuid:523:100892>	speed_50
<kuid:-12:290>	Signal_USA_G_05	<kuid:523:101045>	speed_55
<kuid:-12:21961>	Signal_USA_L01	<kuid:523:100894>	speed_60
<kuid:-12:11902>	Signal_USA_L02	<kuid:523:100895>	speed_65
<kuid:-12:21993>	Signal_USA_L03	<kuid:523:100896>	speed_70
<kuid:3801:23028>	Sign_Warning_20	<kuid:523:101031>	speed_80
<kuid:3801:23029>	Sign_Warning_30	<kuid:523:100898>	speed_90
<kuid:3801:23030>	Sign_Warning_40		
<kuid:3801:23031>	Sign_Warning_50	<kuid:-12:100128>	09
<kuid:3801:23032>	Sign_Warning_60	<kuid:-12:100129>	09
<kuid:3801:23033>	Sign_Warning_70	<kuid:-12:100136>	09
<kuid:3801:23034>	Sign_Warning_80	<kuid:523:23027>	Warn 10
		<kuid:523:23028>	Warn 15
		<kuid:523:23029>	Warn 25
		<kuid:523:23030>	Warn 35
		<kuid:523:23031>	Warn 45
		<kuid:523:100874>	speed_10
		<kuid:523:100882>	speed_100
VERSION 1.5 - ULTIMATE TRAINZ COLLECTION			
Germany			
<kuid:-1:101182>	Tside_Railend		
<kuid:-1:101240>	lever		
Swiss			
<kuid:-1:101359>	speed_100		
<kuid:-1:101360>	speed_50		
<kuid:-1:101361>	speed_60		

ENVIROMENT

VERSION 1

Australia

<kuid:-1:110014> noClouds

<kuid:-1:110015> Water_muddy

English Countryside

<kuid:-1:8000> MediumClouds

<kuid:-1:8002> MoreClouds

<kuid:-1:8001> VeryCloudy

<kuid:-1:110013> VeryVeryCloudy

<kuid:-1:8008> Water1

<kuid:-1:8009> Water2

USA

<kuid:-1:101310> Clouds_USA

<kuid:-1:101293> Water1_USA

VERSION 1.3 - SERVICE PACK 3 (SP3)

<kuid:-12:777> QLD_sky

VERSION 1.5 - ULTIMATE TRAINZ COLLECTION

<kuid:-1:100969> ger_Clouds1

<kuid:-1:100671> ger_Water1

PAINTSHED

New in Ultimate Trainz			
kuid	name	skin kuid	simple stripe template kuid
-13:131	ATSF_baggage	-13:131000	-13:131001
-13:132	ATSF_dome	-13:132000	-13:132001
-13:133	ATSF_uni	-13:133000	-13:133001
-13:134	class218	-13:134000	-13:134001
-13:135	class34-200	-13:135000	-13:135001
-13:136	dl531	-13:136000	-13:136001
-13:137	EF81	-13:137000	-13:137001
-13:138	eurofima	-13:138000	-13:138001
-13:139	gg1	-13:139000	-13:139001
-13:140	gm	-13:140000	-13:140001
-13:141	gp38_hi_nodyn	-13:141000	-13:141001
-13:142	gp38_lo_dyn	-13:142000	-13:142001
-13:143	gp38_lo_nodyn	-13:143000	-13:143001
-13:182	MK1_BR	-13:182000	-13:182001
-13:144	Mk2_BR	-13:144000	-13:144001
-13:183	MK3_BR	-13:183000	-13:183001
-13:145	TGV	-13:145000	-13:145001
-13:146	TGV_unit1	-13:146000	-13:146001
-13:147	TGV_unit2	-13:147000	-13:147001
-13:1465	TGV_unit3	-13:146500	-13:146501

Paintshed Service Pack 1 (included in UTC)					
kuid	name	skin kuid	simple stripe template kuid	Checkers template kuid	swoopy template kuid
-10:179	1044	-13:148000	-13:148001		
-10:195	3dtcar	-13:149000	-13:149001	-13:149002	-13:149003
-10:196	40ft_boxcar	-13:150000	-13:150001	-13:150002	-13:150003
-10:197	50ft_2door_boxcar	-13:151000	-13:151001	-13:151002	-13:151003
-10:198	50ft_gondola	-13:152000	-13:152001	-13:152003	-13:152003
-10:199	54ft_covered_hopper	-13:153000	-13:153001	-13:153002	-13:153003
-10:180	6e1	-13:154000	-13:154001		
-10:181	alco	-13:155000	-13:155001		
-10:200	Bay_window_caboose	-13:156000	-13:156001	-13:156002	-13:156003
-10:182	bb15000	-13:157000	-13:157001		
-10:183	Class_2100	-13:158000	-13:158001		
-10:184	class37	-13:159000	-13:159001		
-10:185	class43	-13:160000	-13:160001		
-10:173	class50	-13:161000	-13:161001	-13:161002	-13:161003
-10:186	class55	-13:162000	-13:162001		
-10:187	dl500_double	-13:163000	-13:163001		
-10:188	dl500_single	-13:164000	-13:164001		
-10:202	eaos	-13:165000	-13:165001	-13:165002	-13:165003

Continues next page...

-10:189	f7	-13:166000	-13:166001		
-10:190	f7b	-13:167000	-13:167001		
-10:203	faoos	-13:168000	-13:168001	-13:168002	-13:168003
-10:204	flatcar	-13:169000	-13:169001	-13:169002	-13:169003
-10:205	gbs	-13:170000	-13:170001	-13:170002	-13:170003
-10:206	gp38	-13:171000	-13:171001	-13:171002	-13:171003
-10:207	gs	-13:172000	-13:172001	-13:172003	-13:172003
-10:208	nohab	-13:173000	-13:173001	-13:173002	-13:173003
-10:191	Rc4	-13:174000	-13:174001		
-10:209	rils	-13:175000	-13:175001	-13:175002	-13:175003
-10:192	sd40_2	-13:176000	-13:176001		
-10:210	sssy716container	-13:177000	-13:177001	-13:177002	-13:177003
-10:211	sw7	-13:178000	-13:178001	-13:178002	-13:178003
-10:212	uh	-13:179000	-13:179001	-13:179002	-13:179003
-10:193	v200	-13:180000	-13:180001		
-10:194	vrs	-13:181000	-13:181001		