

A Guide to Error Fixing Assets for TRS2006

© Paul Hobbs

(Updates available from <http://www.rail-sim.co.uk>
and <http://www.44090digitalmodels.co.uk/>)

Index

- Foreward 3
 - So why do my Assets have errors then?..... 3
 - Why Should Errors be Fixed? 3
 - What is Involved in Fixing Assets?..... 3
 - How Do I Find the Faults?..... 3
- The Error Messages 5
 - Illegal Tags..... 5
 - Missing Tags..... 9
 - Missing Mesh Files..... 10
 - Could not Load Texture..... 12
 - Texture is not a power of 2..... 14
 - Primary and alpha texture are not the same size..... 14
 - The asset specified in 'product-kuid' is not of the correct type 15
 - Unable to load animation file..... 16
- Postscript..... 17

Foreward

So why do my Assets have errors then?

First, a short history of the development of Trainz is necessary. Over the years there have been many different versions of the Trainz program and the assets available were based on the best information available at the time, or in some cases, as Trainz didn't complain the author assumed that the asset was OK. In fact what was happening was that earlier versions of Trainz were working around problems, which TRS2006 does not. Hence assets that apparently worked fine in TRS2004 are being shown as 'faulty' in TRS2006 – the assets always were broken but TRS2004 didn't tell you.

Why Should Errors be Fixed?

In many cases you don't need to fix 'faulty' assets, as they will continue to work in TRS2006. In some cases though a fault will prevent an asset from showing up in Trainz, so if something you have downloaded won't show up in Surveyor the first thing to do is fire up CMP and see if errors are present.

Additionally human nature being what it is if something is described as 'faulty' then it needs fixing. It has been reported though that using error free content can actually improve the stability of Trainz and increase frame rates so it would seem to be worthwhile.

What is Involved in Fixing Assets?

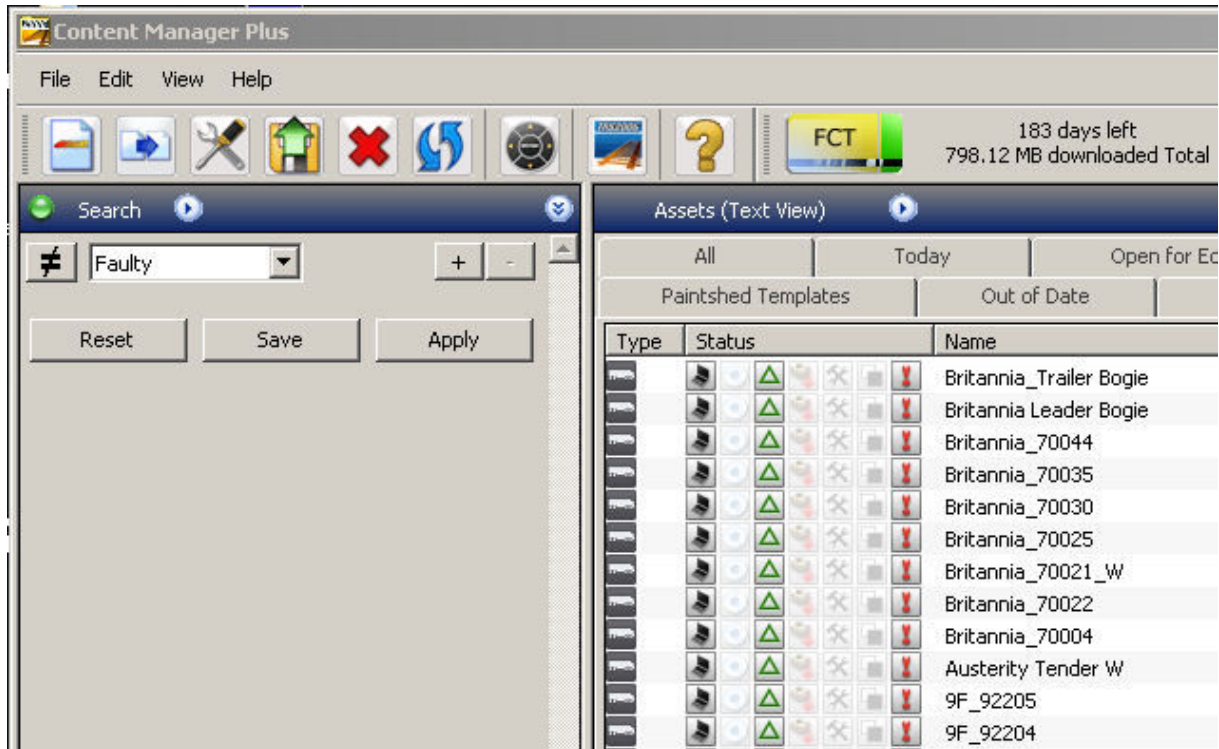
We'll be looking at correcting some common faults later, but it is something that anybody can do. Each Trainz asset has a 'config' text file called (surprise surprise) 'config.txt' (though your PC may well hide the 'txt' file extension in which case it appears as 'config' in an Explorer window), and this is where most of the faults are to be found.

CMP is also very helpful for analysing assets and telling you what is wrong with them. A couple of tips though – firstly CMP has a habit of checking the DLS every so often and opens a dialog box saying it is updating the database or verifying your registration code. When the DLS is busy it can take a long time for these messages to disappear if at all. Simply hit the ESC key to close the dialog box(es) and then you can carry on working.

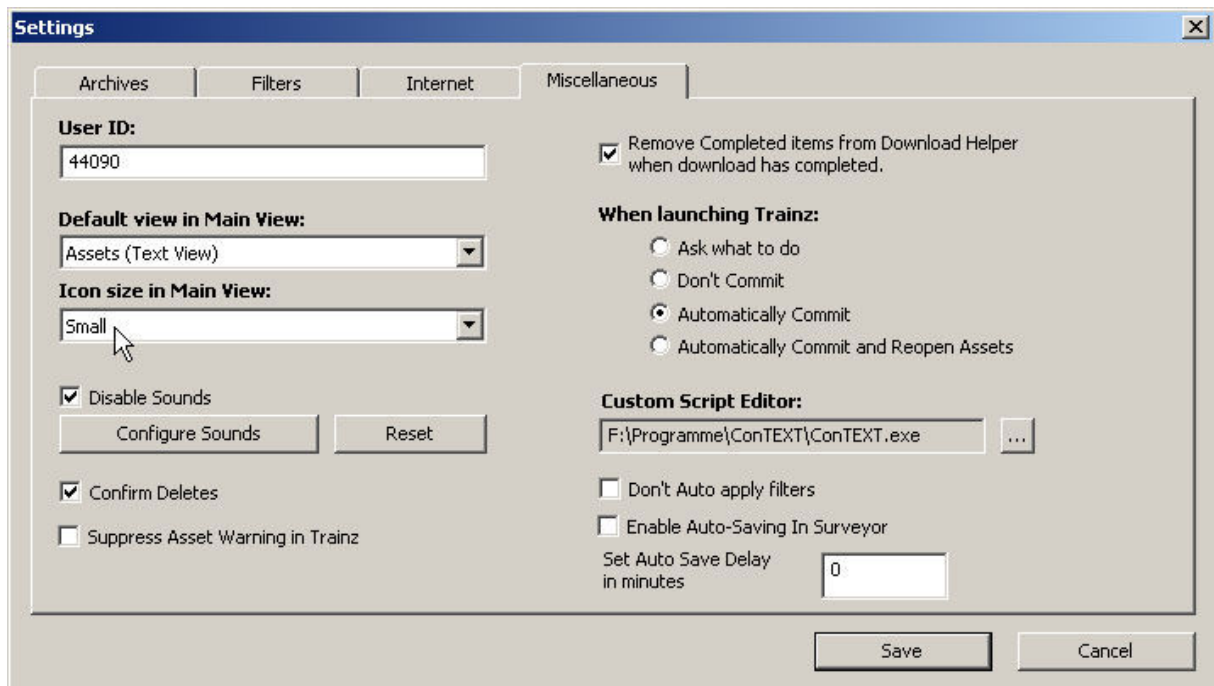
Secondly there are two options available for editing the config files via CMP – always use 'open in Explorer' and not 'open in CCP'. All of the files you need to edit are in the 'local' folder inside the TRS2006 folder, but you should never try and edit anything in the 'local' folder, instead open the asset for editing via CMP.

How Do I Find the Faults?

Start CMP and in the Search Filter choose 'Faulty' like this:



The '!' icon denotes that the asset(s) have faults. Note that this only appears if the 'Icon size in Main View' is set to 'small' and the 'default view in main view' is set to 'Assets (Text View)' in the 'Miscellaneous' tab of the 'Settings' dialog box.



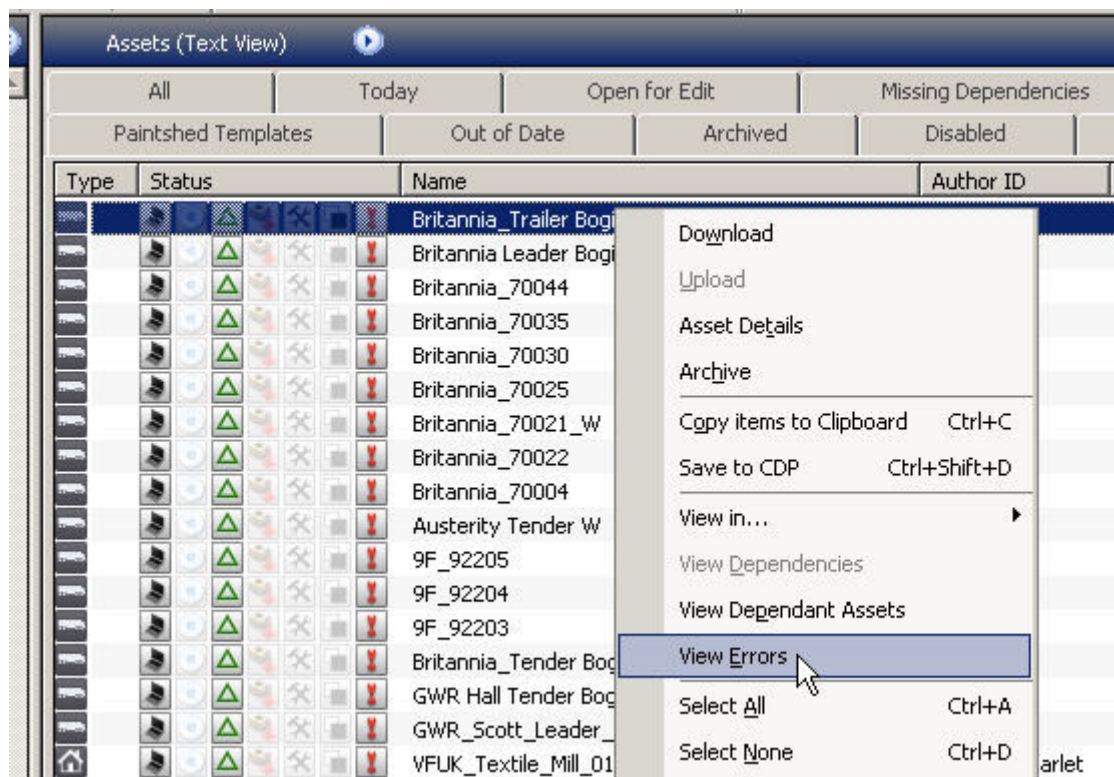
It is also worth noting that CMP itself includes help documentation (found under the 'Help' menu at the top of the window), and the main TRS2006 manual (click on the 'Manual' button on the TRS2006 launcher screen) has a fuller description of CMP.

The Error Messages

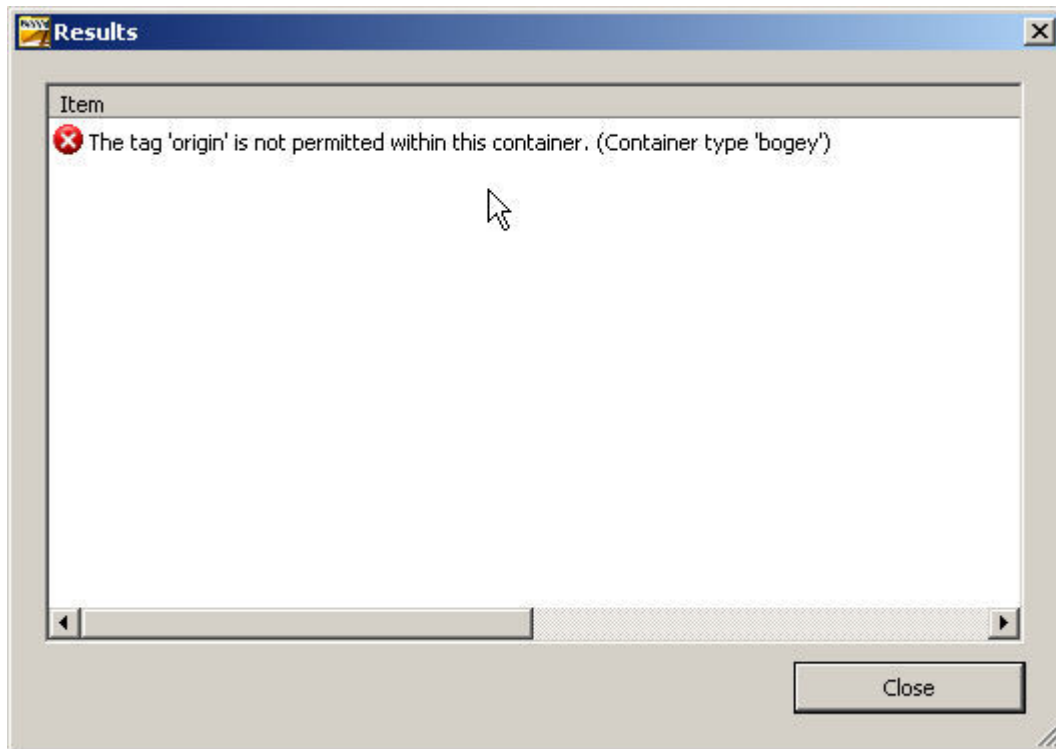
Illegal Tags

The first faulty asset listed above, 'Britannia_trailer_bogey' demonstrates a very common error, an illegal tag. Everything that appears at the start of a line in a config file is a 'tag', a description of what follows the tag itself. Trainz versions prior to TRS2006 simply ignored anything they couldn't understand so tags got used for adding comments or special information that could be used by scripts. Sometimes the tag was even wrongly spelled which causes problems for TRS2006.

So, lets fix this error. Right click the mouse button over the faulty asset and choose 'View Errors' like this:

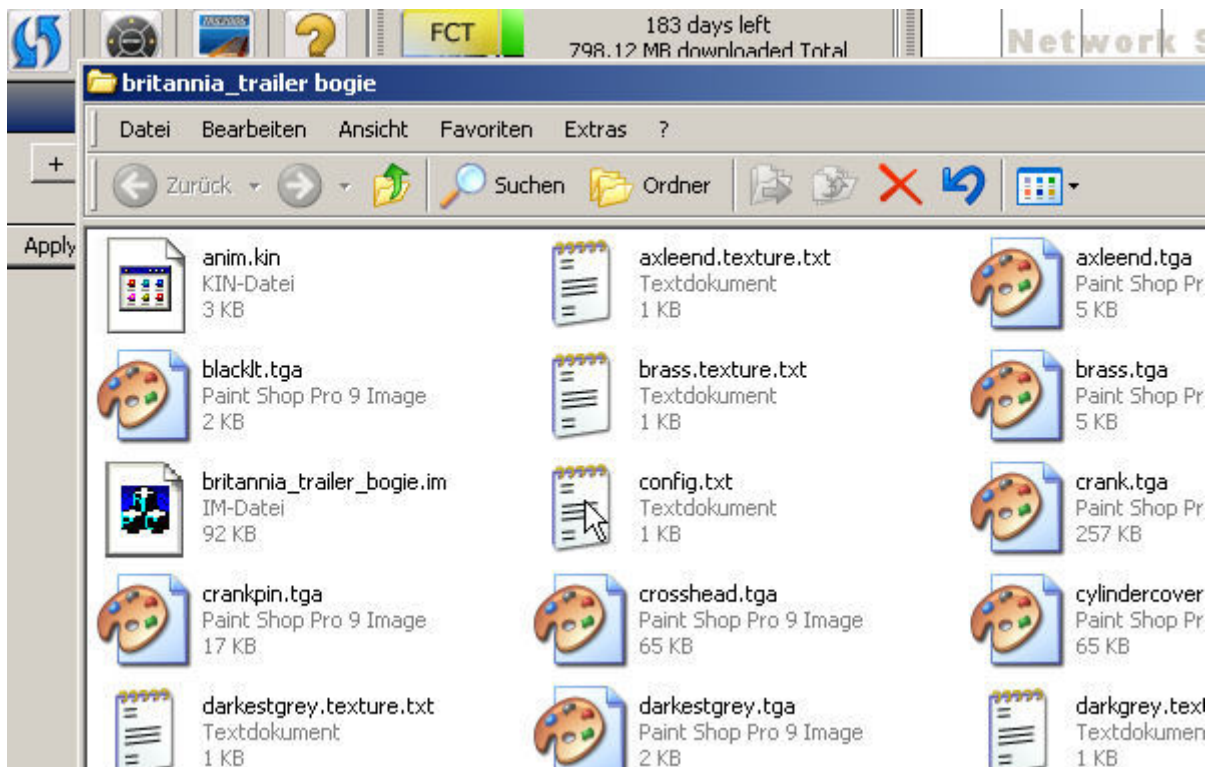


An information window opens:



Now we have to open the asset for editing to correct it. Close this window and select the asset again with the right mouse button. This time choose 'Edit in Explorer'.

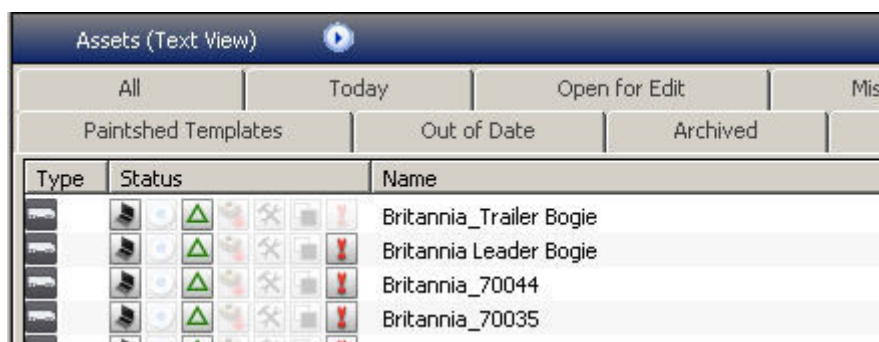
An Explorer window will now open showing the files belonging to this asset.



'config.txt' or 'config' is what we need to change, so double click on it to open it in your usual text editor. Don't use Word or anything too fancy as some programs can add extra invisible characters to the file that can trip TRS2006 up. If in doubt use the standard Windows text editor.

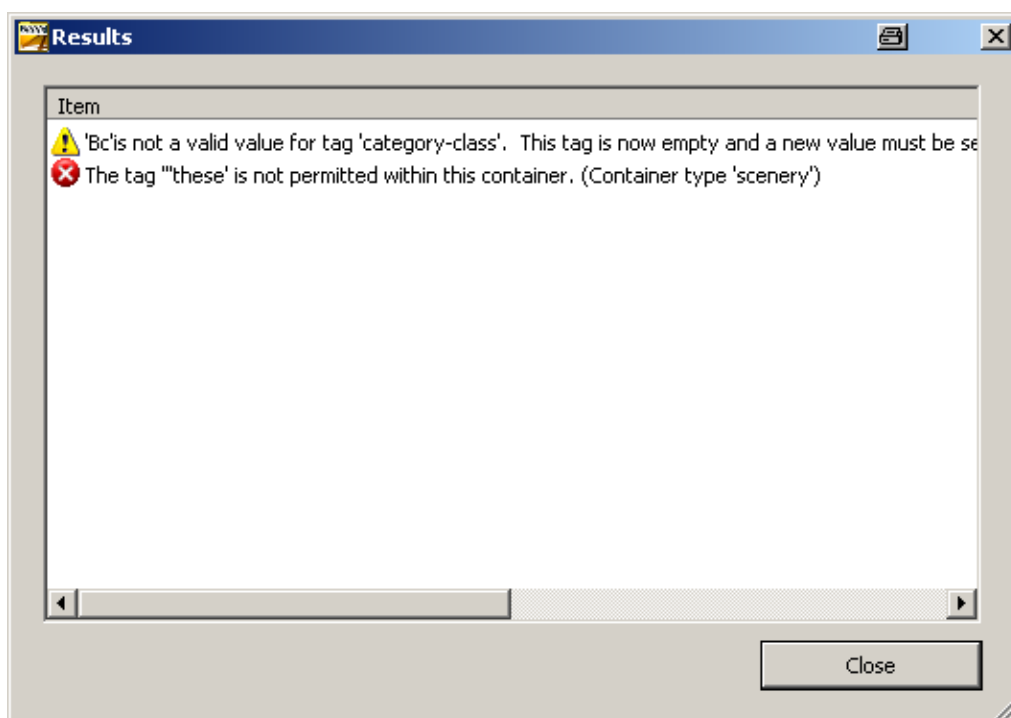
Now delete the line that CMP complained about and resave the file. Select the asset with the left mouse button in the main CMP window if it is not already selected and hit CTRL and M together to commit the asset back into the database. This simply means that CMP will check the asset again to see if any errors are present and will make it available in Surveyor. Assets that are open for editing will not show up in Surveyor unless certain options are set in the 'Settings' dialog box.

Once the asset has been committed the '!' icon should be greyed out denoting that the asset is now error free.



That wasn't too bad was it?

Another case of this error could be that the 'description' tag in the config includes extra "" characters. The first "" defines the start of the description text and the second one marks the end, so if there is an extra "" somewhere in the middle TRS attempts to interpret the rest of the description as tags.



If you get error messages that a tag is not allowed have a good look at the description tag and replace any "" characters in the middle of the text with a single quote character.

Here's an example of a config with this error:

```
Description "A Lunch Break"  
"these "dude's are a product  
of my imagination + AURAN's mesh's / textures + a little bit  
of Gmax from me ..... only available for regestered  
users of Auran TRAINZ Service pack Ect.....on the  
download station no where else these boyz work in UTC  
TRS.....have fun .....Dreddman '"
```

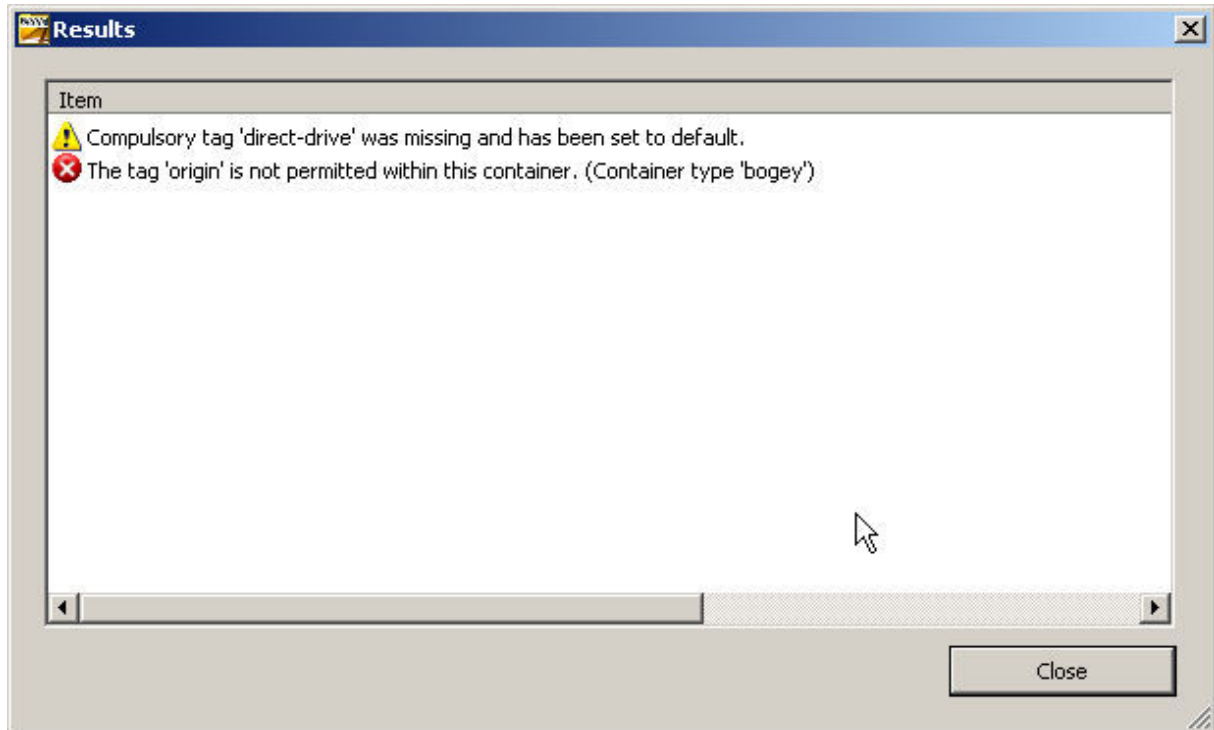
The description should read:

```
Description "A Lunch Break. These dude's are a product of my  
imagination + AURAN's mesh's / textures + a little bit of Gmax  
from me ..... only available for regestered users of  
Auran TRAINZ Service pack Ect.....on the download station  
no where else these boyz work in UTC TRS.....have fun  
.....Dreddman '"
```

All on one line of course.

Missing Tags

The next faulty asset (don't worry, I'm not going through the whole list and there is no significance that the assets are not my own, it's just what is in my CMP at the moment) has a missing tag that needs to be added:



The red icon with the cross denotes a real error and should be fixed in the same way as the previous example. The yellow icon denotes a warning that will not be displayed once the 'red' errors have been fixed, but we may as well fix this as well, so open the config file for editing as before and add the line `'direct-drive 0'`.

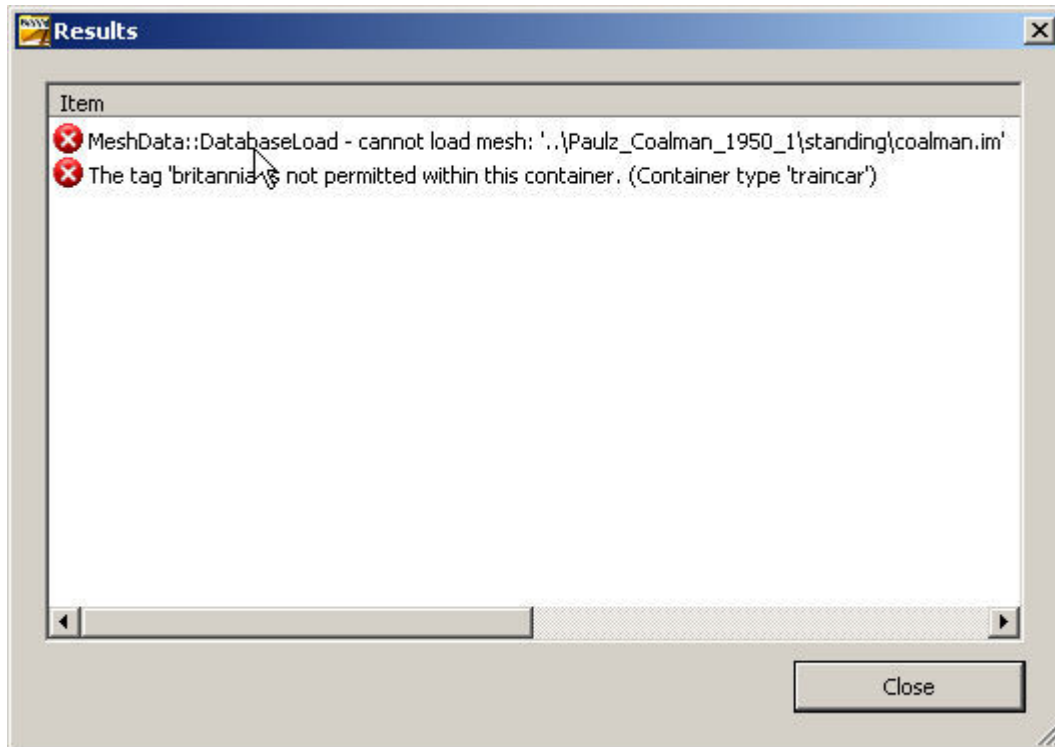
Do not add this to the first, apparently empty line in the file and not somewhere between a pair of curly brackets. If in doubt add new lines directly after the line that starts with `'kuid'`.

This example begs the question how do you know that the new line should read `'direct-drive 0'` and not `'direct-drive'`? The easy answer is to look at a similar asset that is error-free and see what that has in the config file. The more complicated answer is to look in the Content Creation Guide, which will tell you what can follow the `'direct-drive'` tag. In fact in this case `'direct-drive'` will also result in an error-free asset...

Once again, don't forget to commit the asset after editing it.

Missing Mesh Files

The next asset has a missing mesh file as can be seen from the error report:



This is a bit trickier to fix as this asset relies on the data structure that was present in TRS2004. It is an attempt to share the coalman mesh asset between different locos, but unfortunately this would have only worked on the author's PC. This method was never supported or encouraged by Auran for the very good reason that they knew it probably wouldn't work in future versions of Trainz.

It's also interesting in that the asset will still work in both TRS2004 and TRS2006, the only thing that will happen is that the coalman mesh will not be displayed which may well not be noticed.

There's an easy way to fix this, simply delete the references to this mesh in the config file. You'll have to remove the whole block, i.e. remove these lines:

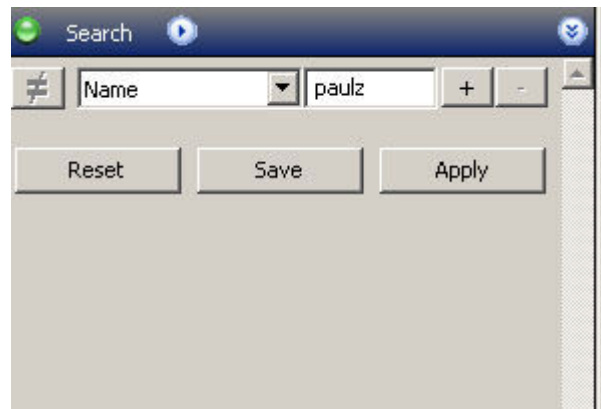
```
coalman {
  mesh          "..\Paulz_Coalman_1950_1\standing\coalman.im"
  auto-create   1
  att-parent    "default"
  att           "a.coalman"
}
```

Resave and commit the asset.

Slightly more complicated is to edit the lines so that they read:

```
coalman {  
  mesh          "standing/coalman.im"  
  auto-create   1  
  att-parent    "default"  
  att           "a.coalman"  
}
```

Now it is necessary to find the missing mesh, which can be done using CMP. First find the asset by searching for part of the name (note that CMP does not use any wildcards in the search fields):



Open this asset for editing and copy the folder called 'standing' into the folder of the loco that we are fixing. This method has the advantage that the coalman mesh should appear now.

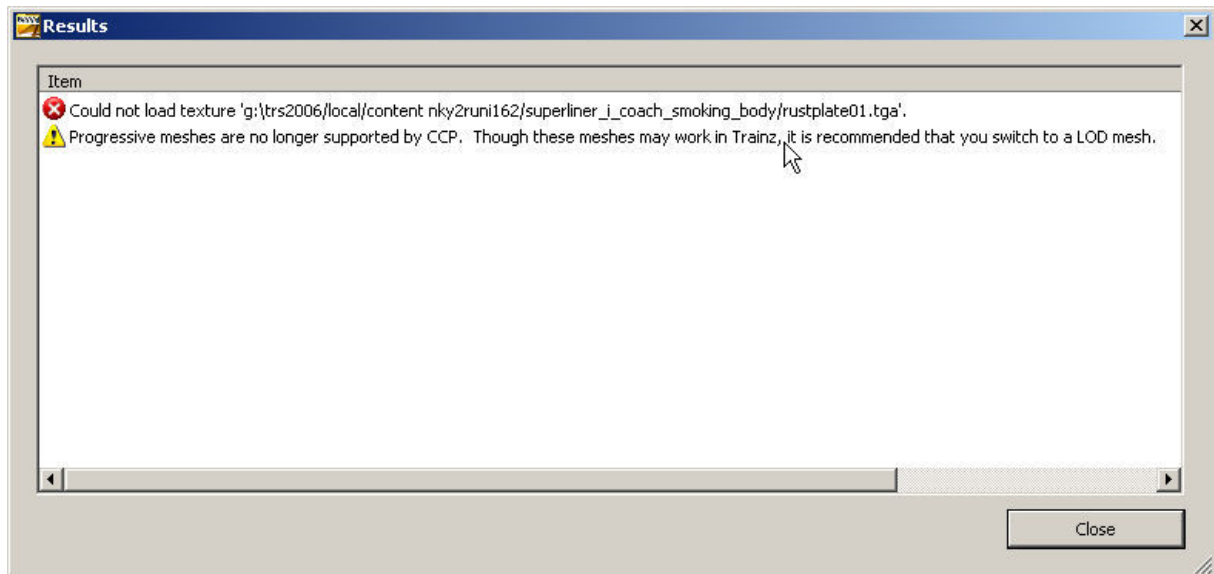
A more likely cause of the missing mesh message would be a typo in the config file, so check that the path and filename in the config really exists. If not you will have to work out which of the *.im files present should be referenced by the config. In some cases it could be that the mesh really is missing and here the choices are to remove the references to the mesh (and hope it wasn't anything important!) or to contact the author.

Type	Status	Name	Author ID	Asset KUID
[Icon]	[Icons]	177548 24048 Paulz_Driver_1950_1	barn700	<kuid:177548:24048>
[Icon]	[Icons]	177548 24045 Paulz_Driver_1925_5	barn700	<kuid:177548:24045>
[Icon]	[Icons]	Paulz_Coalman_1950_1	barn700	<kuid:177548:240021>
[Icon]	[Icons]	Paulz_Coalman_1925_1	barn700	<kuid:177548:24046>
[Icon]	[Icons]	177548 24043 Paulz_Driver_1925_3	barn700	<kuid:177548:24043>
[Icon]	[Icons]	177548 24042 Paulz_Driver_1925_2	barn700	<kuid:177548:24042>

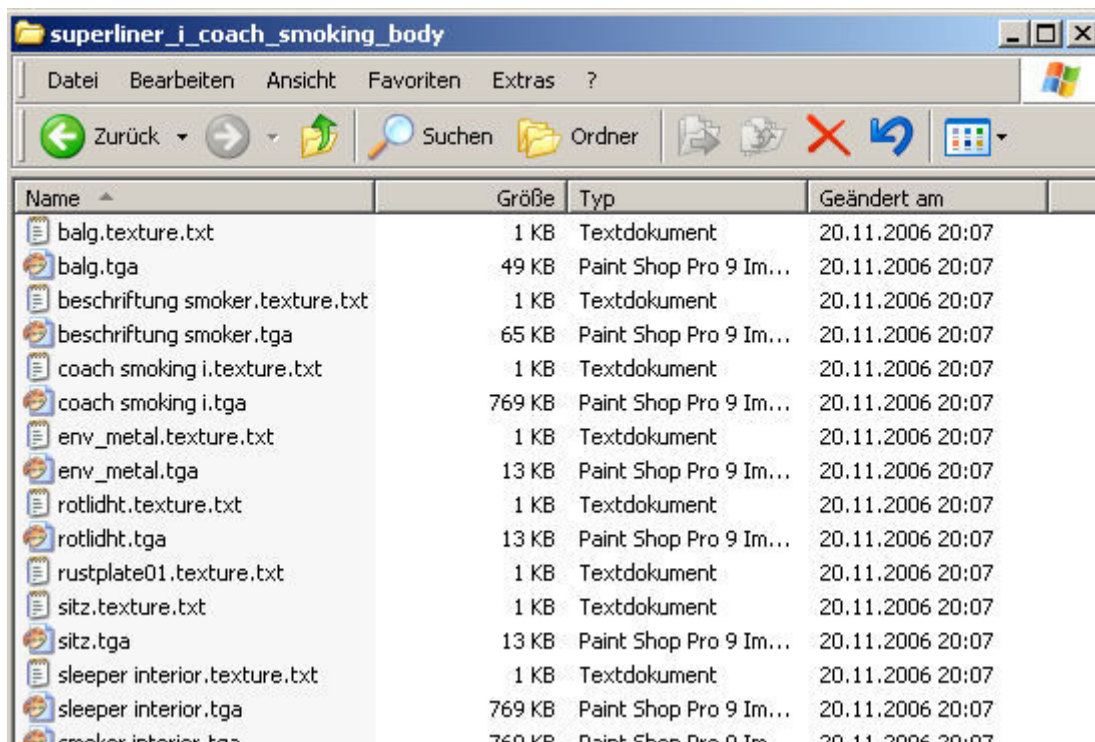
Before taking any drastic steps like removing parts of the config making a backup copy would be a very good idea.

Could not Load Texture

Now so something easier. A very common error is 'Could not Load Texture' as in this example:



Opening the asset for edit and listing the contents of the 'superliner_1_coach_smoking_body' in Explorer gives the following result:



Looking at this you can see that each texture file has a corresponding *.texture.txt file except for 'rustplate01' where no texture file is present. There are a couple of possibilities here, for the brave just delete the file 'rustplate01.texture.txt'. If you are feeling less brave open the 'rustplate01.texture.txt' file to see what type of file it is looking for:

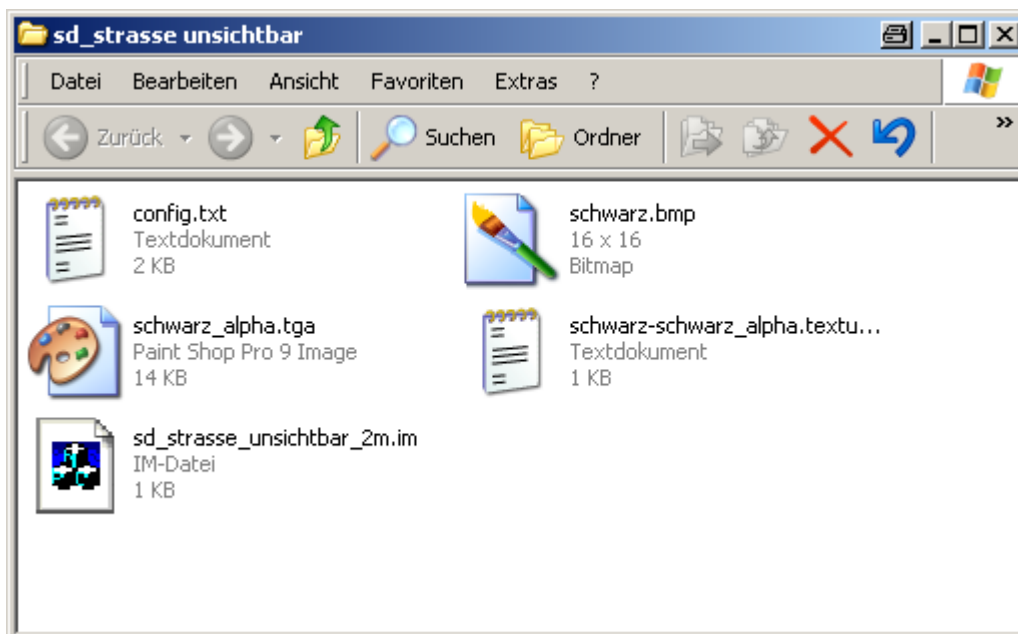
```
1 Primary=rustplate01.tga
2 Tile=st
```

OK, a *.tga. Find the smallest *.tga file in the asset folder, in this case we can use the 'env_metal.tga' file and copy it, changing the name to 'rustplate01.tga'. If you feel confident enough to create your own texture files you can make one using PaintShop Pro or a similar graphics program, but ensure that *.tga files are saved uncompressed. Fill the new *.tga file with a bright colour like pink and save it under the name 'rustplate01.tga' in the folder where the 'rustplate01.texture.txt' file is stored.

Commit the asset and try it out in TRS2006 – if you can't see any bright pink bits it is probably safe to delete both the 'rustplate01.tga' and the 'rustplate01.texture.txt' files. Leaving them there will affect the performance very slightly but it will not cause any major problems.

Should any pink bits be visible then it means that the texture file is needed and was indeed missing in which case the new pink texture can be recoloured to something more appropriate or you could try contacting the author for the missing texture. If all else fails looking for similar assets by the same author might find the missing one.

If you get the 'Could not Load Texture' error and the texture file is present there are a couple of possibilities. Firstly the *.tga file has been saved in a compressed format but this is unlikely to happen as the author should have spotted this himself/herself. Secondly the file could be of the wrong type – here is an example:

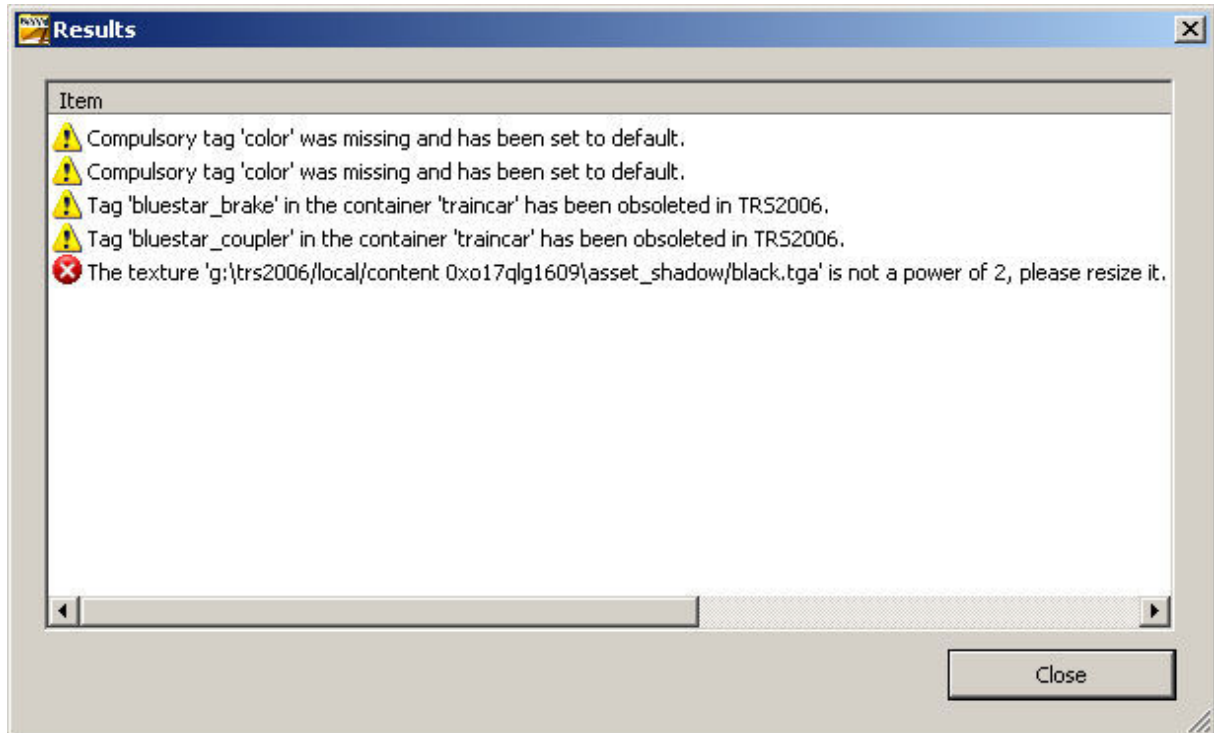


Here the message was that 'Schwarz.tga' could not be loaded – unsurprising really as you can see from the file extension that it is a *.bmp file not a *.tga. Surprising is that the author didn't spot this as the asset couldn't have worked in the first place. The reference to 'schwarz.tga' is in the file 'schwarz-schwarz_alpha.texture.txt' and

this need to be edited to point to 'schwarz.bmp' instead. Incidentally when you see a *.texture.txt file like this with two texture names separated by a hyphen this means that both are referenced within it. One is the texture and the other a separate alpha map texture.

Texture is not a power of 2

Another error that is fairly unlikely to occur with assets that you have downloaded (but it does happen sometimes) is that the texture size is not a power of 2. The width and height of a texture must be a power of 2, i.e. 1, 2, 4, 8, 16, 32, 64, 128 etc.

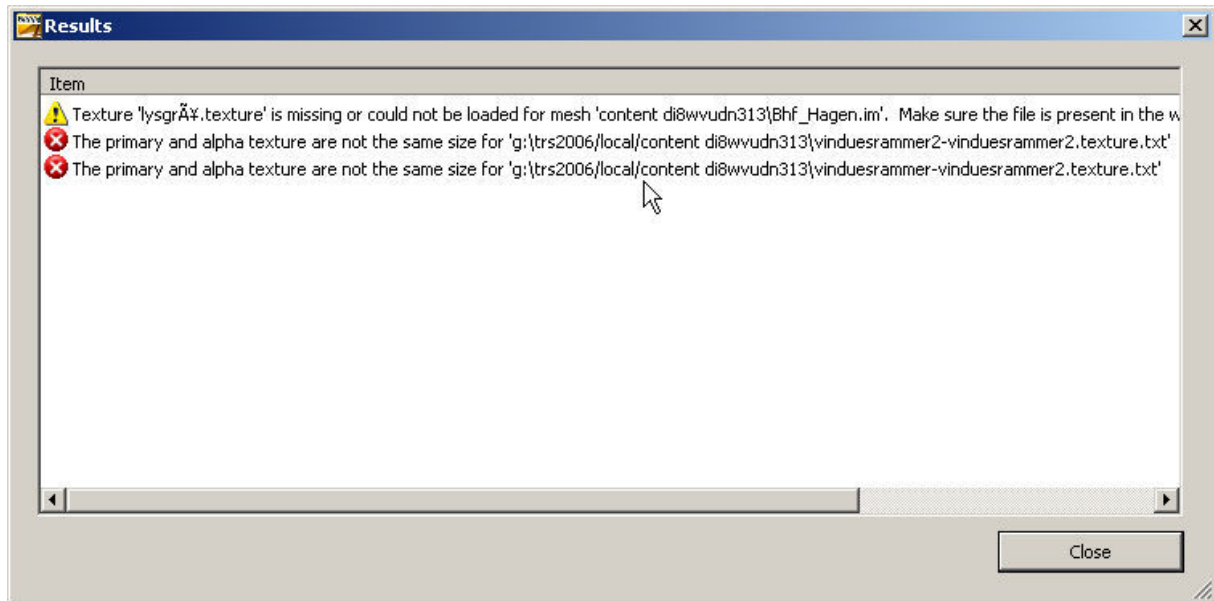


Check the size of the texture by first opening the asset for editing in Explorer and then opening the offending texture file in a graphics program like PaintShop Pro. In this case the texture was 3x3 so it needs resizing – in PaintShop Pro is done with SHIFT-S and typing in the desired sizes. Resave the texture and commit the asset.

Also note the four warnings shown in the error box, once the texture has been resized these will not be shown.

Primary and alpha texture are not the same size

Some assets use texture files that have an extra *.bmp file to define the alpha map – this is used to create see through areas on a model without actually cutting holes in the mesh. An example would be a ladder or a fence, which could be modelled as a single plane with an alpha map.



Prior to TRS2006 the textures did not have to be the same size in pixels, but they do now. The solution is to resize the alpha texture to the same size as the main texture, which can be done using a graphics program such as PaintShop Pro.

Resave the texture and commit the asset.

The asset specified in 'product-kuid' is not of the correct type

The queues part of the config looks like this in this case:

```

queues {
  load0 {
    passenger-queue          1
    size                      25
    initial-count            0
    product-kuid              <kuid:-3:10091>
    attachment-points {
      0      "a.passit00"
      1      "a.passit01"
      2      "a.passit02"
      3      "a.passit03"
      4      "a.passit04"
      5      "a.passit05"
      6      "a.passit06"
      7      "a.passit07"
      8      "a.passit08"
      9      "a.passit09"
      10     "a.passit10"
      11     "a.passit11"
      12     "a.passit12"
      13     "a.passit13"
      14     "a.passit14"
      15     "a.passit15"
      16     "a.passit16"
    }
  }
}

```



```

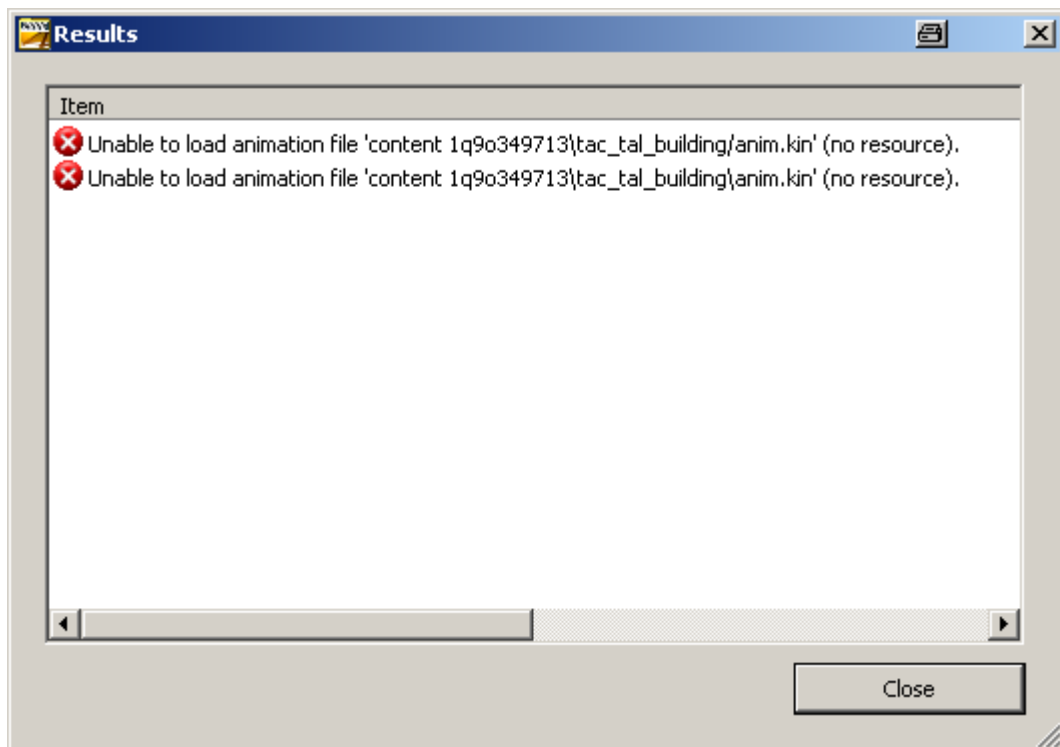
17         "a.passit17"
18         "a.passit18"
19         "a.passit19"
20         "a.passit20"
21         "a.passit21"
22         "a.passit22"
23         "a.passit23"
24         "a.passit24"
    }
    allowed-products {
        0         <kuid:-3:10060>
    }
}

```

The problem is with the line ,product-kuid <kuid:-3:10091>' this is a 'product-category' (passenger), not a 'product-kuid'. The passenger product is <kuid:-3:10060>, so all we need to do is change '10091' to '10060' like this:

```
product-kuid         <kuid:-3:10060>
```

Unable to load animation file

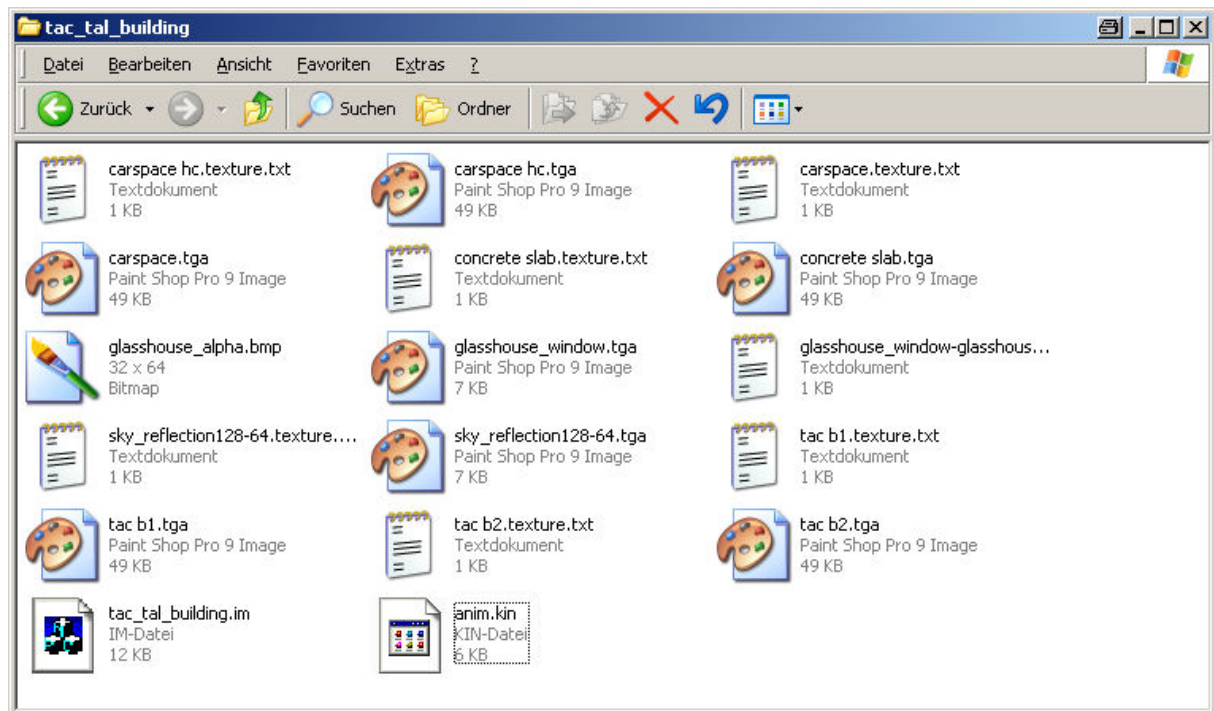


This error is seen in scenery objects that are of kind 'mocrossing' that were developed for earlier versions of Trainz such as UTC. The example seen here is a building with attachment points for a road spline, in order to do this it tries to fool Trainz that it is a level crossing. Unfortunately TRS2006 is not so easily fooled and expects a level crossing to have animated gates or booms and tries to find the animation file 'anim.kin'.

As the author wasn't expecting this he didn't provide one (and why should he, there are no animated parts in the asset).

Fortunately this is easily fixed, we just need to copy an 'anim.kin' file into the same directory as the mesh file. It's not hard to find this file in another asset such as an animated bogey or indeed a level crossing. One can be downloaded [here](#).

The directory where the mesh file is stored should now look like this (note the 'anim.kin' file that I have added). The asset can now be committed.



Postscript

Hopefully this document has helped you to find and fix the more common errors in TRS assets for use in TRS2006. If there are any that I have missed or points that are unclear get in touch <mailto:paul.hobbs@web.de> and I'll post a revised version of this document on the Rail-sim site. Please make sure to give any emails a meaningful subject line as they may well land in my spam trap.