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.

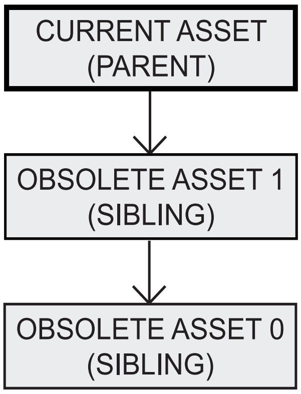
5: Upload your new content to the Auran Download Station (Mandatory)

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 heirachy - 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 and 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 superceded 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.