Trainz Content Utilities

IMPORTANT

These tools are not polished whatsoever. They come supplied "as is" - no support, no bugfixes, not guaranteed to follow best practices or generate correct results.

Downloads:

Trainz Content Utilities Tool

Version 1.0.15 - Download here (NOTE: some functionality not documented yet)

Blender Plugins

Trainz LODz - Download here
Super Batch Export - Download here
Machine3Tools - Download here

Latest external downloads as the above hosted files might be out of date:
Super Batch Export URL: https://github.com/mrtripie/Blender-Super-Batch-Export
Machin3Tools Download Link: https://github.com/machin3io/MACHIN3tools
Machin3Tools Installation Guide: https://machin3.io/MACHIN3tools/docs/installation/

3DS Max Script

Version 0.2 - Download here *(NOTE: functionality not documented yet)*

Functionality:

NOTES:

You can leave any of the fields blank to avoid having them used in the output. If something is required and you have not entered any input in the field the application should tell you in the logs. *(see "log files" section for more information on this)*

.FBX / .IM Filenames:

Any .fbx or .im files MUST use a suffix of "lodXn" where X is a number. The highest polygon LOD will be 0 (i.e *[name]*lod0n.fbx). The lowest polygon LOD will be the highest number (i.e *[name]*lod3n.fbx). So your mesh files might look like this for example:

office_building_lod0n.fbx
office_building_lod1n.fbx
office_building_lod2n.fbx
office_building_lod3n.fbx

lodXn = summer seasonal assets.
lodXnw = winter seasonal assets
lodXna = autumn seasonal assets
lodXnr = spring seasonal assets
lodXns = snow line seasonal assets
lodXnsw = snow line winter seasonal assets
lodXnsa = snow line autumn seasonal assets
lodXnsr = snow line spring seasonal assets
See "Scenery Seasonal Assets" below for more details on seasonal.

YOU MUST USE THESE SUFFIX IN ORDER FOR THE APPLICATION TO FORMAT YOUR CONFIG.TXT FILE CORRECTLY

For different meshes (not LODs of mesh, we are talking completely different meshes) you must supply a unique prefix filename as the application uses the uniqueness of the filename prefix to determine when to write out a new asset config.txt.
This means you can have lots of mesh files (with their LODs) in one library asset and this application will create all the assets based on what meshes are in the library.
For example if you have the following in folder "library 1":
office_building_lod0n.fbx
office_building_lod1n.fbx

office_building_lod2n.fbx
office_building_lod3n.fbx
car_a_lod0n.fbx
car_a _lod2n.fbx
car_a _lod3n.fbx

The application will create 2 folders in its output directory which are “office building” and “car a”. These will be 2 new assets that you can drop straight into Content Manager. NOTE: you must specify the mesh-asset kuid so these assets will know what library to reference (know where the mesh and textures are they need).

.TGA Filenames:

Make sure the following filenames exist for your texture files.
Albedo textures contain the word 'albedo'.
Normal textures contain the word 'normal'.
Parameter textures contain the word 'parameter'.
Other keywords to use in your filename are 'detail' for detail maps (albedo and normal), 'snow' for seasonal snow line textures and 'winter' for seasonal winter textures.
You can combine keywords into the one filename, for example 'name_normal_detail.tga' for a normal map detail texture.

Art Recommendations:

Everyone should have a read through this trainz wiki page as it has a mountain of information on standards for Trainz content creation:
http://online.ts2009.com/mediaWiki/index.php/Art_Recommendations

Root Directory:

This is the full directory path that one or more folders are present in. The application will scan these folders to process each sub-folder using the chosen Option.

Generate config.txt

This application will search a directory and it's children to find specific files based on the Option you select in the Options drop down list.

Output folders and files will be under the 'Output' folders of each root and child directories.

Options:

Scenery

The option will scan the root and sub-folders to find any .fbx or .im files and output config.txt files accordingly. You MUST use the .FBX / .IM Filename approach listed above for this tool to work.

GroundTexture

The option will scan these folders to find any .tga files and output config.txt files accordingly. You MUST use the .TGA Filename approach listed above for this tool to work.

Kuid:

Specify the kuid of the first asset to be generated. For example <kuid:444:666>.
If more than one asset is to be generated (like in the above example of “office building” and “car a”) the application will take the kuid you give it and increment the second part of it (666) by one each time it needs a new kuid.
Office Building = <kuid:444:666>
Car A = <kuid:444:667>
and so on …

Kuid Version Num:

Allows you to specify a version number if you don’t want it to use an original kuid you enter.
For example if you enter the Kuid: <kuid:444:666> and you enter Kuid Version Num: 3 then all kuids that are generated in the config.txt file will be version 3 - <kuid2:444:666:3>, <kuid2:444:667:3> and so on ...

Mesh-Asset Kuid:

This is the kuid of the mesh library asset that the newly generated assets will need to know about if you plan on using a mesh library. For more information on mesh libraries visit the trainz wiki:
http://online.ts2009.com/mediaWiki/index.php/Mesh_Libraries
And the trainzportal blog here:
https://www.trainzportal.com/blog/view/content-creation-blog/mesh-libraries-using-the-mesh-asset-tag

Category-Class:

http://online.ts2009.com/mediaWiki/index.php/Category-class

Inc. Number to end of Username:

This will generate a number on the end of a username. With every groundtexture assets the application process the higher the generated number increases. This allows you to put a number of an asset that uses the same username. For example if you have 5 different “dirt” groundtexture assets it will name them:

Dirt 1
Dirt 2
Dirt 3
Dirt 4
Dirt 5

Scenery Seasonal Assets:

You just simply make sure your mesh files follow these naming conventions and the season-selector container will appear in your generated config.txt.

Make sure the following “suffix” is on every mesh (and their LODs) so the application knows what to generate:

(X is the LOD number)

lodXn = summer seasonal assets.
lodXnw = winter seasonal assets
lodXna = autumn seasonal assets
lodXnr = spring seasonal assets
lodXns = snow line seasonal assets

lodXnsw = snow line winter seasonal assets
lodXnsa = snow line autumn seasonal assets
lodXnsr = snow line spring seasonal assets

Example:
Summer Meshes:
office_building_lod0n.fbx
office_building_lod1n.fbx

Winter Meshes:
office_building_lod0nw.fbx
office_building_lod1nw.fbx

Snow line Summer Meshes:
office_building_lod0ns.fbx
office_building_lod1ns.fbx

Snow line Winter Meshes:
office_building_lod0nsw.fbx
office_building_lod1nsw.fbx

Keep in mind you don't have to use all these combinations. For example a house might only need the following mesh suffixes::
Summer = lodXn
and
Snow line Summer = lodXns

Foliage can definitely benefit from many of the above combination of seasons. Keep in mind the more textures you use the larger your asset will be on disk. See wiki for more explanation of seasonal support:
http://online.ts2009.com/mediaWiki/index.php/%22season-selector%22_container#.22Season-selector.22_Example

Hide Asset in Winter:

This allows you to have the mesh be completely hidden during winter. For example a shrub might die in the winter so instead of adding a blank mesh which is wasteful you just enable this checkbox in the tool and it will write out a winter sub-container under the season-selector container that no mesh-table entry references. If you have a .fbx file that contains "lodXnw" in the filename then this will override the hide asset in winter option and this mesh will be shown.

Hide Asset Above Snow line:

This allows you to have the mesh be completely hidden above the snow line. For example a shrub might die when it gets too cold so instead of adding a blank mesh which is wasteful you

just enable this checkbox in the tool and it will write out a snow line sub-container under the season-selector container that no mesh-table entry references. If you have a .fbx file that contains any of these “lodXns”, “lodXnsw”, “lodXnsa”, “lodXnsr” in the filename then each of these will override the hide asset above snow line option and the mesh that corresponds to the season will be shown.

Anything else not listed is deemed to be self-explanatory. If you don’t understand a input field it is best you search for it on the trainz wiki as it will definitely be listed there to give you more information: http://online.ts2009.com/mediaWiki/index.php/KIND_TrainzBaseSpec

Generate metadata.txt

This tool will search a directory and it's children to find any .fbx files and generate the [name of fbx file].txt metadata file(s) from its findings.

LOD Transition Array

This defines the number of LOD transitions. You should have the same number of .fbx files as you have distances in the LOD Transitions Array. If the Array has “10,40,100,500” then that is 4 different distances therefore you should have 4 different .fbx LOD meshes:
Namelod0n.fbx
Namelod1n.fbx
Namelod2n.fbx
Namelod3n.fbx

The LOD Transitions Array MUST be the same as what is found in your config.txt file.

Options:

All LODs use POP transitions

This is the default behaviour of how LODs work in Trainz so if you don’t plan on using m.clutter materials always use this option.

m.clutter high LODs POP / lowest LOD fade

You must define at least one [name].m.clutter material to use this option. This option will find all the LOD levels for a set of FBX files and generate metadata.txt files for the lowest polygon LOD to tell it to fade out on the last distance in the LOD Transition Array you enter.

This means if you enter a LOD Transitions Array of “10,40,100,500” then the lowest polygon LOD which might be namelod3n.fbx will have a metadata.txt files of namelod3n.txt and a material container of [name].m.clutter with a tag fadeOutEndDistance = 500.

To understand m.clutter materials in relation to metadata.txt files read the trainz wiki here: http://online.ts2009.com/mediaWiki/index.php/M.clutter

m.clutter all LODs use fade transitions

You must define at least one [name].m.clutter material to use this option. This option will find all the LOD levels for a set of FBX files and generate metadata.txt files for all the .fbx LODs to tell them when to fade in and out based on the LOD Transition Array you enter.

This means if you enter a LOD Transitions Array of “10,40,100,500” then each LOD will get a metadata.txt file which contains the correct fadeInEndDistance and fadeOutEndDistance values based on what you entered in the LOD Transitions Array and the number of .fbx files in the directory it parsed.

If you have multiple LODs in your clutter asset you will want to tick the “Append LOD num to material name” checkbox.

Append LOD num to material name

For Option “m.clutter all LODs use fade transitions” it adds a number and the letter ‘a’ or ‘b’ to any material names. This is a complicated topic so a video tutorial will be added soon. You can try generating metadata.txt files using this option to see what the output structure is and manually change your material names that get generated in the metadata.txt files as needed for now if they are not named exactly how you need them. Basically take “body.m.clutter”, it ends up in the metadata.txt file as body0a.m.clutter and also another container body0b.m.clutter since the fade transitions for each LOD (except for the lowest LOD) should have 2 meshes in it using different materials.

For Option “m.clutter high LODs POP / lowest LOD fade” it does the same thing as above but doesn’t output the ‘a’ or ‘b’ character on the end of the material name since we are only working with the lowest LOD and it only contains 1 mesh.

Materials:

All metadata.txt files require material names so that you can determine what tags to use to manipulate the material(s) in questions. This field allows you to enter in one or more materials to add tags for.

Example: body.m.pbrmetal, anothermaterial.m.pbrmetaldetail, somethingelse.m.pbrmetal

NOTES:

To understand m.clutter materials in relation to metadata.txt files read the trainz wiki here: http://online.ts2009.com/mediaWiki/index.php/M.clutter

Anything else not listed is deemed to be self-explanatory. If you don’t understand a input field it is best you search for it on the trainz wiki as it will definitely be listed there to give you more information:

http://online.ts2009.com/mediaWiki/index.php/Mesh_metadata_file
http://online.ts2009.com/mediaWiki/index.php/M.pbrmetaldetail#Material_Attributes

Generate texture.txt

While Trainz automatically generates texture.txt files for your .tga files on commit in Content Manager, sometimes it is handy to generate these yourself with different settings.

This application will search a directory and it's children to find any .tga files and generate a texture.txt file for each tga file it finds.

To find out more about texture.txt files visit the trainz wiki here:
http://online.ts2009.com/mediaWiki/index.php/Texture_file

Update config.txt

Parses the root and all sub-folders and finds their config.txt files.

Trainz-Build:

Updates the trainz-build number to whatever you put here. Leave blank and it will be skipped.

Inc KUID Version:

Increases the kuid version by 1. Untick and it will be skipped.

LOD Transition Array

Adds the new “mesh-table-lod-transition-distances” while striping out any old LOD tags. You should have the same number of .fbx files as you have distances in the LOD Transitions Array. If the Array has “10,40,100,500” then that is 4 different distances therefore you should have 4 different .fbx LOD meshes:
Namelod0n.fbx
Namelod1n.fbx
Namelod2n.fbx
Namelod3n.fbx

This will only be written into config.txt files if the trainz-build is 4.6 or greater.

Must Contain:

Enter in one of more strings separated by a comma (,) to only edit the config.txt files that contain these strings. For example you might only want config.txt files with a certain tag, username or word in the description. Leave blank to change all config.txt files it finds under the root and sub-folders.

.fbx files manipulation

Parses the root folder only and looks for .fbx files.

Options:

Split into 2 Folders

This option will put all .fbx files that contain the string you enter into a folder called “FolderA” and the rest of the .fbx files it finds into a folder called “Folder B”. For example this allows you to move all summer seasonal .fbx files (lodXn.fbx) into one directory for processing (metadata.txt files on) and the winter seasonal .fbx files (lodXnw) into another directory for processing.

String:

The string in which to search on.

Log Files

These logs are VERY useful. If you are having any issues make sure you look under the directory you are parsing for the “output” directory as that is where the log.txt file for your current execution will reside.

Double check it is the correct log file you are opening as many log files are generated when parsing many sub-folders.