DTS Exporter Documentation – Nodes Listing

Version 1.2
June 5th, 2003

Introduction:

This document is a simple directory listing that collects together most of the known nodes that can be used in various DTS shapes (static objects and players, weapons, and vehicles) within the Torque Game Engine. The listing is provided so that you know the names (with correct spelling), function, and usage of the various nodes.

The list itself has been broken down categorically so that it is easy to find specific nodes. It does not contain node references for 3rd party resources, only nodes that the standard "vanilla" Torque install uses. If you require information on nodes that other resources use, you are asked to contact the developer of that resource.

Please note that this documentation is not a tutorial, it is simply a reference guide. If you require a tutorial on how to export data from your 3D modeling application to a DTS file, you are asked to read Joe Maruschak's excellent and in-depth tutorial on the DTS Exporter found on the GarageGames web site.

Lastly, this is a living document and should be treated as such, please do not keep local copies of this document as the information that it may contain could be outdated due to changes that take place within the Torque Game Engine.

Document Contributors:

Logan Foster
Joe Maruschak

Table of Content

Character & Static Shape Nodes

Bounds	Page 5
Cam	Page 5
Col	Page 5
Collision	Page 6
Detail	Page 6
Eye	Page 6
Light	Page 6
LOS	Page 6
LOSCol	Page 6
Mount	Page 6
MultiRes::	Page 7
Ski	Page 7
SORT::	Page 7

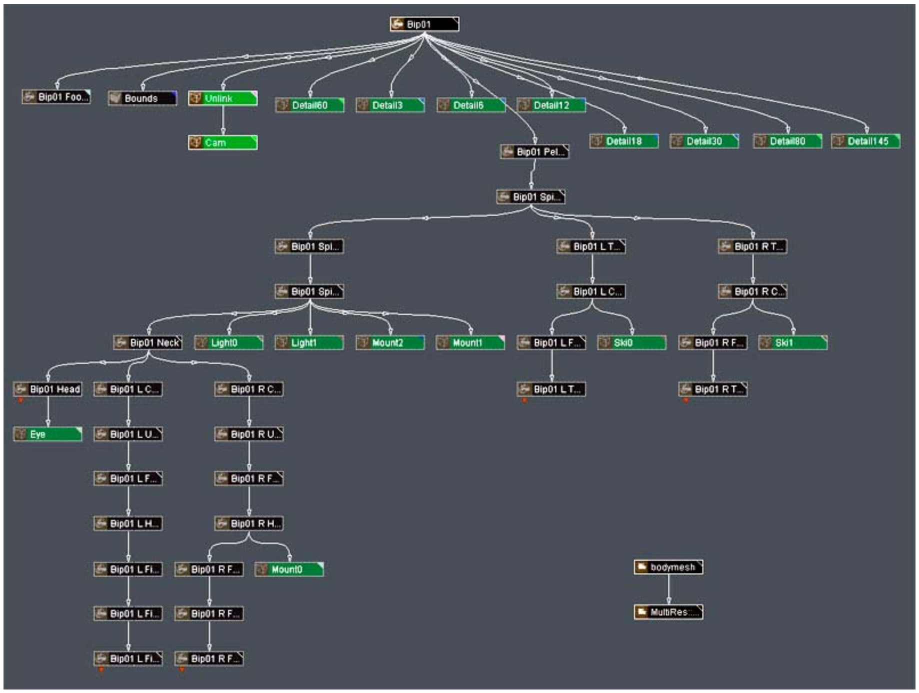
Weapon Nodes

Bounds	Page 8
Col	Page 9
Collision	Page 9
Detail	Page 9
Ejectpoint	Page 10
Mountpont	Page 10
Muzzlepoint	Page 10
Sequence Helpers	Page 10

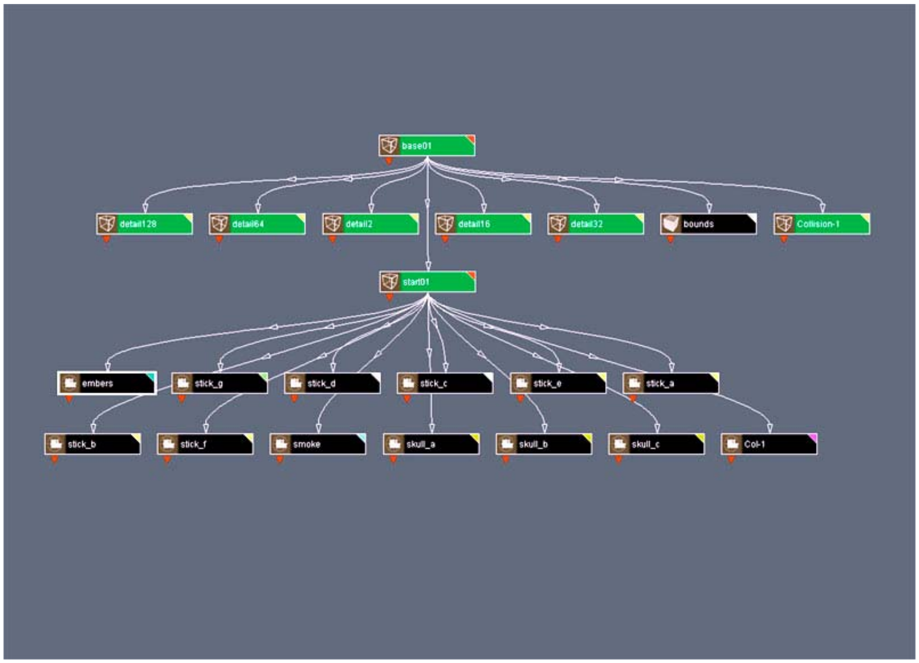
Vehicle Nodes and Information

Bounds	Page 11
Cam	Page 12
Col	Page 12
Collision	Page 12
Contrail	Page 12
Detail	Page 13
Eye	Page 13
Hub	Page 13
Jetnozzle	Page 13
Light	Page 13
LOS	Page 13
LOSCol	Page 13
Mount	Page 14
Smoke_node	Page 14

Character & Static Shape Nodes



A example of a typical hierarchical chart of a proper Torque character model (from 3DS Max)



An example of a typical hierarchical chart of a proper Torque static object, shape (from 3DS Max)

bounds – Bounding Geometry

This is essentially a bounding box that must encompass at all times your shape. The exporter uses the bounding box to decide which geometry it should send to the DTS file.

The “bounds” shape is also used as a ground transform (where your shape will sit/stand when placed in the engine) and the default location for any missing nodes in your scene. The location of the ground transform is specifically based off of the location of the “bounds” nodes transform or pivot point in your 3D scene.

Note this node is not required when using Milkshape to export your 3D scene.

Example Usage- bounds

cam – 3rd person Camera Placement

Required only for playable characters. The “cam” node is used to tell the engine where to view the model from if the camera perspective is placed into 3rd person mode. Without this node you will view from the transform coordinates of the “bounds” shape.

Example Usage- cam

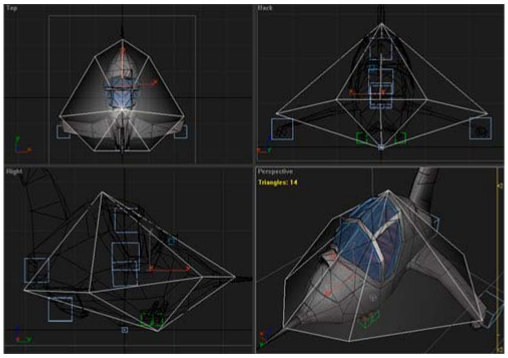
Col-# – Collision Geometry

“Col” nodes are a geometric shape that is used to perform the collision test within Torque Game Engine. Since this is the collision shape that will be used, you will want to ensure that this is a very simple shape (20 polygons or less or else you will affect performance) and that is not concave (i.e. it does not indent or push into itself at any location).

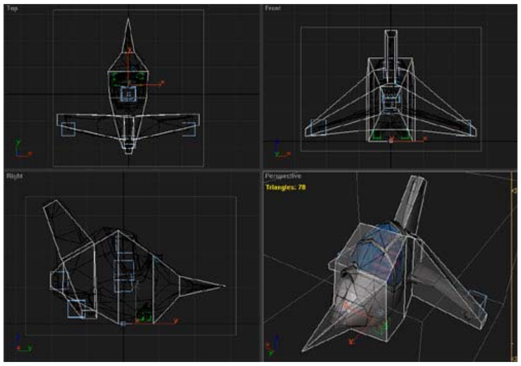
Note: The shape of the collision geometry is very important as it can influence whether or not your vehicle can get stuck on objects in your scene or if it will bounce off of them. Also note that all collision geometry must be convex and not concave in shape or you will generate an error and the exporter will not export your mesh.

All “Col” geometry must be named with this convention and the trailing number that follows must match much a number on a corresponding “Collision” node.

Example Usage- Col-1, Col-2, Col-9



An example of a good collision mesh for your DTS shape: Low polygon count and convex shape



An example of an incorrect collision mesh for your DTS shape: High polygon count and many concave areas.

Collision-# – Collision Marker

“Collision” nodes are very similar to “details” nodes as they tell the DTS Exporter that a collision shape exists and that it should be used. These nodes are specifically named “Collision-#” where number (#) is a value between 1 and 9.

Example Usage- Collision-1, Collision-2, Collision-9

detail# – Detail Marker

The detail markers are placed in your scene as a method to tell the engine the various LOD settings that can be placed into the DTS file when it is exported (as denoted by the trailing number, which replaces the # sign). You can have as many detail markers in your scene as you wish.

The trailing number after “detail” denotes the pixel size of the model on the player’s screen. You do not want to use a number smaller than 2 as this would cause a model to be drawn infinitely.

Example Usage- detail75, detail2

eye – 1st person Camera Marker

Required only for playable characters, this node is essentially just a set of transform coordinates that tells the engine where to place the camera for viewing from 1st person perspective. Without this node you will view from the transform coordinates of the “bounds” shape.

Example Usage- eye

Light# - Light Emitter Marker

Example Usage – Light0, Light1

LOS-# – Line of Sight marker

Markers for line of sight, or 'bullet' collision shapes. Uses # 9 through 15

Example Usage – LOS-9, LOS-10, LOS-15

LOSCol – Line of sight geometry

Geometry for line of sight collision. Uses # 9 through 15

Example Usage – LOSCol-9, LOSCol-1, LOSCol-15

mount# - Mount Marker

Mount markers are used to tell the engine that something can be attached to your model at this position. For example you would use a mount marker to tell the engine where to place a weapon on your player model.

Example Usage: mount0, mount1, mount2

MultiRes:: - Specialized Name Marker

For 3DS Max users for use on character models only. This is a special case marker that is used only for character models that have the Skin modifier on it.

“MultiRes::” is placed before the name of a snapshot of your character mesh. It is used to tell the engine that this particular mesh has the LOD information for the character node and should be used to generate the LOD information for the DTS file on export.

Example Usage- MultiRes::bodymesh, MultiRes::character

Ski# - Ski Marker

Example Usage- Ski0, Ski1

SORT:: - Specialized Name Marker

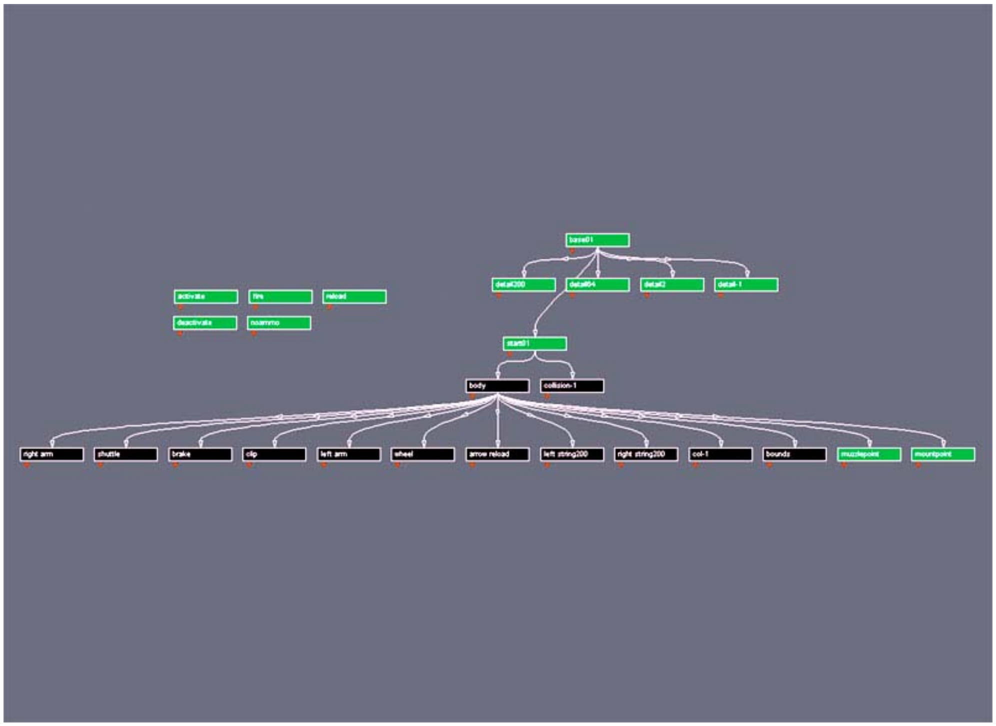
“SORT::” is a specialized flag that is placed in front of your geometric nodes name that the exporter catches and is used when you have a small (i.e. three units or less) polygonal object with transparency that is rendering incorrectly (almost appears backwards or inside out).

When the exporter catches the “SORT::” tag it runs the mesh through a specialized polygonal splitting algorithm that will allow the shape to be rendered correctly in the engine due to its small size/scale.

Note: It is recommended that you only use SORT:: on meshes with small polygon counts (50 or less). If possible you are advised to make any transparent polygons their own mesh if they need SORT:: as this will speed up the exporter process immensely.

Example Usage- SORT::mymesh, SORT::treeleaves

Weapons Nodes



A hierarchical example of a weapon shape for Torque (from 3DS Max)

bounds – Bounding Geometry

> This is essentially a bounding box that must encompass at all times your shape. The exporter uses the bounding box to decide which geometry it should send to the DTS file.
>
> The “bounds” shape is also used as a ground transform (where your shape will sit/stand when placed in the engine) and the default location for any missing nodes in your scene. The location of the ground transform is specifically based off of the location of the “bounds” nodes transform or pivot point in your 3D scene.
>
> Note this node is not required when using Milkshape to export your 3D scene.
>
> Example Usage- bounds

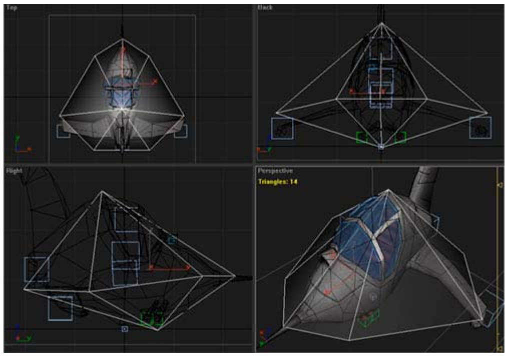
Col-# – Collision Geometry

“Col” nodes are a geometric shape that is used to perform the collision test within Torque Game Engine. Since this is the collision shape that will be used, you will want to ensure that this is a very simple shape (20 polygons or less or else you will affect performance) and that is not concave (i.e. it does not indent or push into itself at any location).

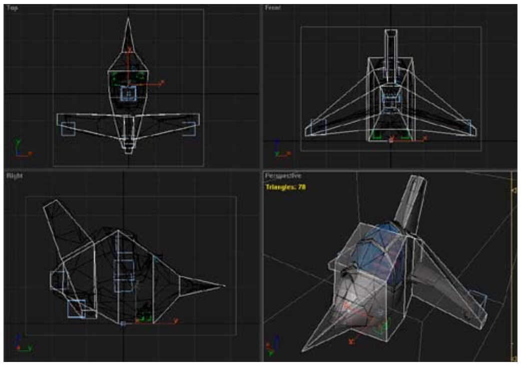
Note: All collision geometry must be convex and not concave in shape or you will generate an error and the exporter will not export your mesh.

All “Col” geometry must be named with this convention and the trailing number that follows must match much a number on a corresponding “Collision” node.

Example Usage- Col-1, Col-2, Col-9



An example of a good collision mesh for your DTS shape: Low polygon count and convex shape



An example of an incorrect collision mesh for your DTS shape: High polygon count and many concave areas.

Collision-# – Collision Marker

“Collision” nodes are very similar to “details” nodes as they tell the DTS Exporter that a collision shape exists and that it should be used. These nodes are specifically named “Collision-#” where number (#) is a value between 1 and 9.

Example Usage- Collision-1, Collision-2, Collision-9

detail# – Detail Marker

The detail markers are placed in your scene as a method to tell the engine the various LOD settings that can be placed into the DTS file when it is exported (as denoted by the trailing number, which replaces the # sign). You can have as many detail markers in your scene as you wish.

The trailing number after “detail” denotes the pixel size of the model on the player’s screen. You do not want to use a number smaller than 2 as this would cause a model to be drawn infinitely.

Example Usage- detail75, detail2

ejectpoint – Weapon Projectile Emitter Marker

The “ejectpoint” node tells the engine where to emit the shell casing DTS file from when the weapon is fired.

Example Usage - ejectpoint

mountpoint – Weapon Location Marker

This node tells the engine where to attach the weapon to the player model. The weapon is attached specifically at the location of the “mountpoint” to the corresponding “mount” node on the player character.

If there is no mountpoint in the DTS file or if the node is not exported, the engine will default to the origin (0,0,0) of the DTS file when mounting.

Example Usage - MountPoint

muzzlepoint – Weapon Muzzle Flash Marker

The “muzzlepoint” node tells the engine where to show the ‘flash’ or ‘bang’ DTS file when the weapon is fired.

Example Usage - muzzlepoint

Sequence Helpers

Activate – Animation that is played when the weapon is activated and placed on the player character. Options: Cyclical sequence (no), Complete cycle (no), Blend sequence (no), FPS (60), Transform Animation.

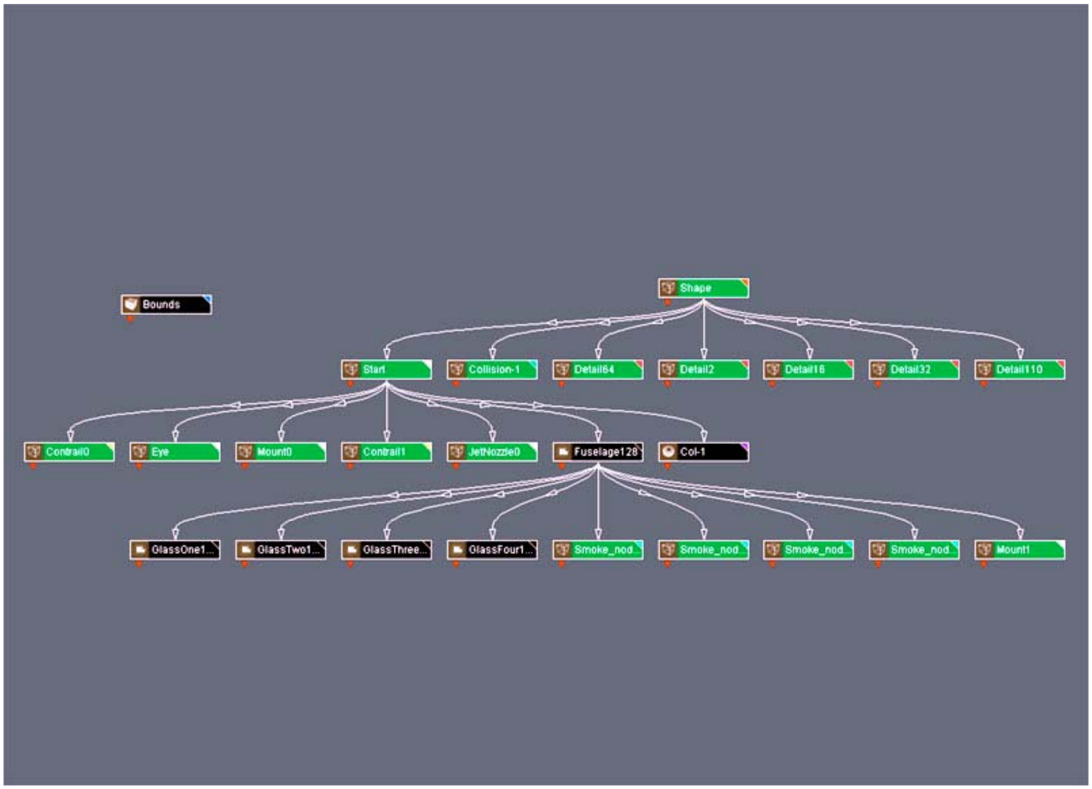
Deactivate – Animation that is played when the weapon is deactivated and put away. Options: Cyclical sequence (no), Complete cycle (no), Blend sequence (no), FPS (60), Transform Animation.

Fire – Animation that is played when the player would fire the weapon. Options: Cyclical sequence (no), Complete cycle (no), Blend sequence (no), FPS (60), Transform Animation.

NoAmmo – Animation that is played when the player has no available ammo. Options: Cyclical sequence (no), Complete cycle (no), Blend sequence (no), FPS (60), Transform Animation.

Reload - Animation that is played when the player is reloading the weapon. Options: Cyclical sequence (no), Complete cycle (no), Blend sequence (no), FPS (60), Transform Animation.

Vehicle Nodes



An example of the vehicle hierarchy for Torque (from 3DS Max)

bounds – Bounding Geometry

This is essentially a bounding box that must encompass at all times your shape. The exporter uses the bounding box to decide which geometry it should send to the DTS file.

The "bounds" shape is also used as a ground transform (where your shape will sit/stand when placed in the engine) and the default location for any missing nodes in your scene. The location of the ground transform is specifically based off of the location of the "bounds" nodes transform or pivot point in your 3D scene.

Note this node is not required when using Milkshape to export your 3D scene.

Example Usage- bounds

cam – 3rd person Camera Placement

Required only for playable characters. The "cam" node is used to tell the engine where to view the model from if the camera perspective is placed into 3rd person mode. Without this node you will view from the transform coordinates of the "bounds" shape.

Example Usage- cam

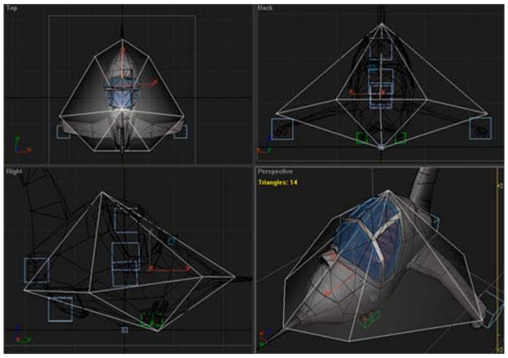
Col-# – Collision Geometry

“Col” nodes are a geometric shape that is used to perform the collision test within Torque Game Engine. Since this is the collision shape that will be used, you will want to ensure that this is a very simple shape (20 polygons or less or else you will affect performance) and that is not concave (i.e. it does not indent or push into itself at any location).

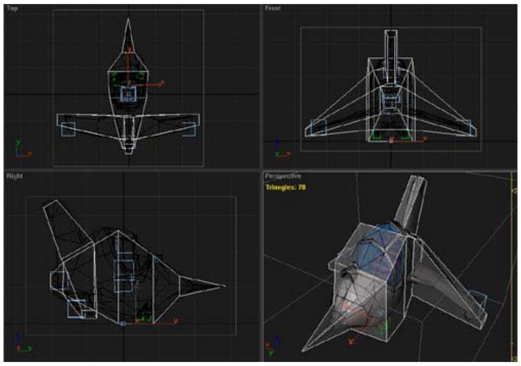
Note: The shape of the collision geometry is very important as it can influence whether or not your vehicle can get stuck on objects in your scene or if it will bounce off of them. Also note that all collision geometry must be convex and not concave in shape or you will generate an error and the exporter will not export your mesh.

All “Col” geometry must be named with this convention and the trailing number that follows must match much a number on a corresponding “Collision” node.

Example Usage- Col-1, Col-2, Col-9



An example of a good collision mesh for your DTS shape: Low polygon count and convex shape



An example of an incorrect collision mesh for your DTS shape: High polygon count and many concave areas.

Collision-# – Collision Marker

“Collision” nodes are very similar to “details” nodes as they tell the DTS Exporter that a collision shape exists and that it should be used. These nodes are specifically named “Collision-#” where number (#) is a value between 1 and 9.

Example Usage- Collision-1, Collision-2, Collision-9

Contrail#

This emitter creates contrails, which are those little wisps that you see that emit from the tips of wings on airplanes.

Example Usage - Contrail0, Contrail1

detail# – Detail Marker

The detail markers are placed in your scene as a method to tell the engine the various LOD settings that can be placed into the DTS file when it is exported (as denoted by the trailing number, which replaces the # sign). You can have as many detail markers in your scene as you wish.

The trailing number after “detail” denotes the pixel size of the model on the player’s screen. You do not want to use a number smaller than 2 as this would cause a model to be drawn infinitely.

Example Usage- detail75, detail2

eye – 1st person Camera Marker

Required only for playable characters, this node is essentially just a set of transform coordinates that tells the engine where to place the camera for viewing from 1st person perspective. Without this node you will view from the transform coordinates of the “bounds” shape.

Example Usage- eye

hub# - Wheel Placement Marker

Wheeled Vehicles Only. Hub nodes are used to define where a wheel DTS shape will attach to the mesh when it is in the engine.

Note: For wheeled vehicles it is recommended that you use multiple hub nodes, this will stabilize your vehicle and make it easier for the end user to drive around in a scene. It is not recommended to have a vehicle with only one hub node!

Example Usage: hub0, hub1, hub2, hub3

JetNozzle#

This node tells the engine where to emit a jet engine exhaust from.

Example Usage – JetNozzle0, JetNozzle1

Light# - Light Emitter Marker

Example Usage – Light0, Light1

LOS-# – Line of Sight marker

Markers for line of sight, or 'bullet' collision shapes. Uses # 9 through 15

Example Usage – LOS-9, LOS-10, LOS-15

LOSCol – Line of sight geometry

Geometry for line of sight collision. Uses # 9 through 15

Example Usage – LOSCol-9, LOSCol-1, LOSCol-15 mount# - Mount Nodes

Mount nodes are used for two things with vehicles in TGE, either as a player attachment point or as a weapon location. Typically mount nodes will be used or assigned to the following functions:

mount0 – pilot of vehicle
mount1 – navigator/gunner
mount2 – passenger
mount3 – passenger
mount4 – passenger
mount5 – passenger
mount6 – passenger
mount7 – passenger
mount8 – passenger
mount9 – Weapon (bomb bay)
mount10 – Weapon (gun)

Smoke_node# – Particle Emitter Placement Marker

These nodes (named “smoke_node#) denote the location of a particle system to the engine. This particular particle system emits particles based on the amount of damage that your vehicle has taken. To place this node on your vehicle, create a dummy helper object and link it to the location on your mesh where you would like the damage smoke to emit from in the game.

Example Usage – Smoke_node0, Smoke_node1