

Modding-Doku

Graphics

v1.0

27.02.04

Contents

1. Setting up a scene
 - a) Bodies
 - b) Reduced bodies
 - c) Scenes
2. Material properties
 - a) Material definitions exported from 3ds max
 - b) Material settings in .bod files
 - c) Bump map creation
3. Exporter usage
 - a) Exporting bodies
 - b) Exporting scenes
 - c) Combined scenes
 - d) Binary and verbose options
4. Types definitions
 - a) Particles
 - b) Effects
 - c) Ships
5. Special bodies list

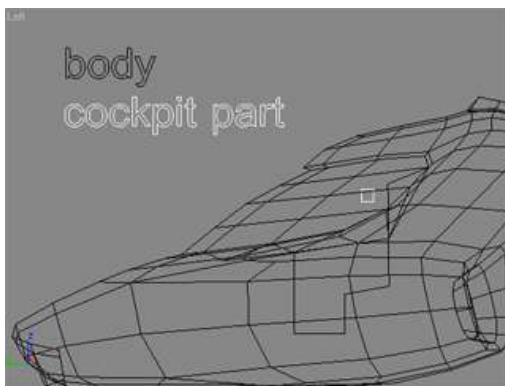
1. Setting up a scene

a) bodies

A scene like a ship or station consists of several smaller units that we call bodies. We try to make them reusable for different objects, for example engine or cockpit bodies used by several ships or the living section that most stations of one race have in common.

The name for a Body always starts with “**B_**“. The rest doesn’t matter, but do not use any numbers here.

Bodies themselves can be divided into **parts**. To do this you need to detach some part of the model and then link it to the rest. If you use a special name for the part, it will be used as a position marker. Like:



Cockpit

defines the position of the cockpit. The origin of the cockpit scene will be placed at the center of the cockpit part. As the cockpit has to be placed somewhere inside the model use a box / triangle to define its position.

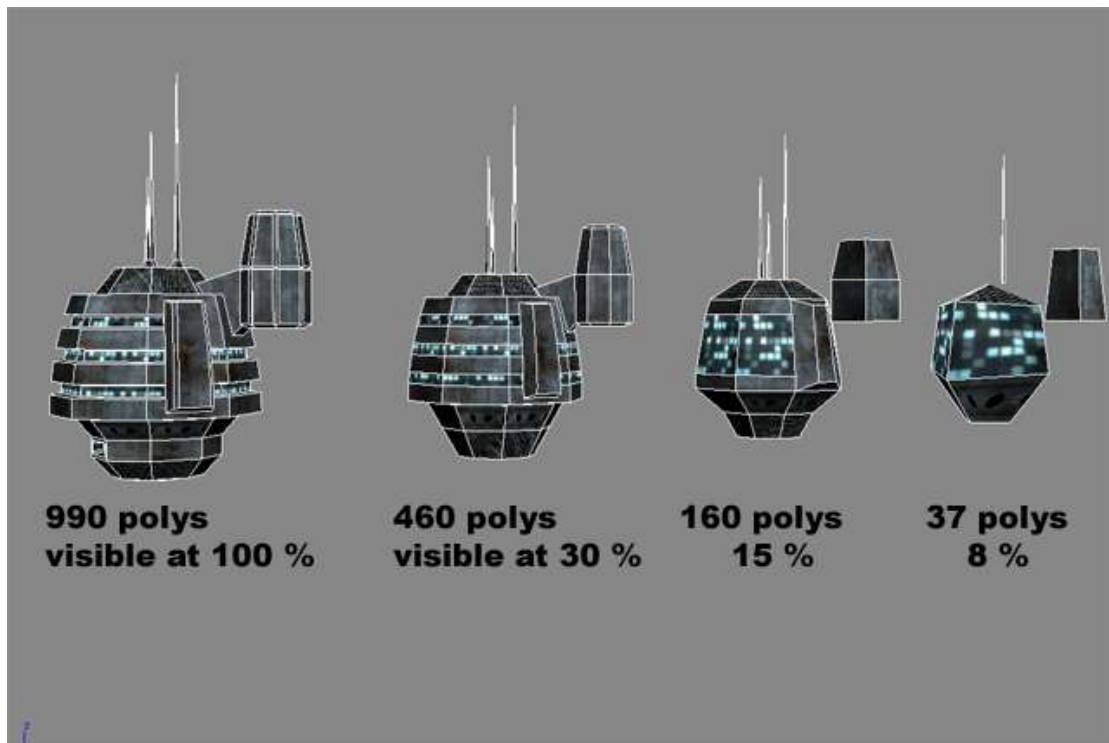
Before exporting a body always make sure that his pivot is centered and in the Origin (0;0;0;)

All scale transformations must be collapsed. You do this by applying a “**Reset-X Form - Modifier**” and then collapsing the modifier stack.

Smoothing groups are exported. They can help a lot in keeping a low poly count.

b) Reduced bodies

When modeling a body always have in mind that you will need to make several levels of detail (LOD) for it. As a rule every reduced version should have half of the polys than the more detailed version (or less).

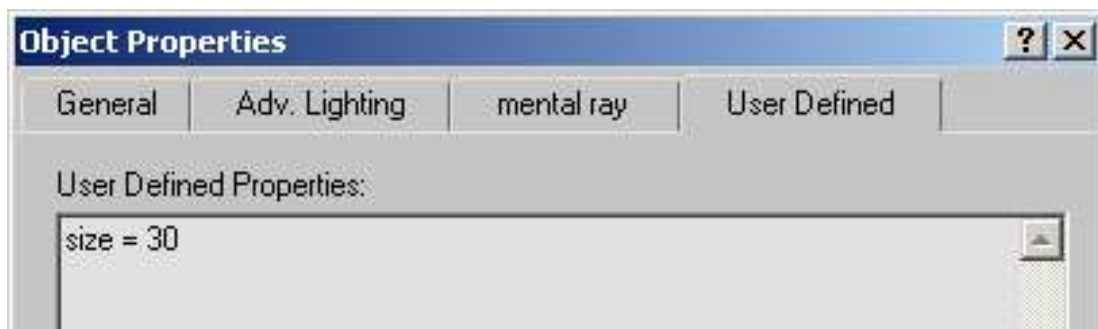


The last version should have as few polygons as possible, with a maximum around 80. The shape and extents of the model must stay the same. If you have to remove something that also reduces the size, place a single triangle at the location to restore the size and proportions.

The names for the reduced bodies should be “**B_**xxx01” to “**B_**xxx04” if you have 4 LOD’s. Normally 4-5 reduced versions are enough.

Special parts like the Cockpit-part do not need to be in the reduced versions.

The engine then uses these reduced versions at a given scale on the monitor. To define this go to the objects properties -> user define. Here you enter for example:



This means: at a size less than 30 percent of the screen this body is used. Normal size schemes are: (30, 15, 8, 3) or for more detailed objects (50, 25, 13, 7, 3).

The body used for 100 or less doesn't need to be defined.

c) Scenes

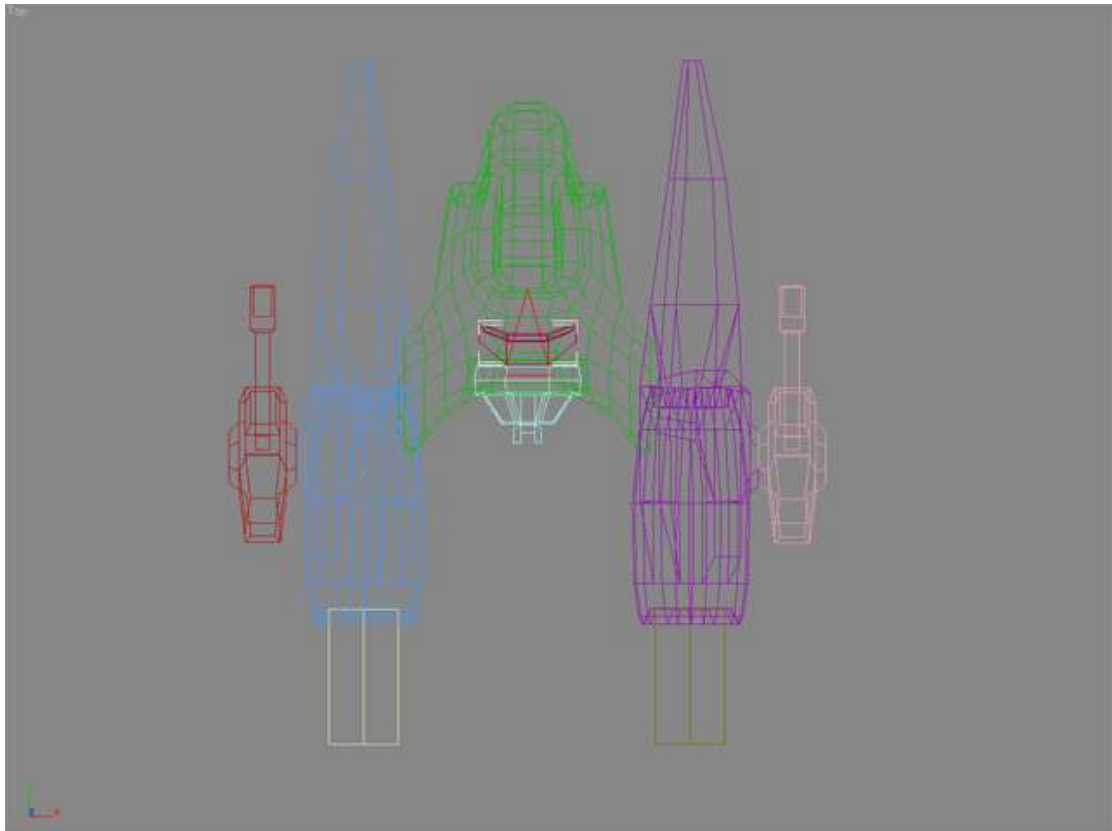
In scenes you can use the models of the bodies you have created or a box object with the same proportions (If you use boxes think of the right orientation!).

They are used as placeholders and just their position, orientation and animation are exported. No geometry or material settings of scenes are exported! As names for the objects you enter the corresponding body number (see exporting bodies).

If you animate objects in scenes you best use the tcb animation controller, as it is most similar to the games controller. Every other controller should work as well, and you always can convert them to tcb via trajectories -> collapse.

One thing that is required: If you have animations like a 360° rotation you need to place a key every 180° .

Here's an example; every body used in the scene has a different color:



The lasers and wings on the left side are the same bodies as on the right side, rotated 180° .

The red triangle is body 940, which marks the center of the scene. All transformations done in the game engine use this as center. Every ship needs to have this body in its scene. For stations body 1 is used instead. Place this body always to the origin of the scene.

When using bodies on 2 sides (like the wings and lasers) remember that mirroring isn't possible. Therefore these objects have to be designed symmetrically.

Laser and cockpit definition

The main cockpit body always has the number one, and therefore is named, for example, 914b_1. The _1 at the end defines it as cockpit number one (Important : it **has** to be at the end!). The corresponding lasers are also named with _1 at the end (i.e. 106b_1).

Like this you can define one main cockpit and up to 6 turret cockpits. See special bodies list for body numbers of turrets and turret lasers.

Changing things like the number of turrets or cockpits will require you to adjust the ship types, see “types definitions” for details.

Thruster and engine definitions

To define engine streaks and effects you will have to rewrite the types settings. But defining their place and behavior needs to be done in the ships scene. The main engine is named Bxxx_1. The thruster bodies are named after the following mask:

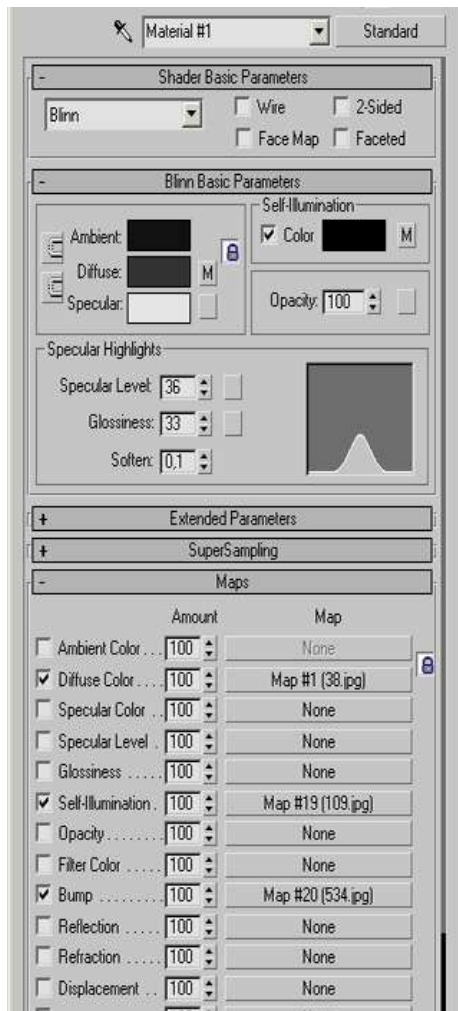
```
ENGINE_JET_MAIN                0x0001
ENGINE_JET_REVERSE             0x0002
ENGINE_JET_ALPHA_POS           0x0004
ENGINE_JET_ALPHA_NEG           0x0008
ENGINE_JET_BETA_POS            0x0010
ENGINE_JET_BETA_NEG            0x0020
ENGINE_JET_GAMMA_POS           0x0040
ENGINE_JET_GAMMA_NEG           0x0080
// following 7's are added to the upper...
ENGINE_JET_UP                  0x0100
ENGINE_JET_DOWN                0x0200
ENGINE_JET_LEFT                0x0400
ENGINE_JET_RIGHT               0x0800
ENGINE_JET_CONTROL             0x0FFE// All rotation control jets
ENGINE_JET_SHOW_JET            0x1000// Show jet body (default: off)
ENGINE_JET_NOSHOW_TRAIL        0x2000// don't show the trail (trail is
enabled if this bit is NOT set) */
```

Add up the effects you want to combine, then convert them to decimal numbers and add them to the thruster bodies names in the scene (i.e.: 237b_32)

2. Material properties

a) Material definitions exported from 3ds max

Material parameters can be edited directly in 3ds max. As base material use a normal **blinn** or **phong** shaded material. Here is a list of exported parameters:



- ambient color
- diffuse color
- specular color
- self illumination strength
- specular level
- opacity

Maps are exported from the following slots:

- diffuse color
- self illumination
- bumpmap (see 3.c for details)

In general the maps are jpg's. If you want to use an alpha channel you have to use the tga file format (24 bit with RLE compression).

There are some texture settings that can only be changed in the source-code; therefore you won't be able to change something on that. For example if you want to use a new texture with alpha channel, you have to replace one of the old tga's.

Blending modes are defined for some textures (multiply, destination, source ...). This isn't possible from 3ds max, transparent materials always use additive (destination) blend mode.

b) Material settings in the .bod file

In .bod files you can see and change material properties after exporting from 3ds max. These settings are now already converted into the engines format, so they differ from what you have seen in max.

Here are two examples:

```
MATERIAL3: 0; 38; 150;150;150; 150;150;150; 229;229;229; 0; 0; 10;
0; 0;0;0; 100; 0; 0; 0; 0;
```

```
MATERIAL5: 0;796; 16; 16; 16; 149;149;149; 229;229;229; 1;100; 10;
0; 1;1;0; 100; 0; 0; 0; 0; 0; 0;
```

Notice that there are 2 different definitions: MATERIAL3 is a definition that uses standard settings defined in the source-code. All values entered here have no effect as the standard values are used. Bumpmaps and lightmaps don't need to be defined. This is useful if you want to use the standard textures supplied with the game and not bother about material properties.

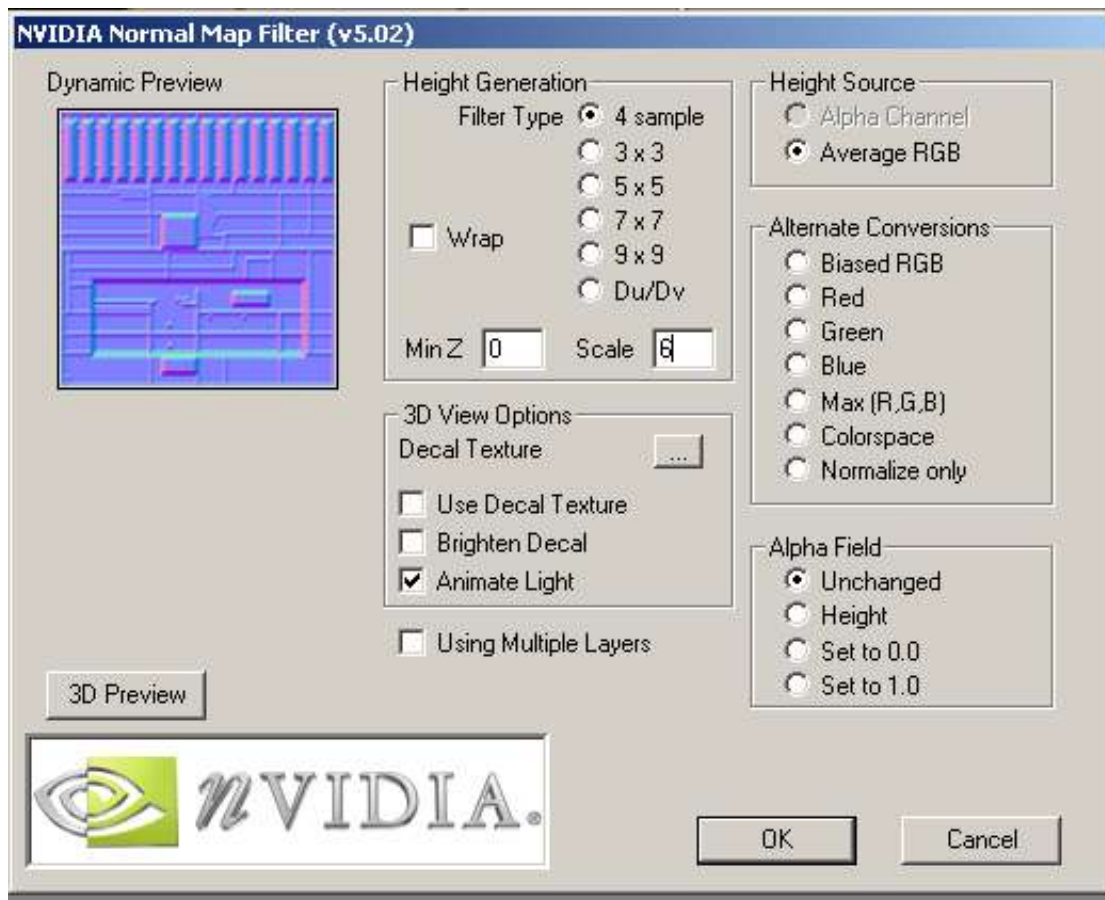
MATERIAL5 is the format that is exported from max by standard. If you want to use MATERIAL3 instead you have to rename it in the .bod file and delete the last 2 numbers (these variables are not supported by MATERIAL3). You can't mix material definitions in a file, you have to choose one.

Now let's look at the variables:

MATERIAL5:	definition format as explained
0;	material number
796;	texture number (file 796.jpg/tga)
16; 16; 16;	ambient (R;G;B)
149;149;149;	diffuse (R;G;B)
229;229;229;	specular (R;G;B)
1;	transparency
100;	self illumination
10;0;	shininess; shinestrength
1;1;0;	destinationblend;2sided;wireframe
100;	texture value (not used)
0;0;	environmentmapID; environmentmapstrength
0;0;	bumpMapID; bumpmapstrength
0;0;	lighmapID; lightmapstrength

What you can see from this : this is a transparent(additive), 2sided texture with 100% self illumination.

c) Bump map creation



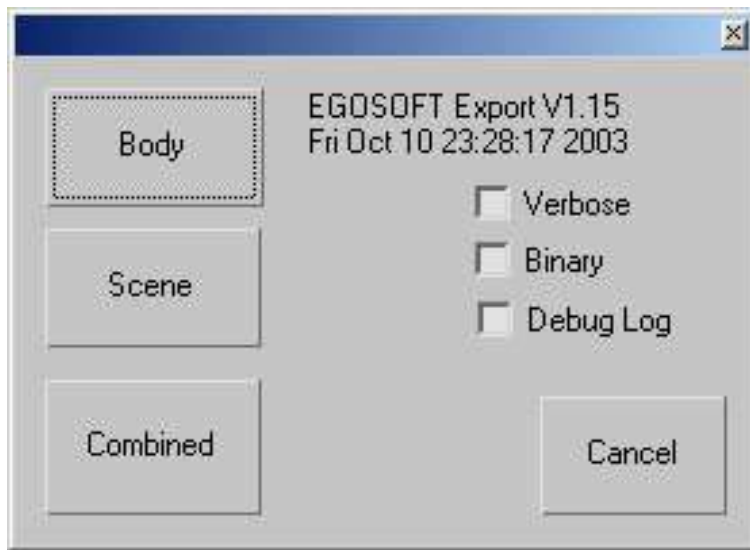
The engine uses DP3 bump mapping. You need to create normal maps for this technique. We use the NVIDIA normal map – filter for Photoshop to create them, but there are also other tools.

To make a normal map you start off by creating a grayscale height map that can then be transformed into the normal map.

As for the blending modes you have to replace existing normal maps to insert new ones. A pair of color map and normal map that correspond would be best.

All textures are located in the subfolder “**tex>true**”

3. Exporter usage



a) exporting bodies

To export a body you first need to copy the **EGOxportX.dle** to your **plugins** folder for 3ds max 5 or 6.

Before you export remember the things mentioned above (pivot to center of body and in origin, reset Xform). Then go to export, select “EGOSOFT Body” from formats and choose your X² location. Bodies are stored in a subfolder named “v”.

Body names are numbers from **00000** to **00999** and from **10000** to **10450**. You can view them via the body viewer in the start menu. If you want to replace existing bodies just put a new body into your v directory and start X² with the parameter “-r”. Bodies in the v directory then have a higher priority and will be loaded. To add new bodies use numbers from **10500** upwards. Just click on body to export.

In body files you can adjust several parameters like the above mentioned material settings. The first number after the material settings is the size of the object.

It gives you the length from the objects center to its farthest point.

```
30000; / Automatic Object Size
```

Between the reduced versions of the body, you can find this:

```
-99; 000000000000010001; /Mark end of part
-99; 0000000000000000; /Mark end of body

/ ---- Next Body of Bodyarray! ---
10; / Automatic pixel threshsize
```

The first two lines are flags that define the end of a part and body.

The automatic pixel threshsize is the value you gave to the reduced version. The next body will be used at less than 10 % screen size in this example.

b) Exporting scenes

Scenes are exported like bodies; their directory is named “**cut**”. Scene names range from **01000.bod** to **07700.bod**. Station and ship models range from **04000** to **04804**. You can view them via the factory viewer in the start menu.

In the exporter just select scene and that’s it.

In the scene file there isn’t much you can change manually. The only easy thing to change is the reference to bodies or scenes. It can be found here at the top of each position / animation block like this :

```
P 0; B 448;  
  { 2; 0; -100; 0; 0.250000; -1.000000; 0.000000; 0.000000; -1;  
-1; } // 0
```

B 448; is the reference to the body 448. It can be exchanged to another body or scene number. Following are the transform informations and time indices (for animation).

c) Combined scenes

Combined scenes are just like normal scenes, but they can also contain geometry and material definitions. If there is an object in a combined scene that has no number as a name, it is stored with its geometry and materials into the scene file.

This has some disadvantages. The model isn’t reusable and you can’t define several LOD’s for it.

Normally this type of scene is only used for cutscenes.

d) Binary and verbose options

binary export writes the files in a compressed form that can be loaded faster and needs less space. The disadvantage is that it is not a readable format so you can’t change things like size or material properties after exporting.

verbose exports the data with a lot of information required during the export process. The files get really huge and shouldn’t be included in a release.

4. Types definitions

a) Particles

Example:

```
1; 3; NULL;

395;0.000000;0.000000;0.500000;0.500000;0.5; 0.1; 0.0; 0.0; -1.0;
1000; 10000; 1200; 300; 2500; lifetimevariation 300;spreadx
0.05;spready 0.05;fadeduration 700;

406;0.000000;0.000000;0.250000;0.250000;0.5; 0.05; 0.0; 0.0; -1.0;
1000; 20000; 1000; 240; 2200; lifetimevariation 150;spreadx
0.01;spready 0.01;fadeduration 300;

406;0.250000;0.500000;0.500000;0.750000;0.4; 0.05; 0.0; 0.0; -1.0;
100; 800; 250; 150; 1700; lifetimevariation 50; rotspeed 2;
```

Short variable description:

PGDescID;NumParticleEmitters;PGDFlags;

MatID;u1;v1;u2;v2;Density;RndVariation;DirX;DirY;DirZ;MinSpeed;MaxSpeed;
Lifetime;MinSize;MaxSize;

Lifetimevariation, spreadx, spready, fadeduration and rotspeed are optional definitions

b) Effects

Example:

```
35;6;0;3000;4.0; // Effect 35 explosion definition M3-M5

EEDF_TIMEOUT|EEDF_CUT|EEDF_GROW|EEDF_RANDOMROT|EEDF_FADEOUT; 0;
4263; 0.015; 0.061; 1500; 0.000; 0; 0; 0; -1; 0;
0.250; 350;

EEDF_TIMEOUT|EEDF_CUT|EEDF_GROW|EEDF_RANDOMROT|EEDF_FADEOUT; 0;
4266; 0.015; 0.076; 2500; 0.000; 0; 0; 0; -1; 0;
0.250; 150;
EEDF_TIMEOUT|EEDF_CUT|EEDF_GROW|EEDF_FADEOUT; 0;
4490; 0.015; 0.030; 2000; 0.000; 0; 0; 0; -1; 0;
0.250; 200;
EEDF_TIMEOUT|EEDF_CUT|EEDF_GROW|EEDF_RANDOMROT|EEDF_FADEOUT; 0;
4265; 0.610; 0.916; 1800; 0.100; 0; 0; 0; -1; 10;
0.250; 100;
EEDF_TIMEOUT|EEDF_GROW|EEDF_FADEOUT; I3DF_2D;
751; 1.068; 1.373; 2000; 0.100; 0; 0; 0; -1; 10;
0.250; 0;
EEDF_GROWANDSHRINK|EEDF_TIMEOUT|EEDF_LIGHT; 0;
14; 0.153; 3.052; 300; 0.800; 0; 0; 0; -1; 0;
0.250; 0;
```

Short variable description:

effectnr; numelements; effectflags; partdelay; partspeedmultiplier;

flags; i3dflags;
bodyid; minsize; maxsize; duration; rndsize; xoffset; yoffset; zoffset; lensflarenr;
atten; halfspeed; time offset;

flags :

EEDF_TIMEOUT
EEDF_CUT
EEDF_GROW
EEDF_GROWANDSHRINK
EEDF_RANDOMROT
EEDF_FADEOUT
EEDF_LIGHT
EEDF_SPEEDDEPENDENT
EEDF_RELATIVESIZE
EEDF_LENSFLARE
EEDF_FADEOUTBODY
EEDF_BULLETRT

I3dflags:

I3DF_HASLENSFLARE
I3DF_2D

c) Ships

example:

13;114;
940;0;0.006;0.0061;0.01;0;3161;25000;1500;102;1;11;373;1000;112;247;4
102;4516;4607;2;3;5;4;9;10;15;21000;21000;73;3;73;4;72;2;0;0;0;0;0;0;
12;5;1;70000;37;38;0;1;700000;25;1;0;SS_SH_A_TL;

Short variable description:

file version; number of lines;

The **file version** is currently 13. If the ship types are extended with new parameters, 2 will be added to the version number. If the ship type file is converted to extended version 14, additional laser, cockpit and turret info is added to the types file.

number of lines is the number of ship records (lines) following this first line.

3dbody;	Center Body number for this object (if the ship has a scene)
3dcompbody;	Body number used for detail display inside menus.
Rotation Alpha;	The maximum turning rate (in rotations per second) for Yaw.
Rotation Beta;	The maximum turning rate (in rotations per second) for Pitch.
Rotation Gamma;	The maximum turning rate (in rotations per second) for Roll.

GSubType;	Galaxy SubType. For ships this is the ship class (TL, TS, M0, M1, M2, M3, M4, M5, TL_P, TS_P, GO, M6, TP)
TextID;	textid of ship class name in textpage 17. textid+1 is the long description of the ship.
SpeedMax;	Maximum speed in units/second.
SpeedDeltaPerSec;	Acceleration in units/s ²
SoundIDOnFlyBy;	3d soundeffect of ship engine.
AverageReactDelay;	
EngineEffect;	Visual effect number used at ship engines
GlowEffect;	
PowerGeneration;	Generated energy per second.
3DSoundVolMin;	Minimum sound volume for ship with speed
3DSoundVolMax;	Maximum sound volume for maximum speed
CutID;	CutID used for ship scene.
CockpitBody;	BodyID or CutID used for Cockpit.
LaserMask;	Bitmask with usable laser subtypes for this ship.
MaxNumLasers;	Max. number of lasers - valid only for converted type
file	
MaxShieldType;	Max. usable shield subtype.
MaxNumShields;	Max. number of installable shields.
MaxRocketType;	Max. usable rocket subtype.
MaxNumRockets;	unused.
MaxExtraSpeed;	Max. extra speed in 10% steps (10 = +100% Upgrade)
MaxExtraRotationSpeed;	Max. extra rotation speed in 10% steps (10 = +100% Upgrade)
TradeContainerSize;	Default number of ware containers to carry.
MaxTradeContainerSize;	Maximum of installable containers (with container upgrades).
CockpitType1;	Index into static cockpit types for extra cockpit #1
CockpitTextID1;	TextID of extra cockpit #1 (0=Main,1=Front,2=Back,3=Left,4=Right,5=Up,6=Down).
CockpitType2;	
CockpitTextID2;	
CockpitType3;	
CockpitTextID3;	
CockpitType4;	
CockpitTextID4;	
CockpitType5;	
CockpitTextID5;	
CockpitType6;	
CockpitTextID6;	
DockBaySize;	Number of ships which can dock at this (carrier) ship.
MaxTKClass;	Maximum Transport Class ship can carry.
DefaultRace;	Default race for this ship type.
HullStrength;	
ExplosionEffect;	Effect for main body of exploding ship.
PartExplosionEffect;	Effect for smaller explosions of ship parts (subbodies)
EngineTrail;	Type of trail - index into types\particles3.txt.

only for converted version 14:

NumberOfCockpits;

for each cockpit:

Class;	unique ID defined in ship scene.
TurretID;	
BodyID;	
PathID;	
NumberOfTurrets;	

for each turret:

LaserPos;	Laser position of first laser in this turret
NumLasers;	
Class;	unique control ID, which cockpit can control which turrets.
NumGuns;	

for each gun:

LaserPos;	
NumLasers;	
BodyID;	body id of gun body in ship main scene
PathID;	path id of gun body in ship main scene
BodyID2;	body id of gun barrel in gun scene (or -1)
PathID2;	path id of gun barrel in gun scene (or -1)

common entries for all object types:

Volume;	Volume of one cargo unit.
RelValue;	RelValue * valuefactor for a ship = ship price.
PriMaxP;	price variation as primary resource
SecMaxP;	price variation as secondary resource
TK;	Transport Class (as ware)

5. Special bodies list

jet bodies: these bodies define the position of the engine trails. To use them place them and name them with **_1** after the number, i.e.: **905_1b**

905, 912, 773, 770, 779, 782, 783, 67, 102, 105, 108, 114, 127, 237, 239, 272, 273, 566, 10209, 10318, 10320, 10434

2D bodies: are always facing the camera. Used for nebulae and glows

590, 591, 509, 555, 585, 420, 528, 529, 531, 532, 536, 537, 538, 539, 579, 709, 710, 711, 712, 713, 726, 727, 728, 729, 730, 751, 775, 10001, 10002, 10003, 10004, 10005, 10006, 10007, 10008, 10009, 10010, 10011, 10012, 10078, 10079, 10083, 10084, 10085, 10086, 10087, 10088, 10089, 10090, 10091, 10092, 10093, 10094, 10095, 10096, 10097, 10098, 10099, 10100, 10101, 10102, 10103, 10104, 10105, 10106, 10107, 10108, 10109, 10110, 10111, 10112, 10113, 10114, 10115, 10116, 10117, 10118, 10119, 10120, 10121, 10122, 10123, 10124, 10125, 10126, 10127, 10128, 10129, 10130, 10131, 10132, 10133, 10134, 10135, 10136, 10137, 10138, 10139, 10140, 10141, 10142, 10143, 10144, 10145, 10146, 10147, 10148, 10149, 10150, 10151, 10152, 10153, 10154, 10155, 10156, 10157, 10158, 10159, 10160, 10161, 10162, 10163, 10164, 10164, 10165, 10166, 10167, 10168, 10169, 10170, 10171, 10172, 10173, 10174, 10175, 10176, 10177, 10178, 10179, 10180, 10181, 10182, 10183, 10184, 10185, 10186, 10187, 10188, 10189, 10190, 10191, 10192, 10193, 10194, 10195, 10196, 10197, 10198, 10199, 10200, 10201, 10202, 10203

11, 189, 213, 274, 282, 284, 573, 574, 578, 436, 440, 449, 552, 733, 734, 10322, 10323, 10324, 10325, 10326, 10327, 10328, 10329, 10434

Logo bodies: used to display the player logo on stations and ships

425, 80, 476, 424, 452

Dock bodies and corresponding scenes

71, DWF_OUTERDOCK, { 600, 601, 602, 603 }
79, DWF_OUTERDOCK, { 604, 605, 606, 607 }
77, DWF_OUTERDOCK, { 608, 609, 610, 611 }
10213, DWF_OUTERDOCK, { 650, 651, 652, 653 }
10216, DWF_OUTERDOCK, { 654, 655, 656, 657 }

875, DWF_INNERDOCK, { 639, 640, 642, 641 }
10224, DWF_INNERDOCK, { 658, 659, 660, 661 }
10226, DWF_INNERDOCK, { 662, 663, 664, 665 }

Gun (-tower) bodies with scenes for recoil and rotation definitions for the bodies.

106, GUNF_FIXED, { 4291, 4292 }
107, GUNF_FIXED, { 4293, 4294 }

91, GUNF_ROTATE, { 4290 }, {{90, GUNPARTF_ROTATEALPHA},{89, GUNPARTF_ROTATEALPHA|GUNPARTF_ROTATEBETA}}

708, GUNF_ROTATE, { 4295 }, {{980, GUNPARTF_ROTATEGAMMA},{981, GUNPARTF_ROTATEBETA|GUNPARTF_ROTATEGAMMA}}

Turret Bodies

939, 950, 812, 708

Display bodies

52, 104, 115, 192, 714, 715, 716, 717, 718, 889

Big hangar bodies

53, 856, 10376