

# STEEL DIVISION II

## MODDING MANUAL



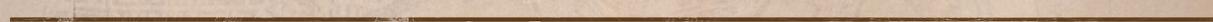
# INTALLATING MODDING TOOLS FOR WARGAMING

## AND GOG CUSTOMERS

For those who have purchased their game on Wargaming or GoG, you must download the modding tools at this address:

<https://www.eugensystems.com/downloads/ModdingAssets.zip>  
<https://www.eugensystems.com/downloads/ModdingTools.zip>

130-17  
119



# CREATING A NEW MOD

Go to the game installation directory (in steam: Library -> Steel Division 2 -> Properties -> Local Files -> Browse Local Files), and enter the Mods subdirectory

Open a console in the `Mods` directory (either by using `cd <path_to_Mods_folder>` or by shift + right click in windows explorer > «open console here») and run `CreateNewMod.bat` with the mod name as argument.  
Example:

```
CreateNewMod.bat MyAwesomeMod
```

The name must be unique ; it will not work if a folder already exists with that name. If all goes well, you should see a message indicating that the mod was created, and a new folder named `MyAwesomeMod` has appeared. It should contain the following directories:

- AssetsModding
- CommonData
- GameData
- Gen

and the following files:

- base.zip
- GenerateMod.bat
- UpdateMod.bat
- UploadMod.bat

These 4 files (`base.zip` and the 3 `.bat`) are part of the modding tools, please do not modify them or you will likely compromise your mod. If something is missing, try to delete the mod folder and recreate it. You are now able to start working on your mod, please read on!

---

# GENERATING A MOD

A generation step is needed before a mod can be activated in-game or uploaded to the Steam Workshop. To proceed, launch **GenerateMod.bat** in your mod directory. The screen will become black for a short period of time.

Once you have generated your mod a default SteamID will be given to it. You need to upload your mod to give him a real SteamID. Even if you haven't finish your work, you can upload it and hide it on your workshop mod page.

# EDITING A MOD CONFIGURATION FILE

Go to the <C:\Users\USERNAME\Saved Games\EugenSystems\SteelDivision2\mod\MyAwesomeMod> folder, and open the Config.ini file:

```
Name = MyAwesomeMod ; The name of your mod
Description = My Mod is Awesome ; The description of
your mod
TagList = ; Don't edit that field
PreviewImagePath = C:\my_mod_image.png ; An image lo-
cated on your hard drive
CosmeticOnly = 0 ; Does your mod affect gameplay or not
? If activated, the mod will not be shared when crea-
ting a lobby
ID = 902495510 ; Mod Steam ID, do not modify
Version = 0 ; Value to increment when an update in this
mod is incompatible with the current version
    DeckFormatVersion = 0 ; Increment this value to
    invalidate all decks for this mod
    ModGenVersion = 0 ; Don't edit that field
```

If you want to change the mod name, you have to : change the «Name» field in the configuration file, change the folder name in <C:\Users\USER->

---

NAME\Saved Games\EugenSystems\SteelDivision2\mod, change the folder name in the steam directory.

The «ID» and «TagList» are filled when UploadMod.bat is launched.

## UPLOADING A MOD

To upload your mod on steam, launch UploadMod.bat in your mod directory. The screen will become black for a short period of time.

This process will update «TagList», «ID» and «ModGenVersion» fields. Once you've uploaded your mod, it's ready to use on steam workshop, you can access its own page, set its visibility, description, image, etc...

## UPDATING A MOD – WHEN A NEW GAME PATCH IS RELEASED

With the game updating through Steam, it will be needed from time to time to update your mod so that it still work with new versions. We created a small tool to help you with this, requiring minimal manual work. Before talking about it, let's see what the problem is.

## WHY DO WE NEED TO UPDATE MODS?

Imagine you have a mod that modifies file A.ndf . When this mod was created, the original file (from game data) was:

---

```
Thing is TThing
(
    SomeInt = 42
)
```

And you modified it to be:

```
Thing is TThing
(
    SomeInt = 43
)
```

No problem here, that's the point of modding. Now what if we at Eugen need to modify this file as well, and end up adding content to it so it looks like this in the new version of the game:

```
Thing is TThing
(
    SomeInt = 42
)
```

```
Thing2 is TThing
(
    SomeInt = 80
)
```

You will want to keep your change to Thing/SomeInt's value, and have the new Thing2 as well. Wanted result for your mod:

```
Thing is TThing
(
    SomeInt = 43
)
```

```
Thing2 is TThing
(
    SomeInt = 80
)
```

## UPDATING MODS

The solution is quite easy, just run UpdateMod.bat in your mod folder. This will merge your changes with the last version of the game data. If you're lucky, the process will work automatically and display a message indicating success. However, it is possible that you encounter conflicts during the operation. Conflicts arise when both the new version of the

---

game and your mod modify the same part of a NDF file. In this case, the process can't guess what result you want, and need manual intervention.

## SOLVING CONFLICTS

Imagine once again that you're modifying A.ndf.

Original version (game data when the mod was created):

```
Thing is TThing
(
    SomeInt = 42
)
```

Modded version:

```
Thing is TThing
(
    SomeInt = 43
)
```

And with the following update, the game's version of the file becomes this:

```
Thing is TThing
(
    SomeInt = 80
)
```

Both the game and the mod have modified the same line. You can easily see that it is entirely up to you what final value should Thing/SomeInt have in your mod, and that no tool can decide it on its own.

The updating tool will stop and warn you about the files that contain conflicts. To handle this case, the tool will mark conflicts like this one with special delimiters:

```
Thing is TThing
(
<<<<<<<
    SomeInt = 80
|||||||
    SomeInt = 42
=====
    SomeInt = 43
>>>>>>>
)
```

---

- After <<<<<<< is the new version from the game's data
- After ||||| is the base common ancestor (in our case original game's data)
- After ===== is your mod's version
- >>>>>> marks the end of the conflict

Resolving the conflicts consists in modifying the file so that it makes sense to the game that will load it (ie. get rid of the delimiters and have a single `SomeInt = ...` line) and has the wanted value for your mod. You are free to keep your mod value (43), to go back to the original value (42), to get the new value (80) or to decide an entirely new.

The process is very similar to a three-way merge like found in version control software such as git.

# MODS BACKUP

Inside your mod directory you will find 2 essentials scripts :

- `CreateModBackup.bat` : which will allow you to create a backup of the current status of your mod. Backups are basically a copy of your mod files archived and stored in `Backup/` folder. A backup will be generated every time you upload your mod, considering that if you upload your mod that means that you're currently on a stable version. You can generate a backup yourself by using this script it will ask for a name, you can leave the entry empty it will use a generated name based on current time.
  - `RetrieveModBackup.bat` : this script will allow you to reset your mod to a previous version stored in `Backup/`, it will simply copy and force paste the backup version in your current working space so be careful before any `RetrieveModBackup`. The script will prompt you for a name, you can leave the entry empty it will select the latest backup generated.
-

# NDF MAP

Every files location you could want to edit.

## INTERFACE

If you want to mod the interface, you gonna need to look at GameData/UserInterface/.

All files under Style/ folder will be related to global objects and reused style accross different screens.

All files under Use/ will describe each screen or part of screen, separated in folder following their usage in game :

- InGame : everything after a match session is launch
- OutGame : everything related to main menu, loading, etc...
- Common : everything common to both In and OutGame.
- ShowRoom : well it speaks for itself, it's everything related to the showRoom.
- Strategic : everything related to Steelman mode

## ADDING A NEW TEXTURE

If you want to add a new texture to replace one already used in game, you can change its declaration by adding 'Modded' at the end of its type ( TUIResourceTexture\_InGame will become TUIResourceTexture\_InGameModded, TUIResourceTexture\_Outgame will become TUIResourceTexture\_OutGameModded, etc.), then set its path relative from AssetsModding/.

All your interface assets must be located under AssetsModding/ and your FileName should be relative to this directory.

For exemple, if you want to change the riflemen's label texture you need to find Texture\_RTS\_Rifle in GameData/Generated/Gameplay/UI/Resources/Textures/RTSNatoTextures.ndf. It looks like this :

---

## TEXTURE\_RTS\_RIFLE\_BEFORE

```
Texture_RTS_Rifle    is    TUIResourceTextureWithUse-
fulnessMask
(
    TypeMask          =    TypeMask/Common
    CivMask           =    CivMask/Common
    FileName          =    `Assets\2D\Interface\UseInGame\
LabelIcons\RTS\default.png`
)
```

To change this texture you will need to modify TUIResourceTextureWithUsefulnessMask to TUIResourceTextureWithUsefulnessMaskModded and the FileName to 'MyAwesomeTexture.png', it should look like :

## TEXTURE\_RTS\_RIFLE\_AFTER

```
Texture_RTS_Rifle    is    TUIResourceTextureWithUse-
fulnessMaskModded
(
    TypeMask          =    TypeMask/Common
    CivMask           =    CivMask/Common
    FileName          =    `MyAwesomeTexture.png`
)
```

So the full path to MyAwesomeTexture.png is [SteamLibraryPath]/SteelDivision2/Mods/[YourModName]/AssetsModding/MyAwesomeTexture.png but like FileName is relative to [SteamLibraryPath]/SteelDivision2/Mods/[YourModName]/AssetsModding/ you just have to set the path from this directory.

And that's it, your texture will replace the label texture for riflemen.

## LOCALISATION

You will sometime find variables called TextToken or NameToken. Every time you see a name containing «Token», it means that the string is supposed to be localised and the content of the variable is a reference token for localisation files.

You can find the mod localisation file here : [GameData/Localisation/](#)

---

## MODDING.csv

You can either create your own token to replace some of our tokens or you can override our tokens' value. All you have to do is set the token value under the TOKEN column and set its content under REFTEXT column.

### TokenSize

Your token size cannot exceed 10 characters!

You can find all localisation entries and translations in Mods\Utils\LocalisationEntries\:

- entries.csv : contains all Tokens, refTexts (not necessarily the version used in game) and the ID associated to the Token which will help you to find the translated value.
- RefStrings.csv : contains informations about each string : context and comment.
- Translation.csv : contains each translation for each string, to find the localised version of the token you need its ID, there is no token anymore in this file.

# GAMEPLAY

## UNITS

All unit descriptors are located in GameData/Generated/Gameplay/Gfx/UnitDescriptor.ndf. You will find various other interesting descriptors in the same folder, like WeaponDescriptor.ndf and AmmunitionDescriptor.ndf.

## DIVISIONS

---

Divisions are located in `GameData/Generated/Gameplay/Decks/Divisions.ndf`. Divisions are composed by packs, a pack is a bundle of the same units, all packs are located in `GameData/Generated/Gameplay/Decks/Packs.ndf`.

## GAMEPLAY CONSTANTS

`GameData/Gameplay/Constantes/` contains all files defining generic gameplay features, for example:

- in `GDConstantes.ndf` you will find the property `GhostDefaultColor` which define the color for ghosts
- `RelativeBonusFluxByIADifficultyAndPhase` and `RelativeBonusPhaseStartByIADifficulty` allow you to give more income money to the AI.

## WEAPONS

Weapons are mainly described in `WeaponDescriptor.ndf` and `Ammunition.ndf`, located in `GameData/Generated/Gameplay/Gfx`.

The first file describe, for each units, how its turrets are configured, and which weapons are mounted. For example, you'll find the properties defining the number of salvos for each weapon, the angle ranges and rotation speeds of the turrets, and some UI-related parameters. You will also find reference to `TAmmunitionDescriptor` objects, which are defined in `Ammunition.ndf`.

In these objects there are many useful properties, here are some of them:

- `Arme`: the type of damage the weapon will inflict. For AP weapon, you can use values from `AP_1` to `AP_30` (see `CommonData/Gameplay/Constantes/Enumerations/ArmeType.ndf`).
  - `PorteeMaximale`: range of the weapon. Some of you already figured out how the meaning of the unit (cf. this post).
  - `HitRollRuleDescriptor`: describes how the hit and pierce chance are computed (see `GameData/Gameplay/Constantes/HitRollConstants.ndf`).
  - `IsHarmlessForAllies`: enables/disables friendly fire.
-

There are also other files that you might want to take a look at:

- `GameData/Gameplay/Unit/CriticalModules/CriticalEffectModule_GroundUnit.ndf`: describes the critical effects distribution on ground units (typically tanks).
- `GameData/Generated/Gameplay/Gfx/ArmesArmures.ndf`: the damage factor used given the firing weapon and the target's armor.

## AI

`GameData/Gameplay/Skirmish` contains files about the AI. An artificial intelligence is called strategy.

There are multiple strategies (used in skirmish, multiplayer, historical battles and Steelman) depending on game difficulty and settings, located in `GameData/Gameplay/Skirmish/Strategies/` subfolders.

Strategies are plugged in the `GameData/Gameplay/Skirmish/IASkirmishDescriptors.ndf` file.

A strategy contains a list of Generators and Transitions. The AI navigates from one `TSequenceGeneratorDescriptor` to another depending on `TIAGeneralStrategyTransition`.

A `TSequenceGeneratorDescriptor` describes the AI's priorities while it is active. MacroAction behavioral descriptors used in the Generators are located in `GameData/Gameplay/Skirmish/SkirmishMacros` folder. Most MacroActions control unit groups with specific objectives (attack, defend, etc.). The required units for each action are described in `TPoolModel` objects which are freely modifiable. Note: new `TPoolModel` objects need distinct GUID fields.

Global settings about the AI are defined in `GameData/Gameplay/Constantes/IAStratConstantes.ndf` and affect both skirmish and campaign AI.

# SOUND

---

You can add your own sound to SteelDivision 2. To do so, you need to put them under GameData/Assets/SoundModding/. (Create folders if they are not already here)

To redirect an existing sound to your own sound, you need to replace its TSoundStream, Template\_GlobalAcknow, Template\_UnitAcknow or TemplateSoundDescriptor by the modded version :

TSoundStream should be replaced by SoundStreamModded

Template\_GlobalAcknow should be replaced by Template\_GlobalAcknowModded

Template\_UnitAcknow should be replaced by Template\_UnitAcknowModded

TemplateSoundDescriptor should be replaced by TemplateSoundDescriptorModded

Then set the FileName to your sound (the path is relative to GameData/Assets/SoundModding/).

Example: you have your GameData/Assets/SoundModding/My20mmShot.wav, to replace the CC\_AutoCanon\_20mm sound you should find this template :

```
template          Template_SoundDescriptor_CC_Autocanon_20mm
[
    UseSpecialLoop,
]
is TSoundDescriptor
(
    TheSoundStream          = TSoundStream ( FileName = 'GameData\Assets\Sons\SFX_Normandie\CC\CC_Autocanon_20mm.wav' )
    SoundSettings           = $/SoundSettings/SFX_SoundSettings
    UseSpecialLoop          = <UseSpecialLoop>
)
```

Then replace TSoundStream by SoundStreamModded and set the FileName to 'My20mmShot.wav'. It should look like this :

```
template          Template_SoundDescriptor_CC_Autocanon_20mm
[
    UseSpecialLoop,
]
is TSoundDescriptor
```

---

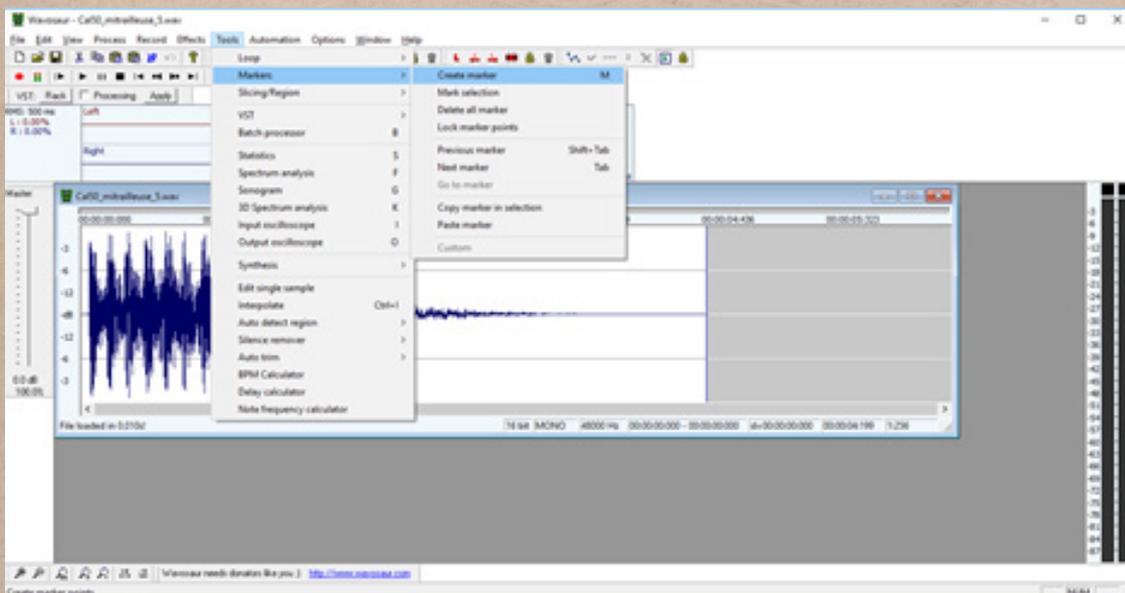
```
(
    TheSoundStream          = SoundStreamModded(
FileName = 'My20mmShot.wav' )
    SoundSettings           = $/SoundSettings/SFX_
SoundSettings
    UseSpecialLoop         = <UseSpecialLoop>
)
```

## SPECS

**SoundFormat**  
Only OGG and WAV files are supported!

Sound FX (shot fired sound, impacts, etc...) must be in a specific format :  
16 bits MONO WAV (44100Hz).

## SPECIFIC CASE : CONTINUOUS FIRE

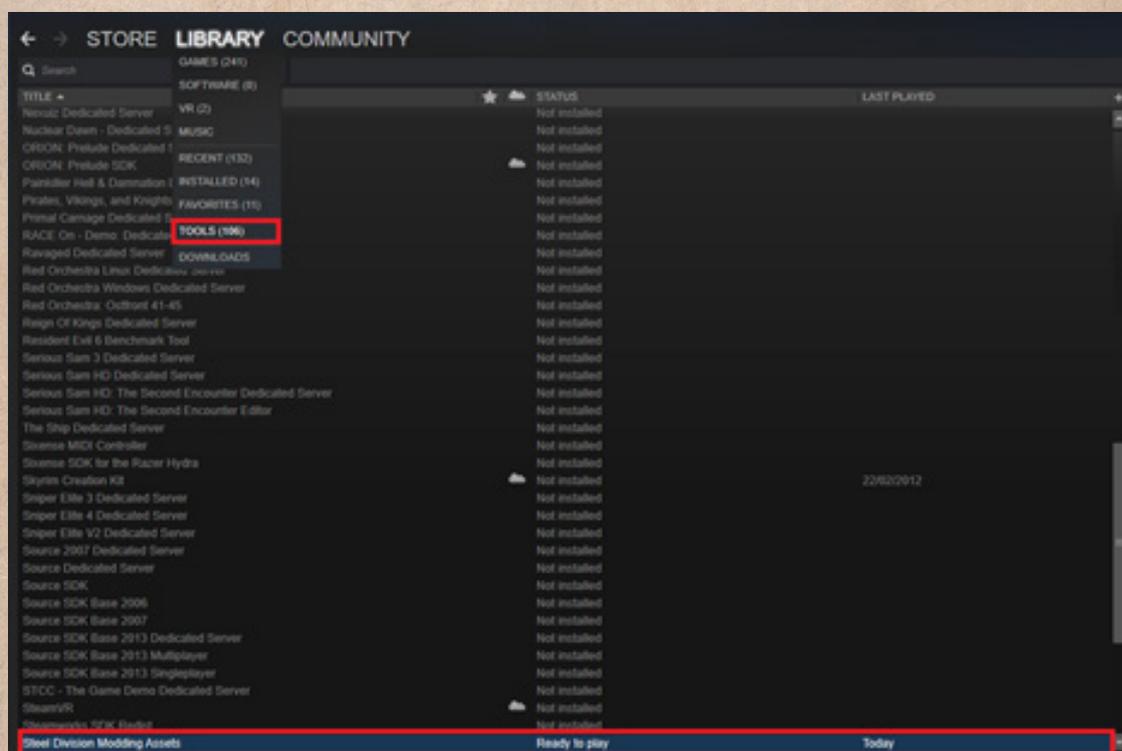


For continuous firing weapons you will need to mark a loop into your file. You need to get the software Wavosaur, load your file then tools>Markers>Create a marker (shortcut 'M'). You need to mark the loop begin and end. That way, the sound will begin

at the start of your file, then loop between the 2 markers until it ends, then it will play the end of the sound.

# SKINNING

In order to reskin our units you will need to download a new Modding tool located on steam under Library > Tools > Steel Division 2 Modding Assets



By downloading this tool you will be able to access our models textures and Ase2NdfBin (necessary to generate your mod). You will find them here : [steamapps/common/Steel Division 2 Modding Assets/Mods/Assets/3D/Units](https://steamapps.com/common/Steel%20Division%20Modding%20Assets/Mods/Assets/3D/Units).

Each type of units (char, avion, infanterie, vehicule) is archived, you will need to unarchive it before selecting which unit you want to reskin. Once you have selected your units to reskin, you will need to copy the unit's files (all png or jpg and Ase2NdfBin) and past them in your mod, with the same path.

Example : you want to reskin the US M4A1 Sherman, you go to [steamapps/common/Steel Division 2 Modding Assets/Mods/Assets/3D/Units/USA/Char.zip](https://steamapps.com/common/Steel%20Division%20Modding%20Assets/Mods/Assets/3D/Units/USA/Char.zip), you unzip it, you go to Char\M4A1\_Sherman\ copy all files and past them in [steamapps/common/Steel Division 2/Mods/\[YourMod\]/Ga](https://steamapps.com/common/Steel%20Division%20Modding%20Assets/Mods/[YourMod]/Ga)

meData/Assets/3D/Units/USA/Char/M4A1\_Sherman/.

### **Warning LODs !**

Don't forget to copy and modify unit's LODs and MID versions, otherwise you might have some surprise in game.

Once you have copied and modified the unit's textures, you have to generate your mod. You can enjoy your mod now.

# **MESH**

## **SETUP YOUR ENVIRONMENT**

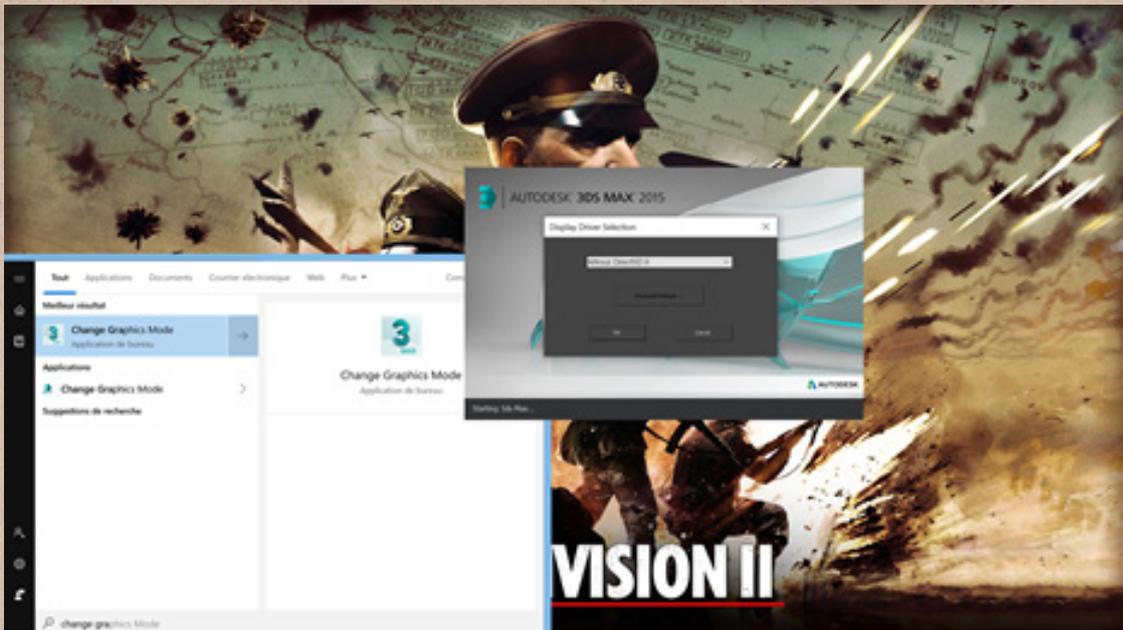
To create or modify mesh in order to integrate them in Steel Division 2 you will need 2 plugins :

- Ase2NdfBinPlugin\_Install.exe
- eugMaterial\_Install.exe

Quit 3ds Max if it's running, install those plugins, you can find them under steamapps/common/SteelDivision2/Mods/Utils/Meshes/PluginsInstallers/, then restart 3ds Max.

Once you have done that you will need to setup your viewport settings. To do so, use Autodesk 3dsMax 2015/Change Graphics Mode and set it to «Nitrous Direct3D 9»

---



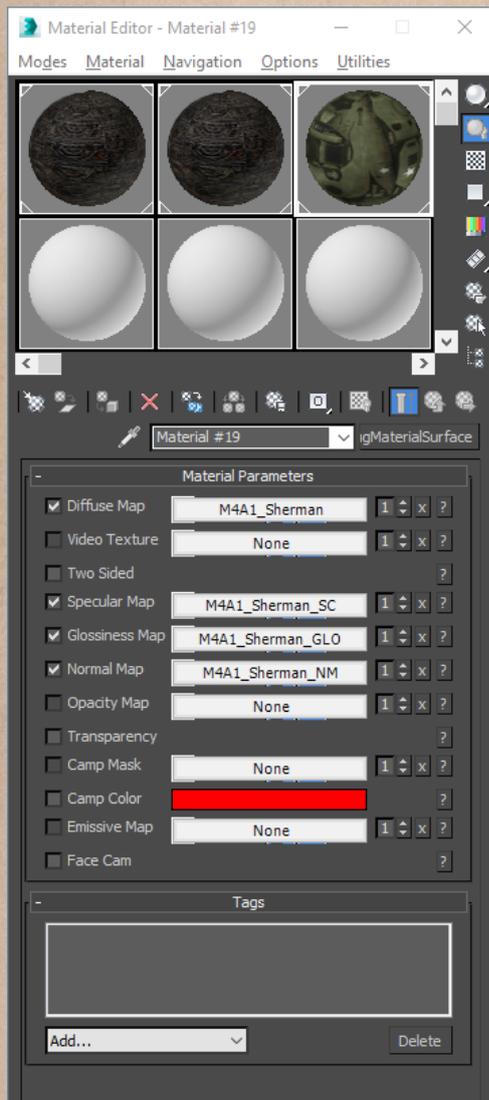
## ASE2NDFBIN

The file need to be exported from 3dsMax in this format to be used ingame.

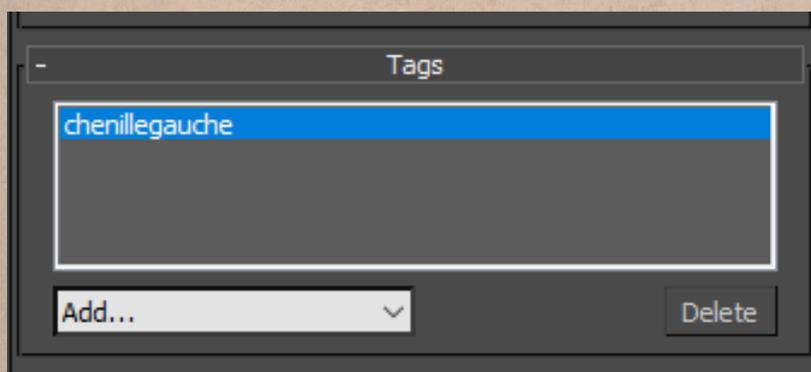
## EUGMATERIAL

EugMaterial =in 3dsMax, we don't use Standard Material. We use this specific material that simulates the ingame visual of 3d models.

---



Along with classic channel (Diffuse, Specular, Glossiness, Normal Map and Opacity), you'll notice a "Tags" section. You can add tags on some elements to have specific behavior. For example, by adding 2 different materials (with its own separated textures) with tags "chenillegauche" and "chenilledroite" on left track and right track of a tank, you'll see the tracks move ingame when units is moving.



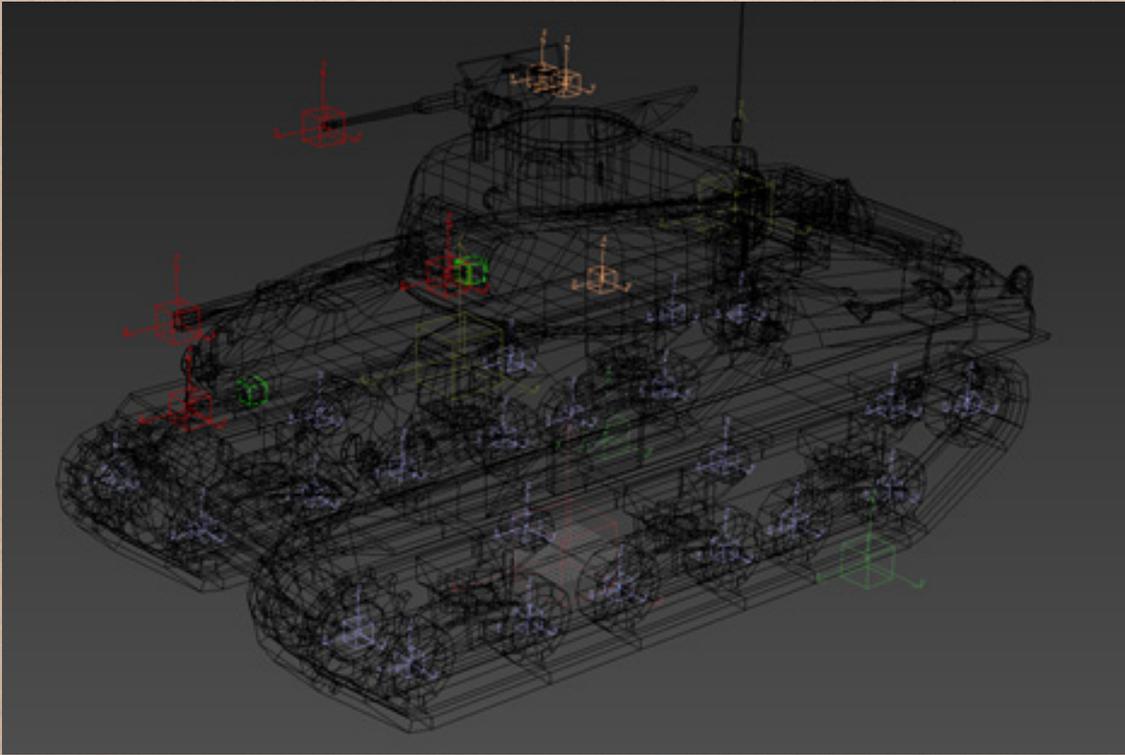
# 3D MODEL DESCRIPTION - EXAMPLE WITH M4A1\_SHERMAN MESH

To work properly ingame, a unit's 3D model need different elements with a specific nomenclature (some words are in French).  
Let's take the example of the M4A1 Sherman tank.



If you check inside the 3d file, you can see 2 sort of elements that compose the model :

- 3d meshes parts : the guns, the turret, wheels, tracks, etc. These elements are visible ingame.
  - "Point helper" boxes : These boxes are invisible ingame. We use the position (but also scale and orientation) of these boxes for different things like FX spawning points or procedural animations on some visible elements.
-



## 3D MODEL SPECS

### 3D MESH

LODs are not generated, so you need to make your own version of High, mid and low poly:

High = 6/7000 triangles max.

Mid = around 2000 triangles.

Low = less than 500 triangles.

### TEXTURES

High = 2048 x 2048 px Diffuse, Glossiness (\_GLO), Specular (\_SC), Nor-

---

mal Map (NM) and sometimes Alpha (\_A) + 256 x 256 px separated textures for tank's tracks. This texture must be separated because when unit is moving, UVs coordinates of tracks meshes will move too, to give the illusion of real tracks.

Mid = 1024 x 1024 px textures. This time, tracks textures must be inside the main textures. They don't move at this level of details.

Low = 256 x 256 px textures.

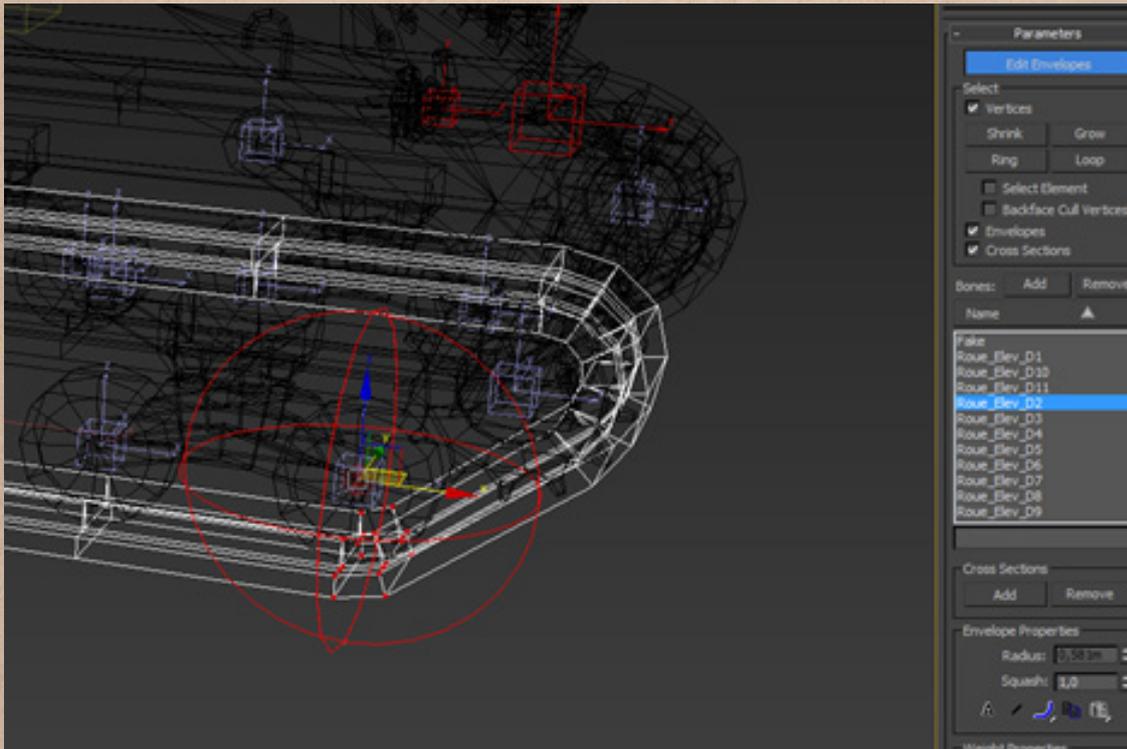
## DIFFERENT USE OF HELPERS BOXES

### FX HELPERS

- • FX\_Tourelle\_x\_tir\_0x = When unit is firing, weapon FX will spawn here.
- • FX\_Stress\_01/02 = Spawn position of smoke FX when unit is taking damages.
- • FX\_Fumee\_Chenille\_D1/G1 = Smoke FX when unit is moving.

### SKINNING HELPERS

- Roue\_Elev\_Dx/Gx = Wheels meshes (Roue\_Dx/Gx) are linked to this type of boxes. When unit is moving, this box move vertically and pull up/down the wheel and also some vertices of the tracks, thanks to the "Skin Modifier". The rotation is applied directly on the wheel mesh.
  - Fake helper = All tracks vertices not "skinned" to a wheel are attached to this box. This box is linked to the main part "Chassis". When the unit will move/stop, it'll drag those vertices and we'll have a more realistic movement.
-



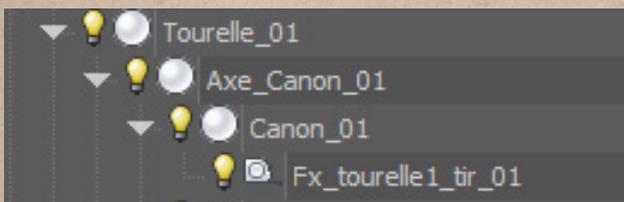
## 3D MODEL HIERARCHY CONSTRAINTS

“Chassis” is the main element, it’s linked to nothing, but lot of other elements are linked to it.

A “Turret” works always this way (-> means “linked to”) : Fx\_Tourelle1\_tir\_01 -> Canon\_01 -> Axe\_Canon\_01 -> Tourelle\_01.

Then the “Tourelle\_01” element is linked to the chassis (or to another turret if it’s a secondary turret).

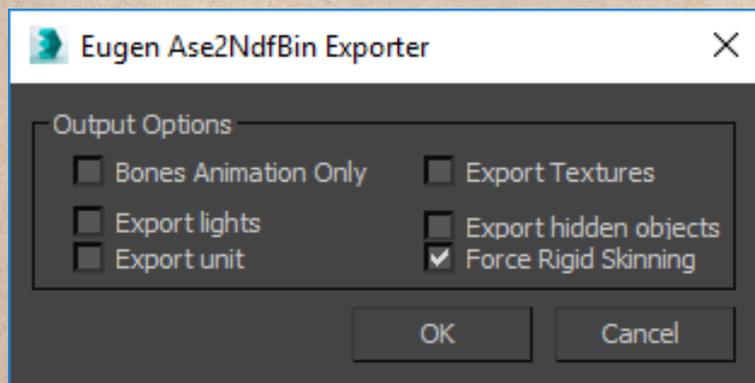
Helpers (smoke, stress) can be linked to the chassis or to a turret, or to nothing.



# EXPORT

As told earlier, once the model is ready, you must export in “.Ase2NdfBin” format.

The exporter will ask you what you want to export, you just have to check the box «Force Rigid Skinning».



# ADDING YOUR MESH TO YOUR MOD

Once you have your Ase2NdfBin, you have to move it with its textures under GameData/Assets/ (you may have to create the path yourself). Otherwise your mesh will not be generated and will not be usable in the game.

Now you can add references to your mesh. For exemple you can replace the US Sherman M4A1 by modifying the unit mesh description located under GameData/Gameplay/Gfx/Units/US/Char/Sherman\_M4A1\_US.ndf. You have to change all FileName to redirect to your own mesh and replace the CivMask variable by “CivMask/MeshModding”.

# KNOWN ISSUES

- In case you have scaled your mesh, you need to run a ResetXForm after that.
-

- Some 3ds Max Modifiers can cause some errors during export, to be sure collapse as many Modifiers as possible before the export.

7 1  
130 17  
119  
1