Bandits: Phoenix Rising

The Unofficial Modding Manual

V.0.4

Created By: Kristian ”Wolfy/Sajber” Nilsen

Index

Introduction

I actually started my relationship with Bandits many years ago when is was on work practice on a distribution company called Pan Vision. While there I got the opportunity to work as a beta tester for Bandits and its creators GRIN.
I actually ended up in the written manual under the section “Thanks to:”.
Little did I know that I one day would mod the game and even less write a modding manual for it, but here I am!

***

This manual is divided in sections, starting with tutorials and ending with a glossary.

The image editing tutorials are written for GIMP (www.gimp.org) and always will, if no one else writes one for Photoshop or any other image editing program.

And before I forget, this manual is provided “as is” an leave no kind of guaranties or promises, make sure to create backup files before you start modding. The manual is also a “Work In Progress” and will evolve over time. The latest version of the manual can always be found on:
www.moddb.com/games/bandits-phoenix-rising

Please also remember that It's probably a BAD idea to use this manual if you are a beginner on modding or programming, at least in this stage of the manuals development.

Now, go and create something fun!
And don't break anything I wouldn't break. ;-)

Regards
Kristian “Wolfy/Sajber” Nilsen

Questions, praise, typos, help etc, can be sent to: sajbermejl(AT)yahoo.com .

A more close look at Wolfys Mod or other modding work I have done check out,
www.moddb.com/members/sajber

For more information on GRIN and what they are up to now, check www.grin.se

File types & File system

*.xml

Can be opened in any text editor. Uses a script system, similar to HTML.

All *.XML files uses the ScriptDatabase import, except:

- The weapon definitions file (Chapter: “Weapon Definition File *.XML”)
- The bindings file, bindings.xml (Chapter: bindings.xml)

To Get help for Database based files, got the chapter “ScriptDatabase”.

*.dsf

*.dsf files can be opened in any text editor of your choice, just as the *.xml files. Editing them is very similar to programming C or C++.

*.DIESEL

Diesel files are used for 3d models, map definitions, etc.

There is currently no way of editing an or creating a *.DIESEL file that works with Bandits.

This makes it currently impossible to add new 3d models to the game.

However there is an 3d studio plugin for exporting diesel files for Ghost Recon Advanced Warfighter.
Note that the exporter ONLY works with 3D studio max 7 and only with Ghost Recon Advanced Warfighter, not Bandits
To export successfully to Ghost Recon you also need an *.xml definition file.

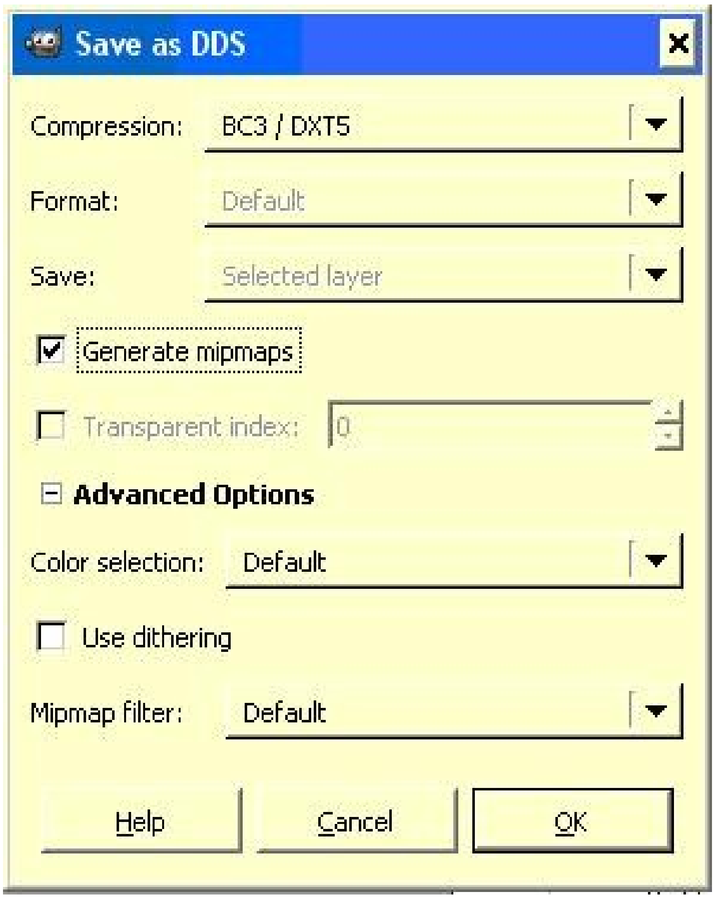
*.dds

A *.dds file is used for textures in Bandits.

To open *.dds files use the gimp plugin “dds.exe” (http://nifelheim.dyndns.org/~cocidius/dds/) or (http://registry.gimp.org/node/70) to open and save.

Saving *.DDS in GIMP:
Use the same settings as the picture below.

Other allowed Compressions:
BC1/DXT1
BC2/DXT3
BC3/DXT5 (Recommended)



Save as DDS

Compression: BC3 / DXT5

Format: Default

Save: Selected layer

☑ Generate mipmaps

□ Transparent index: 0

⊟ **Advanced Options**

Color selection: Default

□ Use dithering

Mipmap filter: Default

Any other type of compression will make Bandits crash when it tries to load the file. These saving options is global, meaning that they need to be used for ALL edited *.dds files.

*.tga

Also used for textures, mostly related textures used in various menus.
Opens with gimp.

Main_menu textures are saved with the following options:
Merge Visibile Layers
RLE compression OFF
Origin: Bottom left

Needs different saving options for different things. No universal saving option, see respective chapter for more info.

**.dll*

*.dll files can be viewed in Biew or any other hex editor.
If a *.dll is compressed, you can use UPX to decompress it, after that your on your own.

Bandits dlls

fmod
version: 3.3.3.0

fmod is a sound library dll. It seams very specific for Bandits, any attempt to upgrade to a newer version results in the following error message:

bandits.exe – entry Point Not Found
****
The procedure entry point _FSOUND_3D_Listener_SetRolloffFactor@4 could not be located in the dynamic link library fmod.dll

nvlinker.dll
version: 2.0.0.3
Developer: Nvidia Corporation
other: Vertex Shader Combiner

Tutorials

Tutorial: Creating a new map (Incomplete).

The first thing you need to do is to create a new folder inside *\Data\Maps* with a map name of your choice.

It's recommended to copy another folder folder map and rename it, to make sure your map works. If you do this, follow the step “To make your map work ingame” so that you are able to test your map while you work on it.

The _displacement file

The displacement file is a *.tga file and needs to have the following name:
“<map_folder_name>_displacement” to work.

The displacement file handles how the textures will be placed on the map.

*****

The displacement file needs to be save with the following flags in gimp:
flatten

The _hightmap file

The hightmap file is a *.tga file and needs to have the following name:
“<map_folder_name>_hightmap” to work.

The hightmap file tells bandits how the to draw mountains, chasms, etc.

The _map file

_map

****
The map file needs to be save with the following flags in gimp:
merge

The _light file & _light.text

_light

****
The light file needs to be save with the following flags in gimp:
merge

The _shade file

_shade
****
The shade file needs to be save with the following flags in gimp:
merge

Spawn points, enemies, pickups and objects.

No information currently available.

*.wp file

Gravel

Gravel is small objects spread across the map. Gravel is used for different foliage and usually non collision objects such as grass, birds etc.

The the amount of gravel and “relative” postion for the gravel is declared in a file called <your_map_name>_gravel.txt

You can import any *.diesel file in this text file and they can have the following settings.: “collisionobject” and “object”.

To get some birds flying around on the map, we just type inside <mapname>_gravel.txt:

Example:
object: birds 0 0 10 birds

This will create 10, non collision birds near the player spawn.
The “object:” tells the game that its suppose to create non collision objects, “birds” declare what object it is “0 0” tells the game that the objects is suppose to spawn 0 (X) and 0 (y) ordinates relative to the player spawn point. If you want the birds to be collision, just change “object:” to “collisionobject:”.

Now, what do we do if we want to have our own objects on our map?

Well, first of all, all objects in the subfolders of “\Data\Graphics\environment\”. The only critera is that the folder and the *.diesel file have an id name.

Example:

The folder “airport” have files inside starting with “airport”.
*Some folders also have alternative *.diesel files. “airport” have 5 files starting with “airport”*

Example 2:
As you can see inside the “\Data\Graphics\environment\airport\ folder, there is multiple files there, “airport-a.DIESEL” to “airport-e.DIESEL.”. They imported randomly and does not require any extra lines, just type “airport” as usual.

To create our own object, create a subfolder inside “\Data\Graphics\environment\”, then put your *.DIESEL file(s) inside the map, make sure to name the file “<foldernname>-a” for the first file “<foldername>-b” for the second file, and so on

To make your map work ingame

What you need to do now is to make the game show your map in the game menus so that you can play it. There is two ways of doing this.

The first is to rename the folders and files to make the game think that your map is “mission01” This is enough when your are testing your map during the building phase. This also the ONLY way of getting a singelplayer map into the original bandits game.

For the finalizing stage on how to get your map into multiplayer, is found below.

But the only way to actually make you able to “pick” your map for singleplayer is to use Wolfys mod. (see below).

Testing

Rename the folder “mission01” in the *\Data\Maps* folder to “mission01_old”, then renaming your own map folder to “mission01”.

You will also need to rename all the files inside your map folder that starts with “mission” to “mission01”.

Example:
“mission25.DIESEL” gets renamed to “mission01.DIESEL”.

You will also need to rename your map missionxx.xml script file inside the *\Data\profiles* folder, to “mission01.xml”

This will make the game run your map when you start a new game.

Multiplayer

Submitting the map to Wolfys mod.

Just zip your map folder and send it to me if you want it to be evaluated for being released with the next official version of “Wolfys Mod”. Don't forget to include a text file with credits, map description, briefing, map name and what game mode(s) its created for.

Tutorial: Creating a new player car (incomplete)

Adding a new player vehicle can be pretty difficult since Bandits doesn't seam to to be coded to be able to handle more than three playable vehicles. Therefore adding a new car requires a lot of small changes and tweaks.

Step by step

Step 1
*Create a new folder inside \Data\Graphics\vehicles\player\, with the 3d model and the ph_decal.dds for the vehicle.

Step 2
*Create a new folder inside with all the effects for the car.

Step 3
*create the settings files and place them in \Data\profiles\

Step 4
*add the car to garage.xml in \Data\profiles\

Step 5
make the 3d model show up in the vehicle selection menu. (Currently not possible)

File system:

\Data\Graphics\vehicles\player\<car name>
    chassis
    and ph_decal

\Data\effects\car_effects\<car name>_effects

<string name="drive_wheels" value="1 1 1 1"/>
what kind of drive... 4x4 etc..

To make your car work in game

This step can be skipped if you have created a car for “Wolfys Mod”.

Submitting the map to Wolfys mod.

Don't forget to include a text file with a credits, car description, car name and what game mode(s) its created for.

Tutorial: Creating a new weapon

Creating a weapon is very similar to creating a new car.
Its a very frustrating process and a lot of things can go wrong.

Step by step

First of all you need to decide what type of weapon you want to make. The following examples will be borrowed from the light machinegun for the light car (Badger).

Step 1
Add 3d models for the weapon in the folder
\Data\Graphics\weapons\turrets\player_turrets\<weapon_name>\

<weapon_name> can be any name you like, but its easier for the game and for you if you name it with _heavy, _medium or _small at the end, this makes it easier for you and bandits to understand what car the weapon is for. Its also recommended to add p_ in the beginning of the name to signal that the weapon is a player weapon.

Example:
p_sniper_light
p_cannon_heavy

This indicates a player sniper weapon for the light car (Badger) or a player cannon for the heavy car (Ogre)

Its also important to add the weapon folder in the \Data\Graphics\weapons\turrets\player_turrets\ if you make a turret weapon or in the fixxed-machineguns if you make a fixed weapon (front weapons), grenade-launchers if you make a launcher weapon or in the mine-layer folder if you make a weapon that drops something in the game world.

Step 2
Create the properties.txt and add it to the
\Data\Graphics\weapons\turrets\player_turrets\<weapon_name>\ folder

The properties.txt can be made in a wide variety of ways. I will just use an example here (from p_machinegun_light), but further studies can be done in the properties.txt section further down in this chapter.

Example:

```
gun_type: tower
projectile_type: tower_player_smallbullet

jacket_type: casing762
alt_jacket_type: casing762

rate_of_fire: 12
```

```
base_speed: 0.2
base_maxangle: 180
base_minangle: -180
base_mesh: gunbase

range: 800
near: 1

gun_speed: 0.2
gun_maxangle: 25
gun_minangle: -25
gun_mesh: gun

recoil: 15
alt_recoil: 10
```

Step 3
Create effects and them to a folder in \Data\effects\player_weapons\turrets\ (if you make a turret weapon)

Here I will just use an example for the machine gun, for more studies around effects look inside the “Creating Effects” chapter.

Step 4
add effects to the game_effects.xml
Once again an example will be used to show how to add the effects for the machine gun. The Creating Effects chapter explains how to add and create the effects from scratch.

The effects for the machinegun is placed in \Data\effects\player_weapons\turrets\machinegun\
You can pick any name for the folder for your weapon.

Step 5
Its now time to add the effects to the game so it can be recognized by bandits. For the machine gun, open game_projectiles.xml

Scroll down to:

```
<!-- player machinegun tower-->
        <effect name="player_tower_machinegun">
                <effect_group>
                <!-- sound -->
                        <sound3d file="weapons\player_tower_machinegun.wav"/>
                <!-- flame -->
                        <link file="player_weapons\turrets\machinegun\machinegun_flame.xml"/>
                </effect_group>
         </effect>
<!-- player machinegun tower special ammo-->
        <effect name="player_tower_machinegun_special">
```

```
                <effect_group>
                <!-- sound -->
                        <sound3d file="weapons\player_tower_machinegun.wav"/>
                <!-- flame -->
                        <link file="player_weapons\turrets\machinegun\machinegun_special.xml"/>
                </effect_group>
        </effect>
```

This is how the effects for a weapon is defined, for this weapon there is also other types of effects for the special ammo. You can also se here how to define what sounds the weapon uses.

Step 6
create a weapon definitions file (*.xml) in \Data\weapons\player_weapons
For this tutorial we will use machinegun_turret.xml but for a more in depth inside look about the weapons definitions file look in the section called, Weapon projectile definition file, *.xml, in the Glossary.

Step 7
Now its time to add your weapon to \Data\profiles\garage.xml file. This is step makes sure that the weapon is connected to the right car.

Open the garage.xml file inside \Data\profiles.
Scroll down to:

```
<block name="weapons">
```

Add the following line right under:

```
<string name="1" value="Machinegun"/>
```

Note: This is just an example, if you add a totally new weapon you need to change the number from 1 to the on higher than the highest weapon number.
In an unmodded bandits game, the highest weapon number is 13. (<string name="13" value="Shotgun"/>) this means that when you add a totally new weapon it has to have the number 14, like this:

```
<string name="14" value=”My Uber cool Weapon"/>
```

The other section is the name that your weapon will have in game.

******

Now scroll up to the line:

```
<block name="weapon_numbers">
```

Same thing here. If you add a totally new weapon you need to type:

```
<string name="<weapon folder name>" value="14"/>
```

But to be more clear once again we use the machine gun:

```
<string name="p_machinegun" value="1"/>
```

As you can see the p_machinegun refers to our graphics folder inside,

```
\Data\Graphics\weapons\turrets\player_turrets\
```

If we made a new shotgun, the folder would be found in:

```
\Data\Graphics\weapons\fixxed-machineguns\player-fixxedguns\
```

and the line would look like this:

```
<string name="shotgun" value="13"/>
```

******

Next up you need to add this under all blocks that has something with:

```
<block name="missionXX_weapons">
```

Add the following to all mission blocks

```
<string name="1" value="yes"/>
```

or if you make a totally new weapon

```
<string name="14" value="yes"/>
```

This is to make your weapons available on the maps. If you forget to add the line to one of the maps, you might not be able to select it..or in worst case, the game crash.

******

Now its time to add the weapon to the right car, and the right part of that car.
All cars have a primary slot, this is the tower. The machine gun is a primary weapon and a tower.

Scroll to the block called:

```
<block name="heavy_back_a">
```

This is the back part of the heavy car (Ogre). That is not where we want to put the machine gun, but we still need to tell the game that we don't want it there so we still need to add a line here just as with the maps.

There for we add:

```
<string name="1" value="no"/>
```

or if we make a totally new weapon

```
<string name="14" value="no"/>
```

You need to add this line to all car part blocks that is not suppose to have this weapon, but when we get to this line:

```
<block name="heavy_primary">
```

we add:

```
<string name="1" value="yes"/>
```

or

```
<string name="14" value="yes"/>
```

Step 8

Add your weapon in the freaks_equip menu. This to make sure that the weapon is available in the game menus.

Go to \Data\menu\freaks_equip\ and open the file freaks_equip.dsf in an text editor.

Since there is currently no way to create your own 3d models, you need to “copy” the model and the weapon icon from another weapon

Scroll down to:

```
import weapon_choice1, weapon_choice2;
```

When you create a new “copied” model and icon its wise to give it a good name, so type something like this right under the line above:

```
declare my_uber_weapon;
declare my_uber_weapon_mesh;
```

Then scroll down to

```
init () {
```

under it type:

```
my_uber_weapon:=none_icon_1;
my_uber_weapon_mesh:=weapon_mesh_1;
```

none_icon_1 is and empty icon, which is appropriate since we don't have an icon our selves.
weapon_mesh_1 is the model for the machine gun.

****

Now scroll down to:

```
//Displaying availible primary choices
```

(Yes, its misspelled).

The following piece of code is a loop, it checks the garage file and checks for all the weapons:

```
        while (i<13) {
                        i:=i+1;
                        if (ScriptDatabase.value("mission"+mission+"_weapons", ""+i)="yes") {
                                if (ScriptDatabase.value(""+mesh+"_primary", ""+i)="yes") {
                                        Menu.text_box_print("choice"+choice,
""+ScriptDatabase.value("weapons", ""+i));
                                        eval "w"+choice+":=(\"\"+i);";
                                        choice:=choice+1;
                                }
                        }
                }
```

If you where to make a totally new weapon you just add the 13 to 14 since there is one more weapon now. The machine gun is in the game, so we don't need to change anything in this example.

Then scroll down to:

```
hide_weapon_icons() {
```

Add the following line right under:

```
my_uber_weapon.set_visibility(false);
```

Then scroll down to:

```
hide_guns() {
```

Add the following line right under:

```
my_uber_weapon:mesh.set_visibility(false);
```

Step 9
Add a weapon text file in \Data\menu\freaks_equip\weapons\

When you add a new weapon the number should be w14.txt and counting one step up every time you add a new weapon.

I will once again use the machine gun as an example (w1.txt).

```
<text: "weapon_info">
<color: 241 241 216 255>
<size:  0.03>
Name: Machinegun

Type: Machinegun

Ammo: Primary lead (unlimited)

Position: Turret mount gives
360 degrees action

Range: Average

Damage: Light

Special: This baby can load
special armour piercing ammo
too, which also increases
the range and accuracy.
</text>
```

Simply put, its the info text about the weapon. You must add it..or the game will crash when you select the weapon.

Step 10
Adding your weapon to start_level.dsf and restart.dsf.

Go to \Data\menu\start_level\ and open start_level.dsf
Scroll down to :

```
if (selected_car="light") {
```

of heavy or medium, depending what car your weapon is for.

Then under the line

```
else if (primary="2") {	ScriptDatabase.set_value("primary","add_weapon","p_sniper_light"); }
```

(if you make a primary weapon), add:

```
else if (primary="14")
{ ScriptDatabase.set_value("primary","add_weapon","p_my_uber_weapon_light"); }
```

Sometimes the same weapons can have different models of properties for different cars this is what the “_light” in the folder means. As a rule add _light, _heavy or _medium at the end of your folders just to be on the safe side.

When your done with start_level.dsf, add the exact somethings to restart.dsf. You can find restart.dsf inside \Data\menu\restart\.

Your weapon should now be ready to kick ass!

Filesystem

Bla, bla bla.

Graphics

Folder: Data\Graphics\weapons\... properties.txt

The properties.txt goes in the right weapon folder.
Example: Data\Graphics\weapons\turrets\player_turrets\p_agun_light\properties.txt

properties.txt

A tower weapon is made of two parts the gunbase, and the gun itself. This is important to know when you set the angles max and min angels the parts can move, so that your able to tilt and rotate the tower.

Testing your weapon.

Setting up a good testing environment for your weapon is easy.
Open start_level.dsf inside \Data\menu\start_level\ . Scroll down to

```
if (selected_car="light")
```

Look inside the paragraph and find:

```
if (primary="1") {	ScriptDatabase.set_value("primary","add_weapon","p_machinegun_light"); }
```

This means that if the player selects the “light” car (Badger) and the first primary weapon (machine gun), the game is suppose to add the weapon “p_machinegun_light” when the game starts. p_machinegun_light is actually a folder inside *\Data\Graphics\weapons\turrets\player_turrets*.

So if your new weapons folder name is for example named “p_laserturret_heavy”. Just change the above line to:

```
if (primary="1") {      ScriptDatabase.set_value("primary","add_weapon","p_laserturret_heavy"); }
```

This will make the game pick your weapon every time you pick the badger with the machine gun.

Remember that this will only work when you start a new mission. Change the restart.dsf inside \Data\menu\restart\, in the exact same way to make it work when you restart an already started mission ingame. Normally restart.dsf and start_

Finalizing your weapon.

This step can be skipped if you have created a weapon for “Wolfys Mod”.

Submitting the map to Wolfys mod.

Don't forget to include a text file with a credits, weapon description, car name and what game mode(s) its created for.

Creating Effects

The effects in Bandits.

Note: the *.xml.enc files is created by the game and does not need to be edited or created. The will be updated automatically by the game every time the game starts.

Adding effects to game_effects.xml

Adding your own music to Bandits

This the tutorial for adding music to the original Bandits.

The following steps are needed:
*Add the music files to the *\Data\Music* folder.
*Edit the hud.dsf file inside *\Data\interface\Hud.*
*Edit the music.xml file inside *Data\profiles.*

Adding the music files

Add the music you want top add to the *\Data\Music* in your Bandits folder. *.mp3 files is known to work, *.wav should work. I also believe *.wma works to but I'm not sure.

Editing the hud.dsf file.

When you have added the music you want scroll your way to the *\Data\interface\Hud* folder and open “hud.dsf”.

Add the following before “*init() {*“ inside hud.dsf,

```
declare channel_5_songs;
declare channel_5_name;
declare m5_1;
declare mn5_1;
```

This will add the variables we need to add ONE new channel and ONE new song.

if you want to add more than one song. just add, *declare m5_2; declare mn5_2;* for song number two and *declare m5_3; declare mn5_3;* for song number three etc..

Then scroll down to:

```
        if (game_completed=1) {
                channel_1_songs:=7;
           channel_2_songs:=7;
                channel_3_songs:=5;
                channel_4_songs:=8;
        }
```

and add the following above the paragraph.

```
channel5_name:="My new channel!";
channel_5_songs:=1;
```

```
m5_1:="data\music\something.mp3";
mn5_1:="My new song!";
```

This will set the name for our channel, the path for our song and the name for our song.

If you have more songs than one..just copy the to last lines and change the number “1” to “2” for the second song and “3” for the third, etc.

Then scroll down too the line that says

```
“//////////////////////////////////////// musicplayer ////////////////////////////////////////”
```

scroll down to:

```
    player_song(channel, song_playing) {
        declare chan, song;
        set_nos();
        chan:=1;
        song:=1;
        if (channel>0 and channel<5) {
            chan:=channel;
    }
```

and since we only have one new channel change the “5” to “6”.
If you add more channels, just count it upwards for every channel you add.

then scroll down to

```
    set_nos() {
        if (channel=1) {
            number_of_songs:=channel_1_songs;
        }
        else if (channel=2) {
            number_of_songs:=channel_2_songs;
        }
        else if (channel=3) {
            number_of_songs:=channel_3_songs;
        }
        else if (channel=4) {
            number_of_songs:=channel_4_songs;
        }
    }
```

and make it look like this.

```
    set_nos() {
        if (channel=1) {
            number_of_songs:=channel_1_songs;
        }
        else if (channel=2) {
            number_of_songs:=channel_2_songs;
        }
        else if (channel=3) {
            number_of_songs:=channel_3_songs;
        }
        else if (channel=4) {
            number_of_songs:=channel_4_songs;
        }
        else if (channel=5) {
            number_of_songs:=channel_5_songs;
        }
    }
```

This also if you just have added one channel. If you have adde mor channels repeat the sam procedure and change the “5” to a “6”, and so on.

scroll down to

```
    change_channel(how,play) {
        if (how="step") {
            if (channel<4) {
                channel:=channel+1;
             }
            else {
                channel:=1;
            }
        }
```

change the “4” to “5” if you just add one channel, if you add more...well count one time upwards for every new channel you want to add.

And lastly scroll down to “change_cd_info(type,name,name2) {“
and find:

```
        else if (type="channel") {
            if (name="1") {
                cd_info.addLine(channel1_name);
            }
            else if (name="2") {
                cd_info.addLine(channel2_name);
            }
            else if (name="3") {
                cd_info.addLine(channel3_name);
            }
                else if (name="4") {
                cd_info.addLine(channel4_name);
            }
        }
```

and make it look like this:

```
        else if (type="channel") {
            if (name="1") {
                cd_info.addLine(channel1_name);
            }
            else if (name="2") {
                cd_info.addLine(channel2_name);
            }
            else if (name="3") {
                cd_info.addLine(channel3_name);
            }
            else if (name="4") {
                cd_info.addLine(channel4_name);
            }
            else if (name="5") {
                cd_info.addLine(channel5_name);
            }
        }
```

Same thing here. Repeat the process if you add more channels than one.

Now we are done with the hud.dsf file.

Editing music.xml

Next, scroll to the \data\profiles\ folder, open music.xml and add the following under all the <block name="mission*"> where (* == 01-24)

```
<integer name="channel_5_songs" value="2"/>
```

This is the same as before, if you add more channels than one, just count one time upward for every new song you add.

This will make the player able to play your song on every map in the game. If you want, you can make your song available on just some maps..if thats the case just dont add the above line under that “block name”.

Your song should now show up ingame.
Good Luck!

Glossary

Effect files, *.xml

Weapon properties file, *.txt

gun_type:	tower
projectile_type:	*tower_player_smallbullet*
jacket_type:	*casing762*
alt_jacket_type:	*casing762*
rate_of_fire:	0-500
base_speed:	0-100
base_maxangle:	0-180
base_minangle:	-180-0
base_mesh:	*gunbase*
range:	0-9999
near:	0-1 (?)
gun_speed:	0-999
gun_maxangle:	0-180
gun_minangle:	-180-0
gun_mesh:	*gun*
recoil:	0-100
alt_recoil:	0-100

Weapon projectile definition file, *.xml

Folder: *“data\weapons\player_weapons\*.xml”*
Check here for an example: *“\data\weapons\player_weapons\machinegun_turret.xml”*

Variable Tree:

```
<projectile>
        <ammo/>
        <on_fire>
                <effect/>
                <spread>
                <raycast>
                        <ballistics>
                                <object_impact>
                                        <spread>
                                                <effect/>
                                        <spread/>
                                        <range_damage/>
                                        <crater/>
                                        <effect/>
                                </object_impact>
                                <car_impact>
                                        <spread>
                                                <effect/>
                                        <spread/>
                                        <range_damage/>
                                        <crater/>
                                        <effect/>
                                </car_impact>
                                <landscape_impact>
                                        <spread>
                                                <effect/>
                                        <spread/>
                                        <range_damage/>
                                        <crater/>
                                        <effect/>
                                </landscape_impact>
                                <on_die>
                                        <effect/>
                                <on_die/>
                        <ballistics/>
                </raycast>
                </spread>
        </on_fire>
        <on_click>
                <effect/>
                <raycast>
                        <ballistics>
                                <object_impact>
```

```
                    <spread>
                        <effect/>
                    <spread/>
                    <range_damage/>
                    <crater/>
                    <effect/>
                </object_impact>
                <car_impact>
                    <spread>
                        <effect/>
                    <spread/>
                    <range_damage/>
                    <crater/>
                    <effect/>
                </car_impact>
                <landscape_impact>
                    <spread>
                        <effect/>
                    <spread/>
                    <range_damage/>
                    <crater/>
                    <effect/>
                </landscape_impact>
                <on_die>
                    <effect/>
                <on_die/>
            <ballistics/>
        </raycast>
    </on_click>
</projectile>
```

Weapon Projectile definition variables:

<ammo/>	
<ammo type="<string> required="<int>""/>	
Variable:	**value:**
<ammo type=”<string>”/>	special
<ammo required=”<int>”/>	0-999

<effect/>
Defines an effect to play. <effect name=”<file path>”/>

Variable:	**value:**
<effect name=”<file path>”/>	

<crater/>	
<crater height_limit="<int>" type="<string>"/>	
Variable:	**value:**
<crater height_limit=”<int>”/>	0 - 9999
<crater type=”<string>”/>	small, medium, large

<raycast> & <raycast/>	
<raycast nothing="<boolean>" damage="<int>" speed="<int>" range="<int>" reported_speed="<int>" tracer_rate="<int>" tracer_width="<int>" tracer_time="<double>" tracer_color="<string>"> <raycast/>	
Variable:	**value:**
nothing	true, false
damage	0-9999
speed	0-9999
range	0-9999
reported_speed	0-9999
tracer_rate	0-9999
tracer_width	0-9999
tracer_time	0-9999
tracer_color	

<range_damage> & <range_damage/>	
<range_damage range="**<int>**" damage_max="**<int>**" damage_min="**<int>**"> <range_damage/>	
Variable:	**value:**

range	0-9999
damage_max	0-9999
damage_min	0-9999

<on_fire> & <on_fire/>
Used for the definition when firing “special” ammo projectiles

<spread> & <spread/>	
Defines the spread of the weapon. <spread max_angle="**<int>**" min_angle="**<int>**" multiplier”**<int>**"	
Variable:	**value:**
max_angle	0-360
min_angle	0-360
multiplier	0-999

<on_die> & <on_die/>

<on_click> & <on_click/>

<projectile> & <projectile/>
Defines the whole definition area.

<landscape_impact> & <landscape_impact/>

<car_impact> & <car_impact/>

<object_impact> & <object_impact/

bindings.xml

Sets all binded keys for the game.

```
<bindings>
	<controll>
		<binding/>
	</controll>
	<controll>
		<binding/>
	</controll>
</bindings>
```

<controll> & </controll>			
<controll center=”<int>” extends=”<int> name=”<string>” r_max="<int>" r_min="-1.000000" controll remap="true" controll type="axis">			
Variable:	**value:**		
center	0-1 (0.100000 standard)		
extents	0-1 (0.900000 standard)		
name	Primary Fire	Turn Left/Right"	Look Horisontal
	Secondary Fire	Accelerate	Look Vertical
	Secondary Weapon 1	Brake/Reverse	Select Command
	Secondary Weapon 2	Handbrake	Turret Spin View
	Secondary Weapon 3	Primary Fire	Toggle Turret Spin View
	Secondary Weapon 4	Secondary Fire	Zoom In
	Music Mode	Secondary Weapon 1	Zoom Out
	Change Music Channel	Secondary Weapon 2	Turn Left/Right"
	Chat Message	Secondary Weapon 3	Accelerate
	Pause Game	Secondary Weapon 4	Brake/Reverse
	Show Scores	Music Volume Up	Secondary Weapon 5
	Ingame Menu	Music Volume Down	Next Secondary Weapon
	Prev Music Track	Next Music Track	Prev Secondary Weapon
	Nitro	Issue Command	

	Screenshot	Handbrake	
r_max	0-1 (0.900000 standard)		
r_min		-1-0 (-1.000000 standard)	
remap	Can it be remapped from inside the game?	true, false	
type		axis, button	

<binding/>		
<binding device="<device>" Invert="<boolean>" object="<unicode key>"/>		
Variable:	**value:**	
device	SWHL 1	
	MOUS 1	
	JOYS 1	
	KEBD 1	
	GMPD 1	
Invert	true, false	
	Device :	**Value**:
object	**KEBD 1**	Unicodes (see chapter: keys)
	MOUS 1	X Axis 1
		Y Axis 1
	JOYS 1	
	GMPD 1	
	SWHL 1	R X Axis
		R Y Axis
		Slider

```
aDevice_name    db 'device_name',0
aObject_name    db 'object_name',0
```

```
aRel_axis_state db 'rel_axis_state',0   ; DATA XREF:
aAxis_state     db 'axis_state
aButton_was_rel db 'button_was_released',0
aButton_was_pre db 'button_was_pressed',0
aButton_pressed db 'button_pressed',0
aSlider         db  ',0
aButton_0       db 'Button ',0
aKey            db 'Key ',0
aPov            db 'Pov ',0
aKey_0          db 'Key ',27h,0
aDown           db ' Down'
a_?avgamepad_id db '.?AVGamepad_IDev@dsl@@
a_?avsteeringwh db '.?AVSteeringWheel_IDev@dsl@@
a_?avjoystick_i db '.?AVJoystick_IDev@dsl@@
FORCEFEEDBACKEFFECT
PLAYING
DOWNLOAD
SET_GEN_DURATION
SET_GEN_TRIGGER
SET_GEN_TRIGGER_REPEAT
aUpload_typespe db 'UPLOAD_TYPESPEC_PARAMS
```

Key Table

Does **ONLY** work inside bindings.xml.

Keys				**Special**	**Media**	**Unknown**
1	Q	F1	ESCAPE	APOSTROPHE	NEXTTRACK	KANA
2	W	F2	BACKSLASH	SEMICOLON	PREVTRACK	KANJI
3	E	F3	SPACE	LSHIFT	MYCOMPUTER	OEM_102
4	R	F4	RETURN	RSHIFT	MEDIASELECT	ABNT_C2
5	T	F5	NUMPAD1	LCONTROL	WEBSTOP	ABNT_C1
6	Y	F6	NUMPAD2	SUBTRACT	CALCULATOR	PRIOR
7	U	F7	NUMPAD3	LBRACKET	WEBHOME	GRAVE
8	I	F8	NUMPAD4	RBRACKET	VOLUMEUP	CONVERT
9	O	F9	NUMPAD5	SCROLL	VOLUMEDOWN	AX
0	P	F10	NUMPAD6	NUMLOCK	MEDIASTOP	UNLABELED
UNDERLINE	A	F11	NUMPAD7	HOME	PLAYPAUSE	NOCONVERT
DECIMAL	S	F12	NUMPAD8	RWIN	MUTE	AT
PERIOD	D	F13	NUMPAD9	LWIN	MAIL	
COMMA	F	F14	NUMPAD0	RMENU	APPS	
DIVIDE	G	F15	NUMPADEQUALS	PAUSE	WEBBACK	
EQUALS	H		NUMPADCOMMA	CAPITAL	STOP	
MINUS	J		NUMPADENTER	SYSRQ	WEBFORWARD	
MULTIPLY	K		LEFT	RCONTROL	WEBREFRESH	
COLON	L		RIGHT	DELETE	WEBFAVORITES	
	Z		DOWN	SLEEP	WEBSEARCH	
	X		UP	WAKE	NEXT	
	C			POWER	BACK	
	V			INSERT		
	B			LCONTROL		
	N			LMENU		
	M					

ScriptDatabase

ScriptDataBase names:

freaks_briefing		
mission_briefings		
settings		
mission_names		

<ScriptDataBase name="settings"> (settings.xml)

<block name="bana">		
<string name="bana" value="<string>"/>	MISSION01	MISSION12
	MISSION02	MISSION13
	MISSION03	MISSION14
	MISSION04	MISSION15
	MISSION05	MISSION16
	MISSION06	MISSION17
	MISSION07	MISSION18
	MISSION08	MISSION19
	MISSION09	MISSION20
	MISSION10	MISSION21
	MISSION11	MISSION22
</block>		

<block name="controls">	
<integer name="mouse_speed" value="21"/>	0-100
<string name="invert_mouse" value="no"/>	yes, no
<string name="set" value="setting2"/>	setting1, setting2
</block>	

<block name="custom">	
<integer name="effect_detail" value="0"/>	
<integer name="gravel" value="0"/>	
<integer name="jacket" value="1"/>	
<integer name="landscape" value="0"/>	
<integer name="lod_far" value="500"/>	
<integer name="motiontrail" value="1"/>	
<integer name="object_detail" value="0"/>	
<integer name="shadow" value="1"/>	
</block>	

<block name="difficulty">	
<string name="difficulty" value="medium"/>	easy, medium, hard
</block>	

<block name="graphics">	
<integer name="effect_detail" value="0"/>	
<integer name="gravel" value="0"/>	
<integer name="jacket" value="1"/>	
<integer name="landscape" value="0"/>	
<integer name="lod_far" value="400"/>	
<integer name="motiontrail" value="1"/>	
<integer name="object_detail" value="0"/>	
<integer name="shadow" value="1"/>	Shadows on/off 1=on 0=off
<string name="detail_level" value="custom"/>	
</block>	

<block name="high">	
<integer name="effect_detail" value="0"/>	
<integer name="gravel" value="0"/>	
<integer name="jacket" value="1"/>	

<integer name="landscape" value="0"/>	
<integer name="lod_far" value="400"/>	
<integer name="motiontrail" value="1"/>	
<integer name="object_detail" value="0"/>	
<integer name="shadow" value="1"/>	
</block>	

<block name="low">	
<integer name="effect_detail" value="0"/>	
<integer name="gravel" value="0"/>	
<integer name="jacket" value="1"/>	
<integer name="landscape" value="0"/>	
<integer name="lod_far" value="400"/>	
<integer name="motiontrail" value="1"/>	
<integer name="object_detail" value="0"/>	
<integer name="shadow" value="1"/>	
</block>	

<block name="medium">	
<integer name="effect_detail" value="0"/>	
<integer name="gravel" value="0"/>	
<integer name="jacket" value="1"/>	
<integer name="landscape" value="0"/>	
<integer name="lod_far" value="400"/>	
<integer name="motiontrail" value="1"/>	
<integer name="object_detail" value="0"/>	
<integer name="shadow" value="1"/>	
</block>	

<block name="player_name">	
<string name="player_name" value="<string>"/>	Player name

</block>

<block name="sound">		
<integer name="music_volume" value="**<int>**"/>	The music volume	0-100 (Standard: 30)
<integer name="sfx_volume" value="**<int>**"/>	Sound volume	0-100 (Standard: 84)
<integer name="voice_amount" value="**<int>**"/>	Voice volume	0-100 (Standard 0)
<string name="ammo_comment" value="**<string>**"/>	Toggles ammo comments	on, off (standard: on)
<string name="intro_movie" value="**<string>**"/>	Toggles intro movie	on, off (Standard: on)
</block>		

<block name=”unknown”>	
ratatatata	.data:006D3E1C aRatatatata db 'ratatatata',0
suck on this	.data:006D3E28 aSuckOnThis db 'suck on this',0
particle	.data:006D3DDC aParticle db 'particle'
rapid fire off	.data:006D3E38 aRapidFireOff db 'rapid fire off',
rapid fire disabled	.data:006D3E48 aRapidFireDisab db 'rapid fire disabled',0
like an artist	.data:006D3E6C ; char aLikeAnArtist[] .data:006D3E6C aLikeAnArtist db 'like an artist',0
trigger happy	.data:006D3E5C ; char aTriggerHappy[] .data:006D3E5C aTriggerHappy db 'trigger happy',0
boom	aBoom db 'boom',0
heavy weapon	aHeavyWeapon db 'heavy weapon',0
quad off	aQuadOff db 'quad off',0
noooo	aNoooo db 'noooo',0
quad	char aQuad[] quad
artist mode on	.data:006D3EB0 ; char aArtistModeOn[] .data:006D3EB0 aArtistModeOn db 'artist mode on',0
cheater	.data:006D3EC8 ; char aCheater[] .data:006D3EC8 aCheater db 'cheater',0
virgin	.data:006D3EC0 aVirgin db 'virgin',0
chicken	.data:006D3ED0 aChicken db 'chicken',0
grin mode	.data:006D3ED8 aGrinMode db 'grin mode',0
good boy	good boy

</block>

<ScriptDataBase name="freaks_briefing”> (freaks_briefing.xml)

<block name="briefing">		
<string name="newgame" value="**<string>**");	Newgame?	yes, no (standard)
<string name="resume” value="**<string>**"/>	Toggles Resume button	yes, no(standard: no)
<string name="car" value="<string>"/	Type of car.	light, medium, heavy (Standard: light
<string name="show_brief_panels" value="true"/>		
<string name="slot" value="**<int>**"/>	Default saving slot	slot1 – slot7(Standard: slot1)
</block>		

<block name="mission">		
<string name="mission" value=”**<int>**");	Varable used to save the current mission number.	01-22)
</block>		

Programming *.dsf files

variables:
declare

int
string
boolean

variable functions:
contains
if(message.contains("+test")){
returns true

floor //shows full numbers only

upper_case

lower_case

round
returns
true/false

is_boolean

is_integer

is_float

is_number

is_string

is_void

if else if:
not
and

Importing Objects

This is a descripton of the many imported objects and most of their properties, functions and variables. These objects can only be imported in the *.dsf files.

text_box		
add_text_box(<string>, <int>, <int>, <int>);	Creates a text_box object.	object name
		x pos
		y pos
		text size 0.4 standard
text_box_print(<string> <string>);		

list_box		
new_list_box		
setPos(level_a, level_b);		
setColor(255, 219, 210, 181);		
setSelectionColor(255, 159, 150, 121);		
setFont("rosiban");		
setTextSize(0.045);		
setVisibility(false);		
multiselect:=true;		
multiline:=true;		
lines()		
selectLine()		
addLine(“String”)		
clear();		

GameWorld		
GameWorld.render_radar(<boolean>);		
GameWorld.add_radar_blip(<string>,<boolean>);		
GameWorld.remove_radar_blip(<string>);		
GameWorld.set_blip_color(<string>, <int>, <int>, <int>);		0 - 255
GameWorld.set_blip_size(<string>, <float>);		
GameWorld.set_blip_text(<string>, <string>);		
GameWorld.set_blip_texture(<string>, <string>);		
GameWorld.set_display_arrow(<string>, <boolean>);		
GameWorld.set_floating_arrow(<string>, <boolean>);		
GameWorld.set_arrow_size(<string>, <boolean>);		
GameWorld.set_arrow_radius(<string>, <float>);		
GameWorld.set_arrow_texture(<string>, <string>);		
GameWorld.set_display_text(<string>, <boolean>);		
GameWorld.set_display_dist(<string>, <boolean>);		
GameWorld.set_docked_text(<string>, <boolean>);		
GameWorld.set_text_size(<string>, <boolean>);		
GameWorld.set_text_font(<string>. <string>);		
GameWorld.Cheater(<string>, <boolean>);	obsolete	chicken
		grin mode
		good boy
		grinkfa

Context		
Context.play_movie(<file path>);	plays a movie file, tested types, avi, divx	relative file path

Context.controll();		
Context.controll_disp_name();		
Context.change_menu(<string>);		start_level
		freaks_map
		main_menu
Context.exit();		
Context.start_game(map);		

Keyboard

Keyboard.key_down();		returns true or false,

SoundManager

Funtion call:	**Comments:**	**Inputs:**
SoundManager.set_stream_volume();		0-100
SoundManager. set_sfx_volume();		0-100
SoundManager.stop();		

SoundManager.play();		SoundManager.play("data\Menu\sfx\click8.wav",0);
		SoundManager.play("**<file path>**",0);

SoundManager.is_music_playing();		returns, true or false

SoundManager.set_music_volume();		SoundManager.set_music_volume(music_volume_now);
		SoundManager.set_music_volume(**<int>**);

SoundManager.stop_music();	Stops the music that SoundManager is currently playing	none

SoundManager.play_music();	Probably uses “eval” to make it possible to use a local variable directly.	eval "SoundManager.play_music(m"+chan+"_"+song+");";
		SoundManager.play_music(**<file path>**);
		eval "SoundManager.play_music(“**<file path>**”);";

ConnectionManager		
ConnectionManager.requestSpawn();		
ConnectionManager.disconnect();		

Playfield		
Playfield.turn_on_fire();		
Playfield.turn_off_fire();		
Playfield.unpause();		
Playfield.pause();		
Playfield.camera_flight("menu_cam", -1);		
Playfield.cancel_camera_flight();		

Playfield.spawn_group(“enemy_squad_a”, “player_spawn”);		
Playfield.order_group(enemy, "attack");		
Playfield.fire_projectile("<string>");		
Playfield.kill_em_all();		
Playfield.kill_car("enemy_squad_a");		

Math		
Math.random(1,37);		
Math.Rotation(0,0,1.0*rot_factor)		
Math.Vector();		
Math.atol(string, int);		
Math.sqrt(int);		
Math.asin(int);		
Math.acos(int);		

HUD as script_hud (in hud.dsf)

HUD can be written to and extended with script_hud. And called directly with HUD.

```
interface script_hud {
        command_entered(message) {
                chat_background.set_visibility(false);
                ConnectionManager.sendTextMessage(message);
                HUD.input_data:="";
                HUD.release_input();
        }
        command_escaped(message) {
                chat_background.set_visibility(false);
                HUD.input_data:="";
                HUD.release_input();
        }

}
```

HUD variables

HUD.input_data	HUD.input_data:="";
HUD.cinema_mode:=<boolean>	

HUD functions

HUD.release_input();	
HUD_car_life();	current_life:=HUD.car_life(); Function with an **<int>** output/return. Returns the current amount of life in the player car. (armour).
HUD.weapon_ammo();	last_ammo_count:=HUD.weapon_ammo(1); Function with an <int> input and an <int> output/return. Returns the amount of ammo.
HUD.new_list_box();	player_name:=HUD.new_list_box("player_name"); Creates a new list_box object with the name “player_name” and returns it to the player_name variable;
HUD.set_radar_visibility(boolean);	
HUD.textures_in_use();	Returns what textures that are in use
HUD.triangle_count();	Returns current triangle count
HUD.vertex_memory_in_use();	
HUD.set_radar_pos(int);	
HUD.set_radar_text_pos(string);	

Unknown Imported Objects			
Prev_Music_Track	Sight	Show_Scores	
Change_Music_Channel	l_danger	yes_no	MessageList
Music_Mode	w_primary_machinegun	keys as keys	Chat_Message
Music_Volume_Down	w_primary_sniper	indicators as indicators	GameDialog as game_dialog,
Music_Volume_Up	w_primary_cannon	IplayerList	
win_b	w_secondary_rocket_a	TeamList	Ingame_Menu
r_background	w_secondary_rocket_b	camera_hud	Next_Music_Track,
d_background	w_secondary_rocket_c	movie_a	brief_a_1
w_primary	w_secondary_grenade_a	movie_b	brief_a_2
w_secondary	w_secondary_grenade_b	loose_a	brief_b_1
l_armor	w_secondary_grenade_c	loose_b	brief_b_2
l_armor01	w_secondary_mine_a	win_a	brief_c_1
l_armor02	w_secondary_mine_b	weapons as weapons	brief_c_2
l_armor03	w_secondary_fixed_a	musicplayer as musicplayer	brief_d_1
l_glow	w_secondary_fixed_b	hudcontrol as hudcontrol	brief_d_2
score_topic_a	d_pos_a	mouse as mouse	brief_e_1
score_topic_b	d_pos_b	nitro	brief_e_2
team_score_topic_a	r_text_pos	nitro_pos_a	maps_1
team_score_topic_b	menus as menus	nitro_pos_b	maps_2
loading_map_a	MOUSE_Y	nitro_back	brief_f_1
loading_map_b	MOUSE_WHEEL_UP	versatile as versatile	brief_f_2
reason_a	MOUSE_WHEEL_DOWN	player_list_a	brief_g_1
reason_b	MOUSE_B1	player_list_b	brief_g_2
winner_a	MOUSE_B2	team_list_a, team_list_b	player_name_a
winner_b	MOUSE_B3	quit_box_a_1	player_name_b
s_map_a	cd_info_a	quit_box_a_2	m_background
s_map_b	cd_info_b	quit_box_b_1	mouse_pointer
s_load_a	m21frag_a	quit_box_b_2	m_scorebackground
s_load_b	m21frag_b	quit_box_c_1	chat_background
c_client_a	brief_p_2	quit_box_c_2	MOUSE_X
s_prev_map_1	briefing_background	RemoteCall as missioncall	brief_h_1
s_prev_map_2	quit_yes_1	quit_box_d_1	brief_h_2
s_next_map_1	quit_yes_2	quit_box_d_2	brief_i_1

s_next_map_2	quit_no_1	allowed_cars_topic_2	brief_i_2
	quit_no_2	allowed_weap_topic_1	brief_j_1
	briefing_1	allowed_weap_topic_2	brief_j_2
	briefing_2	brief_o_1	brief_k_1
	allowed_weap	brief_o_2	
	allowed_weap_2	brief_p_1	
	allowed_cars_1		
	allowed_cars_2		

Thanks to:

I dedicate this work to Jakob the love of my life, Lynx, Elice, Jens, my whole family, mom, dad, brother and pets (living and dead), Pao you showed me that real friends do exist and I love you for it, elmindreda you are an amazing person and you give me a lot of positive energy every time I see you.

Special thanks to everyone else that have ever supported me and helped me trough rough times.

Very special thanks to the players of Wolfys Mod, Gunnar at Grin, and the whole Grin team.

Disclaimer:

I do not take any kind of responsibility for what people might do with the information in this manual and I cannot be held responsible for any thing that might happen when someone uses this modding manual.