



Basic Modding Guide

1 Folder structure

The first thing to know before you start to change things in Zombie Driver is to get acquainted with the game folder structure. After installing ZD, go to your installation folder. For steam users you can locate the game in your steam folder:

```
..\Steam\SteamApps\common\zombie driver
```

Inside you should see the following folders:

- Audio
- DirectX
- Documentation
- OpenAl
- Packs
- PhysX
- Release
- Save
- Video

The interesting ones are: Audio, Packs and Release.

2 Audio

The audio folder contains all sounds and music included with the game. You can pretty much start modding it as it is. Just replace the files with your versions and it will work. Remember about the file format and extension. We are using simple uncompressed wav files and ogg formats.

3 Packs

The packs folder contains two pak files: OgreCore.zip and Pak0.zip. The Pak0.zip file is basically a virtual file system which is used to compress everything together. This benefits in greatly reduced loading times on most of the machines.

The Pak0.zip contains almost all game content excluding audio and movies. You can simply unpack it using any zip decompressor and a new folder structure will emerge:

- Fonts
- Materials
- Meshes
- Scripts

3.1 Fonts

Fonts folder contains fonts used in the main menu and their definitions. If you would like to change them replace the ttf file with your versions or add new ones in a similar way. New fonts will also have to be defined in the scripts/frontend definitions for the menu to use them.



3.2 Materials

In the materials folder you can find all materials and textures used in the game. If you would like to change a road, building, car or any other texture used in the game just replace the correct file with your version and that's it. If you would like to change properties of a material or to create new ones, head on to the Materials/Scripts folder and modify one of the material files.

For example open the Manmade.material file located in the Materials/Scripts folder. Inside you should be able to find such definition:

```
material Manmade/car_vw_taxi_yellow: Base/Default
{
    set_texture_alias Diffuse    Materials/Textures/Manmade/car_vw_taxi_yellow.dds
}
```

Those few lines define the material name, shader and texture used on the taxi with which you start the game.

3.3 Meshes

In the meshed folder you can find all 3D models that are used in the game. The 3D models have 2 formats:

- *.mesh
- *.nxs

The mesh file represents the visual model used by the Ogre rendering engine while the nxs file represents the collision model used by the PhysX engine. Many models do not require collisions so in this case you won't find an nxs file for them.

For example go to the Meshes/Cars folder and in there you will be able to find the following files:

- taxi.mesh (this represents the taxi visual model with which you start the game)
- taxi.skeleton (this is the file with taxi bones – currently not used)
- taxi_c1.mesh (this is the file used to create a collision model for the taxi)
- taxi_c1.nxs (this is the collision model created out of taxi_c1.mesh)

3.3.1 City

Besides the regular 3d models used in the game you can check the Meshes/City folder. In there you will find 3 files:

- City.scene
- entities_description.xml
- physics_seg_0_0.nxs

The city.scene file is an xml file containing the whole city layout (except the mission specific things). It is generated automatically but can be modified by hand as well.

The entities_description file is also an xml that contains definitions of all entities used in the city. Whether they are static, destructible, collidable etc. it is defined in this file.

The physics_seg_0_0.nxs is a collision file for all static props in the city.



3.4 Scripts

In the scripts folder you will find all kinds of scripts in a text format which can be very easily modified. This is probably the fastest way to create a cool mod.

For example go to the Scripts/Cars folder and open the taxi.xml file. In there you will see a bunch of parameters that control our taxi behavior. You can for example quite easily make it faster, slower or to turn much more aggressively.

In other scripts you can find and easily modify weapons, zombies, missions etc.

4 Setup for modding

Using pack is rather cumbersome when you are creating your mod but it is a great way to distribute your mod. How to distribute it in a convenient way will be described a bit later on.

The best way to start modding is to unpack the folders contained in a pack (Pak0.zip) to the root folder so you would have a structure like that:

- Audio
- DirectX
- Documentation
- **Fonts**
- **Materials**
- **Meshes**
- OpenAI
- Packs
- PhysX
- Release
- Save
- **Scripts**
- Video

With that structure you can start changing things. Zombie Driver is by default setup to take the files from the Pak0.zip so if you want the game to use your modified files you will have to change the resources.cfg in the Release folder. In that configuration file you will see a line like this:

```
Multi=FileSystem:../,Zip:../Packs/Pak0.zip
```

FileSystem:../ means that the game will use the resources from the file system (unpacked folders and files as described above)

Zip:../Packs/Pak0.zip means that the game will use the resources from the Pak0.zip

If you want to change the setup so that the game will use the unpacked files in the folders swap the order of the zip and filesystem so it looks like that:

```
Multi= Zip:../Packs/Pak0.zip,FileSystem:../
```

The order in which you place those strings indicates what is more important for the game. This is a cascading structure so the last thing that is defined here is the most important one.



Of course you do not have to extract everything. If you just want to change some weapon parameters you can create/extract the proper files/folders in your system, modify them and the rest will be taken from the original pack file.

5 Distributing your mod

As mentioned, pack files are great to distribute your mod. To create a pack file simply compress the files you have modified or added and name the file accordingly (i.e. my_mod.zip).

The file that you created can be now placed in the Pack folder along side with the OgreCore.zip, Pak0.zip and any other mods.

To use the mod you will have to add it in the resources.cfg located in the Release folder. The line defining you mod should look like this:

```
Multi=FileSystem:../Zip:../Packs/Pak0.zip, ,Zip:../Packs/my_mod.zip
```

In the near future we will add a possibility to select mods from the menu before you start the game. This means that you won't have to modify the resources.cfg. You will simply start the game with a mod of your choosing from the ones that are present in the Packs folder.

