
OpenMW Documentation

Release 0.42.0

Bret Curtis

Jul 29, 2017

Contents

1	Sections	1
1.1	User Manuals	1
1.1.1	OpenMW CS User Manual	1
1.2	Reference Material	10
1.2.1	OpenMW Modding Reference	10

User Manuals

OpenMW CS User Manual

The following document is the complete user manual for *OpenMW CS*, the construction set for the OpenMW game engine. It is intended to serve both as an introduction and a reference for the application. Even if you are familiar with modding *The Elder Scrolls III: Morrowind* you should at least read the first few chapters to familiarise yourself with the new interface.

Warning: OpenMW CS is still software in development. The manual does not cover any of its shortcomings, it is written as if everything was working as intended. Please report any software problems as bugs in the software, not errors in the manual.

Foreword

<Some introductory lines here, explain nomenclature and abbreviations>

How to read the manual

The manual can be roughly divided into two parts: a tutorial part consisting of the first two (three) chapters and the reference manual. We recommend all readers to work through the tutorials first, there you will be guided through the creation of a fairly simple mod where you can familiarise yourself with the record-based interface. The tutorials are very simple and teach you only what is necessary for the task, each one can be completed in under half an hour. It is strongly recommended to do the tutorials in order.

When you are familiar with the CS in general and want to write your own mods it is time to move on to the reference part of the manual. The reference chapters can be read out of order, each one covers only one topic.

A Tour through OpenMW CS: making a magic ring

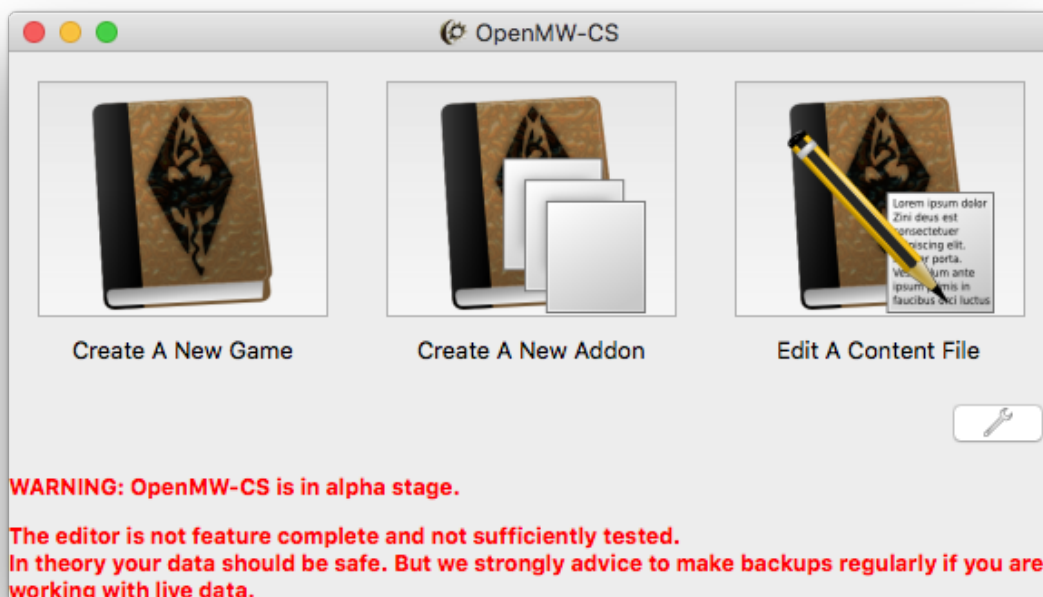
In this first chapter we will create a mod that adds a new ring with a simple enchantment to the game. The ring will give its wearer a permanent Night Vision effect while being worn. You do not need previous Morrowind modding experience, but you should be familiar with the game itself. There will be no scripting necessary, we can achieve everything using just what the base game offers out of the box. Before continuing make sure that OpenMW is properly installed and playable.

Adding the ring to the game's records

In this first section we will define what our new ring is, what it looks like, and what it does. Getting it to work is the first step before we go further.

Starting up OpenMW CS

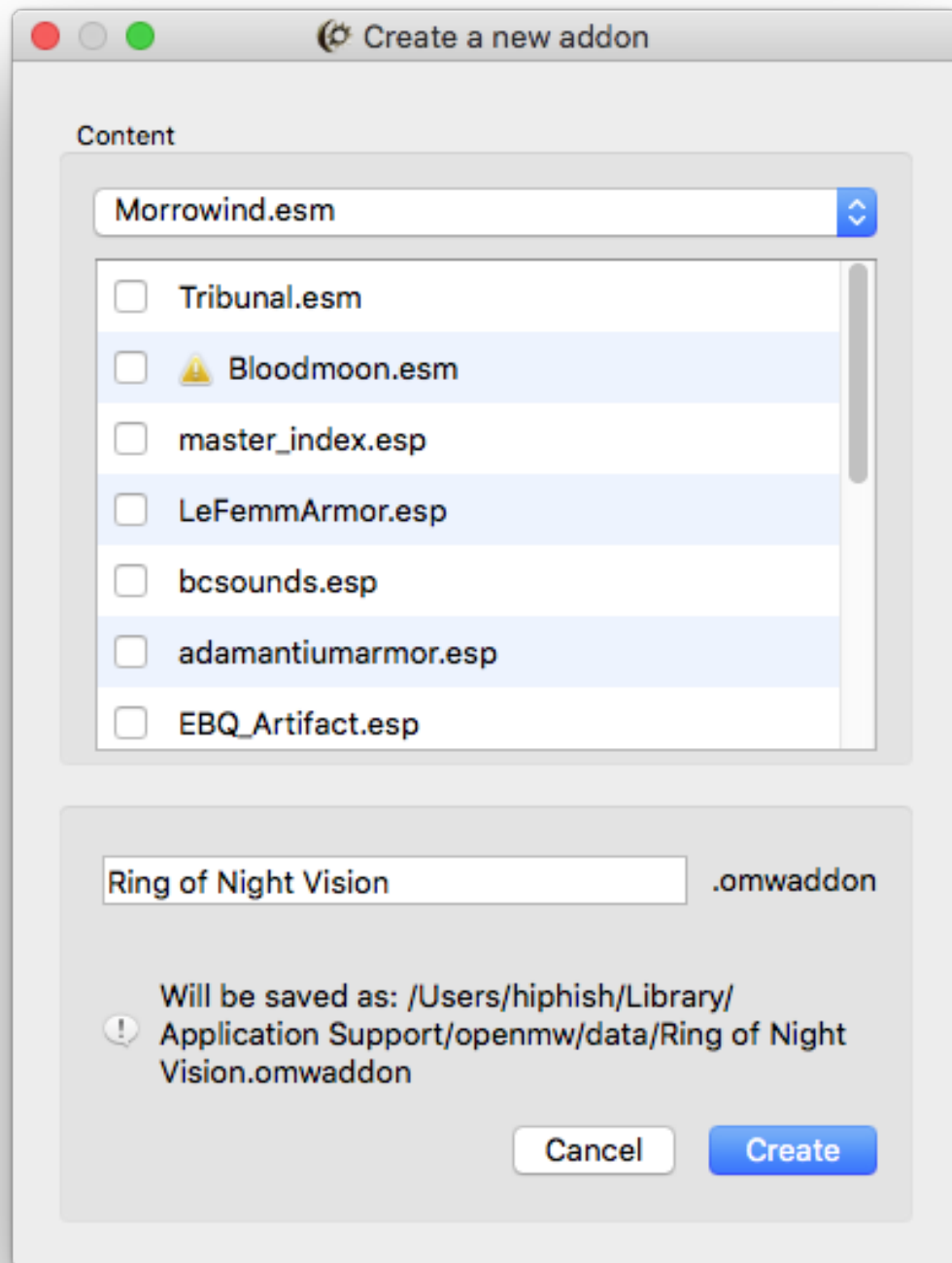
We will start by launching OpenMW CS, the location of the program depends on your operating system. You will be presented with a dialogue with three options: create a new game, create a new addon, edit a content file.



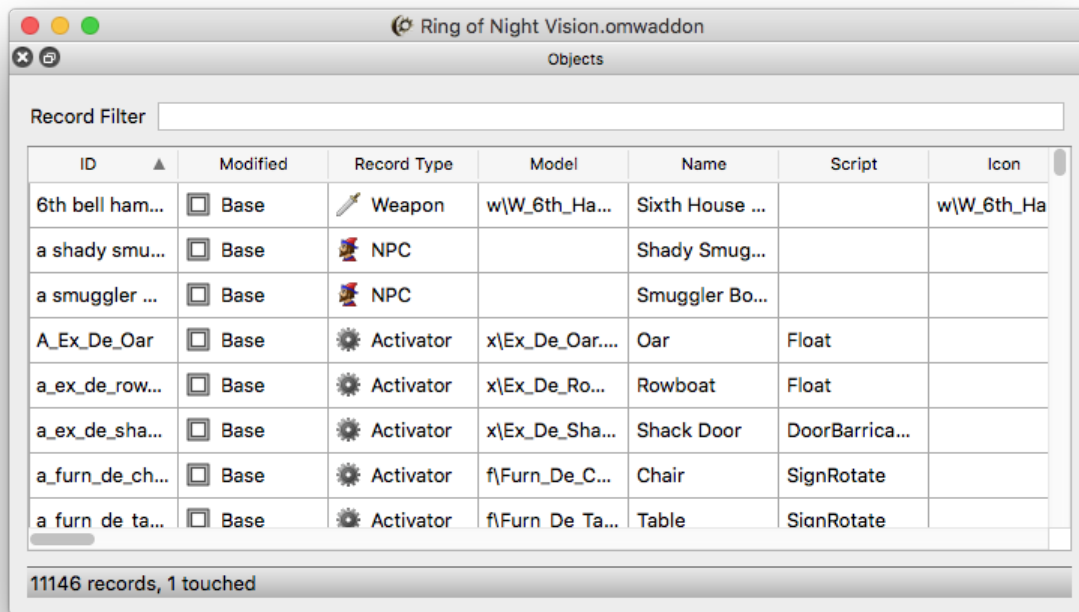
The first option is for creating an entirely new game, that's not what we want. We want to edit an existing game, so choose the second option. When you save your addon you can use the third option to open it again.

You will be presented with another window where you get to choose the content to edit and the name of your project. Then we have to select at least the base game and optionally a number of other addons we want to depend on. The name of the project is arbitrary, it will be used to identify the addon later in the OpenMW launcher.

Choose Morrowind as your content file and enter *Ring of Night Vision* as the name. We could also choose further content files as dependencies if we wanted to, but for this mod the base game is enough.



Once the addon has been created you will be presented with a table. If you see a blank window rather than a table choose *World* → *Objects* from the menu.



Let's talk about the interface for a second. Every window in OpenMW CS has *panels*, these are often but not always tables. You can close a panel by clicking the small "X" on the title bar of the panel, or you can detach it by either dragging the title bar or clicking the icon with the two windows. A detached panel can be re-attached to a window by dragging it by the title bar on top of the window.

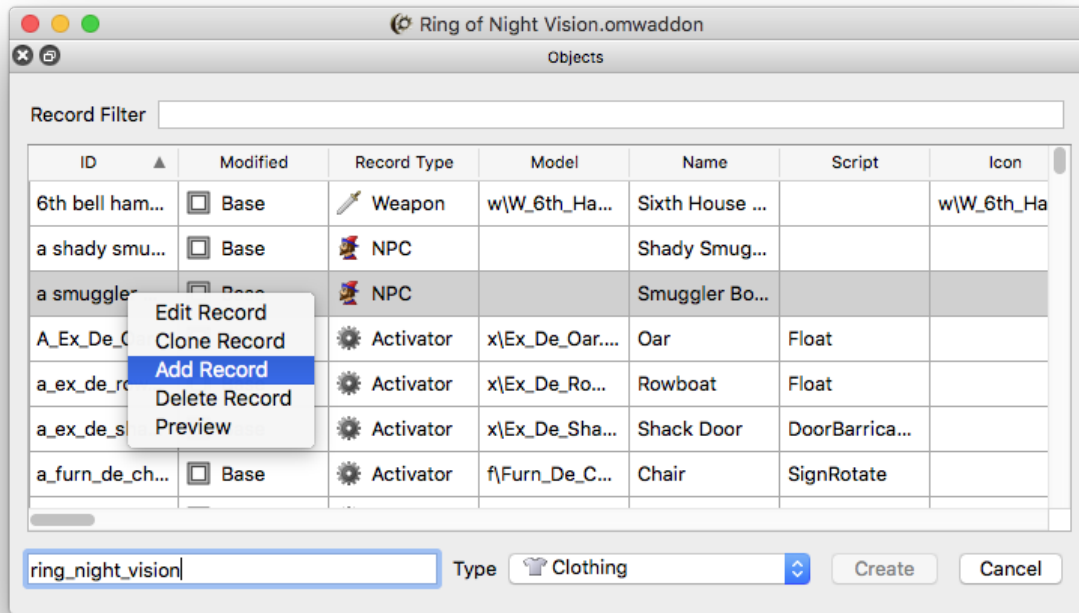
Now let's look at the panel itself: we have a filter text field, a very large table and a status bar. The filter will be very useful when we want to find an entry in the table, but for now it is irrelevant. The table you are looking at contains all objects in the game, these can be items, NPCs, creatures, whatever. Every object is an entry in that table, visible as a row. The columns of the table are the attributes of each object.

Morrowind uses something called a *relational database* for game data. If you are not familiar with the term, it means that every type of thing can be expressed as a *table*: there is a table for objects, a table for enchantments, a table for icons, one for meshes and so on. Properties of an entry must be simple values, like numbers or text strings. If we want a more complicated property we need to reference an entry from another table. There are a few exceptions to this though, some tables do have subtables. The effects of enchantments are one of those exceptions.

Defining a new record

Enough talk, let's create the new ring now. Right-click anywhere in the objects table, choose *Add Record* and the status bar will change into an input field. We need to enter an *ID* (short for *identifier*) and pick the type. The identifier is a unique name by which the ring can later be identified; I have chosen *ring_night_vision*. For the type choose *Clothing*.

The table should jump right to our newly created record, if not read further below how to use filters to find a record by ID. Notice that the *Modified* column now shows that this record is new. Records can also be *Base* (unmodified), *Modified* and *Deleted*. The other fields are still empty since we created this record from nothing. We can double-click



a table cell while holding Shift to edit it (this is a configurable shortcut), but there is a better way: right-click the row of our new record and chose *Edit Record*, a new panel will open.

We can right-click the row of our new record and select *Edit Record*, a new panel will open. Alternatively we can also define a configurable shortcut instead of using the context menu; the default is double-clicking while holding down the shift key.

You can set the name, weight and coin value as you like, I chose *Ring of Night Vision*, *0.1* and *2500* respectively. Make sure you set the *Clothing Type* to *Ring*. We could set the other properties manually as well, but unless you have an exceptional memory for identifiers and never make typos that's not feasible. What we are going to do instead is find the records we want in their respective tables and assign them from there.

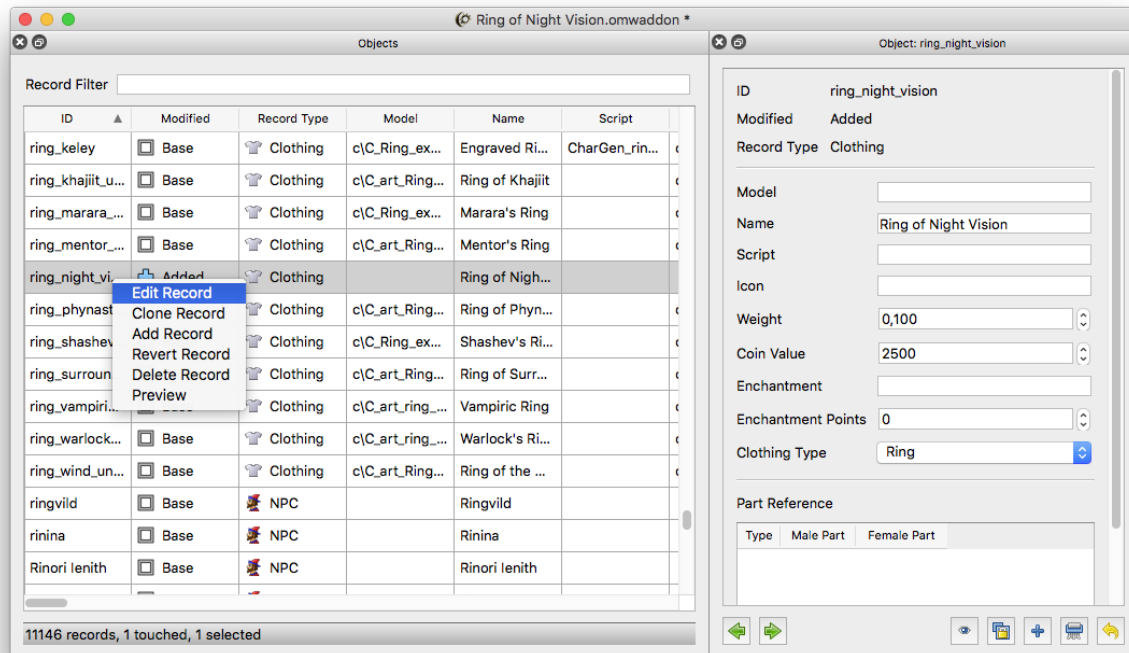
Finding records using filters

We will add an icon first. Open the *Icons* table the same way you opened the *Objects* table: in the menu click *Assets* → *Icons*. If the window gets too crowded remember that you can detach panels. The table is huge and not every ring icon starts with “ring”, so we have to use filters to find what we want.

Filters are a central element of OpenMW CS and a major departure from how the original Morrowind CS was used. In fact, filters are so important that they have their own table as well. We won't be going that far for now though. There are three types of filters: *Project filters* are part of the project and are stored in the project file, *session filter* are only valid until you exit the CS, and finally *instant filter* which are used only once and typed directly into the *Filter* field.

For this tutorial we will only use instant filters. We type the definition of the filter directly into the filter field rather than the name of an existing filter. To signify that we are using an instant filter the have to use *!* as the first character. Type the following into the field:

```
!string("id", ".*ring.*")
```



A filter is defined by a number of *queries* which can be logically linked. For now all that matters is that the *string(<property>, <pattern>)* query will check whether *<property>* matches *<pattern>*. The pattern is a regular expression, if you don't know about them you should learn their syntax. For now all that matters is that *.* stands for any character and *** stands for any amount, even zero. In other words, we are looking for all entries which have an ID that contains the word "ring" somewhere in it. This is a pretty dumb pattern because it will also match words like "ringmail", but it's good enough for now.

If you have typed the filter definition properly the text should change from red to black and our table will be narrowed down a lot. Browse for an icon you like and drag & drop its table row onto the *Icon* field of our new ring.

That's it, you have now assigned a reference to an entry in another table to the ring entry in the *Objects* table. Repeat the same process for the 3D model, you can find the *Meshes* table under *Assets* → *Meshes*.

Adding the enchantment

Putting everything you have learned so far to practice we can add the final and most important part to our new ring: the enchantment. You know enough to perform the following steps without guidance: Open the *Enchantments* table (*Mechanics* → *Enchantments*) and create a new entry with the ID *Cats Eye*. Edit it so that it has *Constant Effect* enchantment type.

To add an effect to the enchantment right-click the *Magic Effects* table and choose *Add new row*. You can edit the effects by right-clicking their table cells. Set the effect to *NightEye*, range to *Self*, and both magnitudes to *50*. The other properties are irrelevant.

Once you are done add the new enchantment to our ring. That's it, we now have a complete enchanted ring to play with. Let's take it for a test ride.

Playing your new addon

Launch OpenMW and in the launcher under *Data Files* check your addon. Load a game and open the console. We have only defined the ring, but we haven't placed any instance of it anywhere in the game world, so we have to create one. In the console type:

```
player->AddItem "ring_night_vision" 1
```

The part in quotation marks is the ID of our ring, you have to adjust it if you chose a different ID. Exit the console and you should find a new ring in your inventory. Equip it and you will instantly receive the *Night Vision* effect for your character.

Conclusion

In this tutorial we have learned how to create a new addon, what tables are and how to create new records. We have also taken a very brief glimpse at the syntax of filters, a feature you will be using a lot when creating larger mods.

This mod is a pure addition, it does not change any of the existing records. However, if you want to actually present appealing content to the player rather than just offering abstract definitions you will have to change the game's content. In the next tutorial we will learn how to place the ring in the game world so the player can find it legitimately.

Adding the ring to the game's world

Now that we have defined the ring it is time add it to the game world so the player can find it legitimately. We will add the ring to a merchant, place it in a chest and put it somewhere in plain sight. To this end we will have to actually modify the contents of the game.

Subsection to come...

Files and Directories

In this chapter of the manual we will cover the usage of files and directories by OpenMW CS. Files and directories are file system concepts of your operating system, so we will not be going into specifics about that, we will only focus on what is relevant to OpenMW CS.

Basics

Directories

OpenMW and OpenMW CS use multiple directories on the file system. First of all there is a *user directory* that holds configuration files and a number of different sub-directories. The location of the user directory is hard-coded into the CS and depends on your operating system.

Operating System	User Directory
GNU/Linux	<whatever>
OS X	~/Library/Application Support/openmw/
Windows	<whatever>

In addition to this single hard-coded directory both OpenMW and OpenMW CS need a place to seek for actual data files of the game: textures, 3D models, sounds and record files that store objects in game; dialogues and so on. These files are called *content files*. We support multiple such paths (we call them *data paths*) as specified in the

configuration. Usually one data path points to the directory where the original Morrowind game is either installed or unpacked to. You are free to specify as many data paths as you would like, however, there is one special data path that, as described later, which is used to store newly created content files.

Content files

The original Morrowind engine by Bethesda Softworks uses two types of content files: *esm* (master) and *esp* (plugin). The distinction between those two is not clear, and often confusing. One would expect the *esm* (master) file to be used to specify one master, which is then modified by the *esp* plugins. And indeed: this is the basic idea. However, the official expansions were also made as ESM files, even though they could essentially be described as really large plugins, and therefore would rather use *esp* files. There were technical reasons behind this decision – somewhat valid in the case of the original engine, but clearly it is better to create a system that can be used in a more sensible way. OpenMW achieves this with our own content file types.

We support both ESM and ESP files, but in order to make use of new features in OpenMW one should consider using new file types designed with our engine in mind: *game* files and *addon* files, collectively called *content files*.

OpenMW content files

The concepts of *Game* and *Addon* files are somewhat similar to the old concept of *ESM* and *ESP*, but more strictly enforced. It is quite straight-forward: If you want to make new game using OpenMW as the engine (a so called *total conversion*) you should create a game file. If you want to create an addon for an existing game file create an addon file. Nothing else matters; the only distinction you should consider is if your project is about changing another game or creating a new one. Simple as that.

Another simple thing about content files are the extensions: we are using `.omwaddon` for addon files and `.omwgame` for game files.

Morrowind content files

Using our content files is recommended for projects that are intended to be used with the OpenMW engine. However, some players might wish to still use the original Morrowind engine. In addition thousands of *ESP/ESM* files were created since 2002, some of them with really outstanding content. Because of this OpenMW CS simply has no other choice but to support *ESP/ESM* files. If you decide to choose *ESP/ESM* file instead of using our own content file types you are most likely aiming at compatibility with the original engine. This subject is covered in its own chapter of this manual.

The actual creation of new files is described in the next chapter. Here we are going to focus only on the details you need to know in order to create your first OpenMW CS file while fully understanding your needs. For now let's just remember that content files are created inside the user directory in the `data` subdirectory (that is the one special data directory mentioned earlier).

Dependencies

Since an addon is supposed to change the game it follows that it also depends on the said game to work. We can conceptualise this with an example: your modification is the changing prize of an iron sword, but what if there is no iron sword in game? That's right: we get nonsense. What you want to do is tie your addon to the files you are changing. Those can be either game files (for example when making an expansion island for a game) or other addon files (making a house on said island). Obviously it is a good idea to be dependent only on files that are really changed in your addon, but sadly there is no other way to achieve this than knowing what you want to do. Again, please remember that this section of the manual does not cover creating the content files – it is only a theoretical introduction

to the subject. For now just keep in mind that dependencies exist, and is up to you to decide whether your content file should depend on other content files.

Game files are not intend to have any dependencies for a very simple reasons: the player is using only one game file (excluding original and the dirty ESP/ESM system) at a time and therefore no game file can depend on other game file, and since a game file makes the base for addon files it can not depend on addon files.

Project files

Project files act as containers for data not used by the OpenMW game engine itself, but still useful for OpenMW CS. The shining example of this data category are without doubt record filters (described in a later chapter of the manual). As a mod author you probably do not need or want to distribute project files at all, they are meant to be used only by you and your team.

As you would imagine, project files make sense only in combination with actual content files. In fact, each time you start to work on new content file and a project file was not found, one will be created. The extension of project files is `.project`. The whole name of the project file is the whole name of the content file with appended extension. For instance a `swords.omwaddon` file is associated with a `swords.omwaddon.project` file.

Project files are stored inside the user directory, in the `projects` subdirectory. This is the path location for both freshly created project files, and a place where OpenMW CS looks for already existing files.

Resource files

Unless we are talking about a fully text based game, like Zork or Rogue, one would expect that a video game is using some media files: 3D models with textures, images acting as icons, sounds and anything else. Since content files, no matter whether they are *ESP*, *ESM* or new OpenMW file type, do not contain any of those, it is clear that they have to be delivered with a different file. It is also clear that this, let's call it "resources file", has to be supported by the engine. Without code handling those files it is nothing more than a mathematical abstraction – something, that lacks meaning for human beings. Therefore this section must cover ways to add resources files to your content file, and point out what is supported. We are going to do just that. Later, you will learn how to make use of those files in your content.

Audio

OpenMW uses [FFmpeg](#) for audio playback, and so we support every audio type supported by that library. This makes a huge list. Below is only small portion of the supported file types.

mp3 (MPEG-1 Part 3 Layer 3) A popular audio file format and de facto standard for storing audio. Used by the Morrowind game.

ogg An open source, multimedia container file using a high quality [Vorbis](#) audio codec. Recommended.

Video

Video As in the case of audio files, we are using [FFmpeg](#) to decode video files. The list of supported files is long, we will cover only the most significant.

bik Videos used by the original Morrowind game.

mp4 A multimedia container which use more advanced codecs (MPEG-4 Parts 2, 3 and 10) with a better audio and video compression rate, but also requiring more CPU intensive decoding – this makes it probably less suited for storing sounds in computer games, but good for videos.

webm A new, shiny and open source video format with excellent compression. It needs quite a lot of processing power to be decoded, but since game logic is not running during cutscenes we can recommend it for use with OpenMW.

ogv Alternative, open source container using [Theora](#) codec for video and Vorbis for audio.

Textures and images

The original Morrowind game uses *DDS* and *TGA* files for all kinds of two dimensional images and textures alike. In addition, the engine supported *BMP* files for some reason (*BMP* is a terrible format for a video game). We also support an extended set of image files – including *JPEG* and *PNG*. *JPEG* and *PNG* files can be useful in some cases, for instance a *JPEG* file is a valid option for skybox texture and *PNG* can be useful for masks. However, please keep in mind that *JPEG* images can grow to large sizes quickly and are not the best option with a DirectX rendering backend. You probably still want to use *DDS* files for textures.

OpenMW CS Starting Dialog

In this chapter we will cover starting up OpenMW CS and the starting interface. Start the CS the way intended for your operating system and you will be presented with window and three main buttons and a small button with a wrench-icon. The wrench will open the configuration dialog which we will cover later. The three main buttons are the following:

Create A New Game Choose this option if you want to create an original game that does not depend on any other content files. The distinction between game and addon in the original Morrowind engine was somewhat blurry, but OpenMW is very strict about it: regardless of how large your addon is, if it depends on another content file it is not an original game.

Create A New Addon Choose this option if you want to create an extension to an existing game. An addon can depend on other addons as well optionally, but it *must* depend on a game.

Edit A Content File Choose this option if you wish to edit an existing content file, regardless of whether it is a game or an addon.

Whether you create a game or an addon, a data file and a project file will be generated for you in your user directory.

You will have to choose a name for your content file and if you chose to create an addon you will also have to choose a number of dependencies. You have to choose exactly one game and you can choose an arbitrary amount of addon dependencies. For the sake of simplicity and maintainability choose only the addons you actually want to depend on. Also keep in mind that your dependencies might have dependencies of their own, you have to depend on those as well. If one of your dependencies needs something it will be indicated by a warning sign and automatically include its dependencies when you choose it.

If you want to edit an existing content file you will be presented with a similar dialog, except you don't get to choose a file name (because you are editing files that already exist).

Reference Material

OpenMW Modding Reference

The following document is the complete reference guide to modifying, or modding, your OpenMW setup. It does not cover content creation itself, only how to alter or add to your OpenMW gameplay experience. To learn more about creating new content for OpenMW, please refer to [OpenMW CS User Manual](#).

Warning: OpenMW is still software in development. This manual does not cover any of its shortcomings. It is written as if everything was working as intended. Please report any software problems as bugs in the software, not errors in the manual.

Foreword

OpenMW is a complete game engine built to be content agnostic. The majority of this guide is applicable to any non-Morrowind project using its engine. That being said, it was designed with the extensive modding community of Morrowind in mind. Therefore, if you are already familiar with modding in Morrowind, you will likely be able to start modding in OpenMW with little to no instruction. We do recommend you at least refer to [Modding OpenMW vs Morrowind](#) to find out about what's different between OpenMW and the original Morrowind engine. For everyone else, or just a good refresher, read on!

Modding OpenMW vs Morrowind

A brief overview of the differences between the two engines.

OpenMW is designed to be able to use all the normal Morrowind mod files such as ESM/ESP plugins, texture replacers, mesh replacers, etc.

Warning: All external programs and libraries that depend on `morrowind.exe` cannot function with OpenMW. This means you should assume mods dependent on Morrowind Graphics Extender, Morrowind Code Patch, Morrowind Script Extender, etc, will *not* work correctly, nor will the tools themselves.

Multiple Data Folders

The largest difference between OpenMW and Morrowind in terms of data structure is OpenMW's support of multiple data folders. This has many advantages, especially when it comes to uninstalling mods and preventing unintentional overwrites of files.

Warning: Most mods can still be installed into the root OpenMW data folder, but this is not recommended.

To install mods via this new feature:

1. Open `openmw.cfg` with your preferred text editor. It is located as described in [Paths](#) and *not* in your OpenMW root directory.
2. Find or search for `data=`. This is located very near the bottom of the file.
3. Add a new line below this line and make a new entry of the format `data="path/to/your/mod"`
4. Make as many of these entries as you need for each mod folder you want to include.
5. Save `openmw.cfg`

Note: All mod folders must adhere to the same file structure as `~/Morrowind/Data Files/`.

To uninstall these mods simply delete that mod's respective `data=` entry. The mods are loaded in the order of these entries, with the top being overwritten by mods added towards the bottom.

Note: Mods that depend on ESM/ESP plugins can be rearranged within the OpenMW Launcher, but mesh/texture replacer mods can only be reordered by moving their `data=` entry.

OpenMW Launcher

The launcher included with OpenMW is similar to the original Morrowind Launcher. Go to the Data Files tab to enable and disable plugins. You can also drag list items to modify the load order. Content lists can be created at the bottom by clicking the New Content List button, creating a list name, then setting up a new modlist. This is helpful for different player profiles and testing out different load orders.

Settings.cfg

The `settings.cfg` file is essentially the same as the INI files for Morrowind. It is located in the same directory as `openmw.cfg`. This is where many video, audio, GUI, input, etc. settings can be modified. Some are available in-game, but many are only available in this configuration file. Please see <https://wiki.openmw.org/index.php?title=Settings> for the complete listing.

Open Source Resources Support

While OpenMW supports all of the original files that Morrowind supported, we've expanded support to many open source file formats. These are summarized below:

<this will be a table of the type of file, the morrowind supported file, and the OpenMW supported file formats>

How To Install and Use Mods

The following is a detailed guide on how to install and enable mods in OpenMW using best practices.

Install

1. Your mod probably comes in some kind of archive, such as `.zip`, `.rar`, `.7z`, or something along those lines. Unpack this archive into its own folder.
2. Ensure the structure of this folder is correct.
 - (a) Locate the plugin files, `.esp` or `.omwaddon`. The folder containing the plugin files we will call your *data folder*
 - (b) Check that all resource folders (`Meshes`, `Textures`, etc.) containing additional resource files (the actual meshes, textures, etc.) are in the *data folder*.

Note: There may be multiple levels of folders, but the location of the plugins must be the same as the resource folders.

3. Open your `openmw.cfg` file in your preferred plain text editor. It is located as described in [Paths](#) and *not* in your OpenMW root directory.

4. Find or search for `data=`. This is located very near the bottom of the file. If you are using Morrowind, this first entry should already point to your Morrowind data directory, `Data Files`; otherwise it will point to your game file, `.omwgame`.
5. Create a new line underneath and type: `data="path/to/your/data folder"` Remember, the *data folder* is where your mod's plugin files are. The double quotes around this path name are *required*.

Note: Some text editors, such as TextEdit on Mac, will autocorrect your double quotes to typographical “curly” quotes instead of leaving them as the proper neutral vertical quotes `" "`.

1. Save your `openmw.cfg` file.

You have now installed your mod. Any simple replacer mods that only contain resource files such as meshes or textures will now automatically be loaded in the order of their `data=*` entry. This is important to note because replacer mods that replace the same resource will overwrite previous ones as you go down the list.

Enable

Any mods that have plugin files must be enabled to work. Master game files and plugin files can only be enabled if they have been properly installed within a *data folder* as described above.

1. Open the OpenMW Launcher.
2. Click on the Data Files tab.
3. In the Content List box, select the content list you wish to modify in the dropdown menu, or make a new one by:
 - (a) Click the New Content List button and enter the name of your content list, then click OK. New lists are useful for keeping track of the mods used for different characters or for different games if you play more than one game using OpenMW.
 - (b) In the Content box, select your game file (`.esm` or `.omwgame`) from the dropdown menu.
4. Now you must activate the plugins you wish to use by checking the box next to their entry in the Content box list.
5. Load order can be changed simply by dragging the entries around within the list. Mods are loaded from the top down, so if one plugin depends on another, it must be lower on the list.
6. Click Play to run OpenMW with your game and enabled mods!

Advanced Settings Configuration

This part of the guide will cover how to make modifications to the more arcane settings in OpenMW, most of which are not available from in-game menus, to optimize or customize your OpenMW experience. If you are familiar with `.ini` tweaks in Morrowind or the other games, this will be quite similar. All settings described in this section are changed in `settings.cfg`, located in your OpenMW user directory. See paths for this location.

Although this guide attempts to be comprehensive and up to date, you will always be able to find the full list of settings available and their default values in `settings-default.cfg` in your main OpenMW installation directory. The ranges I have included with each setting are the physically possible ranges, not recommendations.

Warning: As the title suggests, these are advanced settings. If digging around plain text files and manually editing settings sounds scary to you, you may want to steer clear of altering these files. That being said, this guide should be plenty clear enough that you can find the setting you want to change and safely edit it.

Camera Settings

near clip

Type floating point

Range > 0

Default 1.0

This setting controls the distance to the near clipping plane. The value must be greater than zero. Values greater than approximately 18.0 will occasionally clip objects in the world in front of the character. Values greater than approximately 8.0 will clip the character's hands in first person view and/or the back of their head in third person view.

The default value is 1.0. This setting can only be configured by editing the settings configuration file. The value must be greater than 0.0, but it's unclear if the engine enforces this limitation.

small feature culling

Type boolean

Range True/False

Default True

This setting determines whether objects that render to a few pixels or smaller will be culled (not drawn). It generally improves performance to enable this feature, and by definition the culled objects will be very small on screen. The size in pixels for an object to be considered 'small' is controlled by a separate setting.

The default value is true. This setting can only be configured by editing the settings configuration file.

small feature culling pixel size

Type floating point

Range > 0

Default 2.0

Controls the cutoff in pixels for the 'small feature culling' setting which will have no effect if 'small feature culling' is disabled.

This setting can only be configured by editing the settings configuration file.

viewing distance

Type floating point

Range > 0

Default 6666.0

This value controls the maximum visible distance (also called the far clipping plane). Larger values significantly improve rendering in exterior spaces, but also increase the amount of rendered geometry and significantly reduce the frame rate. This value interacts with the exterior cell load distance setting in that it's probably undesired for this value to provide visibility into cells that have not yet been loaded. When cells are visible before loading, the geometry will “pop-in” suddenly, creating a jarring visual effect. To prevent this effect, this value must be less than:

$$(8192 * \text{exterior cell load distance} - 1024) * 0.93$$

The constant 8192 is the size of a cell, and 1024 is the threshold distance for loading a new cell. Additionally, the field of view setting also interacts with this setting because the view frustum end is a plane, so you can see further at the edges of the screen than you should be able to. This can be observed in game by looking at distant objects and rotating the camera so the objects are near the edge of the screen. As a result, this setting should further be reduced by a factor that depends on the field of view setting. In the default configuration this reduction is 7%, hence the factor of 0.93 above. Using this factor, approximate values recommended for other exterior cell load distance settings are:

Cells	
Viewing Distance	
2 14285	
3 21903	
4 29522	
5 35924	

Reductions of up to 25% or more can be required to completely eliminate pop-in for wide fields of view and long viewing distances near the edges of the screen, but such situations are unusual and probably not worth the performance penalty introduced by loading geometry obscured by fog in the center of the screen. See `RenderingManager::configureFog` for the relevant source code.

Enabling the distant terrain setting is an alternative to increasing exterior cell load distance. Note that the distant land setting does not include rendering of distant static objects, so the resulting visual effect is not the same.

The default value is 6666.0. This setting can be adjusted in game from the ridiculously low value of 2000.0 to a maximum of 6666.0, using the View Distance slider in the Detail tab of the Video panel of the Options menu.

field of view

Type floating point

Range 0-360

Default 55.0

Sets the camera field of view in degrees. Recommended values range from 30 degrees to 110 degrees. Small values provide a very narrow field of view that creates a “zoomed in” effect, while large values cause distortion at the edges of the screen. The “field of view” setting interacts with aspect ratio of your video resolution in that more square aspect ratios (e.g. 4:3) need a wider field of view to more resemble the same field of view on a widescreen (e.g. 16:9) monitor.

The default value is 55.0. This setting can be changed in game using the Field of View slider from the Video tab of the Video panel of the Options menu.

first person field of view

Type floating point

Range 0-360

Default 55.0

This setting controls the field of view for first person meshes such as the player's hands and held objects. It is not recommended to change this value from its default value because the Bethesda provided Morrowind assets do not adapt well to large values, while small values can result in the hands not being visible.

The default value is 55.0. This setting can only be configured by editing the settings configuration file.

Cells Settings

exterior cell load distance

Type integer

Range ≥ 1

Default 1

This setting determines the number of exterior cells adjacent to the character that will be loaded for rendering. Values greater than one may significantly affect loading times when exiting interior spaces or loading additional exterior cells. Caution is advised when increasing this setting.

This setting interacts with viewing distance and field of view settings.

It is generally very wasteful for this value to load geometry that will almost never be visible due to viewing distance and fog. For low frame rate screen shots of scenic vistas, this setting should be set high, and viewing distances adjusted accordingly.

The default value is 1. This value must be greater than or equal to 1. This setting can only be configured by editing the settings configuration file.

preload enabled

Type boolean

Range True/False

Default True

Controls whether textures and objects will be pre-loaded in background threads. This setting being enabled should result in a reduced amount of loading screens, no impact on frame rate and a varying amount of additional RAM usage, depending on how the preloader was configured (see the below settings). The default preloading settings with vanilla game files should only use negligible amounts of RAM, however, when using high-res texture and model replacers it may be necessary to tweak these settings to prevent the game from running out of memory.

The effects of (pre-)loading can be observed on the in-game statistics panel brought up with the 'F4' key.

All settings starting with 'preload' in this section will have no effect if preloading is disabled, and can only be configured by editing the settings configuration file.

preload num threads

Type integer

Range >=1

Default 1

Controls the number of worker threads used for preloading operations. In addition to the preloading threads, OpenMW uses a main thread, a sound streaming thread, and a graphics thread. Therefore, the default setting of one preloading thread will result in a total of 4 threads used which should work well with quad-core CPUs. If you have additional cores to spare, consider increasing the number of preloading threads to 2 or 3 for a boost in preloading performance. Faster preloading will reduce the chance that a cell could not be completely loaded before the player moves into it, and hence reduce the chance of seeing loading screens or frame drops. This may be especially relevant when the player moves at high speed and/or a large number of cells are loaded in via 'exterior cell load distance'.

A value of 4 or higher is not recommended. With 4 or more threads, improvements will start to diminish due to file reading and synchronization bottlenecks.

preload exterior grid

Type boolean

Range True/False

Default True

Controls whether adjacent cells are preloaded when the player moves close to an exterior cell border.

preload fast travel

Type boolean

Range True/False

Default False

Controls whether fast travel destinations are preloaded when the player moves close to a travel service. Because the game can not predict the destination that the player will choose, all possible destinations will be preloaded. This setting is disabled by default due to the adverse effect on memory usage caused by the preloading of all possible destinations.

preload doors

Type boolean

Range True/False

Default True

Controls whether locations behind a door are preloaded when the player moves close to the door.

preload distance

Type floating point

Range >0

Default 1000

Controls the distance in in-game units that is considered the player being ‘close’ to a preloading trigger. Used by all the preloading mechanisms i.e. ‘preload exterior grid’, ‘preload fast travel’ and ‘preload doors’.

For measurement purposes, the distance to an object in-game can be observed by opening the console, clicking on the object and typing ‘getdistance player’.

preload instances

Type boolean

Range True/False

Default True

Controls whether or not objects are also pre-instanced on top of being pre-loaded. Instancing happens when the same object is placed more than once in the cell, and to be sure that any modifications to one instance do not affect the other, the game will create independent copies (instances) of the object. If this setting is enabled, the creation of instances will be done in the preloading thread; otherwise, instancing will only happen in the main thread once the cell is actually loaded.

Enabling this setting should reduce the chance of frame drops when transitioning into a preloaded cell, but will also result in some additional memory usage.

preload cell cache min

Type integer

Range >0

Default 12

The minimum number of preloaded cells that will be kept in the cache. Once the number of preloaded cells in the cache exceeds this setting, the game may start to expire preloaded cells, based on the ‘preload cell expiry delay’ setting, starting with the oldest cell. When a preloaded cell expires, all the assets that were loaded for it will also expire and will have to be loaded again the next time the cell is requested for preloading.

preload cell cache max

Type integer

Range >0

Default 20

The maximum number of cells that will ever be in pre-loaded state simultaneously. This setting is intended to put a cap on the amount of memory that could potentially be used by preload state.

preload cell expiry delay

Type floating point

Range >=0

Default 5

The amount of time (in seconds) that a preloaded cell will stay in cache after it is no longer referenced or required, for example, after the player has moved away from a door without entering it.

cache expiry delay

Type floating point

Range ≥ 0

Default 5

The amount of time (in seconds) that a preloaded texture or object will stay in cache after it is no longer referenced or required, for example, when all cells containing this texture have been unloaded.

Map Settings

global map size

Type integer

Range ≥ 1

Default 18

This setting adjusts the scale of the world map in the GUI mode map window. The value is the width in pixels of each cell in the map, so larger values result in larger more detailed world maps, while smaller values result in smaller less detailed world maps. However, the native resolution of the map source material appears to be 9 pixels per unexplored cell and approximately 18 pixels per explored cell, so values larger than 36 don't produce much additional detail. Similarly, the size of place markers is currently fixed at 12 pixels, so values smaller than this result in overlapping place markers. Values from 12 to 36 are recommended. For reference, Vvardenfell is approximately 41x36 cells.

Warning: Changing this setting affects saved games. The currently explored area is stored as an image in the save file that's overlaid on the default world map in game. When you increase the resolution of the map, the overlay of earlier saved games will be scaled up on load, and appear blurry. When you visit the cell again, the overlay for that cell is regenerated at the new resolution, so the blurry areas can be corrected by revisiting all the cells you've already visited.

The default value for this setting is 18. This setting can not be configured except by editing the settings configuration file.

local map hud widget size

Type integer

Range ≥ 1

Default 256

This setting controls the zoom level for the HUD map widget (the map in the lower right corner of the window). A value of 64 results in the HUD map widget displaying one entire exterior cell. Since the GUI mode map displays 3x3 cells, a value of approximately 21 displays the same area as the GUI mode map. Larger values increase the level of zoom, while smaller values are wasteful since there's no map data to display beyond the 3x3 cell grid.

Note that the actual size of the widget is always the same on the screen unless the scaling factor setting in the "GUI" section is changed. Increasing both the scaling factor of the GUI and this setting does result in a higher resolution HUD map, but unfortunately with a scaled direction pointer on top of it.

The default value for this setting is 256. This setting can not be configured except by editing the settings configuration file.

local map resolution

Type integer

Range ≥ 1

Default 256

This setting controls the resolution of the GUI mode local map window. Larger values generally increase the visible detail in map. If this setting is half the local map widget size or smaller, the map will generally be fairly blurry. Setting both options to the same value results in a map with good detail. Values that exceed the local map widget size setting by more than a factor of two are unlikely to provide much of an improvement in detail since they're subsequently scaled back to the approximately the map widget size before display. The video resolution settings interacts with this setting in that regard.

Warning: Increasing this setting can increase cell load times, because the map is rendered on demand each time you enter a new cell. Large values may exceed video card limits or exhaust VRAM.

The default value for this setting is 256. This setting can not be configured except by editing the settings configuration file.

local map widget size

Type integer

Range ≥ 1

Default 512

This setting controls the canvas size of the GUI mode local map window. Larger values result in a larger physical map size on screen, and typically require more panning to see all available portions of the map. This larger size also enables an overall greater level of detail if the local map resolution setting is also increased.

The default value for this setting is 512. This setting can not be configured except by editing the settings configuration file.

GUI Settings

scaling factor

Type floating point

Range > 0.0

Default 1.0

This setting scales the GUI interface windows. The value must be greater than 0.0. A value of 1.0 results in the normal scale. Values much larger than 2.0 may result in user interface components being inaccessible. Until a gamepad interface is created, increasing this setting is helpful for simulating the larger interface used in console games.

The default value is 1.0. This setting can only be configured by editing the settings configuration file.

menu transparency

Type floating point

Range 0.0 (transparent) to 1.0 (opaque)

Default 0.84

This setting controls the transparency of the GUI windows. The value should be between 0.0 (transparent) and 1.0 (opaque). The default value is 0.84. This setting can be adjusted in game with the Menu Transparency slider in the Prefs panel of the Options menu.

tooltip delay

Type floating point

Range > 0.0

Default 0.0

This value determines the number of seconds between when you begin hovering over an item and when its tooltip appears. This setting only affects the tooltip delay for objects under the crosshair in GUI mode windows. There does not appear to be a setting to control the tool tip delay in outside of GUI mode.

The tooltip displays context sensitive information on the selected GUI element, such as weight, value, damage, armor rating, magical effects, and detailed description.

The default value is 0.0. This setting can be adjusted between 0.0 and 1.0 in game with the Menu Help Delay slider in the Prefs panel of the Options menu.

stretch menu background

Type boolean

Range True/False

Default False

Stretch or shrink the main menu screen, loading splash screens, introductory movie, and cut scenes to fill the specified video resolution, distorting their aspect ratio. The Bethesda provided assets have a 4:3 aspect ratio, but other assets are permitted to have other aspect ratios. If this setting is false, the assets will be centered in their correct aspect ratio, with black bars filling the remainder of the screen.

The default value is false. This setting can only be configured by editing the settings configuration file.

subtitles

Type boolean

Range True/False

Default False

Enable or disable subtitles for NPC spoken dialog (and some sound effects). Subtitles will appear in a tool tip box in the lower center of the screen.

The default value is false. This setting can be toggled in game with the Subtitles button in the Prefs panel of Options menu.

hit fader

Type boolean

Range True/False

Default True

This setting enables or disables the “red flash” overlay that provides a visual clue when the character has taken damage. If this setting is disabled, the player will “bleed” like NPCs do.

The default value is true. This setting can only be configured by editing the settings configuration file.

werewolf overlay

Type boolean

Range True/False

Default True

Enable or disable the werewolf overlay.

The default value is true. This setting can only be configured by editing the settings configuration file.

color background owned

Type RGBA floating point

Range 0.0 to 1.0

Default 0.15 0.0 0.0 1.0

The following two settings determine the background color of the tool tip and the crosshair when hovering over an item owned by an NPC. The color definitions are composed of four floating point values between 0.0 and 1.0 inclusive, representing the red, green, blue and alpha channels. The alpha value is currently ignored. The crosshair color will have no effect if the crosshair setting in the HUD section is disabled.

The default value is “0.15 0.0 0.0 1.0”, which is a dark red color. This setting can only be configured by editing the settings configuration file. This setting has no effect if the show owned setting in the Game Settings Section is false.

color crosshair owned

Type RGBA floating point

Range 0.0 to 1.0

Default 1.0 0.15 0.15 1.0

This setting sets the color of the crosshair when hovering over an item owned by an NPC. The value is composed of four floating point values representing the red, green, blue and alpha channels. The alpha value is currently ignored.

The default value is “1.0 0.15 0.15 1.0” which is a bright red color. This setting can only be configured by editing the settings configuration file. This setting has no effect if the crosshair setting in the HUD Settings Section is false. This setting has no effect if the show owned setting in the Game Settings Section is false.

HUD Settings

crosshair

Type boolean

Range True/False

Default True

This setting determines whether the crosshair or reticle is displayed. Some players perceive that disabling the crosshair provides a more immersive experience. Another common use is to disable the crosshair for screen shots. Enabling the crosshair provides more immediate feedback about which object is currently the focus of actions.

The default value is true. This setting can be toggled with the Crosshair button in the Prefs panel of the Options menu.

Game Settings

show owned

Type integer

Range 0, 1, 2, 3

Default 0

Enable visual clues for items owned by NPCs when the crosshair is on the object. If the setting is 0, no clues are provided which is the default Morrowind behavior. If the setting is 1, the background of the tool tip for the object is highlight in the color specified by the color background owned setting in the GUI Settings Section. If the setting is 2, the crosshair is the color of the color crosshair owned setting in the GUI Settings section. If the setting is 3, both the tool tip background and the crosshair are colored. The crosshair is not visible if crosshair is false.

The default value is 0 (no clues). This setting can only be configured by editing the settings configuration file.

best attack

Type boolean

Range True/False

Default False

If this setting is true, the player character will always use the most powerful attack when striking with a weapon (chop, slash or thrust). If this setting is false, the type of attack is determined by the direction that the character is moving at the time the attack begins.

The default value is false. This setting can be toggled with the Always Use Best Attack button in the Prefs panel of the Options menu.

difficulty

Type integer

Range -500 to 500

Default 0

This setting adjusts the difficulty of the game and is intended to be in the range -100 to 100 inclusive. Given the default game setting for fDifficultyMult of 5.0, a value of -100 results in the player taking 80% of the usual damage, doing 6 times the normal damage. A value of 100 results in the player taking 6 times as much damage, but inflicting only 80% of the usual damage. Values less than -500 will result in the player receiving no damage, and values greater than 500 will result in the player inflicting no damage.

The default value is 0. This setting can be controlled in game with the Difficulty slider in the Prefs panel of the Options menu.

show effect duration

Type boolean

Range True/False

Default False

Show the remaining duration of magic effects and lights if this setting is true. The remaining duration is displayed in the tooltip by hovering over the magical effect.

The default value is false. This setting can only be configured by editing the settings configuration file.

General Settings

anisotropy

Type integer

Range 0 to 16

Default 4

Set the maximum anisotropic filtering on textures. Anisotropic filtering is a method of enhancing the image quality of textures on surfaces that are at oblique viewing angles with respect to the camera. Valid values range from 0 to 16. Modern video cards can often perform 8 or 16 anisotropic filtering with a minimal performance impact. This effect of this setting can be seen in the Video panel of the Options menu by finding a location with straight lines (striped rugs and Balmora cobblestones work well) radiating into the distance, and adjusting the anisotropy slider.

The default value is 4. This setting can be changed in game using the Anisotropy slider in the Detail tab of the Video panel of the Options menu.

screenshot format

Type string

Range jpg, png, tga

Default png

Specify the format for screen shots taken by pressing the screen shot key (bound to F12 by default). This setting should be the file extension commonly associated with the desired format. The formats supported will be determined at compilation, but “jpg”, “png”, and “tga” should be allowed.

The default value is “png”. This setting can only be configured by editing the settings configuration file.

texture filtering

Type string

Range bilinear, trilinear

Default trilinear

Set the isotropic texture filtering mode to bilinear or trilinear. Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are. Bilinear filtering is reasonably accurate until the scaling of the texture gets below half or above double the original size of the texture. Trilinear filtering is an extension of the bilinear texture filtering method, which also performs linear interpolation between mipmaps. Both methods use mipmaps in OpenMW, and the corresponding OpenGL modes are `LINEAR_MIPMAP_NEAREST` and `LINEAR_MIPMAP_LINEAR`. Trilinear filtering produces better texturing at a minimal cost on modern video cards.

The default value is trilinear. This setting can be changed in game using the Texture filtering pull down in the Detail tab of the Video panel of the Options menu.

Input Settings

grab cursor

Type boolean

Range True/False

Default True

OpenMW will capture control of the cursor if this setting is true.

In “look mode”, OpenMW will center the cursor regardless of the value of this setting (since the cursor/crosshair is always centered in the OpenMW window). However, in GUI mode, this setting determines the behavior when the cursor is moved outside the OpenMW window. If true, the cursor movement stops at the edge of the window preventing access to other applications. If false, the cursor is allowed to move freely on the desktop.

This setting does not apply to the screen where escape has been pressed, where the cursor is never captured. Regardless of this setting “Alt-Tab” or some other operating system dependent key sequence can be used to allow the operating system to regain control of the mouse cursor. This setting interacts with the minimize on focus loss setting by affecting what counts as a focus loss. Specifically on a two-screen configuration it may be more convenient to access the second screen with setting disabled.

Note for developers: it’s desirable to have this setting disabled when running the game in a debugger, to prevent the mouse cursor from becoming unusable when the game pauses on a breakpoint.

The default value is true. This setting can only be configured by editing the settings configuration file.

toggle sneak

Type boolean

Range True/False

Default False

This setting causes the behavior of the sneak key (bound to Ctrl by default) to toggle sneaking on and off rather than requiring the key to be held down while sneaking. Players that spend significant time sneaking may find the character easier to control with this option enabled.

The default value is false. This setting can only be configured by editing the settings configuration file.

always run

Type boolean

Range True/False

Default False

If this setting is true, the character is running by default, otherwise the character is walking by default. The shift key will temporarily invert this setting, and the caps lock key will invert this setting while it's "locked". This setting is updated every time you exit the game, based on whether the caps lock key was on or off at the time you exited.

The default value is false. This settings can be toggled in game by pressing the CapsLock key and exiting.

allow third person zoom

Type boolean

Range True/False

Default False

Allow zooming in and out using the middle mouse wheel in third person view. This feature may not work correctly if the mouse wheel is bound to other actions, and may be triggered accidentally in some cases, so is disabled by default. This setting can only be configured by editing the settings configuration file.

camera sensitivity

Type floating point

Range > 0

Default 1.0

This setting controls the overall camera/mouse sensitivity when not in GUI mode. The default sensitivity is 1.0, with smaller values requiring more mouse movement, and larger values requiring less. This setting is multiplicative in magnitude. This setting does not affect mouse speed in GUI mode, which is instead controlled by your operating system mouse speed setting.

The default value is 1.0. This setting can be changed with the Camera Sensitivity slider in the Controls panel of the Options menu.

camera y multiplier

Type floating point

Range > 0

Default 1.0

This setting controls the vertical camera/mouse sensitivity relative to the horizontal sensitivity (see camera sensitivity above). It is multiplicative with the previous setting, meaning that it should remain set at 1.0 unless the player desires to have different sensitivities in the two axes.

The default value is 1.0. This setting can only be configured by editing the settings configuration file.

invert y axis

Type boolean

Range True/False

Default False

Invert the vertical axis while not in GUI mode. If this setting is true, moving the mouse away from the player will look down, while moving it towards the player will look up. This setting does not affect cursor movement in GUI mode.

The default value is false. This setting can be toggled in game with the Invert Y Axis button in the Controls panel of the Options menu.

Saves Settings

character

Type string

Range

Default ""

This setting contains the default character name for loading saved games.

The default value is the empty string, which results in no character being selected by default. This setting is automatically updated from the Load menu when a different character is selected.

autosave

Type boolean

Range True/False

Default True

This setting determines whether the game will be automatically saved when the character rests.

The default value is true. This setting can be toggled in game with the Auto-Save when Rest button in the Prefs panel of the Options menu.

timeplayed

Type boolean

Range True/False

Default False

This setting determines whether the amount of the time the player has spent playing will be displayed for each saved game in the Load menu.

The default value is false. This setting can only be configured by editing the settings configuration file.

Sound Settings

device

Type string

Range

Default ""

This setting determines which audio device to use. A blank or missing setting means to use the default device, which should usually be sufficient, but if you need to explicitly specify a device use this setting.

The names of detected devices can be found in the openmw.log file in your configuration directory.

The default value is the empty string. This setting can only be configured by editing the settings configuration file.

master volume

Type floating point

Range 0.0 to 1.0

Default 1.0

This setting controls the overall volume. The master volume is multiplied with all other volume settings to determine the final volume.

The default value is 1.0. Valid values range from 0.0 (silent) to 1.0 (maximum volume). This setting can be changed in game using the Master slider from the Audio panel of the Options menu.

footsteps volume

Type floating point

Range 0.0 to 1.0

Default 0.2

This setting controls the volume of footsteps from the character and other actors.

The default value is 0.2. Valid values range from 0.0 (silent) to 1.0 (maximum volume). This setting can be changed in game using the Footsteps slider from the Audio panel of the Options menu.

music volume

Type floating point

Range 0.0 to 1.0

Default 0.5

This setting controls the volume for music tracks.

The default value is 0.5. Valid values range from 0.0 (silent) to 1.0 (maximum volume). This setting can be changed in game using the Music slider from the Audio panel of the Options menu.

sfx volume

Type floating point

Range 0.0 to 1.0

Default 1.0

This setting controls the volume for special sound effects such as combat noises.

The default value is 1.0. Valid values range from 0.0 (silent) to 1.0 (maximum volume). This setting can be changed in game using the Effects slider from the Audio panel of the Options menu.

voice volume

Type floating point

Range 0.0 to 1.0

Default 0.8

This setting controls the volume for spoken dialog from NPCs.

The default value is 0.8. Valid values range from 0.0 (silent) to 1.0 (maximum volume). This setting can be changed in game using the Voice slider from the Audio panel of the Options menu.

buffer cache min

Type integer

Range > 0

Default 14

This setting determines the minimum size of the sound buffer cache in megabytes. When the cache reaches the size specified by the buffer cache max setting, old buffers will be unloaded until it's using no more memory than specified by this setting. This setting must be less than or equal to the buffer cache max setting.

The default value is 14. This setting can only be configured by editing the settings configuration file.

buffer cache max

Type integer

Range > 0

Default 16

This setting determines the maximum size of the sound buffer cache in megabytes. When the cache reaches this size, old buffers will be unloaded until it reaches the size specified by the buffer cache min setting. This setting must be greater than or equal to the buffer cache min setting.

The default value is 16. This setting can only be configured by editing the settings configuration file.

hrtf enable

Type integer

Range -1, 0, 1

Default -1

This setting determines whether to enable head-related transfer function (HRTF) audio processing. HRTF audio processing creates the perception of sounds occurring in a three dimensional space when wearing headphones. Enabling HRTF may also require an OpenAL Soft version greater than 1.17.0, and possibly some operating system configuration. A value of 0 disables HRTF processing, while a value of 1 explicitly enables HRTF processing. The default value is -1, which should enable the feature automatically for most users when possible. This setting can only be configured by editing the settings configuration file.

hrtf

Type string

Range

Default ""

This setting specifies which HRTF profile to use when HRTF is enabled. Blank means use the default. This setting has no effect if HRTF is not enabled based on the hrtf enable setting. Allowed values for this field are enumerated in openmw.log file is an HRTF enabled audio system is installed.

The default value is the empty string, which uses the default profile. This setting can only be configured by editing the settings configuration file.

Terrain Settings

distant terrain

Type boolean

Range True/False

Default False

Controls whether the engine will use paging and LOD algorithms to render the terrain of the entire world at all times. Otherwise, only the terrain of the loaded cells is displayed. This setting is best used together with the 'viewing distance' setting in the camera section.

To avoid frame drops as the player moves around, nearby terrain pages are always preloaded in the background, regardless of the preloading settings in the 'Cells' section, but the preloading of terrain behind a door or a travel destination, for example, will still be controlled by cell preloading settings.

The distant terrain engine is currently considered experimental and may receive updates and/or further configuration options in the future. The glaring omission of non-terrain objects in the distance somewhat limits this setting's usefulness.

Video Settings

resolution x

Type integer

Range > 0

Default 800

This setting determines the horizontal resolution of the OpenMW game window. Larger values produce more detailed images within the constraints of your graphics hardware but also significantly reduce the frame rate.

The default value is 800. The window resolution can be selected from a menu of common screen sizes in the Video tab of the Video Panel of the Options menu, or in the Graphics tab of the OpenMW Launcher. The horizontal resolution can also be set to a custom value in the Graphics tab of the OpenMW Launcher.

resolution y

Type integer

Range > 0

Default 600

This setting determines the vertical resolution of the OpenMW game window. Larger values produce more detailed images within the constraints of your graphics hardware but also significantly reduce the frame rate.

The default value is 600. The window resolution can be selected from a menu of common screen sizes in the Video tab of the Video Panel of the Options menu, or in the Graphics tab of the OpenMW Launcher. The vertical resolution can also be set to a custom value in the Graphics tab of the OpenMW Launcher.

fullscreen

Type boolean

Range True/False

Default False

This setting determines whether the entire screen is used for the specified resolution.

The default value is false. This setting can be toggled in game using the Fullscreen button in the Video tab of the Video panel in the Options menu. It can also be toggled with the Full Screen check box in the Graphics tab of the OpenMW Launcher.

screen

Type integer

Range >= 0

Default 0

This setting determines which screen the game will open on in multi-monitor configurations. This setting is particularly important when the fullscreen setting is true, since this is the only way to control which screen is used, but it can also be used to control which screen a normal window or a borderless window opens on as well. The screens are numbered in increasing order, beginning with 0.

The default value is 0. This setting can be selected from a pull down menu in the Graphics tab of the OpenMW Launcher, but cannot be changed during game play.

minimize on focus loss

Type boolean

Range True/False

Default False

Minimize the OpenMW window if it loses cursor focus. This setting is primarily useful for single screen configurations, so that the OpenMW screen in full screen mode can be minimized when the operating system regains control of the mouse and keyboard. On multiple screen configurations, disabling this option makes it easier to switch between screens while playing OpenMW.

Note that a minimized game window consumes less system resources and produces less heat, since the game does not need to render in minimized state. It is therefore advisable to minimize the game during pauses (either via use of this setting, or by minimizing the window manually).

This setting has no effect if the fullscreen setting is false.

Developer note: corresponds to `SDL_HINT_VIDEO_MINIMIZE_ON_FOCUS_LOSS`.

The default value is true. This setting can only be configured by editing the settings configuration file.

window border

Type boolean

Range True/False

Default True

This setting determines whether there's an operating system border drawn around the OpenMW window. If this setting is true, the window can be moved and resized with the operating system window controls. If this setting is false, the window has no operating system border.

This setting has no effect if the fullscreen setting is true.

The default value is true. This setting can be toggled in game using the Window Border button in the Video tab of the Video panel in the Options menu. It can also be toggled with the Window Border check box in the OpenMW Launcher.

antialiasing

Type integer

Range 0, 2, 4, 8, 16

Default 0

This setting controls anti-aliasing. Anti-aliasing is a technique designed to improve the appearance of polygon edges, so they do not appear to be “jagged”. Anti-aliasing can smooth these edges at the cost of a minor reduction in the frame rate. A value of 0 disables anti-aliasing. Other powers of two (e.g. 2, 4, 8, 16) are supported according to the capabilities of your graphics hardware. Higher values do a better job of smoothing out the image but have a greater impact on frame rate.

This setting can be configured from a list of valid choices in the Graphics panel of the OpenMW Launcher, but cannot be changed during game play - due to a technical limitation that may be addressed in a future version of OpenMW.

vsync

Type boolean

Range True/False

Default False

This setting determines whether frame draws are synchronized with the vertical refresh rate of your monitor. Enabling this setting can reduce screen tearing, a visual defect caused by updating the image buffer in the middle of a screen draw. Enabling this option typically implies limiting the framerate to 60 frames per second, but may also introduce additional delays caused by having to wait until the appropriate time (the vertical blanking interval) to draw a frame.

The default value is false. This setting can be adjusted in game using the VSync button in the Video tab of the Video panel in the Options menu. It can also be changed by toggling the Vertical Sync check box in the Graphics tab of the OpenMW Launcher.

framerate limit

Type floating point

Range ≥ 0.0

Default 0.0

This setting determines the maximum frame rate in frames per second. If this setting is 0.0, the frame rate is unlimited.

There are several reasons to consider capping your frame rate, especially if you're already experiencing a relatively high frame rate (greater than 60 frames per second). Lower frame rates will consume less power and generate less heat and noise. Frame rates above 60 frames per second rarely produce perceptible improvements in visual quality, but may improve input responsiveness. Capping the frame rate may in some situations reduce the perception of choppiness (highly variable frame rates during game play) by lowering the peak frame rates.

This setting interacts with the vsync setting in the Video section in the sense that enabling vertical sync limits the frame rate to the refresh rate of your monitor (often 60 frames per second). Choosing to limit the frame rate using this setting instead of vsync may reduce input lag due to the game not having to wait for the vertical blanking interval.

The default value is 0.0. This setting can only be configured by editing the settings configuration file.

contrast

Type floating point

Range > 0.0

Default 1.0

This setting controls the contrast correction for all video in the game.

The default value is 1.0. This setting can only be configured by editing the settings configuration file. This setting does not currently work under Linux.

gamma

Type floating point

Range > 0.0

Default 1.0

This setting controls the gamma correction for all video in the game. Gamma is an exponent that makes colors brighter if greater than 1.0 and darker if less than 1.0.

The default value is 1.0. This setting can be changed in the Detail tab of the Video panel of the Options menu. This setting does not currently work under Linux, and the in-game setting in the Options menu has been disabled.

Water Settings

Note: The settings for the water shader are difficult to describe, but can be seen immediately in the Water tab of the Video panel in the Options menu. Changes there will be saved to these settings. It is suggested to stand on the shore of a moderately broad body of water with trees or other objects on the far shore to test reflection textures, underwater plants in shallow water near by to test refraction textures, and some deep water visible from your location to test deep water visibility.

shader

Type boolean

Range True/False

Default False

This setting enables or disables the water shader, which results in much more realistic looking water surfaces, including reflected objects and a more detailed wavy surface.

The default value is false. This setting can be toggled with the Shader button in the Water tab of the Video panel of the Options menu.

rtt size

Type integer

Range > 0

Default 512

The setting determines the size of the texture used for reflection and refraction (if enabled). For reflection, the texture size determines the detail of reflected images on the surface of the water. For refraction, the texture size determines the detail of the objects on the other side of the plane of water (which have a wavy appearance caused by the refraction). RTT is an acronym for Render to Texture which allows rendering of the scene to be saved as a texture. Higher values produces better visuals and result in a marginally lower frame rate depending on your graphics hardware.

In the Water tab of the Video panel of the Options menu, the choices are Low (512), Medium (1024) and High (2048). This setting has no effect if the shader setting is false. It is recommended to use values that are a power of two because this results in more efficient use of video hardware.

This setting has no effect if the shader setting is false.

refraction

Type boolean

Range True/False

Default False

This setting enables the refraction rendering feature of the water shader. Refraction causes deep water to be more opaque and objects seen through the plane of the water to have a wavy appearance. Enabling this feature results in better visuals, and a marginally lower frame rate depending on your graphics hardware.

This setting has no effect if the shader setting is false.

The default setting is false. This setting can be toggled with the Refraction button in the Water tab of the Video panel of the Options menu.

small feature culling pixel size

Type floating point

Range > 0

Default 20.0

Controls the cutoff in pixels for small feature culling - see the 'Camera' section for more details, however this setting in the 'Water' section applies specifically to objects rendered in water reflection and refraction textures.

The setting 'rtt size' interacts with this setting because it controls how large a pixel on the water texture (technically called a texel) is in pixels on the screen.

This setting will have no effect if the shader setting is false, or the 'small feature culling' (in the 'Camera' section) is disabled.

This setting can only be configured by editing the settings configuration file.

Windows Settings

Type floating point

Range 0.0 to 1.0

This section controls the location and size of each window in GUI mode. Each setting is a floating point number representing a *fraction* of the resolution x or resolution y setting in the Video Settings Section. The X and Y values locate the top left corner of the window, while the W value determines the width of the window and the H value determines the height of the window.

Unlike the documentation for most sections which lists the exact setting name, this page instead lists the names of the windows. For example, to configure the alchemy window, the actual settings would be:

```
alchemy x = 0.25
alchemy y = 0.25
alchemy h = 0.5
alchemy w = 0.5
```

Each window in the GUI mode remembers it's previous location when exiting the game. By far the easiest way to configure these settings is to simply move the windows around in game. Hand editing the configuration file might result in some fine tuning for alignment, but the settings will be overwritten if a window is moved.

Note: To scale the windows, making the widgets proportionally larger, see the scaling factor setting instead.

stats

Default $x = 0.0$ $y = 0.0$ $h = 0.375$ $w = 0.4275$

The stats window, displaying level, race, class, skills and stats. Activated by clicking on any of the three bars in the lower left corner of the HUD.

spells

Default $x = 0.625$ $y = 0.5725$ $h = 0.375$ $w = 0.4275$

The spells window, displaying powers, spells, and magical items. Activated by clicking on the spells widget (third from left) in the bottom left corner of the HUD.

map

Default $x = 0.625$ $y = 0.0$ $h = 0.375$ $w = 0.5725$

The local and world map window. Activated by clicking on the map widget in the bottom right corner of the HUD.

dialogue

Default $x = 0.095$ $y = 0.095$ $h = 0.810$ $w = 0.810$

The dialog window, for talking with NPCs. Activated by clicking on a NPC.

alchemy

Default $x = 0.25$ $y = 0.25$ $h = 0.5$ $w = 0.5$

The alchemy window, for crafting potions. Activated by dragging an alchemy tool on to the rag doll. Unlike most other windows, this window hides all other windows when opened.

console

Default $x = 0.0$ $y = 0.0$ $h = 1.0$ $w = 0.5$

The console command window. Activated by pressing the tilde (~) key.

inventory

Default $x = 0.0$ $y = 0.4275$ $h = 0.6225$ $w = 0.5725$

The inventory window, displaying the paper doll and possessions, when activated by clicking on the inventory widget (second from left) in the bottom left corner of the HUD.

inventory container

Default $x = 0.0$ $y = 0.4275$ $h = 0.6225$ $w = 0.5725$

The player's inventory window while searching a container, showing the contents of the character's inventory. Activated by clicking on a container. The same window is used for searching dead bodies, and pickpocketing people.

inventory barter

Default $x = 0.0$ $y = 0.4275$ $h = 0.6225$ $w = 0.5725$

The player's inventory window while bartering. It displays goods owned by the character while bartering. Activated by clicking on the Barter choice in the dialog window for an NPC.

inventory companion

Default $x = 0.0$ $y = 0.4275$ $h = 0.6225$ $w = 0.5725$

The player's inventory window while interacting with a companion. The companion windows were added in the Tribunal expansion, but are available everywhere in the OpenMW engine.

container

Default $x = 0.25$ $y = 0.0$ $h = 0.75$ $w = 0.375$

The container window, showing the contents of the container. Activated by clicking on a container. The same window is used for searching dead bodies, and pickpocketing people.

barter

Default $x = 0.25$ $y = 0.0$ $h = 0.75$ $w = 0.375$

The NPC bartering window, displaying goods owned by the shopkeeper while bartering. Activated by clicking on the Barter choice in the dialog window for an NPC.

companion

Default $x = 0.25$ $y = 0.0$ $h = 0.75$ $w = 0.375$

The NPC's inventory window while interacting with a companion. The companion windows were added in the Tribunal expansion, but are available everywhere in the OpenMW engine.

Fonts

Morrowind .fnt fonts

Morrowind uses a custom .fnt file format. It is not compatible with the Windows Font File .fnt format, nor compatible with .fnt formats from any other Bethesda games. To our knowledge, the format is undocumented and no tools for viewing or editing these fonts exist.

OpenMW can load this format and convert it on the fly into something usable (see font loader [source code](#)). In OpenMW 0.32, an `-export-fonts` command line option was added to write the converted font (a PNG image and an XML file describing the position of each glyph in the image) to the current directory.

TrueType fonts

Unlike vanilla Morrowind, OpenMW directly supports TrueType (`.ttf`) fonts. This is the recommended way to create new fonts.

- To replace the primary “Magic Cards” font:
 1. Download [Pelagiad](#) by Isak Larborn (aka Isaskar).
 2. Install the `openmw_font.xml` file into `resources/mygui/openmw_font.xml` in your OpenMW installation.
 3. Copy `Pelagiad.ttf` into `resources/mygui/` as well.
 4. If desired, you can now delete the original `Magic_Cards.*` files from your Data Files/Fonts directory.
- You can also replace the Daedric font:
 1. Download [Ayembedt](#) by Georg Duffner.
 2. Install `OMWAyembedt.otf` into `/resources/mygui/` folder in your OpenMW installation.
 3. Add the following lines to `openmw_font.xml`:

```
<Resource type="ResourceTrueTypeFont" name="Daedric">
  <Property key="Source" value="OMWAyembedt.otf"/>
  <Property key="Size" value="24"/>
  <Property key="Resolution" value="50"/>
  <Property key="Antialias" value="false"/>
  <Property key="TabWidth" value="8"/>
  <Property key="OffsetHeight" value="0"/>
  <Codes>
    <Code range="32"/>
    <Code range="65 90"/>
    <Code range="97 122"/>
  </Codes>
</Resource>
```

4. This font is missing a few glyphs (mostly punctuation), but is complete in the primary glyphs. If desired, you can now delete the original `daedric.*` files from your Data Files/Fonts directory.
- Another replacement for the Daedric font is [Oblivion](#) by Dongle.
 1. Install the `Oblivion.ttf` file `resources/mygui/`.
 2. The `openmw_fonts.xml` entry is:

```
<Resource type="ResourceTrueTypeFont" name="Daedric">
  <Property key="Source" value="Oblivion.ttf"/>
  <Property key="Size" value="30"/>
  <Property key="Resolution" value="50"/>
  <Property key="Antialias" value="false"/>
  <Property key="TabWidth" value="8"/>
  <Property key="OffsetHeight" value="0"/>
  <Codes>
    <Code range="32 34"/>
    <Code range="39"/>
  </Codes>
</Resource>
```

```

<Code range="44 46"/>
<Code range="48 59"/>
<Code range="63"/>
<Code range="65 90"/>
<Code range="97 122"/>
<Code range="172 173"/>
<Code range="255"/>
<Code range="376"/>
<Code range="894"/>
<Code range="8211 8212"/>
<Code range="8216 8217"/>
<Code range="8220 8221"/>
</Codes>
</Resource>

```

Bitmap fonts

Morrowind `.fnt` files are essentially a bitmap font, but using them is discouraged because of no Unicode support. MyGUI has its own format for bitmap fonts. An example can be seen by using the `-export-fonts` command line option (see above), which converts Morrowind `.fnt` to a MyGUI bitmap font. This is the recommended format to use if you wish to edit Morrowind's bitmap font or create a new bitmap font.

Normal maps from Morrowind to OpenMW

- *General introduction to normal map conversion*
 - *Normal Mapping in OpenMW*
 - *Activating normal mapping shaders in OpenMW*
 - *Normal mapping in Morrowind with Morrowind Code Patch*
 - *Normal mapping in Morrowind with MGE XE*
- *Converting PeterBitt's Scamp Replacer (Mod made for the MGE XE PBR prototype)*
 - *Tutorial - MGE*
- *Converting Lougian's Hlaalu Bump mapped (MCP's fake bump map function, part 1: without custom models)*
 - *Tutorial - MCP, Part 1*
- *Converting Apel's Various Things - Sacks (MCP's fake bump map function, part 2: with custom models)*
 - *Tutorial - MCP, Part 2*

General introduction to normal map conversion

Authors Joakim (Lysol) Berg

Updated 2016-11-11

This page has general information and tutorials on how normal mapping works in OpenMW and how you can make mods using the old fake normal mapping technique (such as [Netch Bump mapped](#) and [Hlaalu Bump mapped](#), and maybe the most (in)famous one to give shiny rocks in OpenMW, the mod [On the Rocks!](#), featured in MGSO and Morrowind Rebirth) work in OpenMW.

Note: The conversion made in the *Converting Apol's Various Things - Sacks*-part of this tutorial require the use of the application NifSkope. There are binaries available for Windows, but not for Mac or Linux. Reports say that NifSkope versions 1.X will compile on Linux as long as you have Qt packages installed, while the later 2.X versions will not compile.

Another note: I will use the terms bump mapping and normal mapping simultaneously. Normal mapping is one form of bump mapping. In other words, normal mapping is bump mapping, but bump mapping isn't necessarily normal mapping. There are several techniques for bump mapping, and normal mapping is the most common one today.

So let's get on with it.

Normal Mapping in OpenMW

Normal mapping in OpenMW works in a very simple way: The engine just looks for a texture with a *_n.dds* suffix, and you're done.

So to expand on this a bit, let's take a look at how a model seeks for textures.

Let us assume we have the model *example.nif*. In this model file, there should be a tag (NiSourceTexture) that states what texture it should use and where to find it. Typically, it will point to something like *exampletexture_01.dds*. This texture is supposed to be located directly in the Textures folder since it does not state anything else. If the model is a custom made one, modders tend to group their textures in separate folders, just to easily keep track of them. It might be something like *.Textures/moddername/exampletexture_02.dds*.

When OpenMW finally adds normal mapping, it simply takes the NiSourceTexture file path, e.g., *exampletexture_01.dds*, and looks for a *exampletexture_01_n.dds*. If it can't find this file, no normal mapping is added. If it *does* find this file, the model will use this texture as a normal map. Simple.

Activating normal mapping shaders in OpenMW

Before normal (and specular and parallax) maps will show up in OpenMW, you'll need to activate them in the *settings.cfg*-file. Add these rows where it would make sense:

```
[Shaders]
auto use object normal maps = true
auto use terrain normal maps = true
```

And while we're at it, why not activate specular maps too just for the sake of it?

```
auto use object specular maps = true
auto use terrain specular maps = true
```

Lastly, if you want really nice lights in OpenMW, add these rows:

```
force shaders = true
clamp lighting = false
```

See OpenMW's wiki page about [texture modding](#) to read further about this.

Normal mapping in Morrowind with Morrowind Code Patch

Conversion difficulty: *Varies. Sometimes quick and easy, sometimes time-consuming and hard.*

You might have bumped (pun intended) on a few bump-mapped texture packs for Morrowind that require the Morrowind Code Patch (MCP). You might even be thinking: Why doesn't OpenMW just support these instead of reinventing the wheel? I know it sounds strange, but it will make sense. Here's how MCP handles normal maps:

Morrowind does not recognize normal maps (they weren't really a "thing" yet in 2002), so even if you have a normal map, Morrowind will not load and display it. MCP has a clever way to solve this issue, by using something Morrowind *does* support, namely environment maps. You could add a tag for an environment map and then add a normal map as the environment map, but you'd end up with a shiny ugly model in the game. MCP solves this by turning down the brightness of the environment maps, making the model look *kind of* as if it had a normal map applied to it. I say kind of because it does not really look as good as normal mapping usually does. It was a hacky way to do it, but it was the only way at the time, and therefore the best way.

The biggest problem with this is not that it doesn't look as good as it could – no, the biggest problem in my opinion is that it requires you to state the file paths for your normal map textures *in the models*! For buildings, which often use several textures for one single model file, it could take *ages* to do this, and you had to do it for dozens of model files too. You also had to ship your texture pack with model files, making your mod bigger in file size.

These are basically the reasons why OpenMW does not support fake bump maps like MCP does. It is just a really bad way to enhance your models, all the more when you have the possibility to do it in a better way.

Normal mapping in Morrowind with MGE XE

Conversion difficulty: *Easy*

The most recent feature on this topic is that the Morrowind Graphics Extender (MGE) finally started to support real normal mapping in an experimental version available here: [MGE XE](#) (you can't use MGE with OpenMW!). Not only this but it also adds full support for physically based rendering (PBR), making it one step ahead of OpenMW in terms of texturing techniques. However, OpenMW will probably have this feature in the future too – and let's hope that OpenMW and MGE will handle PBR in a similar fashion in the future so that mods can be used for both MGE and OpenMW without any hassle.

I haven't researched that much on the MGE variant yet but it does support real implementation of normal mapping, making it really easy to convert mods made for MGE into OpenMW (I'm only talking about the normal map textures though). There's some kind of text file if I understood it correctly that MGE uses to find the normal map. OpenMW does not need this, you just have to make sure the normal map has the same name as the diffuse texture but with the correct suffix after.

Now, on to the tutorials.

Converting PeterBitt's Scamp Replacer

Mod made for the MGE XE PBR prototype

Authors Joakim (Lysol) Berg

Updated 2016-11-11

So, let's say you've found out that [PeterBitt](#) makes awesome models and textures featuring physically based rendering (PBR) and normal maps. Let's say that you tried to run his [PBR Scamp Replacer](#) in OpenMW and that you were greatly disappointed when the normal map didn't seem to work. Lastly, let's say you came here, looking for some answers. Am I right? Great. Because you've come to the right place!

A quick note before we begin: Please note that you can only use the normal map texture and not the rest of the materials, since PBR isn't implemented in OpenMW yet. Sometimes PBR textures can look dull without all of the texture files, so have that in mind.

Tutorial - MGE

In this tutorial, I will use PeterBitt's [PBR Scamp Replacer](#) as an example, but any mod featuring PBR that requires the PBR version of MGE will do, provided it also includes a normal map (which it probably does).

So, follow these steps:

1. Go to the Nexus page for PeterBitt's [PBR Scamp Replacer](#)
2. Go to the *files* tab and download the main file and the "PBR materials" file.
3. Extract the main file as if you'd install a normal mod (**Pro tip:** Install using OpenMW's [Multiple data folders](#) function!)
4. **Now, open the PBR materials file:**
 - Go to `./Materials/PB/`.
 - Select the `tx_Scamp_normals.dds` file, which is, obviously, the normal map texture.
 - Extract this file to the place you extracted the main file to, but in the subdirectory `./Textures/PB/`.
5. Rename your newly extracted file (`tx_Scamp_normals.dds`) to `tx_Scamp_n.dds` (which is exactly the same name as the diffuse texture file, except for the added `_n` suffix before the filename extension).
6. You're actually done!

So as you might notice, converting these mods is very simple and takes just a couple of minutes. It's more or less just a matter of renaming and moving a few files.

I totally recommend you to also try this on PeterBitt's Nix Hound replacer and Flash3113's various replacers. It should be the same principle to get those to work.

And let's hope that some one implements PBR shaders to OpenMW too, so that we can use all the material files of these mods in the future.

Converting Lougian's Hlaalu Bump mapped

Mod made for MCP's fake bump function, without custom models

Authors Joakim (Lysol) Berg

Updated 2016-11-11

Converting textures made for the Morrowind Code Patch (MCP) fake bump mapping can be really easy or a real pain, depending on a few circumstances. In this tutorial, we will look at a very easy, although in some cases a bit time-consuming, example.

Tutorial - MCP, Part 1

We will be converting a quite popular texture replacer of the Hlaalu architecture, namely Lougian's [Hlaalu Bump mapped](#). Since this is just a texture pack and not a model replacer, we can convert the mod in a few minutes by just renaming a few dozen files and by *not* extracting the included model (`.nif`) files when installing the mod.

1. Download Lougian's [Hlaalu Bump mapped](#).
2. **Install the mod by extracting the `./Textures` folder to a data folder the way you usually install mods (Pro tip: Install us**

- Again, yes, *only* the `./Textures` folder. Do *not* extract the Meshes folder. They are only there to make the MCP hack work, which is not of any interest to us.
3. Go to your new texture folder. If you installed the mod like I recommended, you won't have any trouble finding the files. If you instead placed all your files in Morrowinds main Data Files folder (sigh), you need to check with the mod's .rar file to see what files you should look for. Because you'll be scrolling through a lot of files.
 4. Find all the textures related to the texture pack in the Textures folder and take note of all the ones that ends with a `_nm.dds`.
 5. **The `_nm.dds` files are normal map files. OpenMW's standard format is to have the normal maps with a `_n.dds` instead. Re**
- As a nice bonus to this tutorial, this pack actually included one specularity texture too. We should use it of course. It's the one called `tx_glass_amber_02_reflection.dds`. For OpenMW to recognize this file and use it as a specular map, you need to change the `_reflection.dds` part to `_spec.dds`, resulting in the name `tx_glass_amber_01_spec.dds`.
6. That should be it. Really simple, but I do know that it takes a few minutes to rename all those files.

Now – if the mod you want to change includes custom made models it gets a bit more complicated I'm afraid. But that is for the next tutorial.

Converting Apel's Various Things - Sacks

Mod made for MCP's fake bump function, with custom models

Authors Joakim (Lysol) Berg

Updated 2016-11-09

In part one of this tutorial, we converted a mod that only included modified Morrowind model (`.nif`) files so that the normal maps could be loaded in Morrowind with MCP. We ignored those model files since they are not needed with OpenMW. In this tutorial however, we will convert a mod that includes new, custom made models. In other words, we cannot just ignore those files this time.

Before we begin, you need to know that unless you want to build the NifSkope application from source yourself, you will be needing a Windows OS to do this part, since the application only has binaries available for Windows.

Tutorial - MCP, Part 2

The sacks included in Apel's [Various Things - Sacks](#) come in two versions – Without bump mapping, and with bump mapping. Since we want the glory of normal mapping in our OpenMW setup, we will go with the bump-mapped version.

1. Start by downloading Apel's [Various Things - Sacks](#) from Nexus.
2. Once downloaded, install it the way you'd normally install your mods (**Pro tip:** Install using OpenMW's [Multiple data folders](#) function!).
3. Now, if you ran the mod right away, your sacks will be made out of [lead](#). This is because the normal map is loaded as an environment map which MCP fixes so that it looks less shiny. We don't use MCP, so therefore, it looks kind of like the shack was made out of lead.
4. We need to fix this by removing some tags in the model files. You need to download [NifSkope](#) for this, which, again, only have binaries available for Windows.
5. **Go the place where you installed the mod and go to `./Meshes/o/` to find the model files.**

- If you installed the mod like I suggested, finding the files will be easy as a pie, but if you installed it by dropping everything into your main Morrowind Data Files folder, then you'll have to scroll a lot to find them. Check the mod's zip file for the file names of the models if this is the case. The same thing applies to when fixing the textures.

6. **Open up each of the models in NifSkope and look for these certain blocks:**

- NiTextureEffect
- NiSourceTexture with the value that appears to be a normal map file, in this mod, they have the suffix `_nm.dds`.

7. Remove all these tags by selecting them one at a time and press right click>Block>Remove.

8. Repeat this on all the affected models.

9. If you launch OpenMW now, you'll **no longer have shiny models**. But one thing is missing. Can you see it? It's actually hard to spot on still pictures, but we have no normal maps here.

10. Now, go back to the root of where you installed the mod. Now go to `./Textures/` and you'll find the texture files in question.

11. OpenMW detects normal maps if they have the same name as the base diffuse texture, but with a `_n.dds` suffix. In this mod, the normal maps has a suffix of `_nm.dds`. Change all the files that ends with `_nm.dds` to instead end with `_n.dds`.

12. Finally, **we are done!**

Since these models have one or two textures applied to them, the fix was not that time-consuming. It gets worse when you have to fix a model that uses loads of textures. The principle is the same, it just requires more manual work which is annoying and takes time.

Paths

The following describes the locations for the various OpenMW file paths for different types of files on different operating systems.

Note: Actual location depends on your computer's setup. Username, harddrive, and language may vary.

Configuration files and log files

Linux `$HOME/.config/openmw`

Mac `$HOME/Library/Preferences/openmw`

Windows `C:\Users\Username\Documents\my games\openmw`

Savegames

Linux `$HOME/.local/share/openmw/saves`

Mac `$HOME/Library/Application Support/openmw/saves`

Windows `C:\Users\Username\Documents\my games\openmw\saves`

Screenshots

Linux `$HOME/.local/share/openmw`

Mac `$HOME/Library/Application Support/openmw`

Windows `C:\Users\Username\Documents\my games\openmw`