

Creating a Full Conversion

[Original guide by Your Computer.](#)

When doing a custom story, you are forced to use all default settings for the player and you can only change the levels and add new assets. If you want to change stuff like player speed, sanity properties, menu style and the like, you need to do a full conversion.

This guide will assume you already know how to set up a custom story.

What is a full conversion mod?

A full/total conversion (often called FC, IFC or just a mod) is a complete overhaul of how the game is launched. Using one will allow you to change many aspects of the game by overriding the main config file used to define the game. A mod will allow you to edit anything that is available in the main game's assets, including all config files and *some* hard-coded resources, however you are still restricted to certain things within the game, like the engine itself. Basically, you can edit anything that is already available to you, but it won't add things outside the scope of the configuration provided by Frictional Games.

In most cases, hard-coded configuration cannot be changed, but some assets can be overridden using files with the same names as the original ones. This is an advanced topic which I won't get into here though.

Mods have many advantages over custom stories, but there's one larger drawback; the installation may be more advanced for the average user. Therefore it's suggested that you stick to a custom story unless you absolutely need to edit the game's properties for your creation.

This guide will show you how to create a fully Isolated Full Conversion (IFC).

Starting off

You can download a template at the bottom of this page if you wish, however these instructions assume otherwise.

So you're about to start setting up a FC mod. You should begin by making a folder for it. Place this folder in your Amnesia install directory ("redist" for retail and "Amnesia The Dark Descent" for Steam). For now, let's call it **my_mod**.

Now, depending on how much custom stuff you want in your mod, you need to create a few more folders within your mod folder. The most important folder is the **config** folder, as it holds all your customized settings. Instead of creating a new folder named config, you can just go back to your Amnesia root folder, **copy the config folder** there, and paste it within your mod. Here's a list of folders you need / might want in your mod:

- config
- maps
- main_menu

- graphics (if you use custom graphics)
- entities (if you use custom entities)
- static_objects (if you use custom static objects)
- textures (if you use custom textures)
- music (if you use custom music tracks)
- sounds (if you use custom sounds)
- models (if you use a custom lantern)

There are more folders you can create if you wish. These mostly reflect on how the Amnesia root folder looks like. Extra folders include particles, misc, lights, fonts, gui, lang and more.

Configuring the mod

Now that you have your setup, open the config folder. Because you copied it from Amnesia, it already contains many files you need. There are a few you can dispose of though, so delete them so the folder only contains these files:

- base_english.lang
- game.cfg
- main_init.cfg
- menu.cfg
- pre_menu.cfg
- lang_main (folder)
 - english.lang (within the lang_main folder)

Another two files you can copy over to your config folder is the **resources.cfg** file and (for advanced users) **materials.cfg** file. They're found lying around in the Amnesia root folder. Originally they're supposed to be located at the top of your mod folder, but I've found it to be cleaner to place them in /config. They still seem to function the exact same.

main_init.cfg

[List of the main_init.cfg file.](#)

Open your main_init.cfg file from /my_mod/config. This file controls the main startup attributes of the game. Under **ConfigFiles**, edit all the paths to point to the custom files you have. For example:

```
Resources = "my_mod/config/resources.cfg"  
Materials = "my_mod/config/materials.cfg"
```

Do this with all the files you do have in your config folder. The rest you can leave default.

Under **Directories**, rename **MainSaveFolder** to whatever you want your mod's saves folder to be named, for example "MyMod." This is important, as it separates your mod's generated files from other mods and the main game. Also edit **BaseLanguageFolder** and **GameLanguageFolder** to match your own config folder.

Under **Variables**, change **GameName** to your mod's name.

Under **StartMap**, change **File** to the name of your mod's starting level, change **Folder** to your mod's maps folder ("my_mod/maps"), change **Pos** to the PlayerStartArea you should first spawn in. This section resembles the custom_story_settings.cfg file.

menu.cfg

In this file, you can specify your own mod's menu. Go down to **BGScene** and change the path to match your menu map. You might not have your menu level yet, but you'll probably make it eventually. Just make sure to have one before starting your mod. [Here's a tutorial](#) that talks more about the menu.cfg file (PS: Edit your mod's menu.cfg instead of the main game's).

One more important thing about the menu file is that the default one (which you copied) does not include an entry for a custom logo. You can add it, however, which will allow you to change it. I would say it's a must for a mod to have its own logo, just like it should have its own name.

Add this line within the **Main** section (below Music if you'd like):

```
MenuLogo = "my_mod/lang/eng/graphics/main_menu/menu_logo_mod.tga"
```

This would be the path structured in the original fashion. It can be located in another folder, but personally I prefer using the same setup. If you do not have a lang folder, add one or change the path to somewhere else. Remember to create your own Targa (.tga) file to display.

resources.cfg

This file is a little different. Its purpose is to tell the game in which folders and locations it should search for the files it uses. For example if you use custom entities, you'll need to add your entities folder to the resources.cfg file, or else the game can't find them. If two files have the same name, and both locations are within the scope of the resources.cfg file, the game might mix them up and use the wrong one. It searches from top to bottom through the list, so add your own mod at the top to make sure the game finds your files first.

Do not remove what's already there unless you know what you're doing. Instead, copy one line and paste it at the top (after the **<Resources>** tag of course), then edit it to match your own mod. EG:

```
<Directory Path="/my_mod" AddSubDirs="true" />
```

game.cfg

[List of the game.cfg file and its contents.](#)

This file is very important if you want to edit how the game plays. I'll leave it up to you to experiment though. There are many settings regarding player speed/jump/stanity/health/oil drain etc. Play around with different settings. All the options might seem a bit intimidating to begin with, but you'll likely enjoy editing them.

base_english.lang and english.lang

base_english.lang is a language file just like english.lang is. It mostly deals with basic text that should be present in any mod. You don't really need to edit it unless you want customized messages for for example actions/hints/lantern and other gameplay-like things.

english.lang on the other hand is your FC mod's version of the extra_english.lang file from custom stories. Add any text here as you would for the custom story, like notes, messages, room names, item names etc.

Launching the mod

To start your mod, you need to make sure you have correctly set up the following:

- A config folder with at least a main_init.cfg file that points to appropriate locations and files.
- A main menu map that can be displayed, as configured in menu.cfg.
- A maps folder with at least 1 map to start as you select "Start new game" from the menu.
- Configured your mod's resources.cfg file to add itself to the game's scope.

Once you've set this all up, it should be possible to start your mod. But how?

The most common way of starting a mod is using the command line. Below are the contents for making executable command line files for the different operating systems. These files need to be placed within your mod's root folder (Amnesia/my_mod). I will also list an alternative method of launching mods using a custom application.

Windows

This is a batch file and thus should be saved as launcher.**bat**

```
cd /d ..  
Launcher.exe my_mod/config/main_init.cfg
```

Mac OS X

This is a Terminal command file. It should be saved as launcher **.command**

```
#!/bin/bash  
cd "`dirname "$@"`"  
open ../Launcher.app --args my_mod/config/main_init.cfg
```

Note: Mac .command files may not run until they are allowed to by the system, by running `chmod a+x <path to .command file>` on a Mac. I recommend including the pre-configured file in the template, rather than creating one on Windows.

Linux

This is a Terminal shell file. Name it `launcher.sh`

```
#!/bin/bash

FCNAME="my_mod"

# 64-bit operating systems
if [ "$(uname -m)" = "x86_64" ]; then
    ../Launcher.bin64 "$FCNAME/config/main_init.cfg"

# 32-bit operating systems
else
    ../Launcher.bin "$FCNAME/config/main_init.cfg"
fi
```

I recommend you include all 3 of these files. You never know what OS the user is going to be using.

Steam

If you use the Steam copy of Amnesia, you can also start mods via the Steam client by doing the following:

- Open Steam, right click Amnesia and select Properties.
- Click "Set Launch Options..." on the General tab.
- Type or paste in the argument to your mod's `main_init.cfg` file.

The argument should look like this:

```
my_mod/config/main_init.cfg
```

There is a way of doing this from a command line file as well. The process is a bit different though. Here's an example for doing so for Windows (although the command is nearly identical on other systems):

```
start steam://rungameid/57300//my_mod%%2Fconfig%%2Fmain_init.cfg
```

If you save this command as a `.bat` file on Windows, then run it, it will prompt Steam (if installed) to run Amnesia with the argument "**my_mod/config/main_init.cfg**". Notice how the argument itself does not include slashes, but instead the code `% 2F` which is the URL-way of defining a slash. A Windows `.bat` functionality requires us to use an extra `'%'` sign before the first in order to escape it (not required on other systems).

When running this, Steam will prompt a warning which must be accepted before it will launch the mod.

Launcher application

Alternatively to the methods above, you can launch mods (on Windows and Mac) using the Amnesia Modloader. If you want to use this method, I still recommend making the other launchers available for end users on Linux, as well as those who do not wish to use this user-made application.

[Read more about the Amnesia Modloader here.](#)

Other notes

Bear in mind that it takes experimenting to fully understand it all. You'll learn from mistakes and improve as you work with it. Once you're ready to publish your mod, [ModDB](#) is a nice place to do so. Also feel free to post it at the [Frictional Games forum](#). The forum is also a nice place to get feedback on your work.

I've prepared a download you can use to get started with a full conversion if you found this confusing. You may use it as a template for your mod. It's already been prepared, but you must still configure it.

[You can download it here.](#)

Happy modding!

~ Mudbill

From:
<https://wiki.frictionalgames.com/> - **Frictional Game Wiki**

Permanent link:
https://wiki.frictionalgames.com/hpl2/amnesia/full_conv

Last update: **2016/11/14 11:52**

