

---

# OMOD-Framework

Nov 25, 2019



---

## Getting started

---

<b>1</b>	<b>Getting started</b>	<b>3</b>
<b>2</b>	<b>Extracting files from OMODs</b>	<b>5</b>
<b>3</b>	<b>Executing the installation script</b>	<b>7</b>



Welcome to the official documentation of the OMOD-Framework.



# CHAPTER 1

---

## Getting started

---

Huge thank you for checking out my project :)

### 1.1 Installation

This Framework is a C# class library (.dll) that you can include in your project. How you want to include this is up to you. You can create a *git submodule* and add the project to your solution or just get the dll and add that as a reference to your project.

### 1.2 Setup

```
using OMODFramework;
```

---

**Note:** Do note that you will find another namespace from OBMM called *OblivionModManager* that is needed for compiling the installation script in *.omod* files.

---

```
Framework f = new Framework();
```

This is the basic setup. You can change some internal variables like the temp path used during extraction and script execution with

**Warning:** When working with paths, always use absolute paths `Path.Combine` when combining paths in this Framework

```
f.SetTempPath("absolute-path-please");
```

or change the emulated OBMM version

```
f.SetOBMMVersion(1, 1, 12);
```

---

**Note:** 1.1.12 is the latest official OBMM version

---

or if you want to use the execute script functions than you might want to set the dll path so that you don't get a mission file exception.

```
f.SetDLLPath("erri120.OMODFramework.dll");
```

What to do next depends on your use case.



## CHAPTER 2

---

### Extracting files from OMODs

---

```
Framework f = new Framework();  
OMOD omod = new OMOD("path.omod", ref f);  
string dataPath = omod.ExtractDataFiles();  
string pluginsPath = omod.ExtractPlugins();
```

---

**Note:** ExtractPlugins returns null if there are no plugins

---

These two little functions will extract all data files and plugins and returns the absolute path to their folder.

You can then copy those folders to somewhere or take a look at them.



## CHAPTER 3

---

### Executing the installation script

---

This is more complex than just extracting some files. You will have to implement all UI elements yourself.

```
Framework f = new Framework();
OMOD omod = new OMOD("path.omod", ref f);

string dataPath = omod.ExtractDataFiles();
string pluginsPath = omod.ExtractPlugins();
```

Start by extracting all files from the omod. You will need the location later on.

You will have to implement the *IScriptRunnerFunctions* yourself. You can create a class like

```
class ScriptFunctions : IScriptRunnerFunctions
{
    // implement all functions here
}
//...

ScriptFunctions sFunc = new ScriptFunctions();
```

The *ScriptRunner* class, responsible for executing the script, requires those functions:

```
ScriptRunner sr = new ScriptRunner(ref omod, a);
// to executing the script:
ScriptReturnData srd = sr.ExecuteScript();
```