```
Version 0.52.1 (2019-06-20)
- Changed: Some minor improvements to the Project Generator files.
- Changed: The Installer will clear the cache upon update and re-install
when user chooses to not remove old install.
- Bugfix: Added missing user-agent to the downloader when requesting
filename from the Content-Disposition header.
- Bugfix: Text display issue with mod updates showing as 0ms.
- Bugfix: Fixed custom keybinds causing crashes in some cases.
- Bugfix: Text display issue with mods reporting incorrect version after a
mod update.
- Bugfix: Renaming assemblies through mod updates would cause a hang.
(Thanks CabbageCrow)


Version 0.52 (2019-06-12)
- New: Brand new website with mod hosting and listing.
- New: Desktop Installer with drop-down list of games to install UMF to.
- New: UMF URI Handler: An automated one click mod downloader and installer.
- New: Uninstall.exe for easier/more obvious uninstalls.
- New: UMF now uses Harmony 2.0. (All mods that use Harmony needs to be
recompiled.)
- New: Tools/UMF Mod Packer packs down compiled mod files into an encrypted
re-distributable .umfmod package.
- New: Tools/UMF Project Generator creates ready to use Visual Studio
project files for UMF mods.
- New: The UMF Menu now automatically scales down on resolutions with a
height lower than 800 pixels. (Temporary solution)
- New: UMF Warnings will now detect and disable mods that are compiled for
the wrong .NET Framework.
- New: The Setup can now generate Visual Studio project files with UMF code
all ready to use.
- New: If a .umfmod is packed into a zip file the zip file will be
automatically extracted into the Mods folder.
- New: Duplicate older versions of mods in the Mods folder are now
automatically deleted by UMF.
- New: Added mod dependency settings 'ModDependencies' and
'ModDependencyVersions' to UMF Mod config settings. Use UMFConfigStringArray
to set them.
- New: Direct support for games using .NET 4.7.
- Changed: UMF will now ask if you want to update when a new update is
found.
- Changed: The Setup menu no longer requires pressing enter to confirm the
choice.
- Changed: The PauseGameOnGUIs setting has been improved and enabled by
default again as it should now work with most games.
- Changed: The 'UMF is loading' text has been replaced to stop people from
waiting for a full day.
- Changed: Set the setting ShowTips to off by default. The current GUI is
due for a major overhaul.
- Removed: Automatic rewriting of missing folder permissions have been
removed from the Setup in an attempt to reduce false Antivirus detections.
```

```
- Bugfix: Fixed the UMF key bindings KeysMenu and KeysConsole not taking
effect without a restart. (Thanks CabbageCrow)
- Bugfix: Fixed an issue with keyboard locale in some games.
- Bugfix: Fixed missing MinVersion and MaxVersion checks after the mod
loader rewrite.
- Bugfix: Fixed FieldOfVew setting not returning to default when being set
to 0. (Thanks CabbageCrow)
- Bugfix: Fixed UMF hanging when loading a packed mod with an unknown
password. (Thanks CabbageCrow)
- Bugfix: Fixed UMF hanging with no internet connection.
- Bugfix: Fixed an issue with the bindings cache by moving bindings to the
new cache system. (Thanks Jon E Boyd)
- API: Added void UMFConfig.DeleteConfig(bool backupOld = false)
- API: UMFConfig.Read and Write now supports Section in key string.
"SectionName/KeyName"
- API: Added void UMFGUI.RegisterPauseHandler(Action<bool> action)
- API: Added string[] UMFCache.ReadStartsWith(string keyStartsWith)
- API: Changed UMFDownload.DownloadFile to void DownloadFile(string url,
string directory, string file = null, bool extract = false, Action<string,
bool, string> action = null)
- API: Added int UMFMod.GetModHarmonyPatchCount(Assembly mod = null)
- API: Added int UMFMod.GetModHarmonyPatchCount(string modName)


Version 0.50.1 (2018-12-24)
- Bugfix: Fixed some issues with locale and mod localization that caused UMF
to freeze in some cases.


Version 0.50 (2018-12-23)
- New: The UMF Setup now has a -packmod command line parameter to pack mods
into .umfmod.
- New: UMF can now read mods directly from packed .umfmod files and zip
files.
- New: .umfmod and .zip mods supports their own packed Assets/<mod name>/
folders.
- New: UMF can now load mods from .cs source files when mono is installed.
- New: Scripted .umfpatch system for overwriting/patching inline methods in
all UnityEngine assemblies.
- New: The load order can now be sorted manually from the UMF Settings.
- New: Added console command hotloadmod "<path>" for loading mods that were
not already loaded, such as new source mods.
- New: Enabling a mod that was disabled when starting the game will now
attempt to immediately load that mod.
- New: If UMF encounters a critical error it will zip the log files and show
the main UMF log.
- New: UMF now targets multiple .NET frameworks (Staring with .NET 3.5 and
.NET 4.6.2), and the old and new style Unity Engines.
- New: UMF now automatically loads all AssetBundles found in the Assets
folders.
- New: UMF can now start up the real UPM side-by-side with UMF, if UMF's
```

```
dummy PluginManager.Core.dll is overwritten by UPM's real one through manual
install.
- New: Added more checks to UMF and the Setup to prevent user errors from
not following the instructions.
- Changed: The UMF Mod Loader was rewritten to improve capabilities.
- Changed: Disabled mods are now stored in disabled.txt and the Enabled
setting has been removed.
- Changed: Major internal changes all over to take advantage of the new UMF
Mod Loader.
- Changed: The UMF Warning window has been greatly improved and will catch
and show detailed warnings for each mod.
- Changed: Mod load order and priority is now sorted the following way:
loadorder.txt > LoadPriorty(High, AboveNormal, Normal, BelowNormal, Low)
- Changed: Improved the UMF DropDown box used in the UMF Menu.
- Changed: Improved the text field entry of numbers in the UMF Menu.
- Changed: The UMF Menu now includes disabled mods settings, so that
disabled mods can be enabled again from the UMF Menu.
- Changed: The UMF Menu mod buttons are now colored green and red based on
if they are enabled or not.
- Changed: If a mod has no settings that can be configured by the player it
is now properly stated in the UMF Menu.
- Changed: The console now supports numpad enter to send console commands.
- Changed: The help console command now supports partial search.
- Bugfix: UMF Font sizes would reset to really small in Unity Engine 2018.x
versions.
- Bugfix: Fixed the missing status text when enabling and disabling mods.
- Bugfix: Fixed an issue with encryption. All previous encryption strings
are no longer valid.
- Bugfix: Fixed issues with empty and None key binds.
- Bugfix: Fixed an issue with automatic UMF updates.
- API: Added new UMFDropDown(Rect rect, GUIContent buttonContent,
GUIContent[] listContent, GUIStyle listStyle = null)
- API: Added new UMFDropDown(Rect rect, GUIContent buttonContent,
GUIContent[] listContent, string buttonStyle, string boxStyle, GUIStyle
listStyle = null)
- API: Added string UMFData.AssetsPath
- API: Added string UMFData.AssetsSharedPath
- API: Added Texture2D UMFAsset.LoadTexture2D(string file, bool shared =
false)
- API: Added Mesh UMFAsset.LoadMesh(string file, bool shared = false)
- API: Added AudioClip UMFAsset.LoadAudioClip(string file, bool shared =
false)
- API: Added void UMFCache.Write(string key, string data)
- API: Added void UMFCache.Expire(string key)
- API: Added string UMFCache.Read(string key)
- API: Added bool UMFCache.IsOlderThan(string key, int seconds)
- API: Added bool UMFCache.Exists(string key)
- API: Added attribute [UMFUnload] used on a method, which is triggered
before a mod is unloaded/disabled by the user.
- API: The old UMFConfig API has now been removed.
- API: Added bool UMFPatch.ApplyPatch(string patchName, string patch)
```

```
- API: Added List<string> UMFData.ModNamesEnabled, UMFData.ModNames now
includes disabled mods as well.
- API: Added string UMFData.TempPath
- API: Added string UMFData.SourceModsPath
- API: Added string UMFData.LuaModsPath (Lua mods are not yet implemented)
- API: Added string UMFData.UMFPatchesPath
- API: Added string UMFData.ResourcesPath
- API: Added string UMFData.ManagedPath
- API: Added string UMFData.LoadOrderFile
- API: Added string UMFData.DisabledModsFile
- API: Added UMFConfigParser UMFConfigColorHexRGBA(Color defaultValues =
default(Color), bool requiresRestart = false)
- API: Added List<string> UMFData.ModNamesActive for loaded and active mods


Version 0.45.1 (2018-10-11)
- Bugfix: UPM mods caused game crashes when being loaded.
- API: UMFData.ChangelogsPath was renamed to UMFData.ModInfosPath


Version 0.45 (2018-10-09)
- New: Configurable fully automated game save backup system.
- New: Full localization support in the framework with an API for mods to
provide localization.
- New: UMF Menu configs will now detect the key pressed instead of having to
manually type it in by name.
- New: The UMF Menu settings now have sliders and dropdown boxes for
settings that support it.
- New: The UMF Menu now has a Enable/Disable Mod button for mods.
- New: UMF controlled key binding system with API for mods to prevent
binding conflicts.
- New: UMF Warning window that pops up to warn user of problems with mods
they have installed.
- New: UMFHarmony mods that supply totalPatches now show up in the UMF
Warning window when the totalPatches are not met.
- New: Mods that use encryption on UpdateURL now show up in the UMF Warning
window when the decryption fails.
- New: All updates and downloads are now logged into a permanent log file
that does not get reset.
- New: UMF now has 3 branches, stable, beta and internal debug. A setting
was added for using the beta branch. Debug remains an internal developer
branch.
- New: Added console command bind <key(+key(+key))> <console command>
(<args>)
- New: Added console command unbind <key(+key(+key))/all/*>
- New: Added console command savebinds "<cfg name>" (<append true/false>)
- New: Added console command listcfgs
- New: Added console command printhelp
- New: Added console command fixedDeltaTime (<num>)
- New: Added console command res (<width> <height> <fullScreen[true/false]>
<refreshRate>)
```

```
       This sets the resolution if the paramters are used. Otherwise it
retrieves it and all resolutions supported by the monitor.
- New: Settings marked with (*) requires a game restart. A note explaining
so was added to the top of each mod config, along with an API for the config
parsers.
- New: The setup will now check folder permissions and attempt to rewrite
them if they are bad on Windows installations.
- New: Added console command backup "<name>"
- New: Added console command restore "<name>"
- Changed: Improved the UMF auto update system linked to the change in
assembly loading.
- Changed: The default UMF Console key binding is now supported by most
keyboard layouts.
- Changed: Console command help now takes an optional paramter for help for
a specific command.
- Changed: Made the setup refuse to start if it isn't inside the
uModFramework folder.
            This is due to people selectively extracting files they want and
not following install instructions.
- Changed: Improved the internal assembly resolver to handle harmony reload
events and reflection resolve events.
- Changed: The setup will now give an administrator warning if the Scan
option does not find any games.
- Changed: UMF Menu config text fields are now twice as wide.
- Changed: The UMF Menu settings header now contains the Product Name of the
mod instead of the ini section.
- Changed: A mod's description attribute is now displayed below the settings
header of the UMF Menu if it contains a description.
- Changed: UMF now uses the DotNetZip library to handle all zip operations.
- Changed: UMF will now load the UMF GUI and register default console
commands before mods are loaded.
- Changed: UMF now loads assemblies into memory instead of from disk to make
it easier to self update through file replacement.
- Changed: changelogs.txt and UMF Changelogs renamed to modinfo.txt and UMF
Mod Info to let mods supply the info they want to to show the user on
install/update.
- Changed: Moved all downloads and the UMFDownloader API into an external
.NET 4.5 downloader for TLS 1.2 support. (Includes GitHub downloads.)
- Changed: Renamed the UseUACForUpdater setting to UseUAC and changed the
description for it.
- Changed: Made the UMF Loader slightly more stable with internal
improvements.
- Bugfix: The new v0.40 setup was not working correctly with mono for Linux
and Mac.
- Bugfix: Some of the UMFConsole API was broken due to an inverted bool.
- Bugfix: UpdateURL responses were not being trimmed for empty spaces and
empty lines. (Thanks CabbageCrow)
- Bugfix: Invalid UpdateURL's were not being caught. (Thanks CabbageCrow)
- Bugfix: Some words in-game Changelog text would not be put on new lines
and go under the scrollbar. (Thanks CabbageCrow)
- Bugfix: If the UMF specific config options were missing in a mod with
```

```
configs, UMF would hang. (Thanks CabbageCrow)
- Bugfix: The news box was showing text outside the box when the news text
was too long instead of the overflow being cut off.
- Bugfix: The setup will now function without an internet connection.
- Bugfix: Unzipping/extracting files on newer Unity 2018 versions with .NET
4.x would not work if the files already existed.
- Bugfix: PauseGameOnGUIs causes problems in the game Graveyard Keeper and
UMF now automatically disables this setting in that game.
- API: Renamed UMFConsole to UMFGUI.
- API: Added bool UMFGUI.DefaultCommandsRegistered
- API: Added int UMFData.LoaderUnixTimeDone
- API: Added void UMFTool.CreateDesktopShortcut(string linkName, string
targetPath, string workingDir = null, string arguments = null, string
description = null)
- API: Added void UMFTool.DeleteDesktopShortcut(string linkName)
- API: Added void UMFTool.CreateShortcut(string location, string linkName,
string targetPath, string workingDir = null, string arguments = null, string
description = null)
- API: Added void UMFTool.DeleteShortcut(string location, string linkName)
- API: Added Assembly UMFMod.GetMod(string modName, bool caseInsensitive =
false)
- API: Added string UMFMod.GetModProductName(string modName, bool
caseInsensitive = false)
- API: Added string UMFMod.GetModDescription(string modName, bool
caseInsensitive = false)
- API: Added string UMFData.LocalePath
- API: Added string UMFData.LocaleModsPath
- API: Added string UMFLocale.Get(string key, params object[] formatArgs)
- API: Added string UMFLocale.Get(string modName, string key, params
object[] formatArgs)
- API: Added string UMFLocale.Add(string locale, string key, string text)
- API: Added string UMFLocale.Add(string modName, string locale, string key,
string text)
- API: Added string UMFLocale.Locale
- API: Added bool UMFGUI.RegisterBind(string bindName, string keys, Action
action)
- API: Added bool UMFGUI.UnregisterBind(string bindName)
- API: Added bool UMFGUI.ModifyBind(string bindName, string newKeys, Action
newAction = null)
- API: Added bool UMFGUI.BindExistsByKeys(string keys, bool allBinds =
false)
- API: Added bool UMFGUI.BindExistsByName(string bindName, bool allBinds =
false)
- API: Changed UMFGUI.RegisterCommand to return a bool and log errors if
command exists.
- API: Changed UMFGUI.RegisterCommand now has a optional paramter of
sendCommand which can send a full console command after registration.
- API: Moved UMFData.IsMenuOpen to UMFGUI.IsMenuOpen
- API: Added Texture2D UMFUnity.ColorToTexture2D(int width, int height,
Color color)
- API: UMFHarmony attribute has a new parameter totalPatches.
```

```
[UMFHarmony(int totalPatches = 0, bool debug = false)]
- API: Added T UMFConfig.Read<T>(string key, UMFConfigParser<T> parser,
params string[] comments)
- API: Added void UMFConfig.Write<T>(string key, UMFConfigParser<T> parser,
params string[] comments)
- API: Added abstract UMFConfigParser<T> Use this as a child class to create
new custom config parsers.
- API: Added UMFConfigParser UMFConfigInt(int defaultValue = default(int),
int minValue = default(int), int maxValue = default(int), int vanillaValue =
default(int), bool requiresRestart = false, params int[] allowedValues)
- API: Added UMFConfigParser UMFConfigFloat(float defaultValue =
default(float), float minValue = default(float), float maxValue =
default(float), int decimals = 1, float vanillaValue = default(float), bool
requiresRestart = false, params float[] allowedValues)
- API: Added UMFConfigParser UMFConfigDouble(double defaultValue =
default(double), double minValue = default(double), double maxValue =
default(double), int decimals = 2, double vanillaValue = default(double),
bool requiresRestart = false, params double[] allowedValues)
- API: Added UMFConfigParser UMFConfigBool(bool? defaultValue = null, bool?
vanillaValue = null, bool requiresRestart = false)
- API: Added UMFConfigParser UMFConfigString(string defaultValue = "",
string vanillaValue = "", bool requiresRestart = false, params string[]
allowedValues)
- API: Added UMFConfigParser UMFConfigStringArray(string[] defaultValues =
default(string[]), bool requiresRestart = false, params string[]
allowedValues)
- API: Added UMFConfigParser UMFConfigKeyCode(KeyCode defaultValue =
KeyCode.None, bool requiresRestart = false, params KeyCode[] allowedValues)
- API: Added UMFConfigParser UMFConfigKeyCodeArray(KeyCode[] defaultValues =
default(KeyCode[]), bool requiresRestart = false, params KeyCode[]
allowedValues)
- API: Added UMFConfigParser UMFConfigDirectory(string defaultValue = "",
bool requiresRestart = false, params string[] allowedValues)
- API: Added UMFConfigParser UMFConfigVersion(Version defaultValue =
default(Version), bool requiresRestart = false)
- API: The old UMFConfig.Read/Write/UMF have been obsoleted and will be
removed in v0.50.
- API: Added List<UMFBind> UMFGUI.GetBinds(string searchPattern = null)
- API: Added List<UMFConsoleCommand> UMFGUI.GetCommands(string searchPattern
= null)
- API: Removed UMFDownload.QueueFile and QueueString and IsDownloadingString
in favor of new TLS 1.2 capable download system.
- API: Added UMFDownload.DownloadFile(string url, string path, bool extract
= false, Action<string, bool, string> action = null)
- API: Changed UMFDownload.IsDownloading now checks if anything is
downloading from the DownloadFile list instead.
- API: Added int UMFData.ModsUpdated
- API: Added long UMFData.ModsUpdatedTime
- API: Added bool UMFTool.Unzip(string pathFrom, string pathTo, bool delete
= false)
- API: Added bool UMFTool.Unzip(string path, bool delete = false)
```

```
- API: Added bool UMFTool.Zip(string fileName, string rootDir, List<string>
files)
- API: Added string UMFData.BackupsPath
- API: Added string UMFData.GameSavePath
- API: Added void UMFTool.Backup(string name) <- This creates a game save
backup.
- API: Added void UMFTool.Restore(string name)
- API: Added void UMFGUI.ToggleConsole()


Version 0.40 (2018-08-21)
The #user-friendly Update
- New: Mods that support it can now automatically update them selves. #user-
friendly
- New: UMF will now download, extract and install UMF updates automatically
on game start. #user-friendly
- New: The setup now creates a shortcut on the users desktop to the setup
and mods folder. #user-friendly
- New: The setup will open the Mods folder after installing. #user-friendly
- New: The setup now has an option to scan for all Unity Engine games.
- New: Command line arguments for the setup to do silent installs.
- New: A tip showing how to open the UMF Menu is now displayed in the top-
right corner at game start. #user-friendly
- New: Added a Reset button to the UMF Menu. #user-friendly
- New: Added CPU, GPU and Memory to the log file.
- New: Added automatic execution of all console commands found in
\Mods\Configs\AutoExec.cfg at game start.
- New: Added console command exec "<cfg name>" which executes all console
commands in <cfg name>.cfg found in the Configs folder.
- New: Added console command cfg <modname> <key> (<value>) which retrieves
or sets the config option for the specified mod or UMF.
- New: Added console command cfgSave <modname> which saves the settings that
have been made by cfg to the config file.
- New: Added console command cfgReload <modname> which reloads the settings
from the config file.
- New: Added console command timescale (<num>) which gets or sets the
current timescale of the game.
- New: If a mod update zip contains "changelog.txt" the changelog will be
showed in a window at game start.
       All changelogs will be automatically renamed and moved to
Mods\Changelogs\<modName>_v<version>_changelog.txt where the user can review
them again later.
- New: Added Mod specific UMF config option ConfigVersion used for cleaning
outdated config files.
- Changed: The start up info text now have transparent boxes under them to
make them easier to read. #user-friendly
- Changed: Renamed console command resetconfig to cfgReset, but still
retains resetconfig as an alias.
- Changed: Made the UMF Menu more compact and slightly better looking.
#user-friendly
- Changed: The Setup will now open the web browser to the Troubleshooting
```

```
Guide on the Support Forum if an error happens. #user-friendly
- Changed: UMF will now wait until OnGUI can be triggered before loading
mods in order to show progress for slow/long downloads. #user-friendly
- Changed: The mod specific UMF configs is no longer exposed in the in-game
settings menu. #user-friendly
- Changed: The setup will now clean up UPM installs into the
UnityEngine.dll/UnityEngine.CoreModule.dll to prevent problems. #user-
friendly
- Changed: Dll files moved to the Lib folder to keep the main uModFramework
folder clean for executables and text files.
- Changed: The setup will now install/uninstall UMF into all Data folders
(ex. 32bit and 64bit) if more than one exists.
- Changed: Improved output information when installing and uninstalling UMF
with the setup.
- Changed: The config values will now be trimmed for leading and trailing
spaces to prevent user errors. #user-friendly
- Bugfix: Unable to input text in the Settings GUI/UMF Menu after deleting
the values entirely.
- Bugfix: Mouse cursor would not show after clicking while on the UMF Menu
in Slime Rancher.
- Bugfix: UMF was not working correctly without an internet connection.
(Thanks CabbageCrow)
- API: New mod specific UMF config: UpdateURL. Supports a optional UMF
Encrypted string as well.
       The response should be: version|url (Example:
1.0|https://somewebsite.com/MyMod_v1.0.zip)
- API: Added UMFDownload.DownloadString(string url)
- API: Added UMFDownload.QueueString(string url,
DownloadStringCompletedEventHandler callbackComplete,
DownloadProgressChangedEventHandler callbackProgress)
- API: Added UMFDownload.IsDownloadingString
- API: Added UMFDownload.QueueFile(string url, string path, bool extract =
false, AsyncCompletedEventHandler callbackComplete = null,
DownloadProgressChangedEventHandler callbackProgress = null)
- API: Added UMFDownload.IsDownloading
- API: Added UMFData.LoaderIsDone
- API: Added UMFData.UnixTime
- API: Added UMFEncryption.Encrypt(string message, bool output = false)
Output will output the string in ModName.txt in the umf folder. Store the
encrypted string in your code.
- API: Added UMFEncryption.Decrypt(string encryptedMessage) Only the mod
that encrypted the string can decrypt it.
- API: Added UMFEncryption.Encrypt(string message, string password, bool
output = false) Encryption with a password you supply.
- API: Added UMFEncryption.Decrypt(string encryptedMessage, string password)
Only the password used to encrypt it can decrypt it.
- API: Added UMFData.AssembliesPath (Points to the folder that contains the
internal libraries.)
- API: Added UMFData.GamePath
- API: Added UMFData.ModsPath
- API: Added UMFData.ConfigsPath
```

```
- API: Added UMFData.LibrariesPath
- API: Added UMFData.LogsPath
- API: Added UMFData.CachePath
- API: Added UMFData.ChangelogsPath
- API: The [UMFHarmony] attribute now supports a bool to enable the harmony
debug log harmony.log at desktop. [UMFHarmony(true)] to enable debug log.
- API: Added UMFData.IsMenuOpen This can be used in OnGUI to show more GUI
elements while the UMF Menu is open.
- API: Added UMFConfig.DeleteConfig()


Version 0.35 (2018-07-28)
- New: Full Linux and Mac compatability. (Thanks to Billanater1 for Mac
testing.)
- New: In-game Settings GUI.
- New: 14 new settings for UMF. (HotKeys, feature disabling, frame rate,
FOV, and more)
- New: In-game full feature console. (With 11 console commands included.)
- New: In-game News straight from the website. (Can be disabled.)
- New: In-game Tips. (Can be disabled.)
- Changed: Mods folder is now inside the uModFramework folder instead to
gather all files in one folder. (Less confusion)
- Changed: The version checker can now be disabled in settings.
- Changed: Added a lot more logging to help with debugging peoples log
files.
- Changed: Improved performance through more code cleaning and improving.
- Changed: Version checker now only checks once an hour and will retrieve
the version from cache.
- API: UMFConfig and UMFLogger are now IDisposable. Use them in a using()
clause instead now.
- API: UMFConsole.SendCommand(string command, bool openConsole = false)
- API: UMFConsole.RegisterCommand(string name, string command, string[]
aliases, int arguments, string description, Action action)
- API: UMFConsole.AddConsoleText(string text, bool openConsole = false)
- API: UMFConsole.Args (Get to process arguments for your commands)
- API: UMFConsole.IsConsoleOpen
- API: UMFData.TimeToStart changed to UMFData.TimeToLoad
- API: UMFData.UMFPath returns the full path to the uModFramework folder.
- API: UMFData.AddTip(string tip) Adds tips to the UMF tips.
- API: UMFData.ModNamesLower returns the loaded mods in lower case for
easier processing.
- API: UMFMod.GetModVersion now also supports both an Assembly and a string
modName to get another mod's version.


Version 0.30 (2018-07-20)
- New: Support for running <a
href="https://github.com/UnityPluginManager/PluginManager">UnityPluginManage
r</a> plugins straight from the Mods folder, no install of UPM needed.
- New: Added version checking to the framework, displayed in the info text.
- New: Added automatic zip archive extraction. You can now drop zipped mods
```

```
directly into the Mods folder.
- Changed: Made the info text that shows for 30 seconds on game start
fancier.
- Changed: Improved logging a bit.
- Changed: Greatly improved mod loading, internal threading and thread
cleaning.
- Changed: Cleaned up the code a lot with various minor performance
improvements.
- Bugfix: Fixes for attribute started mods.
- API: Renamed [UModScript] attribute to [UMFScript] - also it actually
works now.
       Can be used on as many public classes as you want. (With
MonoBehaviour child class.)
- API: Added [UMFStart] attribute, used on a method to start a mod.
       This obsoletes the UMFMod class. Can be used only once per mod.
- API: Added [UMFConfig] attribute, used on a method to load the mods config
option on demand.
       The [UMFHarmony] attribute will call this method before patching.
       Future use includes reloading and changing configs from in game.
- API: Added [UMFHarmony] attribute, works on both class and method.
       Calls the method with the [UMFConfig] attribute if it exists, then
runs all harmony patches the mod has.
       When used on a method it also calls the method.
       This is to simplify mod creation (reduce code/easier). Can be used
only once per mod.
- API: The UMFMod class with Start method is now considered legacy and
obsolete and will be removed in a future release.
- API: Changed the Log api and added several overloads for it.
       The call is now Log(string text, bool clean = false, string file =
null)
- API: Added UMFMod.GetModVersion() which returns the Version info of the
mod that calls it.
       Handy for for self logging purposes, intended for the initial logging
call.
       Log("My Mod Name v" + UMFMod.GetModVersion().ToString(), true)
- API: Added summary comments for all API methods.


Version 0.25 (2018-06-29)
- Initial Public Release
```

From:
https://umodframework.com/wiki/ - **UMF Wiki**

Permanent link:
https://umodframework.com/wiki/changelog?rev=1562050048

Last update: **2019/07/02 06:47**