

Unity

Windows Power Functions

User/Developer Guide

Introduction

Thank you for your interest! You are amazing!

If you bought this package it's likely that you are not only interested in video games and you work on a wider range of applications. In my experience, especially when deploying permanent or semi-permanent software based installations, shutting down/rebooting the system when certain conditions are met is essential for the reliability of the product. With this asset you'll be able to trigger a shutdown/reboot sequence from C#.

Have fun with your work.

Checkout these packages too

[Exploding Asteroids](#)

[ODYSSEY - Space Atmospheres Material Pack](#)

[Maximum Contrast Skyboxes Collection](#)

Installation

Once the package is imported you don't need to do anything. The asset is ready to use.

When you import the package you'll find all the content of Windows Power Functions organised in a folder named "Windows Reboot And Shutdown Plugin".

Note that in older versions of Unity (5 and below) you would have to move the dll corresponding to your Unity architecture (x86 or x64) into the root folder of your project, at the same level of your Assets folder. That's not necessary in newer versions of Unity. All dlls are found in the dlls folder. Unity will use the one that corresponds to Unity's architecture (x86 or x64).

Use of power functions

In your scripts you can use the two functions like in the following lines, which can appear anywhere in your code:

Rebooting

```
WinRebootAndShutdown.DoWindowsReboot();
```

Shutting down

```
WinRebootAndShutdown.DoWindowsShutdown();
```

Logging Off the user (NEW)

```
WinRebootAndShutdown.DoWindowsLogoff();
```

Locking the work station(NEW)

```
WinRebootAndShutdown.DoWindowsLockWorkStation();
```

Be warned that these functions, once called, will start the reboot/shutdown procedure irreversibly. All opened applications will be closed and all unsaved data will be lost.

When you build your project to an application, the file `UnityWindowsPowerFunctions.dll` must be copied into the executable's folder (not the Data folder inside it). Make sure you copy the right file which, in the vast majority of cases should be the one contained in the `dlls/x64` subfolder. In case you built your application targeting x86 (32 bit application) then you should copy the one contained in `dlls/x86`.

Examples

A demo is included in the scene named `RebootAndShutdownExample`. In here, a UI exposes some logic included in `Example.cs`, where the power functions are called.

The `Example.cs` script is surely of help if you are a beginner.

Common Issues

PROBLEM: Internal Compiler Error

More precisely, when you build your project you get the following:

Internal compiler error. See the console log for more information. output was:

Unhandled Exception: System.Reflection.ReflectionTypeLoadException: The classes in the module cannot be loaded.

SOLUTION:

Make sure the dll that comes with the package is present and positioned in the right folder. If you lost the dll file, reimporting the asset should solve the issue.

In case you have similar troubles but in the built version, remember that you should include the `UnityWindowsPowerFunctions.dll` file with the resulting executable.

Support

For further information, support and any requests to make this package better please contact the developer at unity@riccardostecca.net

You're also welcome to explore more of my projects at www.riccardostecca.net