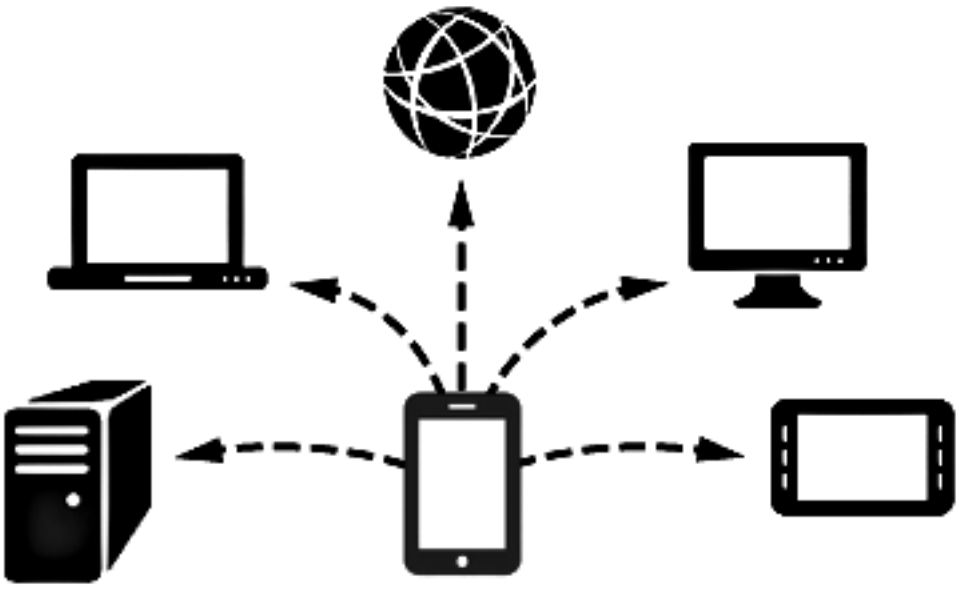
Good ol' Sockets



– for free version of Unity –
– supports iOS & Android –

by

LOST POLYGON

User Manual

v2.0.0

General information

Good ol' Sockets is a drop-in substitute for `System.Net.Sockets` namespace subset. It is designed to make it possible for Unity developers to use sockets on Android and iOS platforms without a Pro license. It also includes an automatic patcher tool that allows converting popular assets (such as **Photon Networking**, **Tasharen Networking**, **UniWeb**, **BestHTTP** and others) in a *single click!*

Good ol' Sockets aim to replicate the API of .NET sockets as close as possible. What this means for you is that you can use pretty much any code that uses `System.Net.Sockets`, including MSDN and hundreds of tutorials over the Web. Two simple commented demo scenes are also included in the package.

Note: Good ol' Sockets is a bit slower than native Unity implementation, and do not implement whole `System.Net.Sockets` and `System.Net` namespaces. However, this *won't* affect most applications, as sockets are rarely the bottleneck, and the chosen API subset is sufficient for almost any needs.

Plugin is tested in Unity 4.x - 5.x. Android and iOS platforms are supported, Pro license is not required.

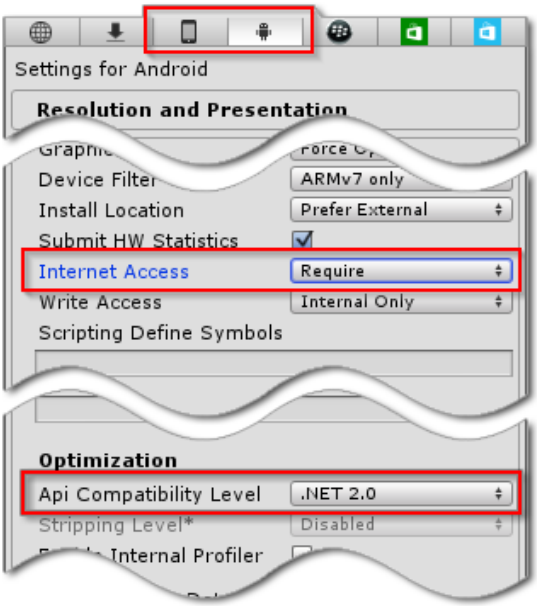
Integration

The process of integrating *Good ol' Sockets* is as simple as it can be.

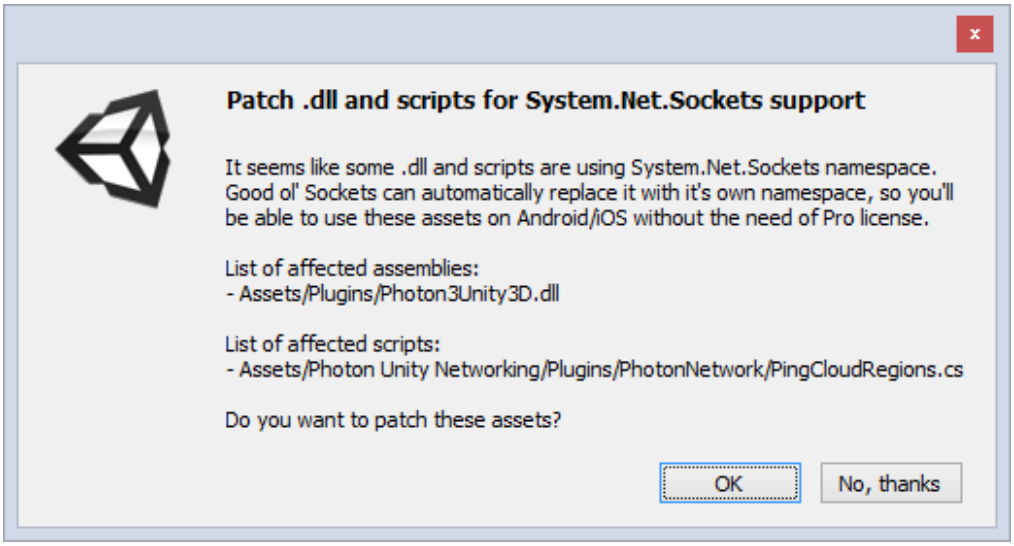
Good ol' Sockets requires some specific player settings to be set in order to function. If incompatible settings are detected, a settings validator dialog will appear, allowing you to fix the settings in a single click. You can also run the settings validator by using:

Tools → Lost Polygon → Good ol' Sockets → Check for Incompatible Settings

You can also set the correct settings manually. First, go to *Build Settings… → Player Settings → Other Settings* and set *"Internet Access"* to *"Require"*. Second, set *"Api Compatibility Level"* to *".NET 2.0"*.

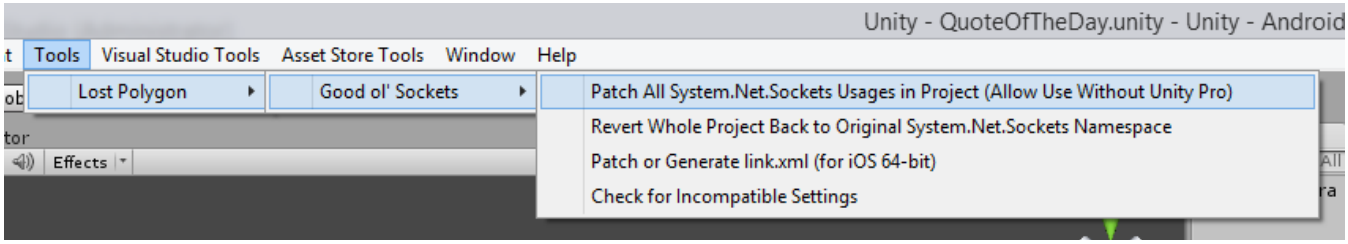


After that, when you import anything that uses `System.Net.Sockets` namespace, a dialog window will appear, allowing you to replace `System.Net.Sockets` (which is not supported without Pro license) with our own. Click OK, and *that's it!*



You can call the automatic patcher manually by using:

Tools → Lost Polygon → Good ol' Sockets → Patch All System.Net.Sockets Usages



Or you can also patch a single file or a whole directory by using "*Assets/Good ol' Sockets*" context menu.

In case of any problems with automatic patcher, or in case you are writing your own socket code, just prefix a namespace usage with "`LostPolygon`" like this:

C#

```
using System.Net;
using System.Net.Sockets;
```

→

```
using LostPolygon.System.Net;
using LostPolygon.System.Net.Sockets;
```

UnityScript/JavaScript

```
import System.Net;
import System.Net.Sockets;
```

→

```
import LostPolygon.System.Net;
import LostPolygon.System.Net.Sockets;
```

Boo

```
import System.Net
import System.Net.Sockets
```

→

```
import LostPolygon.System.Net
import LostPolygon.System.Net.Sockets
```

When building for 64-bit iOS using the IL2CPP runtime, a link.xml file will be created. It is required to avoid stripping of networking classes, as stripping is always enabled when using IL2CPP. More information on link.xml file can be found here.

Reverting the patch

It is also possible to revert the patching process. This is sometimes required when you want to build for a platform that is not supported by *Good ol' Sockets* (Web Player or Windows Phone, for example). In that case, revert the changes by using:

Tools → Lost Polygon → Good ol' Sockets
→ Revert Whole Project Back to Original System.Net.Sockets Namespace

and you'll be able to build your project again.

A bit of clarification: there are no problems with building for Web Player and Windows Phone, but you'll have to do patching/reverting when switching platforms, as *Good ol' Sockets* only directly supports iOS and Android (Editor and Standalone are also supported for debugging purposes). The process is like this:

1. When you want to build for Web Player (or any other platform that is not supported), you have to revert the patch, basically disabling G*ood ol' Sockets*, and the build will use the regular built-in sockets.
2. When you want to build for iOS/Android, you have to apply the patch, so the build will use the replacement sockets, allowing the build to succeed in the free version of Unity.

Known issues

Issue: *On Mac OS X, building the project from MonoDevelop fails with a compiler crash.*
Solution: This happens because MonoDevelop uses a different version of Mono compiler on Mac OS X by default. That compiler has issues with *Good ol' Sockets* library. The solution is to use Unity's built-in compiler instead. To do that:

1) Open MonoDevelop.
2) Go to *Preferences → Projects → .NET runtimes*.
3) Add a new runtime with path *"/Applications/Unity/Unity.app/Contents/Frameworks/Mono"*.
4) Set that compiler to be the default one.
5) Go to *Project → Active Runtime* and select Mono 2.6.5 Unity runtime.

Issue: *My networking code crasheson 64-bit iOS.*
Explanation: Unity uses IL2CPP runtime for building on 64-bit iOS. This runtime is new and has some issues, especially with sockets. If you encounter a bug or incorrect behavior, please build your project using the Mono runtime first. If it works fine with Mono runtime, then it's likely that the issue is on the Unity side, and Good ol' Sockets actually works fine. In that case, please direct all bug reports to Unity Technologies support.

Known incompatibilities

- uLink (reportedly works on Android, but not on iOS)

Implemented System.Net.Sockets counterparts

This is a list of .NET classes, structures and enumerations that are mirrored in *Good ol' Sockets*. You can use this list as a documentation reference – just Ctrl+Click to open your browser at the corresponding MSDN documentation page.

System.Net.Sockets.AddressFamily
System.Net.Sockets.IOControlCode
System.Net.Sockets.IPv6MulticastOption
System.Net.Sockets.IPPacketInformation
System.Net.Sockets.LingerOption
System.Net.Sockets.MulticastOption
System.Net.Sockets.NetworkStream
System.Net.Sockets.ProtocolFamily
System.Net.Sockets.ProtocolType
System.Net.Sockets.SelectMode
System.Net.Sockets.SendPacketsElement
System.Net.Sockets.Socket
System.Net.Sockets.SocketAsyncEventArgs
System.Net.Sockets.SocketAsyncOperation
System.Net.Sockets.SocketError
System.Net.Sockets.SocketException
System.Net.Sockets.SocketFlags
System.Net.Sockets.SocketInformation
System.Net.Sockets.SocketInformationOptions
System.Net.Sockets.SocketOptionLevel
System.Net.Sockets.SocketOptionName
System.Net.Sockets.SocketShutdown
System.Net.Sockets.SocketType
System.Net.Sockets.TcpClient
System.Net.Sockets.TcpListener
System.Net.Sockets.TransmitFileOptions
System.Net.Sockets.UdpClient
System.Net.EndPoint
System.Net.IPEndPoint

System.Net.IPHostEntry
System.Net.IPAddress
System.Net.CookieContainer
System.Net.Dns
System.Net.WebRequest
System.Net.WebResponse
System.Net.HttpWebResponse
System.Net.HttpWebRequest
System.Net.HttpRequestHeader
System.Net.HttpResponseHeader
System.Net.DecompressionMethods
System.Net.SocketAddress
System.Net.WebHeaderCollection
System.Net.NetworkInformation.OperationalStatus
System.Net.NetworkInformation.NetworkInterfaceType
System.Net.NetworkInformation.NetworkInterface
System.Net.NetworkInformation.NetworkInterfaceComponent
System.Net.NetworkInformation.PhysicalAddress
System.Net.NetworkInformation.IPv4InterfaceStatistics
System.Net.NetworkInformation.IPInterfaceProperties
System.Net.NetworkInformation.UnicastIPAddressInformationCollection
System.Net.NetworkInformation.MulticastIPAddressInformationCollection
System.Net.NetworkInformation.MulticastIPAddressInformation
System.Net.NetworkInformation.GatewayIPAddressInformationCollection
System.Net.NetworkInformation.GatewayIPAddressInformation
System.Net.NetworkInformation.IPAddressCollection
System.Net.NetworkInformation.IPAddressInformationCollection
System.Net.NetworkInformation.IPAddressInformation
System.Net.NetworkInformation.UnicastIPAddressInformation
System.Net.NetworkInformation.DuplicateAddressDetectionState
System.Net.NetworkInformation.PrefixOrigin
System.Net.NetworkInformation.SuffixOrigin

Contact

For any questions or concerns about this plugin, feel free to contact me at:

Unity forums thread: http://bit.ly/1x28to6
e-mail: contact@lostpolygon.com
Skype: serhij.yolkin

Changelog

2.0.0:

- Initial IL2CPP support.
- Major internal improvements, edge cases are handled better now.
- Improved patcher UI, performance and compatibility.
- Added validation of player settings to make sure they are always setup correctly.
- Fixed an issue that made some methods always throw a NullReferenceException.
- Added some more specific patching routines.
- Dropped support for Unity 4.2 and older.

1.4.4:

- Reverted some optimizations that introduced instable behavior.

1.4.3:

- Multithreading stability fixes.
- Fixed build for Windows Store Apps failing when Good ol' Sockets was in the project.
- Minor performance optimizations.

1.4.2:

- Fixed build for Windows Phone failing when Good ol' Sockets was in the project.

1.4.1:

- Multiple improvements in handling null reference parameters.

- Multithreading stability fixes.
- Added stub assemblies for Windows Store Apps platform.

1.4.0:
- Improved Unity 5 compatibility.
- Adding **`#define GOODOLDSOCKETS_IGNORE`** to the beginning of .cs file now prohibits patching that file. May be useful for libraries.
- Added specific patches for some popular assets to remove need for most manual post-patch fixes.
- Multiple performance and memory allocation optimizations.
- Fixed an error when assembly patcher found an unknown type in System.Net namespace.
- Assemblies from "Assets/Plugins/Metro/" directory won't be patched anymore.
- Resolved some UI issues.

1.3.1:
- Assemblies from "Assets/Plugins/WP8/" directory won't be patched anymore.
- Fixed incorrect type usage on Windows Phone build leading to a linker error.

1.3.0:
- Added per-directory patching.
- Much improved general .NET sockets compatibility.
- Improved Tasharen Network compatibility.

1.2.4:
- Fixed some regressions introduced in 1.2.3.
- Fixed compatibility with Tasharen Network 1.9.6b.

1.2.3:
- Fixed **`SocketAsyncEventArgs.Completed`** event being absent.
- Fixed some compatibility issues.

1.2:
- Added an option to revert the namespace patch.
- Added **`System.Net.CookieContainer`** support.
- Improved demos.

1.1.3:
- Fixed **`SocketAsyncEventArgs.Completed`** event.

1.1:
- Fixed some compatibility issues.

- Fixed build on Windows Phone platform.
- Improved assembly patcher.

1.0:
- Initial release.