Steamworks for Unity

.NET wrapper for Steamworks

What is this?

Steamworks for *Unity* is a .NET wrapper for the Steamworks API developed by Ludosity. The purpose is to make it easier to implement Steamworks features in games using Unity.

The library aims to be a complete a 1-to-1 mapping of the Steamworks C++ API. This means that the usage pattern is more or less the same as if C++ where used. This also means that the official help resources can be used as most things will work the same. The main differences are how callbacks are handled.

Where to start?

Start by reading the Setup instructions. If you are updating from an older version, check out the Change log.

Then take a look at the Steam class and the example scripts included in the package (Steamworks.cs and Stats.cs). To run the example, just attach these scripts to any GameObject and press Play.

Refer to Common errors if something goes wrong.

Limitations

Only parts of the Steamworks API are exposed at the moment.

- Cloud
- Stats
- Friends
- MatchMaking
- MatchmakingServers
- GameServer
- GameServerStats
- Networking
- Utils

More parts of the API will be exposed in the future. Feel free to contact us if you have any specific feature needs and we will try to make them a priority.

The library is not designed to be thread safe.

Almost all method comments are (for the moment) simply copy-pasted from the regular Steamworks C++ headers. So any names mentioned in these comments refer to the C++ names, but these names are almost the same as the names used in this library.

Software requirements

This library requires access to the Steamworks SDK to work (currently version 1.22).

To be able to use this library on a Windows computer the Microsoft Visual C++ 2010 Redistributable Package (x86) needs to be installed. This applies to both developers and end users. This means that the redist package needs to be included with the final game.

This library is built against Mac OS X 10.6 and have been tested on Mac OS X 10.7.4.

Known bugs

When the library is used from within the editor, the Steam client will think that you are playing the game until the editor is closed, even if play mode is stopped. This will however not effect the usage of this library as it is designed to handle this if used as in the example.

The Game overlay will not always work when running the game from the Unity editor. To make the overlay display correctly you may have to publish a built version of your game to a local content server and run the game via the steam client. Refer to the official Steamworks documentation on how to setup a local content server.

Support

If you have any bug reports, feature requests or questions regarding the wrapper itself, please send an email to **steamworks@ludosity.com**.

Please include the version of the library that you are using (Version Info) and the invoice number from the Unity Asset Store when contacting support. This will enable us to verify your purchase and can significantly speed up turnaround times as we will be able to send you development builds to verify that bug fixes work.

We can unfortunately only help with things that have to do directly with the wrapper, questions about the usage of the Steamworks API itself needs to be directed to the official Steamworks mailing list.

Setup instructions

All Platforms

Download and install the needed version of the Steamworks SDK (currently 1.23).

Create a file named steam_appid.txt and place it on the same level as the Assets folder in the Unity project. For example, if you create a Unity project called Bob, Unity will create a folder called Bob that contains the Assets, Library and Temp folders. Place the steam_appid.txt file directly in the Bob folder.

Enter you AppID (assigned to you by Valve) into the steam_appid.txt file and save it. Make sure that you save the file in regular ANSI encoding, or there may be problems reading the file.

When you build the game for testing, you will have to manually copy the steam_appid.txt to the same folder where the .exe file or the .bundle folder are placed.

Windows

To be able to use the library from the editor, you have to copy the steam_api.dll and SteamworksNative.dll files into the project folder at the same place where you placed the steam_appid.txt file. To be able to build the project you will also have to copy SteamworksManaged.dll into your assets folder.

When you build the game you will also have to manually copy the steam_api.dll file to the same folder where the built .exe file is placed. The folder stucture should be like this:

- Game_Data/
- Game.exe
- steam_api.dll
- steam_appid.txt
- SteamworksNative.dll

Mac

To be able to load the library from the editor and a built game, copy the libsteam_api.dylib file from the Steamworks SDK into the Assets/Plugins/SteamworksNative.bundle/Contents/MacOS folder.

Common errors

- **DllNotFoundException** is thrown even if the SteamworksNative.dll / SteamworksNative.bundle are in the right folder.
    - **On Windows:**
      Make sure that you have copied steam_api.dll to the Unity project folder. See the Windows setup instructions for more detailed information. The error can also occur if the Microsoft Visual C++ 2010 Redistributable Package (x86) is not installed.
    - **On Mac:**
      Make sure that you have copied libsteam_api.dylib into the bundle. See the Mac setup instructions for detailed information.

Change log

Version 1.4.123

- Finished the User interface, based on the ISteamUser class
- Added the Apps interface, based on the ISteamApps class
- Linked against Steamworks SDK 1.23
- Added Cloud::UGCDownloadToLocation (this is one of the functions 1.23 added, we can't add the other one yet as we haven't exposed ISteamScreenshot)
- Fixed Big Picture mode input support
- Added more example code
- Fixed intellisense documentation, last version was lacking in documentation since intellisense was broken and oxygen was removed
- Made sure that every function we've exposed had an alternative version where we don't use the 'out' keyword, but returns a struct

Version 1.3.122

- Linked against Steamworks SDK 1.22 (No support for Big Picture gamepad input yet).
- Added the Utilities interface, based on the ISteamUtils class.
- The game overlay is now fully functional.
- Removed the automatic documentation as it was broken.

Version 1.2.119

- Linked against Steamworks SDK 1.19.
- Renamed the dll/bundle from SteamAPIWrap to SteamworksNative
- Moved all constants to the Constants class
- Added all the remaining features from the ISteamRemoteStorage class to the Cloud interface
- Enabled all the remaining features from the ISteamFriends class to the Friends interface
- Renamed things to better match the C++ API, might cause some compile errors. Sorry, will avoid it in the future.
- Added some missing items to the LeaderboardEntry struct.
- Documentation updates.
- Added the Matchmaking interface, based on the ISteamMatchMaking class
- Added the GameServer interface, based on the ISteamGameServer class

Version 1.1.452

- The game overlay can now be used to open things like the achievements list, the stats page and arbitrary web pages.

Version 1.0.427

- Fixed internal marshaling error when unlocking achievements.
- Added better handling of errors in native code.

Version 1.0.403

- All the structures that wrap handles can now use .Equals() and == to test for equality.
- Added an example of how to use Stats.

Version 1.0.352

- Initial release