Heathen

Lime Lodge Skerries Athy, CO. Kildare, Ireland

Steamworks Complete

Full documentation is available online in the Heathen Knowledge Base.

You can find getting started instructions that will guide you through your first steps with the asset.

Our knowledge base. All code is fully documented in source meaning you can use Visual Studio IDE or any similar IDE to read information on any class, object, struct, method or variable you are interested in.

The following document summarizes the objects available, for full, up-to-date documentation please see our Knowledge Base.

Contents

API

https://kb.heathen.group/assets/steamworks/api

Heathen’s C# centric and Unity centric wrapper around the raw Steam Steamworks API

App.Client

https://kb.heathen.group/assets/steamworks/api/app.client

leverages the ISteamApp interface from Valve's Steam Client API and exposes every feature of that API in a C# and Unity centric way. Beyond a simple API wrapper, the App.Client class provides for basic API initialization for client API and runs the callback update. It is possible to run Steam API without Steamworks Behavior using only this class.

App.Server

https://kb.heathen.group/assets/steamworks/api/app.server

leverages the ISteamApp interface from Valve's Steam Client API and exposes every feature of that API in a C# and Unity centric way. Beyond a simple API wrapper, the App.Client class provides for basic API initialization for client API and runs the callback update. It is possible to run Steam API without Steamworks Behavior using only this class.

App.Web

https://kb.heathen.group/assets/steamworks/api/app.web

leverages the ISteamApp interface from Valve's Steam Web API and exposes the features of that API in a C# and Unity centric way.
Note we do not attempt to expose authorized endpoints. That is much of the Steam Web API requires a publisher key and should be called from a trusted web server not from a game client or a game server. We only expose those web API calls that do not require a publisher key.

Authentication

https://kb.heathen.group/assets/steamworks/api/authentication

The Authentication has both client and server interfaces that are identical. Heathen's Steam API wrap's these in a single call which will call the appropriate client or server interface for you based on the build type in Unity.

BigPicture.Client

https://kb.heathen.group/assets/steamworks/api/bigpicture.client

leverages features of ISteamUtil to simplify working in Steam's BigPicture mode. In particular tests for "IsBigPicutre", "RunningOnStemDeck" and handling Gamepad Text Input.

Clans.Client

https://kb.heathen.group/assets/steamworks/api/clans.client

leverages features of ISteamFriends, ISteamUser and ISteamUGC to simplify working with Steam "clans" aka "groups" aka "guilds" ... known by a lot of names but these are the "communities" you see in Steam which can have chats, news, etc.

Friends.Client

https://kb.heathen.group/assets/steamworks/api/friends.client

You can list the clan owner, its officers, open the clan chat in overlay or join the clan's chat in game. The most common use game developers look for and the most complex is to join the clan chat in game.

Input.Client

https://kb.heathen.group/assets/steamworks/api/input.client

Input API is a flexible action-based API that supports all major controller types - Xbox, PlayStation, Nintendo Switch Pro, and Steam Controllers.

Inventory.Client

https://kb.heathen.group/assets/steamworks/api/inventory.client

Steam Inventory Service is a set of features that allow a game to enable persistent player inventories without having to run special servers to manage users or items.

Leaderbaords.Client

https://kb.heathen.group/assets/steamworks/api/leaderboards.client

Leaderboards are ranked lists of players where a player's score determines their position on the leaderboard. Leaderboards can contain additional data for each entry either as a details array or as an attachment, attachments are useful for playbacks and other large bits of information while details are useful for character builds, player settings, etc.

Matchmaking.Client

https://kb.heathen.group/assets/steamworks/api/matchmaking.client

The matchmaking system is fundamentally a system for getting player's together to play games. The main feature of the system is the Steam Lobby. Steam Lobbies can be searched for based on the metadata of a lobby, they can be advertised via Party Beacons, on the friends list and direct invites can be sent to targeted players.
Steam Lobby can be used for more than a simple game lobby, depending on your games specific needs they can provide for teams, party/groups, session merging and more. Valve allows a user to be a member of 1 "normal" lobby and up to 2 additional "invisible" lobbies. Each lobby has its own set of metadata for the lobby itself and for each of its members and each lobby includes a simple chat system.

Overlay.Client

https://kb.heathen.group/assets/steamworks/api/overlay.client

The overlay interface provides simplified access to Steam's overlay features. The Steam Overlay can be used to show web pages, access friends list, clans and more.

Parties.Client

https://kb.heathen.group/assets/steamworks/api/parties.client

You can use this system as a matchmaking solution where in your user can scan for available beacons to join and can create and publish beacons that others can view and join.

You can attach metadata to a beacon like the metadata concept with Steam Lobby based matchmaking. You can use the party system alongside other forms of matchmaking manually updating the available "slots" a beacon is advertising.

RemotePlay.Client

https://kb.heathen.group/assets/steamworks/api/remoteplay.client

Functions that provide information about Steam Remote Play sessions, streaming your game content to another computer or to a Steam Link app or hardware.

RemoteStorage.Client

https://kb.heathen.group/assets/steamworks/api/remotestorage.client

Remote storage is primarily used to save game data and other user specific information to a cloud storage area. This ensures that data is available wherever the player chooses to play. The API makes it easy to read and write files to and from Steam Remote Storage Aka Steam Cloud save.

Screenshots.Client

https://kb.heathen.group/assets/steamworks/api/screenshots.client

Take screenshots ... that is the main function hence the name.
The main reason to use this is if you’re handling your own advanced screenshot features ... as is all the rage these days.
You can enable the HookScreenshots feature, and Steam will notify you when the user hits the screenshot button, they have configured in Steam client. You can then start the process of taking the screenshot however makes since for your game.

StatsAndAchievements.Client

https://kb.heathen.group/assets/steamworks/api/statsandachievements.client

Read and set the value of stats and achievements for client API.

StatsAndAchievements.Server

https://kb.heathen.group/assets/steamworks/api/statsandachievements.server

Read and set the value of stats and achievements from the server API.

User.Client

https://kb.heathen.group/assets/steamworks/api/user.client

The user interface can be used to modify the local user's rich presence data and read data from other users. In large the features and capabilities of the User interface are available through the related data object

UserGeneratedContent.Client

https://kb.heathen.group/assets/steamworks/api/usergeneratedcontent.client

Provides access to Steam User Generated Content aka Steam Workshop.

Utilities

https://kb.heathen.group/assets/steamworks/api/utilities

The utilities interface contains misc. features that don't fit in well to other interfaces. It can flip images, convert IP addresses from uint, to string and back again and test Steam client's various modes.

Utilities.Client

https://kb.heathen.group/assets/steamworks/api/utilities.client

The client side of the Utilities interface exposes the ISteamUtility interface with features that can read the user's IP Country, detect what mode Steam is running in and similar.

Voice.Client

https://kb.heathen.group/assets/steamworks/api/voice.client

The voice interface enables you to capture, compress and decompress voice data from the user’s active recording device if any is present.

Data Layer

A mid-level abstraction of Steam API and a common code base between Godot and Unity. These `Data` objects such as UserData, LobbyData, AchievementData and more have engine specific features such as returning avatars, icons, etc. in the engine's native "texture" format but will have identical functionality between the two engines.
Data objects are simple structs that are implicitly convertible with the underlying native data type for example one can convert a ulong value to a UserData object

Achievement Data

https://kb.heathen.group/assets/steamworks/data-layer/achievement-data

Represents a Steam Achievement object.

App Data

https://kb.heathen.group/assets/steamworks/data-layer/app-data

Represents a Steam App.

Clan Data

https://kb.heathen.group/assets/steamworks/data-layer/clan-data

Represents a Steam Clan aka Group.

DLC Data

https://kb.heathen.group/assets/steamworks/data-layer/dlc-data

Represents a Steam Downloadable Content entry.

Game Data

https://kb.heathen.group/assets/steamworks/data-layer/game-data

Represents Steam Game and is like Steam App.

Input Action Data

https://kb.heathen.group/assets/steamworks/data-layer/input-action-data

Represents an Input Action from the Steam Input interface.

Input Action Set Data

https://kb.heathen.group/assets/steamworks/data-layer/input-action-set-data

Represents an Input Action Set from the Steam Input interface.

Input Controller Data

https://kb.heathen.group/assets/steamworks/data-layer/input-controller-data

Represents a controller for use in the Steam Input interface.

Item Data

https://kb.heathen.group/assets/steamworks/data-layer/item-data

Provides easy access to detailed item information without needing a ScriptableObject. The ItemData structure can be implicitly converted from an int or item def meaning that if you know the ID of the item you want to work with you can get it without needing to look anything up.

Leaderboard Data

https://kb.heathen.group/assets/steamworks/data-layer/leaderboard-data

A simple struct that provides easy access to a given leaderboard's features and data. The structure is convertible to and from the native and primitive data types for a Steam Leaderboard.

Lobby Data

https://kb.heathen.group/assets/steamworks/data-layer/lobby-data

The LobbyData object is a custom CSteamId that carries tools and functions unique to the Steam Lobby system. This object is common between both Unity and Godot game engine integrations.

Lobby Member Data

https://kb.heathen.group/assets/steamworks/data-layer/lobby-member-data

LobbyMemberData is a structure that wraps around LobbyData and UserData to simplify access to the lobby member metadata and related features for a specific user on a specific lobby. In most cases you would be working with the LobbyMemberData of the local user and can easily create the LobbyMemberData by reading it from the related lobby.

Stat Data

https://kb.heathen.group/assets/steamworks/data-layer/stat-data

A data wrapper around Steam's concept of a stat. This is implicitly convertible from string expecting that string to be the "API Name" of the stat.

User Data

https://kb.heathen.group/assets/steamworks/data-layer/user-data

The UserData object provides quick and efficient access to the details and artifacts pertaining to any given user. Heathen's UserData object is interchangeable with CSteamID and ulong and can be created by simply casting or assigning from those values.

User Leave Data

https://kb.heathen.group/assets/steamworks/data-layer/user-leave-data

Used in the GameConnectedChatLEaveEvent and related event handlers.

Workshop Item Data

https://kb.heathen.group/assets/steamworks/data-layer/workshop-item-data

Used as a tool to create and update UGC Workshop Items.

Objects

The following articles explain in detail the features and capabilities of every object in the Heathen Steamworks integration. In most cases you won't need to read these articles as we have provided full comments in code thus intelliSense and auto-complete in your IDE of course should provide you with much of the same information you will find here.

Authentication Session

https://kb.heathen.group/assets/steamworks/objects/authentication-session

Used by the Authentication interface to represent an active authentication session.

Authentication Ticket

https://kb.heathen.group/assets/steamworks/objects/authentication-ticket

Used by the Authentication interface to represent an active authentication ticket.

Chat Room

https://kb.heathen.group/assets/steamworks/objects/chat-room

The clan chat room structure is used by chat specific features of the Clans interface and is itself a shortcut to the most common chat features.

Clan Chat Msg

https://kb.heathen.group/assets/steamworks/objects/clan-chat-msg

Represents a chat message from the Clans interface.

Currency

https://kb.heathen.group/assets/steamworks/objects/currency

A simple class used to list the supported currency codes available in Steam and to fetch the string symbols of those codes.

Data Model

https://kb.heathen.group/assets/steamworks/objects/data-model

It is intended that you use this to create your core save files as Scriptable Objects. This makes it trivial to query the Steam Remote Storage interface for records of this file type and to load data into a usable point in memory that can easily be referenced by Unity components.
You would typically start by defining your underlying data type e.g. a serializable object that represents the data you wish to save

Favourite Game

https://kb.heathen.group/assets/steamworks/objects/favorite-game

Used by the Matchmaking interface in relation to favorite game servers.

Input Action Data

https://kb.heathen.group/assets/steamworks/objects/input-action-data

Data used when checking the value of a Steam Input Action.

Input Action Update

https://kb.heathen.group/assets/steamworks/objects/input-action-update

Used by InputAction objects to report changes as an event argument. InputAction is a type of Game Event, to learn more about Game Events read our article here.

Inventory Result

https://kb.heathen.group/assets/steamworks/objects/inventory-result

Used by the Inventory interface to return general inventory information.

Item Detail

https://kb.heathen.group/assets/steamworks/objects/item-detail

Used by the Inventory interface and in the Item Definition to represent instances of an item.

Leaderboard Entry

https://kb.heathen.group/assets/steamworks/objects/leaderboard-entry

Represents an entry found on a leaderboard and contains data about that entry such as the user its related to, details and any attachments.

Lobby Chat Msg

https://kb.heathen.group/assets/steamworks/objects/lobby-chat-msg

Represents a chat message sent from the Steam Lobby system. This can be used to identify the sending lobby, user and data and can parse the data to a string or Json serializable object.

Lobby Game Server

https://kb.heathen.group/assets/steamworks/objects/lobby-game-server

Represents Game Server connection information as used by the Steam Lobby system.
Note a server will always have either or both IP/Port and ID, it will never have none. ID e.g. CSteamID is used by SteamNetworking and SteamNetworkingSockets for connections. IP/Port is more offten used with 3rd party transports such as KCP, TCP, UDP, etc. A server may have both as a server may handle both SteamNetworkign and other 3rd party connections.

Party Beacon Details

https://kb.heathen.group/assets/steamworks/objects/party-beacon-details

Used by the Parties interface and represents a given beacons details.

Rank Change

https://kb.heathen.group/assets/steamworks/objects/rank-change

Used by the Leaderboard system when a user's rank changes.

Remote Storage File

https://kb.heathen.group/assets/steamworks/objects/remote-storage-file

Used by the Remote Storage interface to represent a file located on the Steam Remote Storage system.

Steam Game Server Configuration

https://kb.heathen.group/assets/steamworks/objects/steam-game-server-configuration

Used to define a Game Server's configuration for initialization of the Steam API.

UGC Query

https://kb.heathen.group/assets/steamworks/objects/ugc-query

Designed to simplify the act of querying User Generated Content Aka Workshop items from Steam this object can be used to create and manage queries.
The intended use is that a query will be created through one of the static Create methods to properly initialize the base query and you can then use simple calls on the object to modify the query, execute the query and step through query pages.
This is used by the UGC Query Manager which can its self-simplify UGC query even further.

UGC Community Item

https://kb.heathen.group/assets/steamworks/objects/ugc-community-item

Returned by UGC Query and related tools and represents the data of a UGC Item aka Workshop Item.

User Leave Data

https://kb.heathen.group/assets/steamworks/objects/user-leave-data

Used by the Clans interface indicating a user leaving a target chat.

User Lobby Leave Data

https://kb.heathen.group/assets/steamworks/objects/user-lobby-leave-data

Used by the Lobby Manager's evtUserLeft event to indicate who left and why.

Workshop Item Data Create Status

https://kb.heathen.group/assets/steamworks/objects/workshop-item-data-create-status

Used with the User Generated Content interface for 1 line creation of workshop items as the parameter in the callback e.g.

Workshop Item Key Value Tag

https://kb.heathen.group/assets/steamworks/objects/workshop-item-key-value-tag

Used by the WorkshopItemData object.

Workshop Item Preview File

https://kb.heathen.group/assets/steamworks/objects/workshop-item-preview-file

Used by the WorkshopItemData object.