

# Unisave

---

Unisave tries to make the process of saving and loading data super easy.

Definitely check out the [online documentation](#).

## Features of Unisave Local

---

- Automatic field saving (no manual calls to some `Save()` method)
- Saving primitives (int, string, float, ...)
- Saving collections (List, Dictionary)
- Saving geometry (Vector2, Vector3Int, ...)
- Saving your custom classes

Unisave Cloud is currently under development and will be available soon.

## Usage

---

Here is a script that automatically loads its data on `Awake` and saves `OnDestroy` :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Unisave;

public class MyBehaviour : UnisaveLocalBehaviour
{
    [SaveAs("player.title")]
    public string playerTitle = "untitled";

    [SaveAs("player.stats.traveled")]
    private float distanceTraveled = 0.0f;

    void Start()
    {
        Debug.Log(playerTitle);

        distanceTraveled += 10.0f;
    }
}
```

See the `Unisave/Examples/Local/Leaderboard` for a full working leaderboard example.

## API

---

You might want to save or load your script right now and not wait for `OnDestroy` . In that case you skip the inheritance of `UnisaveLocalBehaviour` and specify the calls yourself:

**UnisaveLocal.Load(MonoBehaviour myScript)**

- iterates over all `[SaveAs("foo")]` fields and loads them
- you usually load from within your script, so you put `this` as the argument.

**UnisaveLocal.Save(MonoBehaviour myScript)**

- iterates over all `[SaveAs("foo")]` fields and saves them
- you usually load from within your script, so you put `this` as the argument.

Look at the implementation of `Unisave/Scripts/Local/UnisaveLocalBehaviour` , it's very straight forward.

Also check out the [online documentation](#).

# Lastly

---

It's wonderful that you find Unisave interesting.

If you have any questions, just [send an email](#).

Also [leave a review](#) if you think other people might be interested.

I wish your game success! Goodbye!