

## Save & Load System by VEOdev

### Setup:

All what you have to do is create a new Data object and assign your **profile ID**  
If the profile already exist it will be loaded automatically, you can disable it from the settings.

**Profile ID** is a string that represent the folder name, you may have many profiles.

**Data** is the object we will use to access our saved data or save new data

```
using DataManagement;

public class Example : MonoBehaviour
{
    Data data;

    void Awake()
    {
        data = new Data("MyProfileID");
    }
}
```

### How to set data:

Use the `SetData("YourDataTag", YourData);`

*SetData will update the data if it already exists, if not it will create it.*

```
void AssignExampleData()
{
    // Set example data.
    data.SetData("playerName", "Jake");
    data.SetData("PlayerHealth", 150);
    data.SetData("playerPosition", new Vector2(0,0));

    // SetData takes a data tag and the data, data tag can be named anything you like.
    // we need it later to access the data.
}
```

Doing it this way can be messy for large data like player info or player stats, A better way is to use class that contains the player data and set the whole class at once.

```
..// Make a class for the player data
..public class PlayerData
..{
..    ..public string name = "Jake";
..    ..public int health = 100;
..    ..public Vector2 position = new Vector2(0,0);
..}
..public PlayerData playerData;

..void AssignExampleData()
..{
..    ..// Now we just assign the whole class
..    ..data.SetData("PlayerData", playerData);
..}
```

### How to get the data:

All what you need to do is use the **GetData<Datatype>("YourDataTag");**

```
..void GetExampleData()
..{
..    ..playerData = data.GetData<PlayerData>("PlayerData");

..    ..// GetData will return a data with the specified type
..    ..// Make sure to type the data tag correctly.
..}
```

## How to save the data:

The **SetData** method will not save the data it just assigns it, we recommend you to save the data in check points or before scene change, check points are the better way.

```
..void SaveTheData()
..{
..    ..// Just call SaveProfile and it will save all the data in your profile.
..    ..// if the profile doesn't exist it will be created when you save the first time.
..    ..data.SaveProfile();
..}
```

## How to load the data:

When you create the data object, if there is already a saved data in that profile, it will be automatically loaded, in case you don't want that and you disable auto load from the configuration. You can use **LoadProfile()** method.

```
..void LoadTheData()
..{
..    ..// Make sure the profile exist before trying to load it, or it will show an error
..    ..// To check if profile exist you can use :
..
..    ..if (data.isEmpty == false)
..        ..data.LoadProfile();
..
..    ..// This will load the profile only if is not empty and there is already a saved data there.
..}
```

**What data type can be saved?** Everything.

```
..public class ExampleData
..{
..    ..public string name;
..    ..public int integer;
..    ..public List<string> list;
..    ..public Dictionary<string, int> dict;
..    ..public struct Struct ...
..    ..public class MyClass ...
..
..
..
..
..}
```

## Other useful methods

```
void UsefullMethods()
{
    // This checks if there is a data with this tag, and returns a boolean.
    data.HasData("Data Tag");

    // This will remove a data with that tag from the save file.
    data.DeleteData("Data Tag");

    // This will delete the whole profile, make sure to make a confirmation popup when you use it.
    // When you do it all the saved data will be lost.
    data.DeleteProfile();
}
```

## Other useful parameters

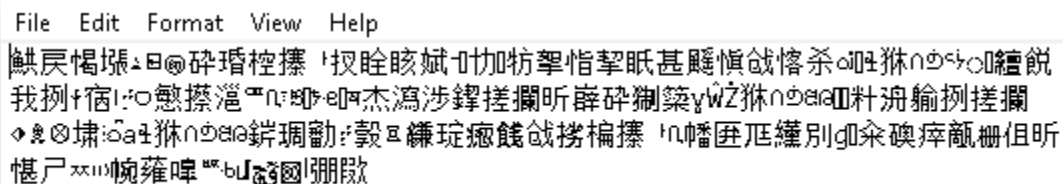
```
public string FilePath
public string BackupFilePath
public string JsonFile
public bool isEmpty
public bool isLoading
public bool isSaved
```

## What if my data file got corrupted?

If the saved data file is corrupted for some reason, the system will use a backup data file.

## Can my users cheat and change the data?

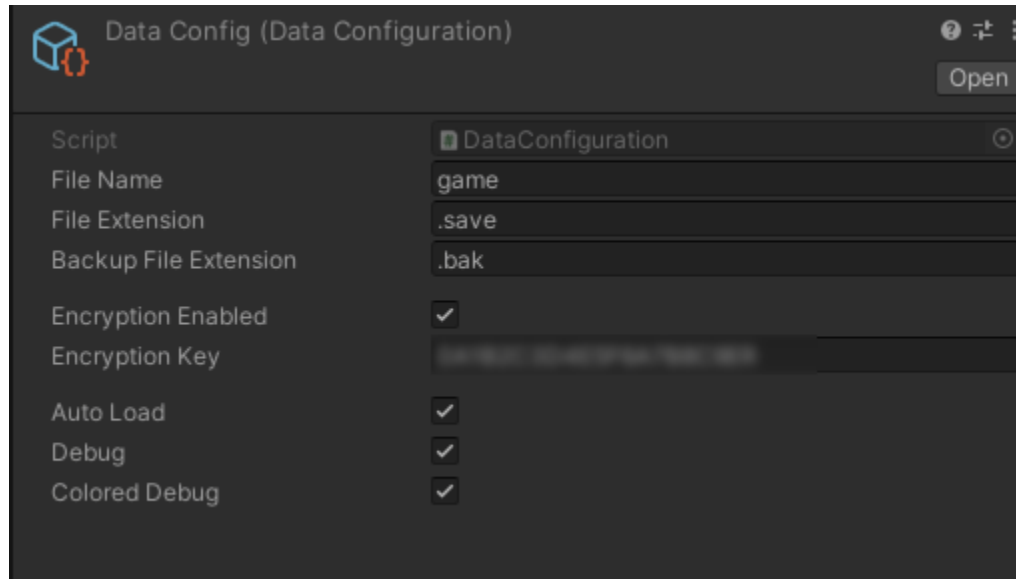
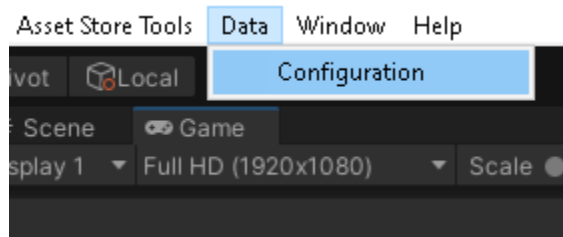
The data is encrypted with a secret key that you chose, which makes it impossible to read. You can disable the data encryption from the config.



File Edit Format View Help  
鯁戾惕張±B@碎璿控擻 扠睦眩斌叶协防掣恠挈眠甚颯慎戔愾杀o咩猓nㄔo縊銳  
我捌†宿!∩慇擦滄°n:ㄔeㄔ杰瀉涉鐸搓攔听辟碎猓築yWZ猓nㄔaㄔ料泐輸捌搓攔  
◆&⊗埔:ǒa±猓nㄔaㄔ鏘琯勸:穀E縑琰癒餞戔撈棉擻 1n幡匪厓縑別gㄔ余碘瘁甌冊但听  
悝尸x111惋薤啤°°bU3圖彌馱

This is how your saved data file will look like.

## How to change the configuration?



## Example:

```
void Examples()
{
    //lets change the player data a little bit.
    playerData.health = 250;
    playerData.position = new Vector2(10,15);

    //now lets set the data.
    data.SetData("PlayerData", playerData);

    //now lets save and close the game.
    data.SaveProfile();
}
```

```

//we call this when we open the game.
void LoadAndPrintData()
{
    //Init the data object with the profile.
    data = new Data("MyProfileID");

    //Check if there is data with this tag just in case.
    if (data.HasData("PlayerData"))
    {
        //If there is we assign our playerData to our saved playerData
        playerData = data.GetData<PlayerData>("PlayerData");
    }
    else
    {
        //If not we create it
        data.SetData("PlayerData", playerData);
    }

    //Now lets try to print the data and see if it changed
    print(playerData.health);

    //output
    //250
}

```