



Game Data Editor

Contact

- [✉ Email Me](#)
- [Tweet to Me](#)
- [GDE on Unity3d Forum](#)
- [Subscribe](#)

Overview

[What is the Game Data Editor?](#)
[What about the Free Version?](#)

Quick Start Guides

Using Spreadsheets



Managing Game Data

Accessing Game Data

Saving Game Data

GDE + Playmaker

FAQ/Troubleshooting

Other Links

[Stay At Home Devs, LLC](#)

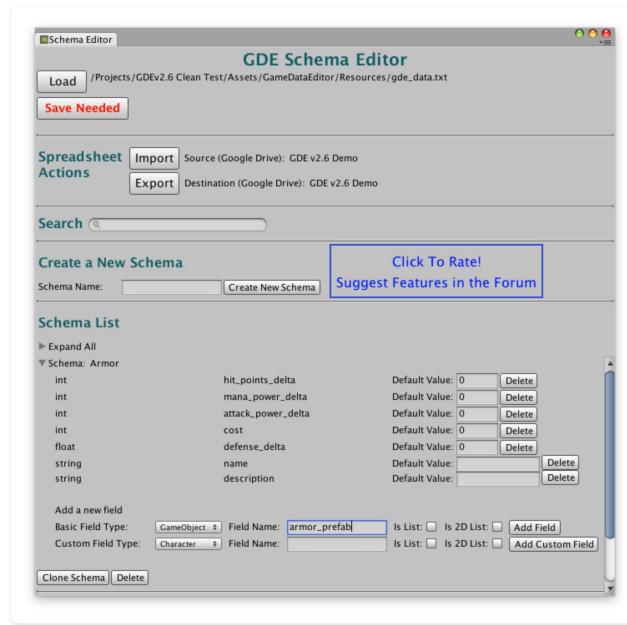
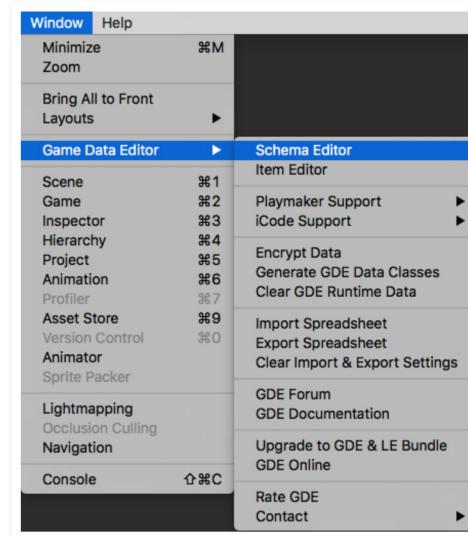
GDE Quick Start Guide v2.6.2

There are two ways to start using the Game Data Editor to manage your data. You can enter the data in manually or import it from a spreadsheet. First I'll go over manually editing your data.

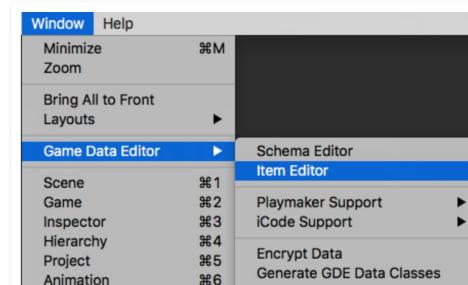
1. Enter Your Game Data

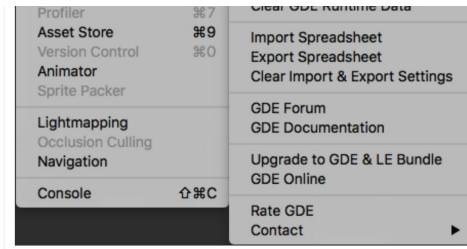
Using GDE Editor:

- First, create a schema by specifying data types with the Schema Editor



- Second, create your data items with the Item Editor





The screenshot shows the GDE Item Editor window. At the top, there are tabs for "Spreadsheet" and "Actions". Below that is a search bar and a "Create a New Item" section where a schema named "Mage" is selected. The main area is titled "Item List" and contains a table of items. One item, "Armor: LeatherTunic", is expanded to show its properties: hit_points_delta (Value: 25), mana_power_delta (Value: 5), attack_power_delta (Value: 5), cost (Value: 20), defense_delta (Value: 0), name (Value: Leather Tunic), and description (Value: Anybody who wears leather is cool.). There are also buttons for "Clone Item" and "Delete". A sidebar on the right says "Click To Rate!" and "Suggest Features in the Forum".

Import Your Game Data From a Spreadsheet:

Choose an import method (Local Excel Workbook or Google Drive) and follow the prompts. Refer to the [Spreadsheet Format](#) before importing.

The screenshot shows the GDE runtime data menu with the "Import Spreadsheet" option highlighted. A modal dialog titled "Choose Import Method" is open, containing three buttons: "Import Local File", "Import Google SpreadSheet", and "Reauthenticate With Google".

Play Mode Saving:

It is possible to save a modified item (or a newly generated item) back to your data file during Play mode.

First update your item in the master data store by calling **UpdateItem** with the item you'd like updated.

```
GDEItemData data = new GDEItemData(GDEItemKeys.Item_1);
data.some_field = "A new value";
GDEDDataManager.UpdateItem(data);
```

When you are done updating items, save your updates to the data file by calling **SaveToDisk**.

```
GDEDDataManager.SaveToDisk();
```

Note: Play mode saving is not available when Web platform is set as the build target.

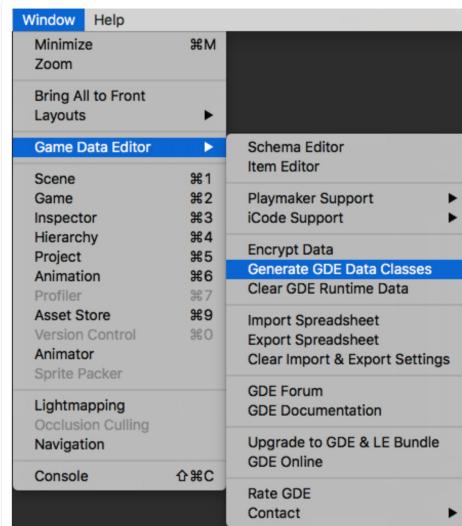
Edit Saved Data:

It is possible to edit saved data directly with the GDE Item editor. Check the "Runtime Values" box to go into runtime values mode. Any edits made to your items will be made to your saved data, not your main data file. This is useful for cheating, or testing hard to attain situations in your game.

| Item List | | <input checked="" type="checkbox"/> Show Runtime Values |
|--|--------------------|---|
| <input type="checkbox"/> Expand All | | |
| <input type="checkbox"/> Armor: LeatherTunic | | |
| int | hit_points_delta | Runtime Value: 25 |
| int | mana_power_delta | Runtime Value: 5 |
| int | attack_power_delta | Runtime Value: 5 |
| int | cost | Runtime Value: 20 |
| float | defense_delta | Runtime Value: 0.1 |
| string | name | Runtime Value: Leather Tunic |
| string | description | Runtime Value: Anybody who wears leather is cool. |
| <input type="button" value="Reset"/> | | |

2. Generate Your Data Classes

To generate the GDE Data Classes use the Generate GDE Data Classes menu action. *Note:* If you are using Playmaker or iCode, skip this step.



3. Include the GameDataEditor namespace

Include the GameDataEditor namespace at the top of any scripts where you will access GDE.

```
using GameDataEditor;
```

4. Initialize GDE

There are 3 ways to initialize GDE. From bundled Resources, from a string, or from a TextAsset.

To initialize with Playmaker, see the [Playmaker Quickstart](#). To initialize with iCode, see the [iCode Quickstart](#).

From Resources:

Include your data file in your game's Resources by placing the data file in a /Resources path. For example:

"Assets/GameDataEditor/Resources/gde_data.txt". Initialize GDE by passing the file name without the file extension:

```
GDEDataManager.Init("gde_data");
```

If you encrypted your data with GDE, add "_enc" to the end of your file name and pass "true" for the second parameter.

```
GDEDataManager.Init("gde_data_enc", true);
```

From a URL:

To initialize GDE with data from a server, first download the file, and then pass the file contents to GDE.

```
IEnumerator InitGDE()
{
    WWW payload = new WWW("http://example.com/gde_data.txt");
    yield return payload;

    if (!GDEDataManager.InitFromText(payload.text)) {
        Debug.LogError("Error initializing!");
    }
}
```

If your data was encrypted with GDE decrypt your data first:

```

IEnumerator InitGDE()
{
    WWW payload = new WWW("http://example.com/gde_data_enc.bytes");
    yield return payload;

    string data = GDEDataManager.DecryptGDEDData(payload.bytes);
    if (!GDEDataManager.InitFromText(data)) {
        Debug.LogError("Error initializing!");
    }
}

```

From a TextAsset:

You can load GDE data from a TextAsset like this:

```

// Adapted from Unity3D AssetBundle Docs
IEnumerator InitGDE()
{
    var payload = new WWW ("http://example.com/myBundle.unity3d");
    yield payload;

    TextAsset gde_data = payload.assetBundle.Load<TextAsset>("gde_data");

    if (!GDEDataManager.Init(gde_data)) {
        Debug.LogError("Error initializing!");
    }
}

```

If you encrypted your data with GDE, call the Init method like this:

```
GDEDataManager.Init(gde_data, true);
```

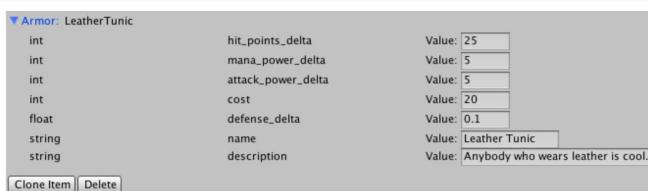
5. Read your data

The generated GDE Data Classes provide properties and methods for reading and saving your items.

To read data with Playmaker, see the [Playmaker Quickstart](#). To read data with iCode, see the [iCode Quickstart](#).

Load a single item

If your Item was defined like this:



Read it by calling the constructor of the data class of the item you'd like to load with the item's key.

```

// Make sure GDE is initialized BEFORE reading any item data
GDEArmorData tunic = new GDEArmorData(GDEItemKeys.Armor_LeatherTunic);

// Do something with tunic data
int totalHP = tunic.hit_points_delta + 100;

```

Note: The `GDEItemKeys` and `GDESchemaKeys` class will contain constant strings of all your keys. Use them to prevent typos in your scripts.

Load a random item

If you'd like to load a random item of a particular type, use the **GetRandom** method.

```
GDEArmorData randomArmor = GDEDataManager.GetRandom<GDEArmorData>();
```

Load all items of a specified schema type

If you'd like to load all items of a particular type, use the **GetAllItems** method.

```
List<GDEArmorData> allArmor = GDEDataManager.GetAllItems<GDEArmorData>();
```

6. Modify, Clone, and Save your data

Saving data to persistent data store

Please note GDE no longer uses PlayerPrefs to persist modified data. To persist your data, call the new Save method when it is convenient in your game. Be sure to save as often as you need. Web builds still use

PlayerPrefs (Save and ClearSaved methods have no effect).

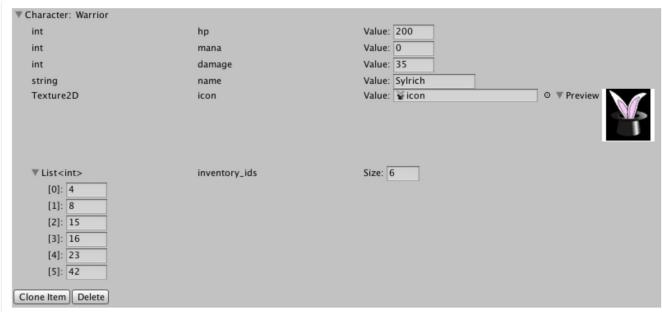
```
// To persist modified data call..  
GDEDataManager.Save();  
  
// To clear any persisted data call..  
GDEDataManager.ClearSaved();
```

Modifying data

To modify your data during runtime, use the [GDE Data Classes you generated](#).

To modify data with Playmaker, see the [Playmaker Quickstart](#). To modify data with iCode, see the [iCode Quickstart](#).

I'll use this Warrior item to demonstrate how to modify its data.



Modifying basic types:

Basic types will be saved automatically. (Unity types are read only)

```
GDECharacterData warrior = new GDECharacterData(GDEItemKeys.Character_Warrior);  
  
// Any changes to basic field types are saved automatically  
warrior.hp += 100;  
warrior.name = "Celestipoo";
```

Modifying list types:

List types will be saved by calling the **Set** method. (Unity type lists are read only)

```
GDECharacterData warrior = new GDECharacterData(GDEItemKeys.Character_Warrior);  
  
// Modify the list any way you like.  
// Add, Remove, or Change entries  
// Then call the generated Set method  
warrior.inventory_ids[0] = 314;  
warrior.inventory_ids.Add(1124);  
warrior.inventory_ids.RemoveAt(4);  
  
// Make sure to call Set!!  
warrior.Set_inventory_ids();
```

The modified warrior inventory from above will always load the modified values until it is reset. You can **reset all fields** on an item or **reset specific fields**.

```
GDECharacterData warrior = new GDECharacterData(GDEItemKeys.Character_Warrior);  
  
// Reset the inventory_ids field  
warrior.Reset_inventory_ids();  
  
// Reset all fields  
warrior.ResetAll();
```

Cloning Items

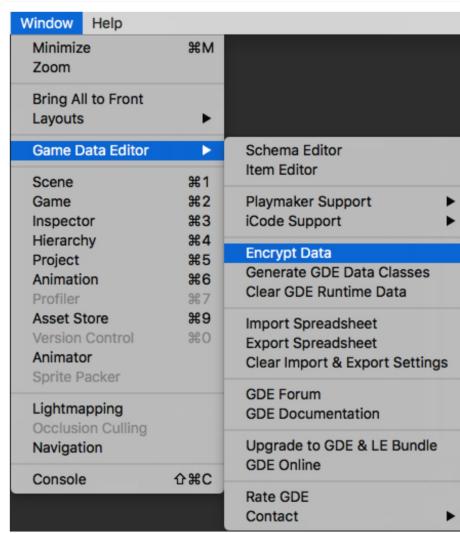
You can clone any item using the generated GDE Data Classes. Cloned items will use a new Guid as a key, and will be persisted. A Shallow Clone will clone all basic fields and lists, but will not clone Custom Types. A Deep Clone will clone all basic fields and lists, as well as Custom Types, all the way down the chain.

```
GDEItemData item = new GDEItemData(GDEItemKeys.Item_1);  
  
// To create a shallow clone, call...  
GDEItemData shallowClone = item.ShallowClone();  
  
// To create a deep clone, call...  
GDEItemData deepClone = item.DeepClone();
```

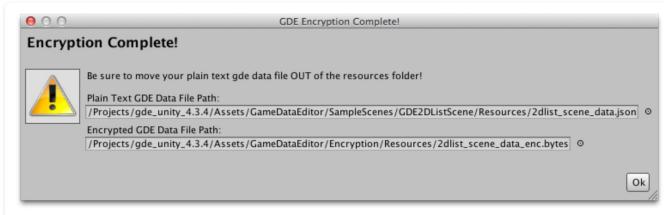
7. Encrypting Your Data (Optional)

I recommend encrypting your data at, or near, the end of your

development cycle. You will need to encrypt your data **after any changes** from the GDE Editors or when importing from a spreadsheet. To encrypt your data, go to the Encrypt Data menu action:

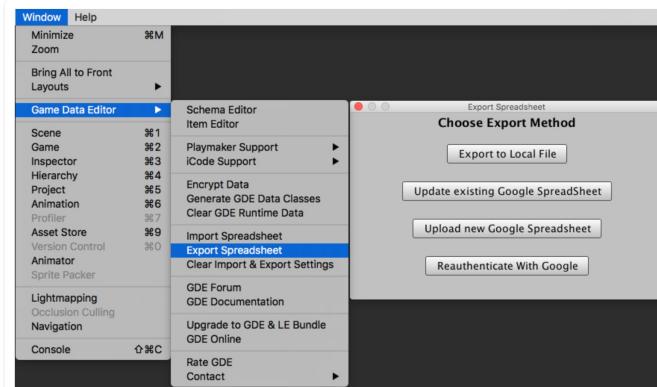


After encryption is complete, you will see a confirmation dialog. You can click the paths in the dialog to find those files in your project. To load your encrypted data see [GDE Init Section](#).



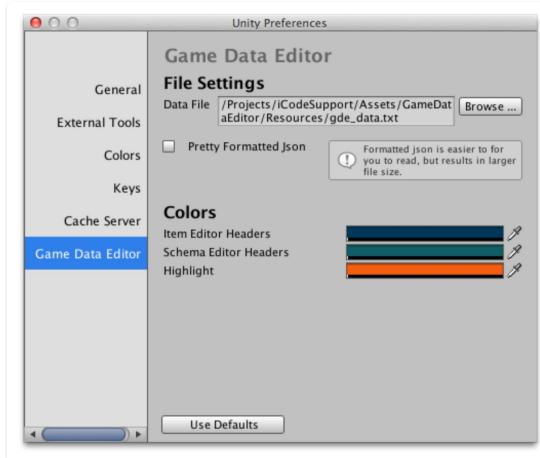
8. Exporting Data (Optional)

You can export your data to a spreadsheet for editing or archiving. Select the Export Spreadsheet menu, Choose an export method (Local Excel Workbook or Google Drive), and follow the prompts.



9. GDE Preferences

GDE Preferences can be found in Unity Preferences pane.



10. That's it!

If you have questions you can post a message on the [forum](#) or send me an [email](#).

Start the GDE Walkthrough: [Importing Spreadsheets](#) →