

UGESystem Custom Reward Guide

🚫 [CRITICAL WARNING] Data Loss on Package Update

This guide instructs you to create files within the `Assets/UGESystem/` folder for convenience.

Updating the UGESystem package may RESET (DELETE) all changes within this folder.

For safety, it is strongly recommended to create custom scripts **OUTSIDE** the package folder (e.g., `Assets/MyGame/Scripts/`), or **BACK UP** your work manually before performing any package updates.

This guide explains how to extend the **Reward System** to grant items, experience points, currency, etc.

🚫 IMPORTANT NOTICE: Web Story Maker

- Custom Rewards are **NOT supported in the Web Story Maker**.
- They can **ONLY** be configured within the **Unity Editor's Inspector window**.

Goal: What are we building?

As an example, we will create a `GiveGoldReward` that grants a specific amount of gold to the player.

Step 1: Create Reward Class

Create a class that defines the reward data (how much) and logic (how to give).

1. **Folder:** `Assets/UGESystem/Core/Scripts/UGESystem/GameEvents/Data/Storyboard/Rewards/` (or your custom folder)
2. **Create File:** `GiveGoldReward.cs` (C# Script)
3. **Write Code:** Copy and paste the code below.

```

using UnityEngine;

namespace UGESystem
{
    [System.Serializable]
    public class GiveGoldReward : AbstractEventReward
    {
        // Amount of gold to set in the Unity Inspector
        [field: SerializeField]
        public int Amount { get; private set; }

        // Default Constructor (Required)
        // Called when adding a new reward in the Editor.
        public GiveGoldReward() : base("Give Gold to Player")
        {
            Amount = 100; // Set default value
        }

        // [Core] Logic to grant the reward
        // Automatically called by the system when the event ends (EndCommand).
        public override void GrantReward(UGEEventTaskRunner runner)
        {
            // [USER IMPLEMENTATION REQUIRED]
            // Call your game project's gold manager or inventory system here.

            // Example Code:
            // if (GameManager.Instance != null)
            // {
            //     GameManager.Instance.AddGold(Amount);
            // }

            // Log for testing
            Debug.Log($"{content}"; [UGESystem Reward] Granted {Amount} Gold to the player.");
        }
    }
}

```

Step 2: Verify and Use

Unlike Command Handlers, Reward classes **do NOT require manual registration**. The system automatically detects them as long as the class file exists.

1. Return to the Unity Editor and wait for compilation to finish.
2. Select the `GameEvent` asset you want to edit.
3. Find and expand the **End Command** at the bottom of the `Commands` list. (Rewards are always granted at the end of an event.)
4. Click the **Add (+)** button in the `Rewards` list. You should see **GiveGoldReward** in the dropdown.
5. Select it and set the gold `Amount`.

Now, when that event finishes in the game, the `GrantReward` function will automatically execute, granting the gold.