

Realistic Car Controller Pro

Thank you for purchasing and using Realistic Car Controller Pro. This documentation will guide you on how the customization system works.

RCCP_Customizer.....	3
Spoiler Manager	4
Siren Manager	4
Upgrade Manager	5
Paint Manager	5
Wheel Manager.....	6
Configuration.....	7
How To Add & Use Customization.....	7
How the UI Buttons and Sliders Work for the Customization?	7
Creating New Managers.....	8
Modular Upgrade Managers	8
How To Customize with Scripting.....	9

Vehicles have 8 different upgradable customizations. They are;

- Spoilers,
- Sirens,
- Upgrades (Engine, brake, and handling),
- Paints,
- Wheels
- Customization (Suspensions, springs, dampers, etc...)
- Decals
- Neons

Customization system works with the attachable component named “[RCCP_Customizer](#)”. Select your vehicle on your scene or prefab and add it to your vehicle. [RCCP_Customizer](#) includes 8 different upgradable customizations. Each of them has been managed by their own manager script.



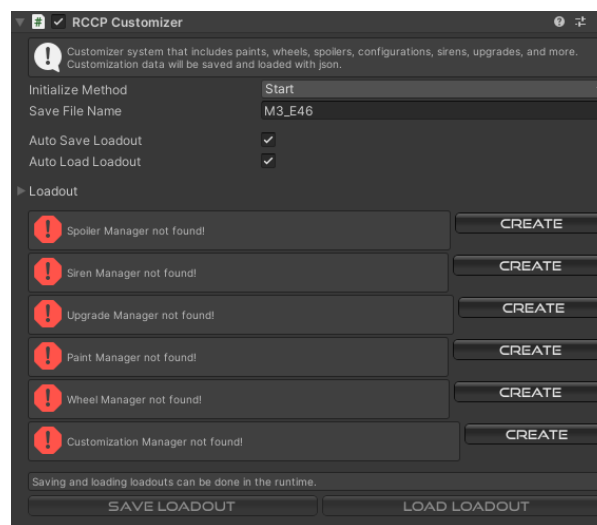
All demo vehicles have this addon component installed. You can use this component to customize your vehicle. For example, there is a “**Customization**” button in the **RCCP_Canvas** (*Options menu*). When you click this button, customization panel will pop up, and you’ll be able to customize your vehicle at runtime. Each category will be explained below.

Also, there is a new demo scene named “[RCCP_Scene_Blank_Customization](#)” with the customization features can be found in the scenes folder. This scene has a trigger marker. When player vehicle triggers it, canControl bool of the vehicle will be disabled, and modification UI menu will pop up. When player pushes the close button, canControl bool of the vehicle will be enabled, and modification UI menu will be disabled.

Each upgrade system in the [RCCP_Customizer](#) has a manager script to control the corresponding upgrade item. All upgrade managers have been managed and observed by the [RCCP_Customizer](#) script attached to the vehicle as an addon component. UI buttons for customization will be related to the player vehicle directly. Creating and editing the upgrade managers won't waste your time, they are plug and play systems. I'll be explaining each manager system below.

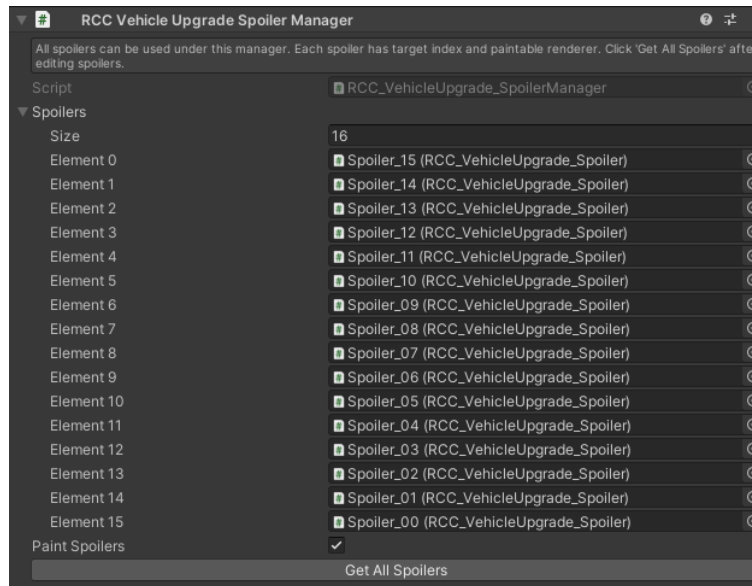
RCCP_Customizer

Simply, select your vehicle and click the customizer button to add / select the customizer component. Save file name will be used to save the customization loadout data with Json. If auto load loadout is enabled, latest saved loadout data will be used to customize the vehicle when it's enabled. If auto save loadout is enabled, customization data will be updated each time when player makes changes on the vehicle. Initialize method can be adjusted as well.



Spoiler Manager

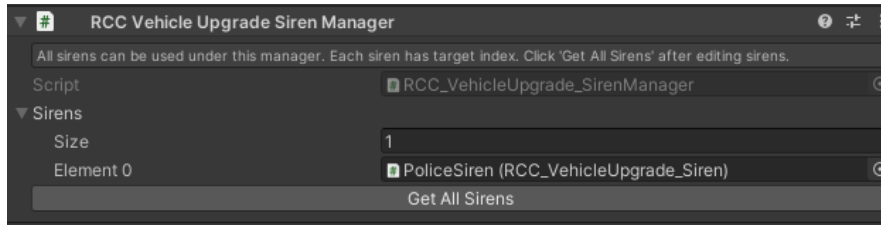
Vehicles may have many spoilers, they are just gameobjects with RCCP_VehicleUpgrade_Spoiler script. Manager will enable only selected one and disable all other spoilers. All you must do is, reposition all spoilers for your vehicles. And adding them to the list.



When you add / remove / change any spoilers, you must click **“Get All Spoilers”**. Otherwise, your changes won't be applied. Your spoiler gameobjects must have RCCP_VehicleUpgrade_Spoiler script.

Siren Manager

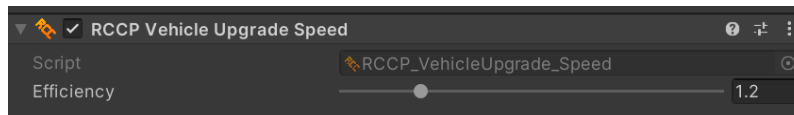
Vehicles may have many sirens, they are just gameobjects with RCCP_VehicleUpgrade_Siren script. Manager will enable only selected one and disable all other sirens. All you must do is, reposition all sirens for your new vehicles. And adding them to the list.



When you add / remove / change any sirens, you must click “**Get All Sirens**”. Otherwise, your changes won’t be applied. Your siren gameobjects must have RCCP_VehicleUpgrade_Siren script.

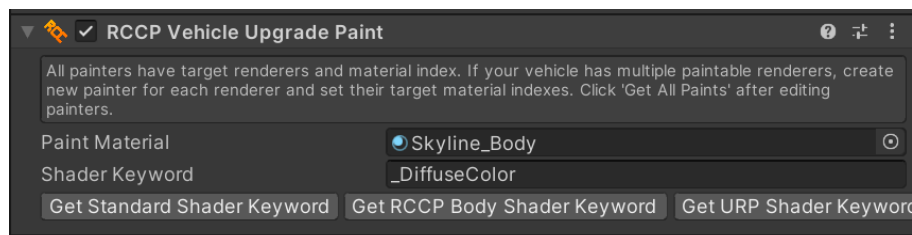
Upgrade Manager

Vehicles can have four upgrade managers. They are engine, brake, speed, and handling. They are placed inside the manager script. You can simply select the upgradable item and set their maximum efficiency values. Manager will take default values of the engine, handling, speed, and brake. And manager has 5 levels for each upgrade system. At 5th level, target value of the engine, handling, and brake will be applied. Default efficiency value is x1.2. This means that, if your engine has 500nm as maximum torque, it will be 600nm if player reaches the fifth upgrade level for the engine. Maximum efficiency value can’t be higher than 2x.

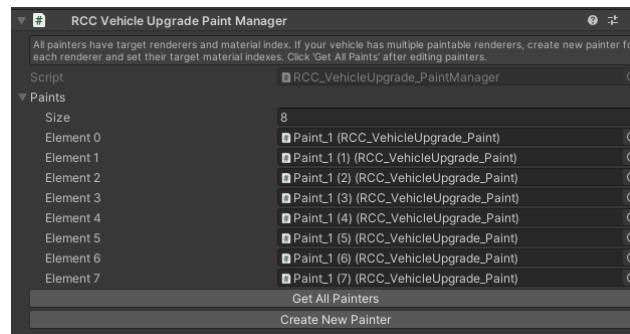


Paint Manager

Vehicles may have many paintable parts. Paint manager includes painter gameobjects with target material. All you must do is, setting target paintable material of the vehicle. Just be sure to click “**Get All Painters**” button when you add more painters. Each painter has target material, and target shader keyword.

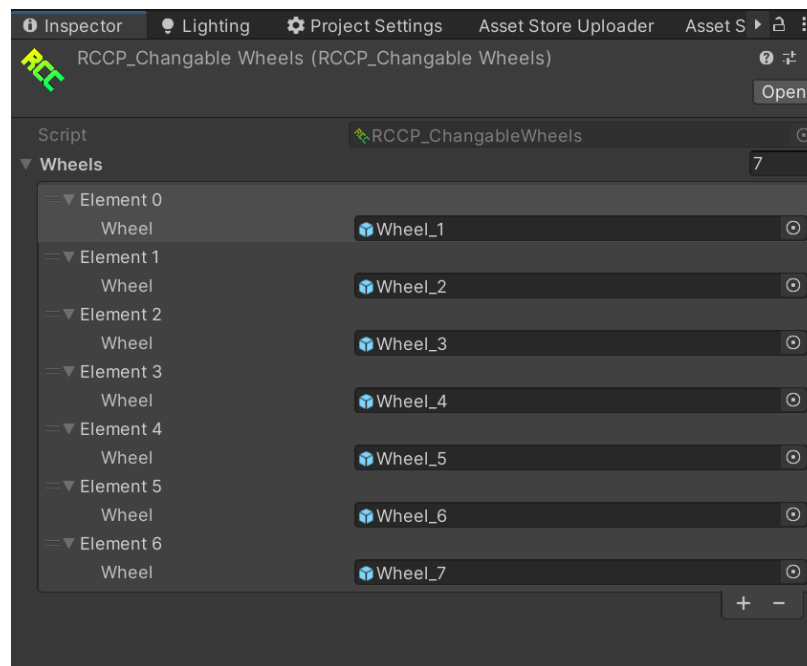


Note: The chosen material for the painter will be used to reference only. It won't edit the material directly in the project. It will be used to find instance of the target materials. If your vehicle has multiple paintable parts, you can duplicate the painters and select other materials as well. Just be sure to click **"Get All Painters"** button when you add more painters to the manager.



Wheel Manager

Wheel manager will take the wheel prefabs from the [RCCP_ConfigurableWheels](#) (**Prefabs/Wheels** or **Tools** → **BCG** → **RCCP** → **Configure Ground Materials**). You can simply change the wheel prefabs, delete, or add your own wheel prefabs. Be sure your wheel model has proper pivot position and axes, otherwise wheel will be positioned at the wrong location. X should be facing to right, Y to up, and Z to forward.



Configuration

Configuration of the vehicle can be edited at runtime with the customization manager. You can access the customization manager of the **RCCP_Customizer** to make this work. You can use any public method in the manager script to customize your vehicle.

How To Add & Use Customization

Your customizable vehicles must have “**RCCP_Customizer**” component. This component is managing all systems itself. Add this component to your vehicle and create corresponding managers. Your scene doesn't have to include any other managers unlike the older versions. **RCCP_Customizer** is responsible for the whole customization process.

How the UI Buttons and Sliders Work for the Customization?

UI buttons in the customization panel have scripts to interact with the **RCCP_Customizer** component attached to the player vehicle. These scripts are;

- **RCCP_UI_Spoiler** (For spoiler buttons. Has target index value.)
- **RCCP_UI_Siren** (For siren buttons. Has target index value.)
- **RCCP_UI_Wheel** (For wheel buttons. Has target index value.)
- **RCCP_UI_Color** (For color buttons. Has target color value.)
- **RCCP_UI_Upgrade** (For upgrade buttons. Has target values for engine, handling, and brake. And have text to display current level.)
- **RCCP_UI_CustomizationSlider** (For customization sliders such as cambers, suspension distances, spring forces, dampers, target positions, and more. Value of the slider will be used.)

These UI buttons and sliders will interact with the player vehicle first. If player vehicle has been found, they will look for the **RCCP_Customizer** component attached to the vehicle. If vehicle has that component, corresponding manager will be used to customize the vehicle (if exists).

Creating New Managers

Once you add the **RCCP_Customizer** component to your vehicle, it won't include any systems yet. You can create them by simply clicking the **"Create"** button for each manager. Creating managers will be using demo assets located in the **"Resources/Customization Setups"**. You can change them as you wish. All you must do is, reposition the visual upgradable items.

Modular Upgrade Managers

Vehicles may not have all managers together. For example, your vehicle may not include sirens, or spoilers. Simply don't create it, UI button at the customization canvas will be disabled for this vehicle.



How To Customize with Scripting

You can use public methods in the **RCCP_Customizer** script to customize the vehicle. To do that, your vehicle must have **RCCP_Customizer** component. It has several main managers for paint, wheels, upgrades, customization, spoilers, sirens, etc... First, you'll need to access to the manager component, and then you can make any changes.

Example;

```
yourVehicle.Customizer.PaintManager.Paint(color);
```

```
yourVehicle.Customizer.WheelManager.UpdateWheel(index);
```

You can use all public methods in the **RCCP_Customizer** script to customize your vehicle in your own script.