

## Spawning New Vehicles With Code

You don't have to use `GameObject.Instantiate()` for spawning new vehicles. You can spawn new vehicles by just one line of code using `RCC.cs` class. You can take a look at API documentation named "[Realistic Car Controller V3.53 API](#)" for all methods in `RCC.cs`. These methods are only used for spawning, registering, de-registering, setting controllable, and setting engine state of the vehicle. `RCC.cs` has many static methods.

## Spawning New Vehicles With Given Position, Rotation, Sets It's Controllable, And Engine State

You can spawn new vehicles by;

```
RCC.SpawnRCC(RCC_CarControllerV3 vehiclePrefab, Vector3 spawnPosition, Quaternion spawnRotation, bool registerAsPlayerVehicle, bool isControllable, bool isEngineRunning);
```

As you can see, you can spawn your vehicle with given configuration by only one line of code just like above. You just only need to have a prefab of your vehicle.

## Registering Vehicle As Player Vehicle

You can register the vehicle as player vehicle by;

```
RCC.RegisterPlayerVehicle(RCC_CarControllerV3 vehicle);
```

```
RCC.RegisterPlayerVehicle(RCC_CarControllerV3 vehicle, bool isControllable);
```

```
RCC.RegisterPlayerVehicle(RCC_CarControllerV3 vehicle, bool isControllable, bool engineState);
```

At this moment, registered vehicle will be active player vehicle with given settings.

## De-Registering Player Vehicle

You can de-register the player vehicle by;

```
RCC.DeRegisterPlayerVehicle ();
```

At this moment, registered vehicle will not be active player vehicle anymore. Player won't be able to control any vehicles.

## Setting Controllable State Of The Vehicle

You can set controllable state of the vehicle by;

```
RCC.SetControl(RCC_CarControllerV3 vehicle, bool controlState)
```

## Setting Engine State Of The Vehicle

You can set engine state of the vehicle by;

```
RCC.SetEngine(RCC_CarControllerV3 vehicle, bool engineState)
```