

# SetVehicleDoorState

This function sets the state of the specified door on a vehicle.

## Syntax

```
bool setVehicleDoorState ( vehicle theVehicle, int door, int state )
```

**OOP Syntax** Help! I don't understand this!

**Method:** *vehicle:setDoorState(...)*  
**Counterpart:** *getVehicleDoorState*

ADDED/UPDATED IN VERSION 1.5.8 r20319:

bool setVehicleDoorState ( vehicle theVehicle, int door, int state [, bool spawnFlyingComponent = true ] )

## Required Arguments

- **theVehicle:** The vehicle that you wish to change the door state of.
- **door:** An integer representing which door to set the state of. Valid values are:
  - **0:** Hood
  - **1:** Trunk
  - **2:** Front left
  - **3:** Front right
  - **4:** Rear left
  - **5:** Rear right
- **state:** An integer representing the state to set the door to. Valid values are:
  - **0:** Shut, intact (aka Closed, undamaged)
  - **1:** Ajar, intact (aka Slightly open, undamaged)
  - **2:** Shut, damaged (aka Closed, damaged)
  - **3:** Ajar, damaged (aka Slightly open, damaged)
  - **4:** Missing

## Optional Arguments

*NOTE:* When using optional arguments, you might need to supply all arguments before the one you wish to use. For more information on optional arguments, see optional arguments.

ADDED/UPDATED IN VERSION 1.5.8 r20319:

**Note:** This parameter doesn't work during the vehicle creation. You need a SetTimer if you need to create the vehicle using this parameter.

- **spawnFlyingComponent:** A boolean, if set to true, spawns flying doors etc. if you remove a component with state == 4.

## Returns

Returns *true* if the door state was successfully set, *false* otherwise.