

# *Realistic Water Physics*



## Manual

### Current version 3.1

Last update: 4/24/2018  
(Tested on Unity 2017.3.1f1)

Unity asset store link:

<https://assetstore.unity.com/packages/tools/physics/realistic-water-physics-3-107126>

Contact:

<http://realisticphysics.com>

Youtube Demo/Preview videos:

[https://www.youtube.com/channel/UCFIZQjz7wRFG0GFJOnUiH\\_g](https://www.youtube.com/channel/UCFIZQjz7wRFG0GFJOnUiH_g)

Web Demo:

Link coming soon

# Realistic Water Physics

## [Manager](#)

[Setup World](#)

[Manage Gameobjects](#)

[Setting up a gameobject](#)

[Support](#)

[Create custom material](#)

[Water Areas](#)

[Settings](#)

[Enable Advanced options:](#)

## [Realistic Component Overview:](#)

[Realistic Component Settings:](#)

[Support options](#)

## [Child Support Component Overview:](#)

## [Water Area Component Overview:](#)

[Water Area Component Settings:](#)

[Water Area Component Advanced:](#)

[Water Area Component Support:](#)

## [Demos:](#)

[Extra scripts:](#)

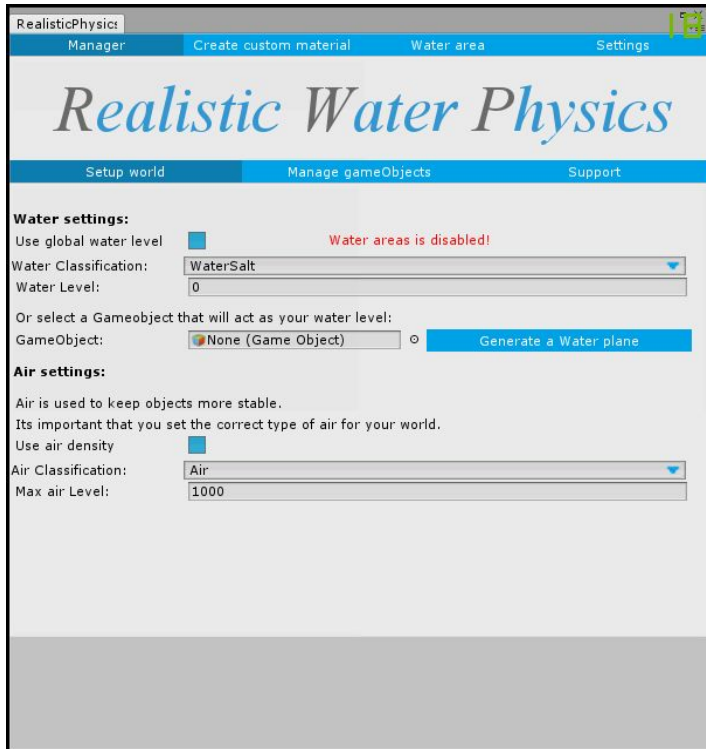
[Example code:](#)

## [All Default Materials](#)

[Knows issues:](#)

[Future updates:](#)

# Manager




The manager is something you will be using most of the time.

The manager includes all the options you will need, including the asset settings.

By default the manager is set to the default world options.

More on each menu below.

 *The manager is build for self repair, If required '.asset' files are deleted it will generate them again. This may however crash the interface, simply close and reopen the interface will sort those issues out.*

# Setup World

RealisticPhysics

Manager Create custom material Water area Settings

## Realistic Water Physics

Setup world Manage gameObjects Support

**Water settings:**

Use global water level  Water areas is disabled!

Water Classification:

Water Level:

Or select a Gameobject that will act as your water level:

GameObject:

**Air settings:**

Air is used to keep objects more stable.  
Its important that you set the correct type of air for your world.

Use air density

Air Classification:

Max air Level:

## Water settings:

### Use global water level:

If enabled, the global water level option below will be used.

### Water Classification:

Select what kind of water your world has, there are 17 included water types for you to use.

### Water Level:

Here you set the world's water level.

### Gameobject:

You can assign a gameobject to act as your water level to, this will overwrite the water level set in the editor.

### Generate a water plane:

This button will generate a water plane for you, it will automatically assign it as your water gameobject.

## Air settings:

### Use air density:


If you have this enabled, Gameobject with a active buoyancy script will react to air as well. This helps to stabilize the gameobjects on the water.

### Air Classification:

Select what kind of air your world has, there are 22 included air types for you to use.

### Max Air Level:


Set a value to where your air exist in your world, if your object will reach above this, air physics will no longer be calculated.

 *The is no update/save button for this menu, it will automatically save your settings to the settings file, the settings file is located at:*

***Assets\RealisticPhysics\RealisticWaterPhysics\Resources\RealisticWaterPhysicsManager.asset***

*But we don't recommend you change the settings through the file, this may lead to unexpected bugs.*

# Manage Gameobjects

RealisticPhysics 

Manager Create custom material Water area Settings

## Realistic Water Physics

Setup world Manage gameObjects Support

**Gameobject settings:**

Floating Quality:

List order:  Show alphabetical  Show by density

Material List:

Material type:

Solid material:


**Other:**

Percentage Solid:

[Search for existing buoyancy objects](#)

Select a gameObject in the Hierarchy

[First select a gameobject.](#)

 Changes have NOT been saved!

## Gameobject settings:

### Floating Quality:

<u>Extreme low</u>	Best used for background objects with locks on the rigidbody rotations as it will spin all over the place.
<u>Low</u>	Best used for simple objects, it will not spin like crazy but it might not be most stable for a boat.
<u>Medium</u>	This might be the best option for good quality with good performance.
<u>High</u>	As it might be useful of ships and small boats it's not recommended to be use on large gameobjects.
<u>Extreme High</u>	This is the best quality with the lowest performance.

### List Order:

This helps sort the material lists below.

### Material List:

Default, This list contains only the build in default materials.

Custom, This list contains only the custom make materials.

### Material Type:

Solid, This will sort the list to only show solid materials

Liquid, This will sort the list to only show liquid materials

Gases, This will sort the list to only show Gas materials

### Solid Material / Liquid Material / Gas Material:

Depending on the option above it will show you all the materials you can use for this gameobject(s)

If you have the default Material list selected you will have access to 53 Solid materials, 17 Liquid materials and 22 Gas materials.

## Other:


### Percentage Solid:

With this option you can set how much of the object is solid, for example a donut has a hole in the middle, this means that your object is not 100 % solid.

### Attempt to Auto-calculate:

This button will help calculating the percentage solid for you.

***For this to work a gameobject must have a collider and mesh.***

 *The materials are **not** Unity Physic materials. The materials used are only used for the buoyancy system.*




# Setting up a gameobject

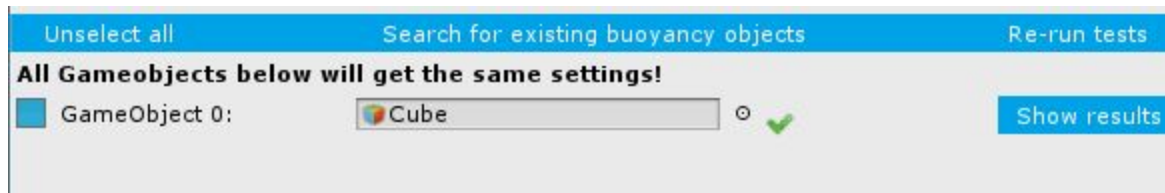
## Search for existing buoyancy objects:

This will alter the Hierarchy to only show existing gameobject with the buoyancy script.

## or:

If you have no existing buoyancy objects, just select a gameobject in your hierarchy that you want to set up.






 *You can select gameobject from your hierarchy with a simple click, or Ctrl or Shift click to select many at the same time, All the selected gameobject will set the exact same settings.*




You will now see your gameobject in the interface.

Once a new gameobject is selected the interface will automatically test your gameobject to see if it's suitable as a buoyancy object.

## The little icons:

	This object can not use used as a buoyancy object
	This object is missing a Mesh Filter.
	This object is missing a Collider.
	There is something wrong with the object.
	This object is good to go!

 *You can still create a buoyancy object if the Mesh Filter OR Collider is missing, however it's not supported!*

**Show results:**

This will show you the results of the test.

**Re-run tests:**

This will re run the tests, if for example your gameobject was missing a collider, you can add the collider to your gameobject without closing the interface, however once the collider is added you need to rerun the tests to update the interface.

**Checkbox:**

There is a little checkbox next to the selected gameobject, once its test was successful it will be automatically checked, if the tested failed the checkbox will uncheck.

If there is no Mesh Filter and not Collider, the checkbox will be unchecked and locked preventing you from creating a buoyancy object that won't float

**Select all / deselect all:**

Because you can select multiple object at the same time a select all and unselect all can be useful.

 *You can bypass this check in Advanced mode, however it's not recommended!*

**Create / Update:**

Once you're happy with the settings you can press the button on the bottom, this will create a script on the selected gameobject(s) if they have the checkbox next to them enabled.

If there is already a script on the gameobject, it will automatically update the script.

# Support

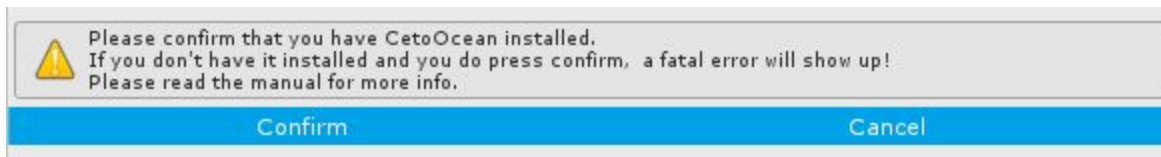
The screenshot shows a game interface for 'RealisticPhysics'. At the top, there is a navigation bar with 'Manager', 'Create custom material', 'Water area', and 'Settings'. Below this is a large title 'Realistic Water Physics' in a blue serif font. A secondary navigation bar contains 'Setup world', 'Manage gameObjects', and 'Support'. A warning box with an exclamation mark icon contains the text: 'Here you can enable and disable support assets. Please make sure to install them before enabling the support option below. For more info please read the manual.' Below the warning box is a list of support assets, each with a checkbox and an 'Unlock support' button. The assets listed are: Ceto: Ocean, Dynamic Water, Aquas Water, PlayWay Water, Ultimate Water System, Realistic Water, Suimono, and Habrador.

Asset Name	Checkbox	Action
Ceto: Ocean	<input type="checkbox"/>	Unlock support
Dynamic Water	<input type="checkbox"/>	Unlock support
Aquas Water	<input type="checkbox"/>	Unlock support
PlayWay Water	<input type="checkbox"/>	Unlock support
Ultimate Water System	<input type="checkbox"/>	Unlock support
Realistic Water	<input type="checkbox"/>	Unlock support
Suimono	<input type="checkbox"/>	Unlock support
Habrador	<input type="checkbox"/>	Unlock support

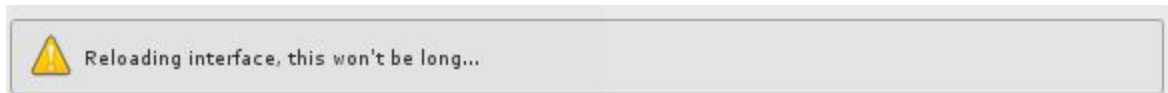
🔵 On the **Support** menu you can enable support for other assets. We are **not** the creators of those assets, we also **don't** claim to always improve the buoyancy system for those assets, however we say that the assets on this menu as tested with the realistic water physics and may or may not have included extra support options. Each of the asset creators did allow us to use part of their code in our system. **Support will be provided if our system has problems with any and only of those assets.**

🔵 If you want support for another asset in the support list, contact me and i will contact the developers. And with a bit of luck add it in the next version.

To enable support for an asset please make sure you have that asset installed already. Click the corresponding **Unlock support** button.



A message will show up asking you if you indeed have the asset installed, please check again if indeed you have the listed asset installed, and press **Confirm**, if not just click **Cancel** and install the asset first.



The interface will now recompile the script's to unlock the support of the asset. This won't be long, however the bigger your project the longer it will take. Please do not close the interface while its reloading.

Once the interface is reloaded the checkbox will unlock, confirming that the support has been unlocked.

Clicking the checkbox will enable the support of the asset.

🔵 Some assets are plug and play however some require extra steps to get working, more info about supported assets .....

# Create custom material

The screenshot shows the 'RealisticPhysics' application window. The title bar includes 'RealisticPhysics' and standard window controls. The menu bar contains 'Manager', 'Create custom material', 'Water area', and 'Settings'. The main title is 'Realistic Water Physics' in a blue serif font. Below the title is a navigation bar with 'Update existing' and 'Create new' tabs. The 'Create new' tab is active. The form contains the following fields and instructions:

- Instruction: "First select a material type you want to create or change:"
- Field: "Material type:" with a dropdown menu showing "Solid".
- Instruction: "First give the new material a name:"
- Field: "Material name:" with an empty text input box.
- Instruction: "Second, set the density of the new material, this value must be in kg/m3"
- Field: "Material density:" with an empty text input box.

A blue button labeled "Create new material" is positioned below the input fields. The bottom portion of the window is a large, empty grey area.

## **Create new:**

### **Material type:**

Select what kind of material your new material is, Solid, Liquid or Gas.

### **Material Name:**

Give your new material a name.

### **Material density:**

This might be a tricky one, the density is used to calculate a objects mass, and will determine if a object will float or not, We recommend you read the wiki on density before entering a number here. <https://en.wikipedia.org/wiki/Density>

### **Create new Material:**

Press this button once you're done.

## **Update Existing:**

### **Material type:**

Select what kind of material list you want to show.

### **Solid / Liquid / gas Material:**

Select what list you want to show.

### **Material density:**

This might be a tricky one, the density is used to calculate a objects mass, and will determine if a object will float or not, We recommend you read the wiki on density before entering a number here. <https://en.wikipedia.org/wiki/Density>

### **Update material:**

Press this button once you're done, this will update material.

### **Delete:**

This will delete the selected material.

! You can **not** alter or delete [default materials](#)

# Water Areas

RealisticPhysics

Manager Create custom material Water area Settings

## Realistic Water Physics

**Setup water area:**

GameObject 0:

List order:  Show alphabetical  Show by density

Material List:

Material type:

Gas material:

**Extra options:**

Trigger type:

! Area density:0.08992

Create / Update

! Changes have NOT been saved!

! If a message shows up: **Water areas is disabled**, then **Disable 'use global water level'** In the Manager [setup world](#) menu.

! If a message shows up: **No gameobject selected**, Select a gameobject in the Hierarchy.

## Setup water area:

To select a gameobject you want to transform into a water area you first need to select it in your editor, simply click the gameobject in the Hierarchy to select it. You can even select more than 1 at the same time.

### Material List:

Default, This list contains only the build in [default materials](#).

Custom, This list contains only the custom make materials.

### Material Type:

Solid, This will sort the list to only show solid materials

Liquid, This will sort the list to only show liquid materials

Gases, This will sort the list to only show Gas materials

### Solid Material / Liquid Material / Gas Material:

Depending on the option above it will show you all the materials you can use for this gameobject(s)

If you have the default Material list selected you will have access to 54 Solid materials, 17 Liquid materials and 22 Gas materials.

### Trigger type:

This will affect how the object will be triggered to interact with the water.

Trigger, is the most reliable option.

Collider, is a lot slower then the Trigger type.

Dimension, Currently work in progress and may not work at all!




### **Create / Update:**

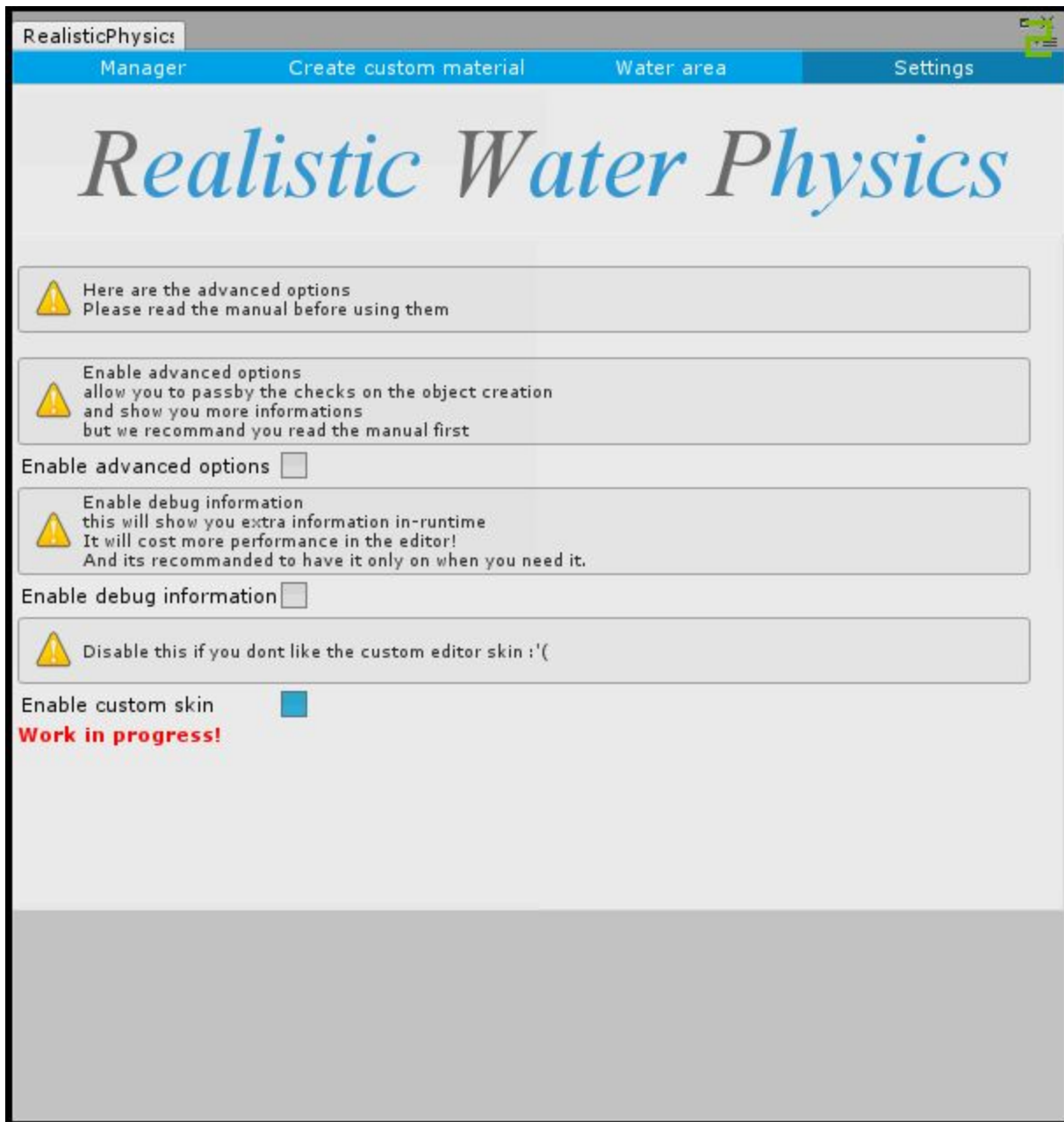
Press this button once your done.

If you're creating a new water area, the interface will automatically sets up a script with your settings on the selected gameobject.

If you're updating an existing water area, the interface will update the script on the selected gameobject.

 *If a message shows up: **Already got a buoyancy script**, The selected gameobject already got a buoyancy system attached to it, it's therefore **not** possible to add a water area to the gameobject.  
Ignoring this will result in a **error!***

# Settings



! The settings menu effects not only the interface but also the scripts and run time experience.

**Enable Advanced options:**

Once enabled you'll be working in the advanced interface, unlocking all the advanced options. What may sound good however this option allows you to bypass gameobject compatibility checks and may result in gameobjects that won't be floating like expected. Unless you're sure about what each option does it's recommended to leave this off.

**Enable debug information:**

Once enabled, you will get all the debug information of the asset. We tried to lower the spam as much as possible but it will give you all the information what each object in the scene does and what every button in the interface does, as this might be a bit too much info, it's recommended to keep this option off unless you have a gameobject that's not working as expected.

**Enable custom skin:**

As you might have already seen the interface uses a custom skin. This is to make it look good, but it has 1 flaw, it's bright! If you're using the pro editor skin you might want to disable this option. Enabled or disabled has no effect on how the interface works.

# Realistic Component Overview:

*Realistic Water Physics*  
*Buoyancy Component*

**Settings**

slicesPerAxis:	2
voxelsLimit:	16
Material name:	Aerogel
Material density:	1
Percentage Solid:	100
Rigidbody Mass:	0.01
Volume:	0
Calculated density:	0

**Modifiers**

Any change will have direct impact of the object.  
But they will not save while in run-time!

Mass:	<input type="text" value="0.01"/>
Percentage solid:	<input type="text" value="100"/>
Water demping	<input type="text" value="0.1"/>
Air demping	<input type="text" value="0.01"/>
Center of mass	X <input type="text" value="0"/> Y <input type="text" value="0"/> Z <input type="text" value="0"/>

**Advanced settings**

Only change this if you know what you are doing!

Mesh filter:	<input type="text" value="None (Mesh Filter)"/>	⊙
Mesh collider:	<input type="text" value="None (Mesh Collider)"/>	⊙
Collider:	<input type="text" value="None (Collider)"/>	⊙


**Support options**


No extra options available right now

In the overview you can read the settings of the realistic component.

! The interface has been changed in version 3.1, we recommend you update your version to match the manual.

# Realistic Component Settings:

 *The settings are applied the moment you change them, its there for best to pause the game if its in runtime before applying them*

 *The settings are not saved while in runtime, however its possible to change all of them and see the effect of it.*

## **Mass:**

Changing this value will after the object's mass without recalculating everything else. This might be useful while testing.

## **Percentage solid:**

This is basicly the save option as in the interface, allowing you to change it without recalculating everything else.

## **Water damping:**

Allows you to change how the water is damping the object.

## **Air damping:**

Allows you to change how the air is damping the object.

## **Center of mass:**

Allows you to change the center of mass of the gameobject.

## **Mesh Filter:**

Here you can assign another mesh filter to act as its own mesh filter.

## **Mesh Collider:**

Here you can assign another mesh collider to act as its own mesh collider.


## **Collider:**

Here you can assign another collider to act as its own collider.


### **Recalculate**

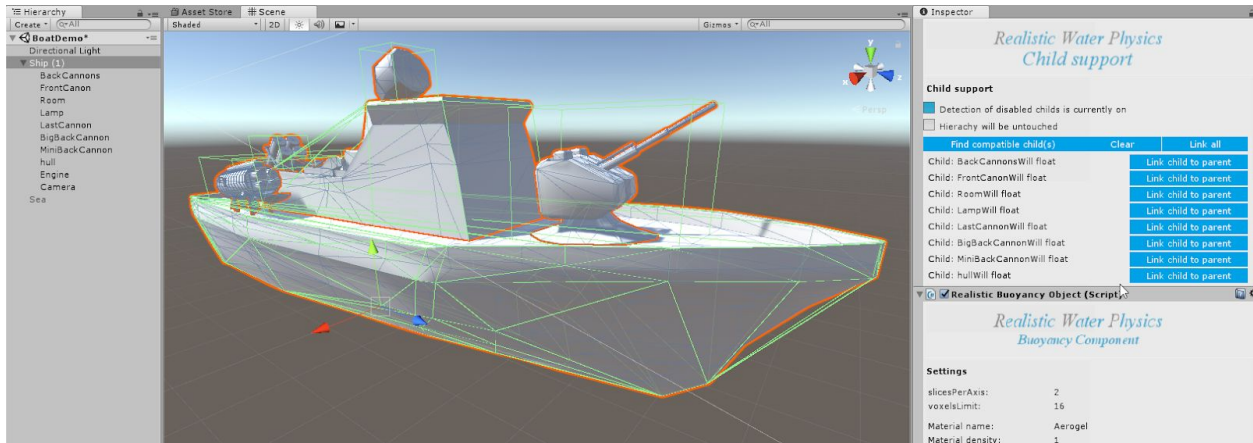
Once you're done, click the recalculate button.

This will force the script to recalculate the object, however it's not recommended to do it often as it will create some lag in runtime.

 *As it might be useful in some cases to use another gameobjects mesh and or collider, it does add an extra risk of bugs to the script.*

### **Support options**

 *Once you enabled support for an asset in the [interface support menu](#) Extra options for the gameobject will show up here. The options are made to look a lot like the options from those assets and they will explain themselves.*



# Child Support Component Overview:




### **Detection of disabled childs**

With this option turned on, the **Find Compatible chil(s)** button will also list disabled gameobjects.

### **Hierarchy will remain untouched**

As its possible to unlink and link gameobject in runtime, if you have this option turned on the gameobject that you unlink from the parent will also be removed as a child object from its parent, and on link, will be part of the gameobject as a child.

This will improve the physics of the objects, however it might not always be what you want.

 *To stop objects that are already a part of the gameobject but is a child of a child to be reconnected to the main parent, on link with this option turned on the gameobject will only be a child of the parent if it currently has no parent.*

### **Find Compatible chil(s)**

This will list all available childs of this gameobject, including the child objects of childs. However a gameobject is only compatible if it has a realistic Buoyancy component attached.

### **Clear**

This will clear the list of all objects, it will NOT unlink them!

This is usefull if you have deleted a child, press **clear** and **Find Compatible chil(s)** again.

### **Link all/ unlink all**

This will link or unlink all the gameobjects in the list

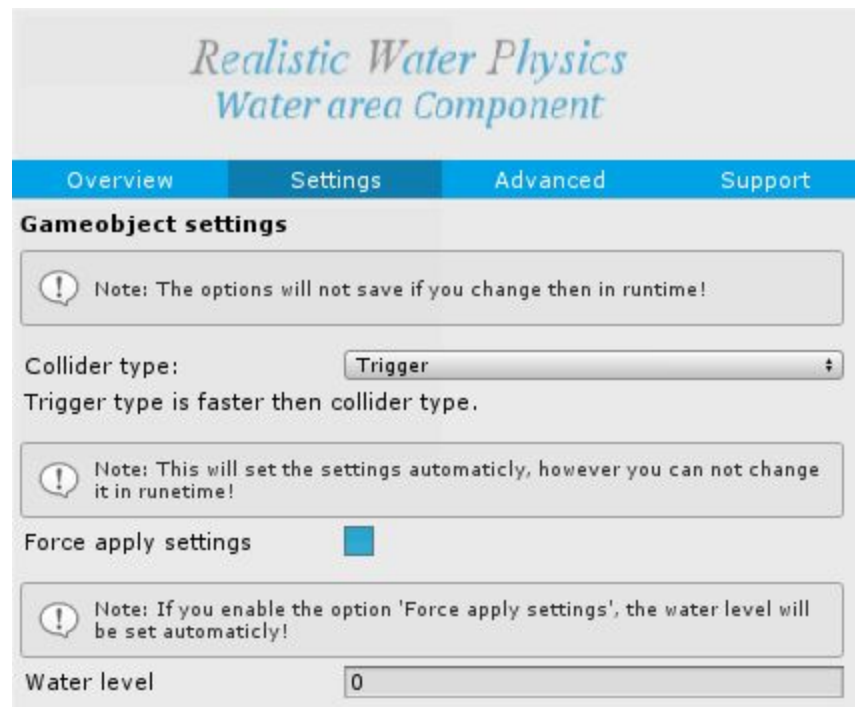


# Water Area Component Overview:



In the overview you can read the settings of the Water area component.

# Water Area Component Settings:



## Trigger type:

This will affect how the object will be triggered to interact with the water.

Trigger, is the most reliable option.

Collider, is a lot slower then the Trigger type.

## Force apply settings:

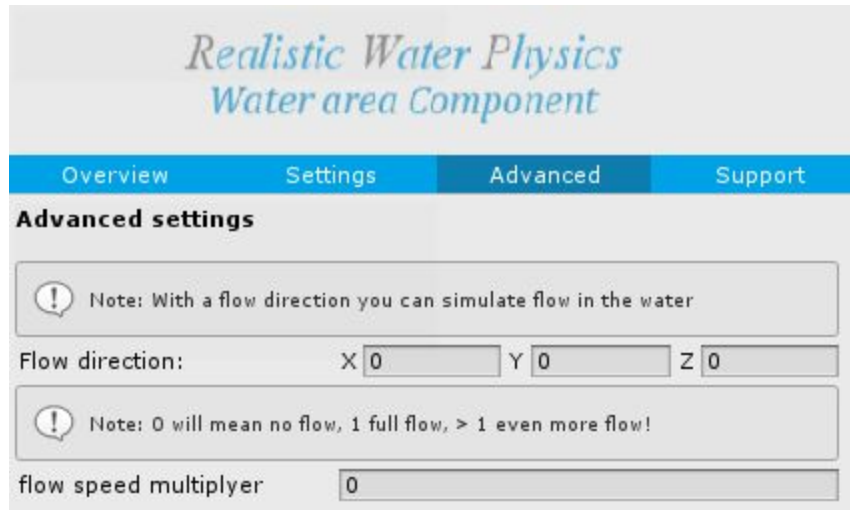
If you have this option enabled the script will make sure everything is setup correctly.

For example, it will make sure that if the type is set to trigger, that there is a collider on this gameobject with is trigger set to true.

## Water Level:

The water level is world space, if force apply settings is enabled the script will make sure this is set correctly.

# Water Area Component Advanced:



The screenshot shows the 'Advanced settings' section of the 'Realistic Water Physics Water area Component' interface. It features a navigation bar with 'Overview', 'Settings', 'Advanced', and 'Support'. Below the navigation bar, there are three input fields: 'Flow direction' with sub-fields for X, Y, and Z, and 'flow speed multiplier'. Each input field has a note icon and a text explanation.

*Realistic Water Physics*  
*Water area Component*

Overview Settings **Advanced** Support

**Advanced settings**

Note: With a flow direction you can simulate flow in the water

Flow direction: X  Y  Z

Note: 0 will mean no flow, 1 full flow, > 1 even more flow!

flow speed multiplier

## Flow direction:

Simulate a flow in water, all object in this water area will be affected.

## Flow speed Multiplier:

0 = no flow,

1 = full speed,

Everything above 1 is even more flow


# Water Area Component Support:

• Once you enabled support for an asset in the [interface support menu](#) Extra options for the gameobject will show up here. The options are made to look a lot like the assets buoyancy options, The options will explain themselves.

# Demos:

We have included a couple of demos for you to play with:

BoatDemo	A simple demo showing you how to control a boat with W,A,S,D and V for turbo. Its a simple demo and the included BoatEngine can be used to help you understand how to do this yourself.
ScriptExample	A simple demo showing you how to create gameobject in runtime, allowing you to bypass the interface, because you don't have access to the interface in runtime. The included RealisticWaterPhysicsScriptExample can be used to help you understand how to do this yourself.
SubChildSupport	This demo includes the boat that's used in the BoatDemo, this boat is using the SubChild system. This system allows you to link up multiple gameobject together to act as 1 object without using fixed joints. There are no example scripts in this demo.
SupportDemo	The supportDemo shows that the game works with all the supported assets, this demo is mostly used for testing while developing and might cause some issues if you try to enable supports that you do not have installed.
WaterAreaDemo	This demo shows of the new water areas, allowing you to have multiple different water types in your scene, the demo shows how the same type of gameobject/material act with each type of material.
PlayerControllerDemo	This demo will show how it's possible to have your player interact with Realistic buoyancy gameobjects.


 You are free to use any/all content of the demo files for your own projects.

## Extra scripts:

In the assets are a few extra scripts, some are free to use others are only useful for the asset scripts themselves, below is a small list with a bit of information what is what and how to use them:

RigidBodyPointer.cs	This is used as a example for the playerControllerDemo, allowing you to walk object a object and experience the forces from the rigidbody its pointing at.
ExternalWaterLevelManager.cs	If you don't want to use a static water level, but something that's a bit more dynamic use this script. Drag and drop it on a GameObject that acts as your water.
BoatEngine.cs	This is a simple example on how to add force your your boat/gameobjects.
RealisticManager.cs	This script is build after the main scripts where build. We stops using static scripts as of the old version 2.2 as this script is build as a example of what you can access.

# Example code:

 Below you can find example code to do some things through code.

 *RealisticBuoyancyObject* is the script that controls the buoyancy on a gameobject

## Setup through code

### RealisticBuoyancyObject.SetupScript()

#### Parameters:

**RealisticPhyscisMaterial** realisticMaterial

**int** materialPercentageSolid

**int** \_SlicesPerAxis

**int** \_VoxelsLimit

**bool** setTroughInterface

**bool** \_Concave = false


**bool** upgrade = false

This will setup your gameobject with a buoyancy component (*RealisticBuoyancyObject*)  
And will return true if successfull.

#### Example:

```
Var rbo = this.gameObject.AddComponent<RealisticBuoyancyObject>();  
Debug.Log("Setup gameobject: " + rbo.SetupScript(new RealisticPhyscisMaterial {  
Name = "AnyName", Density = 500 }, 100, 4, 4, true));
```

 You can not access the *RealisticPhysicsMaterials* from here, so make a new one.

 You sure you have **setTroughInterface** set to **true**, or the script will not have edit options.

## Get mass

### RealisticBuoyancyObject.Mass()

#### Parameters:

None

Return the calculated mass.

#### Example:

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
Debug.Log("Objects mass: "+ rbo.Mass());
```

## Remove mass

### RealisticBuoyancyObject.RemoveMass()

#### Parameters:

Float mass

This will remove that amount of mass from the script, and update its components.

#### Example:

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
rbo.RemoveMass(50);
```

## Add mass

### RealisticBuoyancyObject.AddMass()

#### Parameters:

Float mass

This will add that amount of mass from the script, and update its components.

#### Example:

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
rbo.AddMass(50);
```



## Get center of mass

**RealisticBuoyancyObject.CenterOfMass()**

### Parameters:

none

This will return a Vector3 value that's currently its center of mass.

### Example:

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
Vector3 newVector = rbo.CenterOfMass();
```

## Set center of mass

**RealisticBuoyancyObject.CenterOfMass()**

### Parameters:

**Vector3** newPoint

This will set its new center of mass and add the overwrite center of mass to it.

### Example:

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
rbo.CenterOfMass(new Vector3(0,0,0));
```

## Move to Position

### RealisticBuoyancyObject.MovePosition()

#### Parameters:

**Vector3** position

This will move the gameobject to that position without flipping the object over.

#### *Example:*

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
rbo.MovePosition(new Vector3(0,0,0));
```

## Add force

### RealisticBuoyancyObject.AddForce()

#### Parameters:

**Vector3** force

**ForceMode** mode = ForceMode.Force

#### Parameters:

**float** x

**float** y

**float** z

**ForceMode** mode = ForceMode.Force

This will add force to the gameobject just like Rigidbody.AddForce does, however it will calculate its mass to.

#### *Example:*

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
rbo.AddForce(new Vector3(0,0,0));  
  
rbo.AddForce(0,0,0);
```

## **RealisticBuoyancyObject.AddForceAtPosition()**

### **Parameters:**

**Vector3** force

**Vector3** position

**ForceMode** mode = ForceMode.Force

This will add force to the gameobject just like `Rigidbody.AddForceAtPosition` does, however it will calculate its mass to.

### ***Example:***

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
rbo.AddForce(new Vector3(0,0,0), new Vector3(0,0,0));
```

## **Get Volume**

[This has been removed]

## **Get Density**

### **RealisticBuoyancyObject.Density()**

#### **Parameters:**

**None**

Returns the calculated object density.

### ***Example:***

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
Debug.Log("Objects density: "+ rbo.getDensity());
```

## Get Water density

**RealisticBuoyancyObject.GetWaterDensity()**

**Parameters:**

**None**

Returns the water density this object is current in.

**Example:**

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
Debug.Log("Water density: "+ rbo.GetWaterDensity());
```

## Is In Water

**RealisticBuoyancyObject.isInWater()**

**Parameters:**

**None**

Returns True if it's currently in any type of water

**Example:**

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
Debug.Log("In water: "+ rbo.isInWater());
```

## Get Water Level

### RealisticBuoyancyObject.GetWaterLevel()

#### Parameters:

float x

float z

Returns the water level at the given point

This works with all support types.

It's used for the buoyancy system but can be used elsewhere to.

#### **Example:**

```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
Debug.Log("Water level at:: "+ rbo.GetWaterLevel(this.Transform.Position.x,  
this.Transform.Position.z);
```

## Is Under Water

### RealisticBuoyancyObject.isUnderWater()

#### Parameters:

Vector3 \_Position

This will check if the vector is underwater or not.

#### **Example:**


```
Var rbo = this.gameObject.GetComponent<RealisticBuoyancyObject>();  
Debug.Log("Object is underwater:: "+ rbo.isUnderWater(this.Transform.Position);
```


# All Default Materials

Solid	Liquid	Gas
Aerographite	WaterFresh	Air
Metallic Microlattice	WaterSalt	Acetylene
Aerogel	Liquid Hydrogen	Ammonia
Styrofoam	Tetrachloroethene	Argon
Cork	Mercury	Benzene
Pine	Gasoline	Blast Furnace Gas
Lithium	Petrol	Butane
Wood	Diesel Fuel	Carbon Dioxide
Oak	Methanol	Chlorine
Potassium	Turpentine	Hydrogen
Sodium	Milk	Hydrogen Sulfide
Ice	CrudeOil	Hydrogen Chloride
Nylon	CoconutOil	Methane
Plastics	OliveOil	Natural Gas
Magnesium	Sunflower Oil	Nitric Oxide
Beryllium	SoyBean Oil	Nitrogen
Glycerol	Beer	Oxygen
Concrete		Propane
Silicon		Sulfur Dioxide

Aluminium		Water Vapor
Diiodomethane		Helium
Diamond		Tungsten Hexafluoride
Titanium		
Selenium		
Vanadium		
Antimony		
Zinc		
Chromium		
Tin		
Manganese		
Iron		
Steel		
Niobium		
Brass		
Cadmium		
Cobalt		
Nickel		
Copper		
Bismuth		
Molybdenum		
Silver		
Lead		
Thorium		
Rhodium		
Tantalum		


Tungsten		
Gold		
Plutonium		
Platinum		
Iridium		
Osmium		
Uranium		
Human		

 *If the material you are looking for is not in this list, don't worry, With the interface you can create your own materials.*

 *This version is fully rewritten and tested by 1 person, and bugs may have slipt in. If something is not as expected, buggy or completely not working please **contact us** on the **website**, of on the **forum page**, and I will try to help you out as best I can.*



# Known issues:

 Included in the project is a text file called VersionInfo.txt This file contains the list of known issues for its current version. This way we do not have to update the manual every time we fix one of them, but we try to keep the list of known issues to a minimum.

Below are a few issues we have yet to fix, and the way to bypass them yourself.

## **Aquas water River flow will push object in the wrong direction.**

This issue is because of how the flow is calculated, this issue will disappear if you make sure the entire waterarea is in + vectors, so make sure your gameobject will not end up in a negative world vector in the water. Like (-50,0,0)

## **The interface is not selecting the right materials if i select a gameobject in the editor.**

This issue will show up in the manage gameobjects as well as in the water areas,  
As for the manage gameobject, try selecting a different gameobject first and then go back to your first selected gameobject, this will force the interface to update its information.  
As for the WaterAreas, we have not yet build this in.

## **The interface window is showing me graphics issues**

You can disable the custom skin in the settings tab, Enable custom skin. If disabled it will fallback on its unity default skin. This might some those issues.

## **Changes to the center of mass in runtime will mess up the buoyancy**

We fixed this issue in version 3.1 however we recommend you don't do this in runtime,  
To avoid issues with it if you have them.

## **Disconnecting a child gameobject from its sub child support system in runtime will mess up the buoyancy.**

This has been fixed in version 3.1.

**Creating a subChildSystem setup not in the middle of the world (0,0,0) will messup the placement of the probe points.**

This has been fixed in version 3.1

**I got a empty gameobject with a rigidbody and a realisticBuoyancy Script, it also got a Child support script and i made sure every script is turend on, however my gameobject just ignores the water help!**

The buoyancy script will disable itself internaly if it detects that is a issue with it, however you can simply press the Re-Calculate button on the bottom of the script to check again, make sure no errors show up and you will be fine.


## Future updates:

While in development we will come across a lot of things we still want to add, but as that means the development time will increase we sometime just add it to the next update list.

Below is a list of items we have yet to build in, but are planned for the next version.

### To do for version 3.2 (updated)

- Fix more issues!
- Add a single layer mesh collider for the water area.
- Add more water Dynamics to objects.
- Add better buoyancy support for Ultimate Water System Advanced buoyancy.
- Improve Habrador support with other assets.
- Add child support for Habrador support.
- Improve more support options.
- Add option to use custom materials in the setup world tab
- Add option to show/move and Hide/lock the centerOfMass position.
- Add option to use mutlible materials in procentages in 1 gameobject.
- Add option to make realistic dynamic water force and weight ajustments.
- Add better support for ragdolls, something like automaticly create everything for the user.

 *Some things might be removed or added, but the manuals might not have been updated, In the project you will find a file called VersionInfo.txt this file contains up-to-date information.*