

## how to use volumes

*Last updated: 05 June, 2005*

### Disclaimer

We hope these articles and resources will prove useful, but your projects are your responsibility. Always make any necessary backup of your work to revert to in case things go horribly wrong.

### Introduction

Volumes can be used to give an area of your level specific properties, and come in many varieties. Invisible in-game, they are most commonly used for water, lava, location names, and to block players from going 'out-of-bounds'. In previous versions of Unreal, water, lava, and other effects were created by defining a sealed zone, In Unreal 2003-4 however, volumes now replace this method.

This page will show you how to implement a range of volumes in your map, and make any necessary adjustments to it, and also touches on advanced setups. This page assumes you have basic knowledge of the editor.

### Preparations

To start with, we'll create a simple pool of water. This will give you an idea how volumes work. Skip past this section if you want info on specific volumes. Before we do this however, we'll need a prepared area within a map to work with. You may already have an area for liquid carved out in your level; if this is the case, feel free to use that area. If not, you will need to create something similar to the image below.



Our simplistic area above consists of two subtracted cubes, to create a square 'pool' of water in the middle of a room. It has a player start in one corner for testing, and is lit all over with a few basic lights. Once you have an area to work with, you're all set to add a volume.

### Adding a Volume

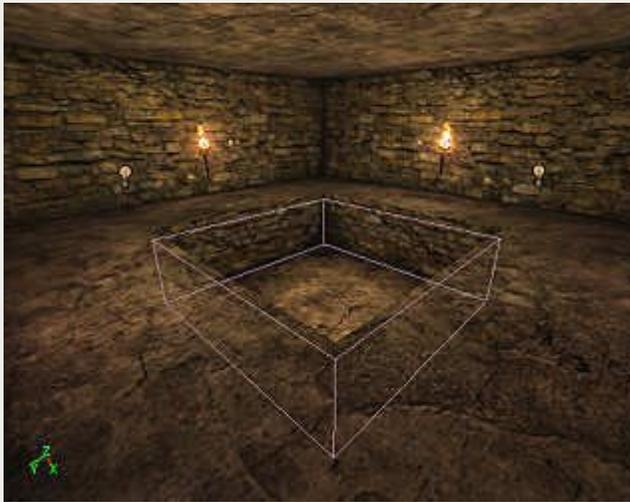
1. Using the cube-shaped red builder brush, move and resize it to completely cover the area you want to be water. Basically; everywhere inside this cube will be water. In the image below, we've 'filled' our pool with the brush, leaving a small rim at the top for realism.

2. Next, we simply need to add our volume. Simply right-click the volume button, and click 'WaterVolume'.



The 'Add Volume' button.

Move the red builder brush away, and you should see a pink wireframe cube, as below (Volumes can be hidden using the 'O' key; if no volume appears, try hitting the 'O' key).

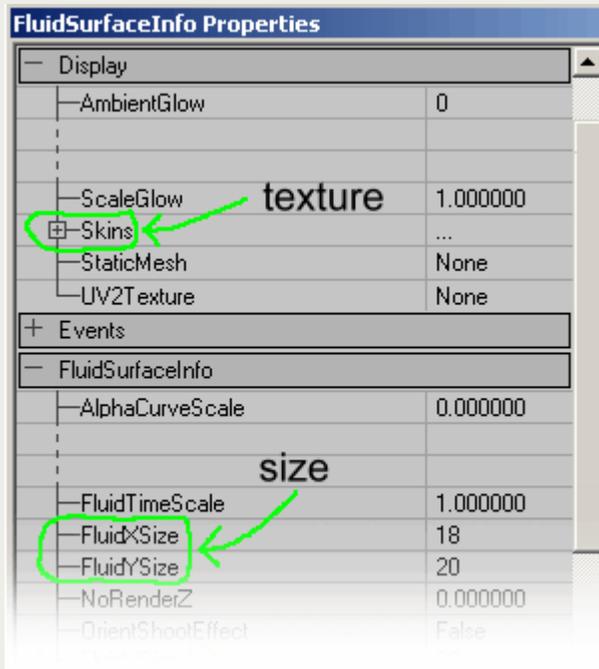


3. This water volume will be invisible in-game, so we obviously need to make it look as if there is actually a liquid there. In previous versions of unreal, this was done by adding a two-sided translucent sheet brush (usually with a animated texture). In UnrealED 3.0 however, much more realistic liquid can be added using a FluidSurfaceInfo actor; which is basically an oscillating mesh. To add a FluidSurfaceInfo actor, open your actor browser (Chess Piece button), expand 'Info' by clicking the plus (+), and click once on FluidSurfaceInfo to highlight it. Then, right-click the ground near your volume in your 3D view and select "Add FluidSurfaceInfo Here". You should be left with something like below.



4. Next, we need to resize and configure the FluidSurfaceInfo to fit our pool of water. Firstly, move the FluidSurfaceInfo over your water volume in your 2D view, so that the FluidInfo actor is roughly centered over the pool, and downward so that the mesh is pretty much level with the top of the volume.

5. Double-click the FluidSurfaceInfo actor to open it's properties. We need to change the following properties...



**Skins** - This changes the texture of your fluid. Firstly, find the texture you want and click on it once to highlight it (A good water texture is FinalWater1, located in the DefaultFluid package). Back in the properties, click on 'Skins' then click the 'Add' button. Click on the row [0] which appears, then click the 'Use' button.

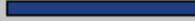
**XSize / YSize** - This is the size of our fluid. Change these values until it just fits your pool.



Our pool of water is complete! Now we simply need to rebuild (Hit F8) and test the map in-game (Remember to add a Player Start if you need to). As you can imagine, volumes and FluidSurfaceInfo can be very versatile, especially considering we've only just touched on the basics, which brings us to the next section...

## Volume Properties

In this section we're going to make some small adjustments to the pool of water we created above. Double-click the pink wireframe volume to open its properties...

WaterVolume Properties	
- PhysicsVolume	
bBounceVelocity	False
bNoInventory	False
<b>bPainCausing</b>	False
bWaterVolume	True
<b>DamagePerSec</b>	0.000000
<b>DamageType</b>	None
EntryActor	None
+ Sound	
- Volume	
AssociatedActorTag	None
DecoList	None
<b>LocationName</b>	under water
LocationPriority	0
- VolumeFog	
bDistanceFog	True
<b>DistanceFogColor</b>	
DistanceFogEnd	2000.000000
<b>DistanceFogStart</b>	8.000000

**bPainCausing, DamagePerSec, DamageType** - You may set these values so that the player inflicts damage when in the water.

**LocationName** - All volumes have a location name, which appears next to your name or on scoreboards so team members know where you are. This can be changed to something more interesting/amusing, for example 'somewhere cold and wet' or 'red base pool'.

**DistanceFog Color** - This is the fog color the player sees whilst underwater.

**DistanceFogStart / DistanceFogEnd** - Specifies the thickness of the fog. The end value is where nothing will be visible beyond that point. You can change these values to alter the thickness of the underwater fogging.

As you can see, it's quite straightforward. Change any of these values you please, and test the map to see the results. There are also more advanced options you can alter here, such as the Physics of the volume, which will be covered later.

## Other Common Volume Types

Now that you know how to add and modify your volumes, you can experiment with the many different volume types in UnrealED. Most are added and modified in the same way as above, so I'll only give a brief description here.

**Blocking Volume** - Players cannot enter this volume. This can be an essential tool in blocking players from areas they shouldn't get to, such as the edge of maps. Try to avoid 'invisible walls' if possible. Can also be used to block specific classes.

**Volume** - Used for location names. Use them in team-based maps so you know where your team members are.

**Ladder Volume** - Apply this against a wall or mesh, and setup the wall direction in its properties to create a ladder or climable wall. Use your 2D Top-Down view to view the direction of the wall, using the red arrow.

**Water and Lava Volume** - Used to create water, slime, and lava (really?!).

**Limitation Volume** - Enables you to control projectiles from weapons, such as the translocator disc. It allows you to specify an effect when one of these comes into contact with the volume.

**Physics Volume** - Allows you to alter the physics within that volume. For example, low gravity.

**Sniping Volume** - Enables you to set up bots to shoot at players. Need to be used with 'UnrealScriptedSequence' actors.

---

[Home](#) | [Projects](#) | [Articles & Resources](#) | [Gallery](#) | [Help](#)



e-mail: [webmaster@outpt.co.uk](mailto:webmaster@outpt.co.uk)

Website, Articles, & Projects by Outpt  
Unreal Series (C) Epic Games & Digital Extremes