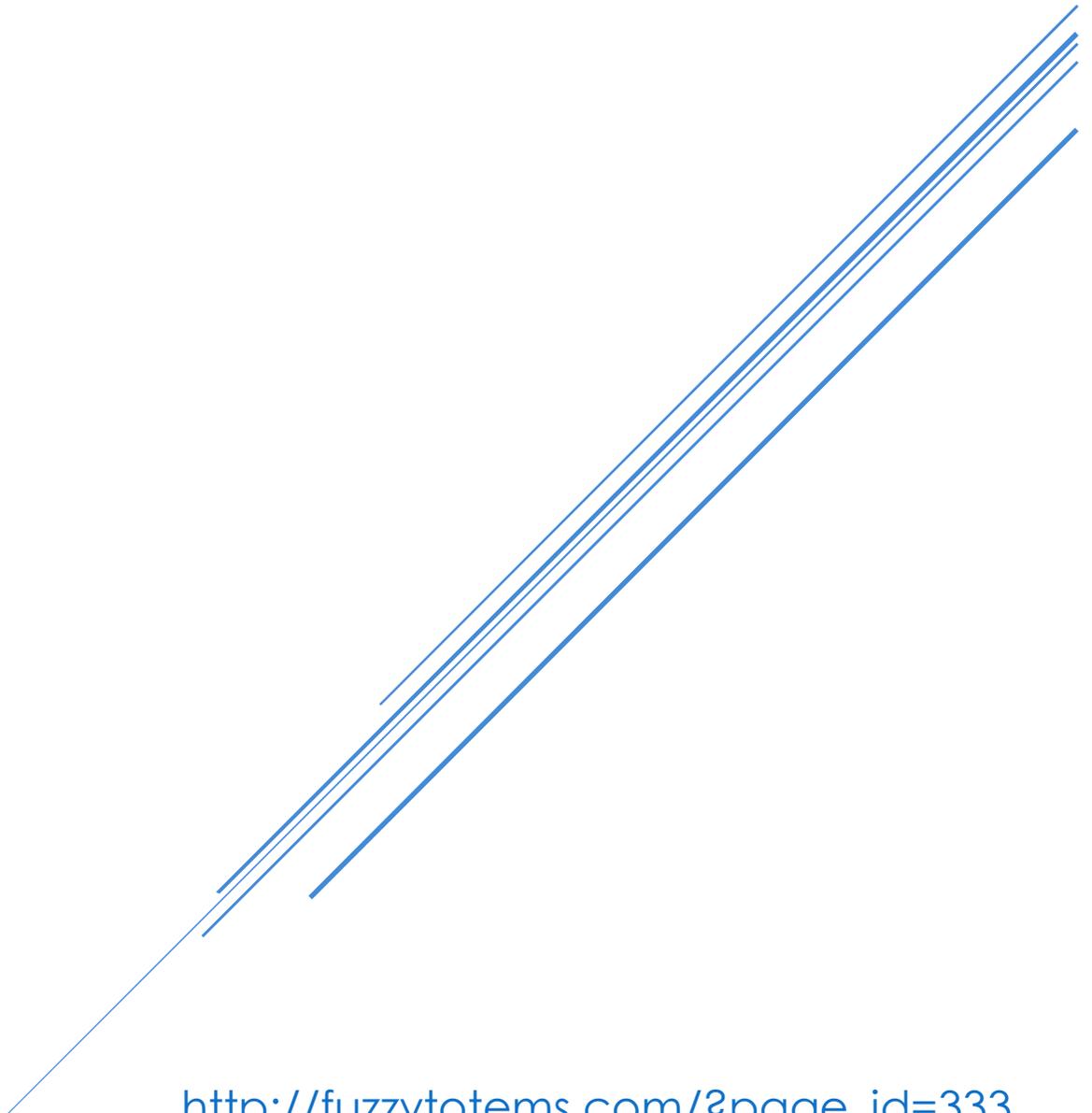


# FUZZYTOTEMS' COOLDOWN MANAGER

September 16th, 2016 build



[http://fuzzytotems.com/?page\\_id=333](http://fuzzytotems.com/?page_id=333)

Status: In Review

## TABLE OF CONTENTS

Features.....	2
The Full Package.....	2
Cooldown Submission and Ticking.....	2
High .....	3
Medium.....	3
Low .....	3
Cooldown Retrieve Information .....	3
High .....	4
Medium.....	4
Low .....	4
Cooldown Modifier .....	4
High .....	4
Medium.....	5
Low .....	5
Cooldown Dilation .....	5
High .....	6
Medium.....	6
Low .....	6
Cooldown Dilation Clamping .....	6
Global Cooldown Dilation Identifier.....	6
On-The-Fly Network Rule Modifications.....	6
Multiple Charges Recharge? .....	7
Fidelity Setting .....	7

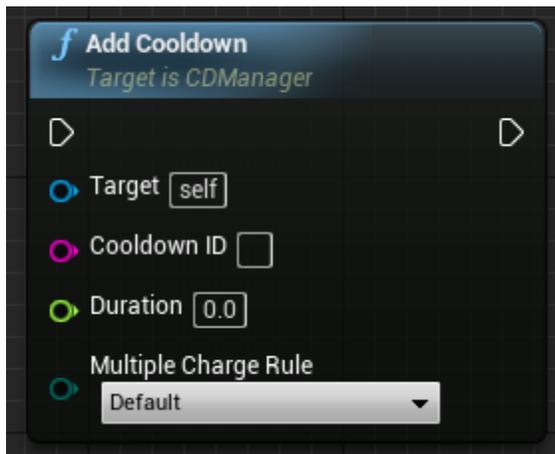
## FEATURES

### THE FULL PACKAGE

This cooldown manager offers a robust set of features to assist in managing cooldowns. It comes with three fidelity settings to help you disable excessive functionality if you don't need quite the detail of the higher end features. A chart that breaks down the fidelities is available at the end of the document.

Author's note: The term cooldown may sometimes be replaced with CD or CDs.

### COOLDOWN SUBMISSION AND TICKING



```
void AddCooldown(FString cooldownID, float duration, EChargeRechargeable multipleChargeRule);
```

This is the portion of the plugin that you submit your cooldowns to.

The cooldown ID is the ID of the spell or ability that you wish to put on CD. This is the same string you will use to retrieve CD information later. The duration is the number of seconds, so 10.5 would be 10 ½ seconds. `bAllChargesRecharge` indicates whether all charges recharge simultaneously.

For a moment, let's assume you have a Fireball spell and the default Multiple Charge Rule is "false". The spell has a 15 second cooldown and 2 charges, meaning you can cast it twice before running into the charge limit. By setting "MultipleChargeRule" to default, it will pull the default rules from the manager, which is "single" in this example, meaning the first Fireball will need to fully recharge before the second begins recharging. By setting "MultipleChargeRule" to "simultaneous", the second Fireball will begin charging even if the first Fireball is still recharging. By setting "MultipleChargeRule" to "single", the first Fireball will need to fully recharge before the second begins recharging.

As a note, the shortest charge will always be the one ticking down. So let's say in the aforementioned example, your Fireball has a varying cooldown. The first cast has a 15 second cooldown and the second cast has a 5 second cooldown. You finish the second cast while the

first cast has 11 seconds left on its cooldown. The second charge will take priority and cool down first.

---

## HIGH

High fidelity is the most CPU intensive, as it must perform individual operations on every CD.

---

## MEDIUM

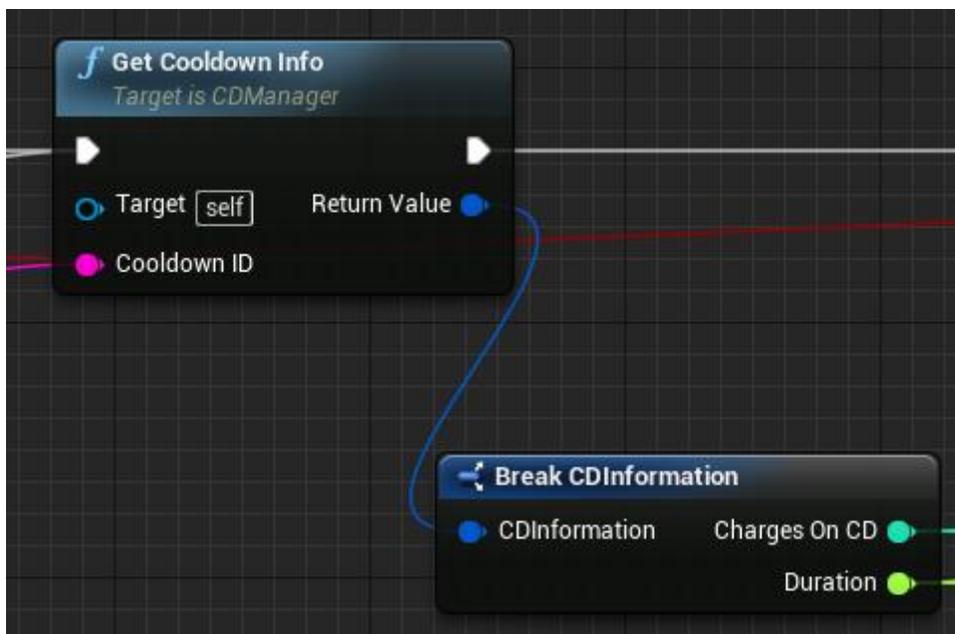
Medium fidelity runs in a fraction of the time of high fidelity, but it loses the ability to modify individual CDs as a result. Medium is able to track individual CDs.

---

## LOW

Low fidelity scales similarly to medium. It also does not track individual frames, meaning it cannot give you readouts on the time until any CD is up, let alone the next one.

## COOLDOWN RETRIEVE INFORMATION



```
FCDInformation GetCDInfo(FString cooldownID);
```

The struct fed back by this (FCDInformation) is a Blueprint friendly struct that contains two pieces of information – Charges On CD and Duration.

Charges On CD indicates how many items with the cooldown ID are currently on cooldown for the player. 0 indicates no charges on CD.

Duration indicates how many seconds are left until the next charge of the spell is off of cooldown for the player. -1 indicates no charges on CD.

---

## HIGH

High fidelity is very low on overhead for retrieval, and it provides maximum information.

---

## MEDIUM

Medium fidelity retrieves the same level of detail from the information as high fidelity with the same runtime.

---

## LOW

Low fidelity retrieves the same level of detail as medium fidelity.

## COOLDOWN MODIFIER



```
void AddCDModifier(FString cooldownID, float amountToReduce);
```

This function allows the user to submit a cooldown and a float amount to reduce by. That is to say a positive number for the float will reduce the amount of time left on the cooldown internally by that amount. A negative number will increase the amount of time left on the cooldown.

AddCDModifier simply subtracts the amount of time indicated by the float from the amount of time remaining on the cooldown specified by cooldownID. At the end of every tick, all cooldown modifiers submitted through this function are emptied, so make sure your cooldown is in memory before submitting this.

If you wish to reduce all cooldowns by a specific amount, there is a global cooldown dilation identifier mentioned later in the document (that can also be found in the Blueprints in the defaults section). By submitting that string as the first parameter, you indicate that every cooldown in the table should have its value reduced by the float provided.

**Individual cooldown modifiers only work in high fidelity. Medium fidelity only supports global cooldown modifiers. Low fidelity supports neither.**

---

## HIGH

High fidelity is the only setting that supports individual cooldown modifiers and global cooldown modifiers. If `bAllChargesRecharge` is set to true for this spell, the individual cooldown modifiers apply to all charges.

---

## MEDIUM

Medium fidelity does not support individual cooldown modifiers, but it does support global cooldown modifiers.

---

## LOW

Low fidelity does not support cooldown modifiers.

## COOLDOWN DILATION



```
Void AddDilation(FString cooldownID, float dilationAmount);
```



```
Void RemoveDiltaiion(FString cooldownID, float dilationAmount);
```

A cooldown dilation is a modification to the speed that a cooldown lapses, not a modification to the amount of time left on the cooldown. If the float provided is `1.0f`, it will double the rate that the cooldown indicated by `cooldownID` will progress. If the float provided is `0.1f`, it will increase the rate the cooldown progresses by 10%. If the float provided is `-1.f`, it will stop the cooldown. If it is under `-1.f`, the cooldown will progress in a negative direction. The dilations stack additively internally. Adding both a `-1.f` dilation and a `1.f` dilation will result in the two negating each other.

Remove Cooldown Dilation requires a matching string **\*\*AND\*\*** a matching amount.

As mentioned in the previous section, there is a global cooldown dilation identifier available in the default properties that can be used to indicate that all spells should dilate a specific amount.

**Individual cooldown dilation is only available with high fidelity. Global cooldown dilation is available in both high and medium. Cooldown dilation is not available in any capacity in low.**

---

## HIGH

High fidelity is the only setting that supports individual cooldown dilations and global cooldown dilations. If `bAllChargesRecharge` is set to true for this spell, the dilation applies to all charges.

---

## MEDIUM

Medium fidelity does not support individual cooldown dilation, but it does support global cooldown dilation.

---

## LOW

Low fidelity does not support cooldown dilation.

## COOLDOWN DILATION CLAMPING

Dilation Clamp Floor	<input type="text" value="0.0"/>
Dilation Clamp Ceiling	<input type="text" value="2000.0"/>

Inside of the cooldown manager are two default properties; Dilation Clamp Floor and Dilation Clamp Ceiling. These are used to make sure that cooldown dilation stays within a range you specify. In the given example, your cooldowns can only be reduced to 0% reduce (turning recharging off), but they can never be made negative. Similarly, they can be increased to recharge 2000x faster but no faster.

## GLOBAL COOLDOWN DILATION IDENTIFIER

Global Cooldown DID	<input type="text" value="GLOBALCOOLDOWN"/>
---------------------	---

This property is the aforementioned global cooldown dilation identifier. If you add a cooldown modifier or dilation using this string, it will apply to all cooldowns.

## ON-THE-FLY NETWORK RULE MODIFICATIONS

Broadcast To All?	<input checked="" type="checkbox"/>
-------------------	-------------------------------------

This property determines whether the information is multicast or not.

If this is set to true, the multicast version of the updates will be called. This means that all of the clients that are aware of the existence of the CD Manager will also have the cooldowns in their memory. This would primarily be useful in situations such as Heroes of the Storm or Master X Master's camp respawn timer. By setting this to true, all clients will know the timer.

By setting this to false, the only client that is updated as to the current status of the CD Manager is the owner of the CD Manager.

**If you believe your CD Manager is broken, check this setting. If it works with "Broadcast To All?" set to true, then it is likely an issue with what actor is being given ownership over the CD Manager.**

#### MULTIPLE CHARGES RECHARGE?

Multiple Charges Recharge?

This property sets the default for all cooldowns that this cooldown manager deals with.

#### FIDELITY SETTING

Fidelity

This is the fidelity setting. It is a very basic setting. It is a defaults only setting available in the CDManager. It disables core functionality in favor of performance (and vice versa).