

# Replacement Filters and Options

Nali Weapons 3 Final

## 1 – Introduction

This mod comes with an all-purpose replacement mutator, and one of its strongest points is the ability to limit or modify the addition and replacement of weapons, ammo and other kinds of inventory using filters and optional properties.

Currently, these have to be set in the *NWConfig.ini* file directly into their respective fields (see [INI NWConfig.pdf](#), [\[NWCoreVIII.NWMainReplacer\]](#) section, list properties *DefaultWeaponItems*, *DefaultPickupItems*, *WeaponItems*, *Ammoltems* and *PickupItems*), as these are meant for advanced tweaking for servers, but they can also be used for personal standalone experience and enjoyment.

## 2 – Options

The options are set in the property called *Options*, and they are formatted as *property=value*, separated by semicolons and cannot contain spaces, for example:

```
Options="ThirdPersonScale=2.5;"
```

Which means “when adding or replacing by this item, set its *ThirdPersonScale* property to 2.5”, which ultimately would lead the item to be rendered at 2.5x its original size in the third person perspective (which would be quite funny).

These properties are the ones the item has in its code or visible from the editor.

However, not all properties will work, due to engine and game limitations, so you need to either check into the item code itself to confirm if it's possible or by trial and error. But the possibility is here for the simple ones at least.

For more custom modifications, if you have at least some basic knowledge of *UnrealScript*, you can set these *Options* to execute a callback function instead so you can do whatever you want with the item after it gets replaced or added to a player's inventory.

All you have to do is to add the property C, such as:

```
Options="C=someID;"
```

And the function *SetPropertiesOptions* from the class *NWMutator* will be called as:

```
SetPropertiesOptions("someID", A);
```

Where *A* is the item to handle, and the other argument is the ID passed to identify where it was called from. The *C* property can have multiple IDs for multiple function calls.

For more details on this callback function, see [NW3 Plugins.pdf](#).

### 3 – Replacement Filters

The replacement filters are set in the property called *Filters*, and they are formatted as *property=value*, separated by semicolons and cannot contain spaces, for example:

*Filters="game=DM;"*

Which means “add or replace by this item only if the current gametype is a normal Death Match”.

For instance, some nuclear weapons from this mod have some map size filters setup by default so such nukes do not get used in small maps like DM-Deck16[] for example, but still may be used in big open maps like CTF-Face.

The list of possible filters are shown and explained below:

Property	Description
<i>team=? , ? , ... ;</i>	List of the only player teams the item <b>can</b> be added to or replaced from. Valid values are <i>RED</i> , <i>BLUE</i> , <i>GREEN</i> , <i>YELLOW</i> , <i>NONE</i> or any of the numbers <i>0</i> , <i>1</i> , <i>2</i> , <i>3</i> and <i>255</i> .
<i>noteam=? , ? , ... ;</i>	List of the only player teams the item <b>cannot</b> be added to or replaced from. Valid values are <i>RED</i> , <i>BLUE</i> , <i>GREEN</i> , <i>YELLOW</i> , <i>NONE</i> or any of the numbers <i>0</i> , <i>1</i> , <i>2</i> , <i>3</i> and <i>255</i> .
<i>game=? , ? , ... ;</i>	List of the only gametypes the item <b>can</b> be added to or replaced from. Valid values are: <i>DM</i> – For Death Match <i>TDM</i> – For Team Death Match <i>CTF</i> – For Capture The Flag <i>AS</i> – For Assault <i>DOM</i> – For Domination <i>LMS</i> – For Last Man Standing <i>MH</i> – For Monster Hunt  Other gametypes not listed here, can still be added by their package and class, for example: <i>Botpack.DeathMatchPlus</i>
<i>nogame=? , ? , ... ;</i>	List of the only gametypes the item <b>cannot</b> be added to or replaced from. Valid values are: <i>DM</i> – For Death Match <i>TDM</i> – For Team Death Match <i>CTF</i> – For Capture The Flag <i>AS</i> – For Assault <i>DOM</i> – For Domination <i>LMS</i> – For Last Man Standing <i>MH</i> – For Monster Hunt  Other gametypes not listed here, can still be added by their package and class, for example: <i>Botpack.DeathMatchPlus</i>
<i>minH=? ;</i>	Map min allowed height (in unreal units).
<i>maxH=? ;</i>	Map max allowed height (in unreal units).
<i>minS=? ;</i>	Map min allowed size (in unreal units).
<i>maxS=? ;</i>	Map max allowed size (in unreal units).
<i>minP=? ;</i>	Map min allowed number of player spawns.
<i>maxP=? ;</i>	Map max allowed number of player spawns.
<i>minLqZ=? ;</i>	Map min allowed number of liquid zones (water + lava + slime).
<i>maxLqZ=? ;</i>	Map max allowed number of liquid zones (water + lava + slime).

Property	Description
<i>minDmgZ=?;</i>	Map min allowed number of damage-able zones.
<i>maxDmgZ=?;</i>	Map max allowed number of damage-able zones.
<i>minWtZ=?;</i>	Map min allowed number of water-only zones.
<i>maxWtZ=?;</i>	Map max allowed number of water-only zones.
<i>minLvZ=?;</i>	Map min allowed number of lava-only zones.
<i>maxLvZ=?;</i>	Map max allowed number of lava-only zones.
<i>minSlmZ=?;</i>	Map min allowed number of slime-only zones.
<i>maxSlmZ=?;</i>	Map max allowed number of slime-only zones.
<i>C=?,?,...;</i>	<p>List of custom function call IDs to decide if the item should be added/replaced or not, or to modify the filtering parameters.  The function called is <i>checkFilters</i> from the class <i>NWMutator</i> (more details in <a href="#"><i>NW3 Plugins.pdf</i></a>).</p> <p>Example:</p> <p style="margin-left: 40px;"><i>C=myID;</i></p> <p style="margin-left: 40px;">will call the <i>NWMutator</i> class function as:</p> <p style="margin-left: 40px;"><i>checkFilters("myID", Filters, Other)</i></p> <p style="margin-left: 40px;">being "<i>myID</i>" the ID value defined in <i>C</i>, <i>Filters</i> the current <i>Filters</i> set for this item replacement/addition and <i>Other</i> being an optional <i>Pawn</i>.</p>