**Protection Receivers Parameter**

This is a database of sorts for the enemies that will receive protection from other enemies in that same battle. We’ll refer to them here onwards as ***Receivers***. The Receivers parameter shall be referred to as the ***Receiver Database***. You have to register them here for the functionality to work.

Receivers have these fields:

**Name:** As it sounds a name. Makes it easier to tell them apart in the database.

**Enemy DB Entry:** Also as it sounds. When an enemy with that entry appears in battle, it is treated as a Receiver with the properties defined in the current entry.

To avoid glitchiness, make sure that no two Receivers have the same Enemy DB Entry.

**Protectors:** List of Protectors that can grant protection to this Receiver. Only those copy-pasted to this list from the Protectors parameter should work.

**Damage Mitigation:** Percent of damage this Receiver avoids while their protection is in effect. Defaults to 100, meaning that they simply take no damage.

Tip: If you want the receiver to be healed while being protected, set this to a value above 100. For example, setting it to 120 means that it heals for 20% of the damage it’d normally take. Setting it ti 180 makes it heal for 80% of said damage. And so on.

If you want the Receiver to take more damage (as a sort of anti-protection effect), you’d set this to a negative value. Setting it to -20 means it takes 20% extra, setting it to -80 means it takes 80% extra, and so on.

**Transfers Damage:** Whether or not it transfers any of the damage it’d take to the Protectors protecting this. By default, the split is even between protectors.

**Damage Transferrable:** How much of the blocked damage can be transferred to the Protectors.

**Protectors Needed:** How many of this Receiver’s Protectors need to be alive for it to get the protection effect. Setting this to 0 or below makes this Receiver get the protection all by itself.

**Protectors Parameter**

Like the Protection Receivers parameter, but for the enemies that provide protection for others. We’ll refer to them from here onwards as ***Protectors***. The Protectors parameter in the plugin’s Plugin Manager entry will be referred to as the ***Protectors Database***. You have to register them here for them to be able to do any protecting.

Protectors have these fields:

**Name:** As it sounds. There to help tell the Protectors apart at a glance.

**Enemy DB Entry:** Just like with Receivers. Also avoid having any two protectors sharing the same DB entry.

**Transfer Damage Mitigation:** Percentage of how much of the damage bounced from the Receiver to this Protector is mitigated.

Same logic as the Receivers’ Damage Mitigation applies; Use a value higher than 100 for this Protector to take healing, use a negative value for this Protector to take extra damage.

***Setting up Receivers and Protectors***

It is assumed you’re following these instructions on a fresh new RMMV project. And that you know the basics of RMMV.

Let’s start with Receivers. Go see the list of Receivers from the Receivers parameter. Double-click an empty entry. A new window should pop up, letting you put in the Receiver’s data.

For this example, let’s put this in:

Name: Bat

Enemy DB ID: Select the Bat from the dropdown (should be ID number 1)

Damage Mitigation: 100

Transfers Damage: true

Transfers Mitigated Damage: true

Damage Transferrable: 100

Protectors Needed: 1

Click OK. Now, let’s set up a Protector for this Bat.

Go to the list of Protectors from the Protectors parameter. Double-click an empty entry. Put in this data into the new window:

Name: Slime

Enemy DB ID: Select the Slime from the dropdown (should be ID number 2)

Transfer Damage Mitigation: 0

Click OK. Highlight this new Protector entry, and do Ctrl-C to copy it. Click OK, then go to the Receiver entry we made earlier. Double-click its Protectors list, and copy-paste the Slime into it. Click OK as many times as you need to to get out of the Plugin Manager.

Set up a Troop with a Bat and at least 2 Slimes. In the first map in the game, set an NPC to trigger a battle with that troop when you interact with it.

In the battle, have your whole party attack the bat. If things are working as they should, the Bat should be taking 0 damage, while the Slimes take a low, equal amount of damage.

Now, kill both Slimes. The Bat should no longer be invulnerable. Kill the Bat.

Now, that proves the base functionality works. But, let’s make this more fun…

In the Enemies database, copy the Slime entry into Slots 3 and 4. Change slot 3’s graphic to the Orc’s and the name to Orc. Change slot 4’s graphic to the Scorpion’s and the name to Scorpion. Apply the changes, and go back to the plugin’s entry in the Plugin Manager.

Make two copies of the Slime’s Protectors Database entry. Set these fields on one of them:

Name: Scorpion

Enemy DB ID: The Scorpion’s entry

Transfer Damage Mitigation: 200

Set these fields on the other copy:

Name: Orc

Enemy DB ID: the Orc’s entry

Transfer Damage Mitigation: -100

Copy paste those new Protector entries into the Bat’s Protectors list. Set up the troop from before to include the Orc and Scorpion.

Fight that troop again, starting off by attacking the Bat a bunch. Look closely at the damage numbers that pop up; the Scorpion should be getting healed for the same amount that the Slime is taking damage. The Orc should be taking double the damage that the Slime is taking.

The reason this happens is because when you set Transfer Damage Mitigation to a value higher than 100, instead of getting damage transferred to it, the Protector heals for a percentage of it. Since we set it to 200 for the Scorpion, it heals for all the damage that’d normally be transferred.

When you set it to a negative number, the damage increases. So since we set it to -100 for the Orc, the damage transferred to it got doubled.

Now, let’s try something else. Set the Damage Mitigation of the Bat’s Receiver entry to 200. Try fighting the troop again, and you’ll see that the only thing happening is the Bat getting healed for all the damage it’d normally take. Since it’s not getting damaged, nothing gets transferred to its Protectors.

So now, that should show you most everything this plugin is capable of. I’m sure that you can think up new uses by considering just how the damage and stuff goes around.

Also, there are other interesting effects you can get through this plugin that I didn’t mention. Try discovering them yourself😉