JonnyFive's WeatherFX mod:

Section 1: About WeatherFX

1.1 Introduction

I came up with the idea for this when I was doing work on my main project, "Cold as Hell," which involves a pretty good amount of snow. When I was creating the mod, I originally used a series of MapSpots with TIDs 900 – 905 to spawn snow objects with randomized movement parameters to give the illusion of drifting snowfall. This worked pretty well most of the time, except that it would lag a slow computer to hell and each of the series of things would spawn its snow object at the same time, which looked crappy in small sectors.

At the same time, CaH had a lot of big sectors with snow in it, so having some way to limit the weather spawn to the area around the player would help performance a lot.

I've seen other weather effects mods that use different methods. One of the best was the method used by solarsnowfall's rain which spawned rain items around a central MapSpot. I like this approach, but I wanted to create something slightly more robust.

1.2 Before You Start

This all assumes you have a functional knowledge of basic ACS and DECORATE editing. If you want to change one of the more advanced parameters, you should have a decent knowledge of ACS and how to use the acc.exe compiler to develop a library file.

1.3 How Do I Run This?

The main zip file includes the following .pk3 file:

j5weatherFX.pk3

Which can be simply dragged over **zdoom.exe** or **gzdoom.exe**, or run with the **–file** command.

It has been tested with ZDoom 2.2.0 as well as GZDoom 1.1.04. It should work with the latest SVN versions of both of these programs as well.

IMPORTANT NOTE: Although only SlumpED can current read .pk3 files natively, remember that they are only renamed .zip files. Rename it to j5weatherFX.zip and you'll be able to view the files using Windows Explorer or your favorite ZIP program!

Section 2: Using WeatherFX

2.1 Configuration

In order to use the weatherFX mod, you'll need to set up your mod with the required DECORATE objects as well as the right scripts. The main weatherFX script will handle all the spawn code, so your job is actually pretty easy.

2.1 DECORATE Setup

The first thing you'll need to do is configure the weather spawners you want to use in your DECORATE file. Feel free to modify or use one of the included spawners, but basically you'll need these things in the DECORATE file:

-One **WeatherSpot** object (editor thing #*30000* by default)

-The **Action_WeatherToggle** and **Flag_NoWeather** objects

-A spawner object for each type of weather effect you want (e.g. a "SnowSpawner" object for snow or a "RainSpawner" object for rain. These don't need thing numbers)

-A particle object for each weather effect (e.g. a "snow" object or a "raindrop" object)

The droplet object's properties will basically define how your weather effect will look when it's spawned, and the spawner will define how it moves when it's spawned.

EXAMPLE SPAWNER:

```
//Basic snow spawner
ACTOR SnowSpawner
{
+Missile
+NoClip
+NoGravity
+NoBlockMap
RenderStyle Translucent
Alpha 0
States
{
Spawn:
RAIN A 1
RAIN A 1 A_SpawnItemEx ("SnowFlake", 0, 0, 0, random(0,5), random(0,5), random(0,2), random(0,255), 0)
Stop
}

}
```

Precipitation object

x,y,z momentum

facing direction

The first three random values are the x, y, and z movement, and the last is the angle. You can adjust these for different speeds of projectile, as well as having rain/snow that moves in one direction only.

EXAMPLE PRECIPITATION OBJECT

```
//Light, drifting snow flake
ACTOR SnowFlake
{

        +NOBLOCKMAP
        +DROPOFF
        +MISSILE
        +NOTELEPORT
        +DontSplash
        RenderStyle Translucent
        Alpha 0.9
        gravity 0.02
        scale 0.5
        States
        {
        Spawn:
                SNOW A 1
                Loop
        Death:
                TNT1 A 1
                Stop
        }

}
```

The object's appearance and gravity are determined here.

2.2 KEYCONF Setup (optional)

If you want the player to be able to toggle weatherFX, include the information from the included **KEYCONF** file in your WAD as well. The player will automatically be given the **Action_WeatherToggle** item by the main script, but they need a key to actually use it.

2.4 Main Script

To use the main script (WEATHER), you'll need to set it up as a library, which means sticking it in your WAD between the A_START and A_END markers, or putting it in a .pk3 file in the "acs/" directory.

IMPORTANT NOTE: The mod *requires* that the player be assigned a TID for many features. This is set to 255 by default, but can be changed by editing **weather.acs** and changing the **PlayerTID** constant to whatever number you like. For multiplayer compatibility, you'll need to make this non-constant and assign each player their own TID.

2.5 Your Level

You will need to put down **WeatherSpot** (thing *#30000*) items to bound the areas you want weather in, one in the upper left corner and one in the lower right corner of the area. The z-height of the upper left **WeatherSpot** will determine where the weather effects are spawned from.

Keep in mind that the weather items will collide with map geometry and will not spawn in any area with a ceiling lower than the z-height of the upper left spot, so you can cover an area with a building that is lower than your outside sky without worrying about having random snow flakes or rain drops spawning inside.

You need to assign each **WeatherSpot** a TID.

NOTE: Technically you don't need to use **WeatherSpot** objects (you could use **MapSpot**s), but I include the **WeatherSpot** because it helps to keep the weatherFX stuff separate from the rest of your map.

2.6 Your Level's ACS

Once you put the main library file in your resource WAD, just stick **#import "weather.acs"** at the top of your level's ACS script (make sure you put a copy of **weather.acs** in the directory your acc.exe compiler is – this can be extracted from the .pk3 file).

In order to get the weatherFX effects working, you'll need to set up several variables in your main script:

VARIABLES:

WFX_WeatherZones	The number of **WeatherSpot** pairs in your level. Each weather zone requires *two* (upper-left and lower-right).

NOTE: Use one set of each of these variables for each weather zone. Start with *i* = 0 for one weather zone and increment once for each additional weather zone. You can have up to 101 weather zones, which should be plenty for most levels.

WFX_Upper_TID[*i*]	The TID of the weather zone's upper-left **WeatherSpot**.
WFX_Lower_TID[*i*]	The TID of the weather zone's lower-right **WeatherSpot**.
WFX_Type[*i*]	The DECORATE spawner type (use a string literal like "RainSpawner") for the weather zone.
WFX_Density[*i*]	The density in **particles/delay time** for the weather zone. Adjust this to control the level of precipitation. Keep in mind that all particles are uniformly and randomly distributed in the spawn zone, and that if **WFX_IgnorePlayer[i]** is set to 0, the spawn zone will be no bigger than a square with each side **WFX_Distance[i]** units from the player (assuming this square is entirely within the weather zone itself).
WFX_Distance[*i*]	The distance in map units from the player that weather effects will spawn.
WFX_IgnorePlayer[*i*]	1 = The effects will always spawn regardless of where the player is. 0 = The player position will be taken into account to limit spawning.
WFX_Active[*i*]	1 = This weather zone is active. 0 = This weather zone is inactive.

These will determine the properties of each weather area (which is bounded by the two WeatherSpot objects).

The last thing you need to do is start the weather spawn code using script 950 in your map:

```
ACS_Execute(950, 0, Delay Time, 0, 0);
```

Delay time is the time in tics between particle spawns. I suggest keeping this low, around 10 – 15 tics.

2.7 Putting it all together

Compile your map's ACS, save the map, and load it up along with the resource WAD where you put the weatherFX DECORATE and library file. You can simply load the weatherFX example file as your resource if you want, but keep in mind that MAP01 is a sample map in the resource file, so load your map *after* the resource file to make sure your level shows up.

Assuming everything is set up properly, you should see your weather effects in the area you bounded by the WeatherSpots.

2.8 Compatibility Issues

This mod uses the following resources (script #'s, etc.). If you project also uses these, you will need to either change your project or change this mod to prevent conflicts:

Resources Used by weatherFX

Script Numbers:
940
941
950

Editor Thing Numbers:
30000

Section 3: Using These Resources in your Project

3.1 General Terms of Use

You are completely free to use any of these resources in your own projects. The included graphics, rain sound effect, ACS/DECORATE work, and example map were all created solely by the author (Jon Washburn). You have the permission of the author to use any of these resources in any way you see fit, provided you give the original author (Jon Washburn) credit somewhere in your project's documentation.

3.2 Modifications of this Resource

You are free to modify this project and its included resources in any way you see fit, provided you give the original author (Jon Washburn) some kind of credit in somewhere in your modification's documentation.

3.3 Author Contact Information

Although I don't care if you contact me first if you want to use this mod, I'm always happy to hear about people's work. You can contact me through the ZDoom forums.