**Basic Setup**

1. **Create a backup of your project.** I’m 96% sure nothing in my plugin will break your project, but it never hurts to back it up before installing a new plugin.
2. Install the TDW Berry Planting Improvements plugin by copying it into your project’s Plugins folder.
3. Move the files in the Graphics folder included into your project’s Plugins folder.
4. If you want to use the new mulch items, add the items listed in the PBS > items\_newmulch.txt to your PBS > items.txt file (or copy the file to your PBS folder for v21+).
5. In order to use most of the features for this plugin, go through your game and find all your BerryPlant events.
   1. In order for them to appear on the Town Map, you'll need to add "map(regionID,x,y)" to the event name. If you forget one, I've added a passive warning that will appear in the console when you open the Town Map that will tell you what map and which event doesn't have this included to help you out.
      1. regionID => the region that corresponds with the map this berry plant should appear on
      2. x => the x position on the map this berry plant should have
      3. y => the y position on the map this berry plant should have
      4. Example: 
   2. If using the Berry Core and Dex plugin, in order for them to have a Zone, you'll need to add "berryzone(name)" to the event name.
      1. name => the name of the Zone to apply to the spot. These can be any string, but must match the same string you use in the berry\_data file
      2. Example: 
6. Explore the settings in Plugins > TDW Planting Improvements > 000\_Settings.

**Features Setup**

**Preferred Weather** (Requires Berry Core and Dex plugin)

1. Add a "PreferredWeather" line to each berry in berry\_data file in PBS. Then list out any weather types you want to be included as a Preferred Weather for that berry. For example, you can add the following line to make it prefer Rain or Snow: PreferredWeather = Rain, Snow
2. Add icons for any of your custom weathers in the Graphics > Pictures > Plant Icons folder if you wish for the weather data to appear in the Berrydex.
3. Change what benefits will apply to berries exposed to preferred weather in 000\_Settings.

**Preferred/Unpreferred Zones** (Requires Berry Core and Dex plugin)

1. Add a "PreferredZones" and/or "UnpreferredZones" line to each berry in berry\_data file in PBS. Then list out the names of Zones you wish to apply to the berry (See the Setup section below to see how to set up Zones). For example, you can add the following line to make it prefer a Field or City zone: PreferredZones = Field, City
2. Add icons for any of your Zones in the Graphics > Pictures> Plant Icons folder if you wish for the zone data to appear in the Berrydex.
3. Change what benefits (or penalties) will apply to berries planted in these zones in 000\_Settings.

**Custom Pest Encounter Tables**

1. If you want to set up custom BerryPlantPest encounter tables for different maps, [you'll do so the same way you'd add any new encounter type for a map](https://essentialsengine.miraheze.org/wiki/Wild_encounters).
   1. You can find the different encounter types defined by default in 003\_PestsEncounters. Ones that have a color at the end (ex: BerryPlantPestRed) are the encounters for a berry plant that yields a berry of that color, which requires the Berry Core and Dex plugin.
2. You can use the BERRY\_PEST\_DEFAULT\_ENCOUNTERS table set in 000\_Settings if you want the same encounters throughout the entire game, or for most places.

**Watering Features - Animation**

1. If you’d like to show watering animations when watering a berry plant, make sure to set BERRY\_WATERING\_SPRITES in 000\_Settings to include file names for each of the trainers in your game. You must also include the appropriate images in the Graphics > Characters folder (I suggest [Ulithium\_Dragon](https://reliccastle.com/resources/342/)’s graphics)
2. You can also use graphics specific to different watering can items. Just add in graphics with the same names as the ones added in Step 1, but with “\_<CAN ID>” appended to the file name, where <CAN ID> is the symbol of the watering can item.
   1. For instance, if your base graphic is called "girl\_watering" and you want to include a unique graphic while using the Sprayduck, add a graphic called “girl\_watering\_SPRAYDUCK”

**Watering Features – Water Can Fill Levels**

1. If you’d like to make it so watering cans have a limited supply of water and need to be filled up once empty, set BERRY\_WATERING\_MUST\_FILL in 000\_Settings to true.
2. Make sure the player has a way to fill up watering cans, such as through the pbFillWateringCans command (see below).

If you are using the Caruban’s BerryPots plugin, and you’d like the fill levels to be respected when watering berries in the pots, do the following:

1. In the BerryPots.rb file of the BerryPots plugin, find the line # Water the growing plant
2. After that line, replace the following lines…

|  |
| --- |
| return if !@item  return if !pbConfirmMessage(\_INTL("Want to sprinkle some water with the {1}?",  GameData::Item.get(@item).name))  berry\_plant.water  pbWateringAnim(idx)  pbUpdateSoilSprite(berry\_plant,idx)  pbMessage(\_INTL("{1} watered the plant.\\wtnp[40]", $player.name))  if Settings::NEW\_BERRY\_PLANTS  pbMessage(\_INTL("There! All happy!"))  else  pbMessage(\_INTL("The plant seemed to be delighted."))  end |

With the line pbBerryPlantWater(berry\_plant,idx)

**Documentation**

**Functions**

pbMulchComposter or pbComposter

* Description: Opens a composter to turn berries into mulch. Requires Berry Core and Dex plugin. Define mulch recipes in 000\_Settings.

pbAutoPlantBerry(Events, MapID)

* Description: Autoplant berries. Requires Berry Core and Dex plugin. If the events given are not BerryPlants, don’t have BerryPlantData set up yet, or are already planted, they will not be planted again. Any berries that aren’t successfully planted are returned to the player.
* Arguments:
  + Events
    - Required. An array of Event IDs that are set up to be BerryPlant events. These are where the berries will attempt to be planted.
  + MapID
    - Optional. Set which Map ID the events you want to plant in are in. If not set, it will default to the map the player is currently on.
* Examples:
  + pbAutoPlantBerry([5,6,7,8])=> Will ask the player to pick some berries from their bag, then attempt to plant them in events 5, 6, 7, 8 on the current map.
  + pbAutoPlantBerry([10,12],16)=> Will ask the player to pick some berries from their bag, then attempt to plant them in events 10 and 12 on Map 16.

pbFillWateringCans(VariableID1, VariableID2)

* Description: Fills all your Watering Cans if BERRY\_WATERING\_MUST\_FILL is true.
* Arguments:
  + VariableID1
    - Optional. The variable ID to save the number of watering can items the player currently has that were filled. If not set, it will save the value to Variable 1.
  + VariableID2
    - Optional. The variable ID to save the name of the watering can the player has, if they only have one. If not set, it will save the name of the item to Variable 3.
* Examples:
  + pbFillWateringCans=> Fills all watering cans the player has. Saves the number of watering cans filled to Variable 1. If the player has only 1 watering can, saves the name of that watering can to Variable 3.
  + pbFillWateringCans(2,4)=> Fills all watering cans the player has. Saves the number of watering cans filled to Variable 2. If the player has only 1 watering can, saves the name of that watering can to Variable 4.
* Uses:
  + On its own, using this function will fill the watering cans but give no feedback to the player. A good way to use this is to set up an event (like a well) that will fill up the watering cans while giving feedback to the player about what was done. For example, if you set up an event page like the following, using the variables set when using this function, you can give flavor text to the player:
  + 