



Secret Bases Remade

User Manual

By Vendily

Table of Contents

Features.....	3
Installing the Script.....	4
Step 0: Installing the Hotfixes.....	4
Step 1: Dropping in the Files.....	4
Step 2: Configuring the Maps.....	5
Importing Maps.....	5
Creating the Dummy Map.....	5
Creating the Base Items Map.....	5
Creating the Map Templates.....	5
Double Check Your Work.....	6
Step 3: Editing the Settings.....	6
Step 4: Configuring GameData::SecretBaseTemplate.....	8
Step 5: Defining Secret Bases.....	10
Step 6: Defining Decorations.....	11
Step 7: Compile and Test.....	12
Useful Variables and Methods.....	12
\$secret_bag.....	12
\$PokemonGlobal.secret_base_rank.....	13
\$stats.moved_secret_base_count.....	13
pbSecretBaseMart(stock, speech=nil, cansell=false).....	13
pbReceiveDecoration(decoration).....	13
pbGetPlayerBaseLocation(mapname=-1,mapid=-1).....	13
Record Mixing and the Cable Club.....	14
Secret Base Skills.....	14

Features

- Creation and Decoration of Secret Bases, like in RSE

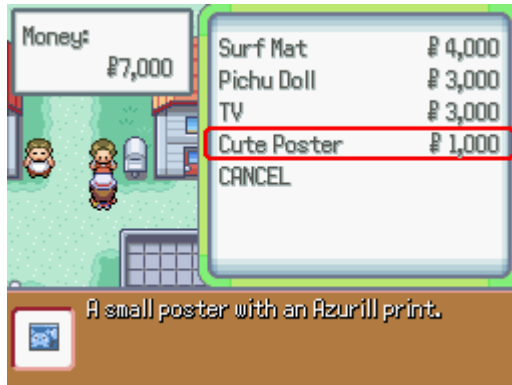


- User configurable Map Templates, Bases and Decorations

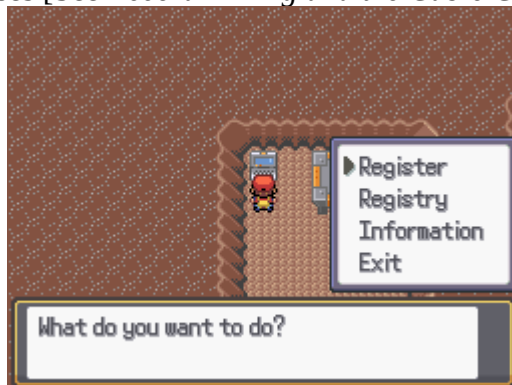
```
GameData::SecretBaseTemplate.register({
  :id          => :CaveBrown3,
  :map_id      => 78,
  :type        => :cave,
  :door_location => [19,16],
  :pc_location  => [21,9],
  :map_borders  => [8,6,22,16]
})

[CaveBlueRoute2]
MapTemplate = CaveBlue4
Location = 21,17,69
[TreeRoute1]
MapTemplate = Tree3
Location = 5,21,11
[ShrubRoute4]
MapTemplate = Shrub2
Location = 39,34,12
```

- Gifting, Buying, and Selling Decorations



- Mix Records and Visit Other Player's Bases [See *Record Mixing and the Cable Club*]



Installing the Script

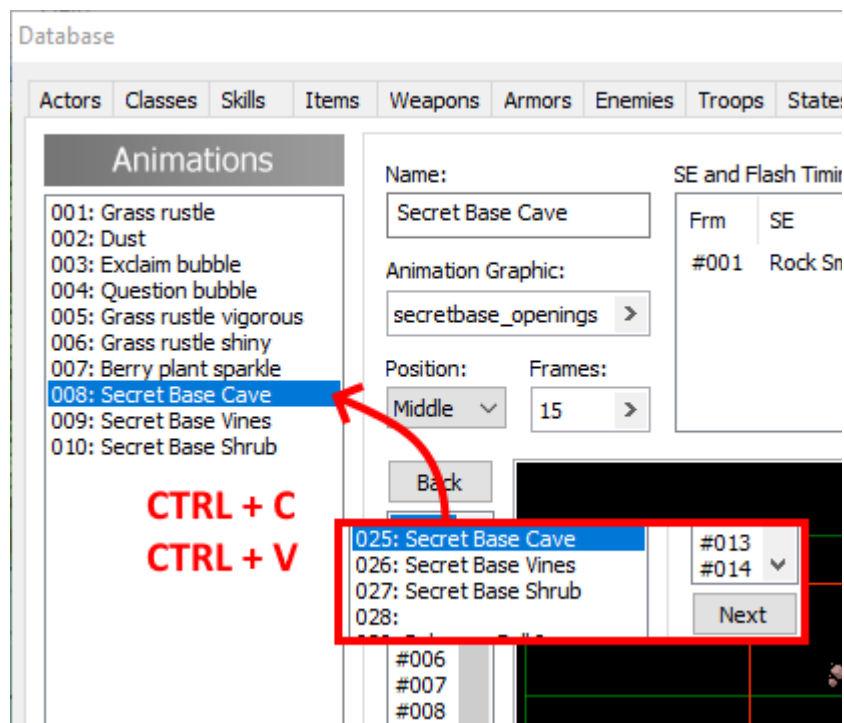
Secret Bases Remade has a decent amount of set up before the script is ready to be used, but once completed, adding new Templates, Bases, and Decorations is very quick.

Step 0: Installing the Hotfixes

This script requires them, so you have to have them installed as well.

Step 1: Dropping in the Files

- ❗ If you have made changes to your `Tilesets.rxdata`, or `Animations.rxdata`, you must copy these files into a new project and open both the project you want to install the script into, and the new project to copy them over in the editor.



Demonstration of copying Database entries.

You can quickly drop in the contents of the folders in this download into the root folder of your project, which will automatically sort into their respective folders. This will install this Plugin, the Graphics, and some of the Data files. The maps are not named correctly to avoid overwriting the ones in your project, so you would have to manually rename and import them if you wish to use them.

Step 2: Configuring the Maps

Importing Maps

You need to rename the Map rxdata files, to import them into your project. An easy way to import a map is to create a map in RMXP, remember the ID of that map, save and close RMXP, and rename the map you want to import to MapXXX.rxdata, where XXX is the ID of the map we just created, overwriting the newly created map.

Creating the Dummy Map

You need to create a blank dummy map for the script. This is the map that is actually used for the purpose of secret bases, and can be completely empty, as the map data is overwritten. To clarify, **only the map's data is overwritten**. The name, tileset, BGM, or BGS given to the dummy map is kept, as well as any map metadata for it.

This map must be set to the Secret Base Tileset.

Creating the Base Items Map

The download included a SecretBaseItemsMap.rxdata in the Data folder. You can import this file if you wish. Otherwise, you can create a new map. The only necessary contents are the assorted events that are used by decorations.



Events should only apply animations to "This Event" or "Player".
Self Switches are not properly saved (as event IDs change on every map reload),
so only Temp Switches should be used in events.

This map must be set to the Secret Base Tileset.

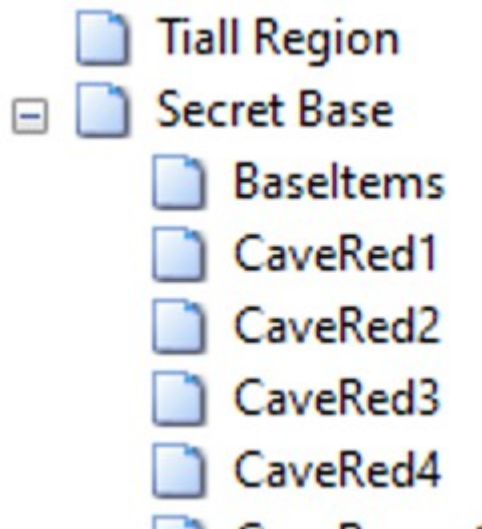
Creating the Map Templates

The download includes optional Map Templates, based off the secret bases from Pokemon Emerald. Each is named after its respective GameData::SecretBaseTemplate ID. Otherwise, you can create a new map, and design the layout of a map to your heart's content. **Holes must be placed on Layer 2.**

These maps must be set to the Secret Base Tileset.

Double Check Your Work

When completed, you should have a layout similar to this, though you are free to organize it to whichever way feels best to you.



A completed maps setup.

Step 3: Editing the Settings

In the folder for the plugin, open `001_Settings.rb` in a text editor. There are a number of constants here with comments.

! There are **three settings that MUST be adjusted** when setting up the script. They are `SECRET_BASE_MAP`, `SECRET_BASE_DECOR_MAP`, and `SECRET_BASE_TILESET`. You must verify that these values are correct, otherwise the script will act in unintended ways.

Below is a longer explanation of each.

Constant Name	Explanation
<code>SECRET_BASE_MAP</code>	The Map ID of the Dummy Map created in Step 2. Do not zero pad this ID (56 not 056).
<code>SECRET_BASE_DECOR_MAP</code>	The Map ID of the Base Items Map you created or imported in Step 2. Do not zero pad this ID (56 not 056).
<code>SECRET_BASE_TILESET</code>	The Tileset ID of the Secret Base Tileset imported in Step 1. Do not zero pad this ID (56 not 056).
<code>SECRET_BASE_MOVE_NEEDED</code>	The Move Symbol needed to create a secret base. Creates <code>HiddenMoveHandlers</code> for this move.

Constant Name	Explanation
SECRET_BASE_MAX_DECORATIONS	<p>The maximum amount of decorations that can be placed in a base.</p> <p>This can not be uncapped, the player will always be limited in the number of decorations they can place.</p>
SECRET_BASE_DECOR_ANYWHERE	<p>Allows the placement of Decor-type items anywhere that counts as floor as well as mats and desks (true) or if they can <i>only</i> be placed on mats and desks (false).</p> <p>To clarify, valid tiles for placing Decor when false is tiles with the SECRET_BASE_DECOR_FLOOR_TAG.</p>
SECRET_BASE_PC_FILENAME	<p>The filename in the Characters folder for the Secret Base PC.</p> <ul style="list-style-type: none"> Facing Down is Off. Facing Left is On.
SECRET_BASE_MESSAGES_ANIM	<p>A hash of Secret Base Entrance Data when interacting with a closed base.</p> <pre>:type => ["On Interact", "On Opening", Animation ID, Appear at Midpoint]</pre> <p>:type is the type set in GameData::SecretBaseTemplate "On Interact" is a string shown when the player inspects the base location. "On Opening" is a string shown after the base is opened. Animation ID is the ID of the animation that plays when the base opens. Should be 15 frames. Appear at Midpoint is a boolean, if it should show the base switching to the open form halfway through the animation (true) or at the end (false).</p>
SECRET_BASE_HOLES	<p>The Tile ID and width and height of the various holes in SECRET_BASE_TILESET.</p> <p>See <i>Step 6: Defining Decorations</i> for how to find the Tile ID.</p> <p>Do not add 384 to the Tile ID to compensate for autotiles.</p>
SECRET_BASE_GROUND_DECOR_TAG	<p>The TerrainTag Symbol for Layer 1 decorations in the base, like the rocks and bushes.</p> <p>You only need to give this to tiles that are on Layer 1, Tiles in Layers 2 and 3 already block placements.</p>
SECRET_BASE_WALL_TAG	<p>The TerrainTag Symbol for valid locations for Wall-type decorations. Wall-type decorations can only be placed if all of their tiles have this terrain tag.</p>

Constant Name	Explanation
SECRET_BASE_DECOR_FLOOR_TAG	The TerrainTag Symbol for valid locations for Decor-type decorations. Decor-type decorations can be placed on top of other decorations if those tiles have this terrain tag.
def self.secret_bag_pocket_names	Returns an array of pocket names for the SecretBag. Also defines the number of pockets.
SECRET_BAG_MAX_POCKET_SIZE	The number of slots in each pocket. Set to -1 to uncap the pocket, allowing for infinite slots.
SECRET_BASE_MAX_SAVED_BASES	The maximum number of bases that can be saved. Set to -1 to uncap, allowing a base to be saved into every Secret Base defined.
SECRET_BASE_MAX_REGISTERED_BASES	The maximum number of bases that can be registered, which prevents them from being overwritten while record mixing. Set to -1 to uncap, which allows every saved base to be registered.
SECRET_SKILLS_TRAINER_TYPE	A hash of Secret Skills for each Trainer Type. Each entry is a TrainerType symbol, and an array of GameData::SecretBaseSkill IDs. Up to (\$PokemonGlobal.secret_base_rank + 1) skills will be shown to the player when interacting with a Base Owner. See <i>Secret Base Skills</i> for more information.

Also in the Settings is some MessageType constants and GameData::TerrainTags used by the script. You only need to change them if they overlap with other scripts. If you change the Terrain Tags, you have to update the Tileset, of course.

Step 4: Configuring GameData::SecretBaseTemplate


In [001] New GameData\001_SecretBaseTemplate.rb, all of the Map Templates are defined.

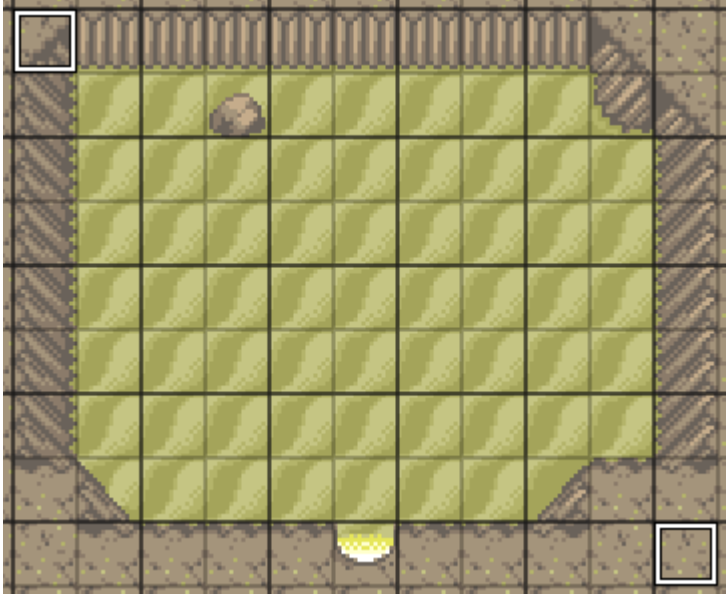


If you are using the included Map Templates, **you need to update the :map_id to point to the new Map ID**, with each map being named after their respective Map Template IDs.

You can delete the definitions if you are not using them.

There are a number of properties defined, some of which are unused. Below is an explanation of the properties.

Property	Explanation
:id	The symbolic representation of this Template. Used in the Compiler for Secret Bases.
:map_id	The Map ID of this Template created in Step 2. Do not zero pad this ID (56 not 056).
:type	The type of base. This determines the graphics for the base's exterior when closed and opened, as well as the SECRET_BASE_MESSAGES_ANIM data.
:door_location	An array consisting of the X and Y positions of the door. If you highlight the tile, the coordinates in the bottom right corner of RMXP's window is the exact spot. 
:pc_location	An array consisting of the X and Y positions of the door. If you highlight the tile, the coordinates in the bottom right corner of RMXP's window is the exact spot. This tile should have a passable tile below it, and it must always be possible to exit the base from this position without any decorations. You don't want to softlock your players!
:owner_location	An array consisting of the X and Y positions of the owner of the base, for record mixing. If unset, the base owner does not appear. Ideally, though not required, the base owner should stand on a base decoration tile, so that the player understands the reason why a "random tile" is blocking them.
:preview_steps	The number of steps the player should walk when entering the base to potentially move into it. Defaults to 2 if unset.
:map_borders	An array consisting of the X and Y positions of the top left and bottom right corner of the base. To clarify, this is not the map itself, but just the portion of the map consisting of the base.

Property	Explanation
	 <p>If your map has a border, the top left is generally (8,6), or (0,0) if it has a void border instead.</p>

Step 5: Defining Secret Bases

Secret Bases are defined in `secret_bases.txt`.

A Secret Base PBS entry consists of 3 parts: a unique ID, a MapTemplate, and a Location.

```
[ID]
MapTemplate = TemplateID
Location = Map ID, X Position, Y Position
```

The ID can be any name you want, so feel free to make it descriptive.

The MapTemplate must be an ID from `GameData::SecretBaseTemplate`.

The Location on the map is the exact placement of the exterior base entrance.

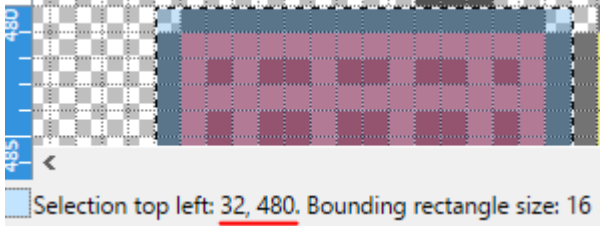
- ❗ The Location must be accessible with the player facing Up, and there must be a ground tile that the player can stand on below this position.
- You don't want to make an inaccessible base, nor softlock your players when they exit. Additionally, bases can only be interacted with when the player is facing Up.
- ❗ Consider the type of base you are creating. Not all base types have a closed graphic, so there must be a way for the player to identify a base location, such as
- Emerald's unique trees.

Step 6: Defining Decorations

Secret Base Decorations are defined in `secret_decorations.txt`. You can skip this step if you are using the provided PBS file.

A Secret Base Decoration PBS entry requires 3 parts: a unique ID, a player facing Name, and a Pocket number.

There are a number of properties defined. Below is an explanation of the properties.

Property	Explanation
[ID]	The symbolic representation for this decoration.
Name	The name for this decoration.
Pocket	The pocket number that this decoration is stored in. Starts at 1. See <code>def self.secret_bag_pocket_names</code> .
Price	The cost of this decoration at a Secret Base Mart. Defaults to 1.
SellPrice	The money the player will receive when selling this decoration at a Secret Base Mart. If the <code>SellPrice</code> is 0, the item can not be sold. Defaults to half of <code>Price</code> .
Description	The description for this decoration.
TileOffset	<p>The Tile ID of the top left tile in <code>SECRET_BASE_TILESET</code>. A simple formula is $(Y_Pixels/16)*8 + (X_Pixels/16)$, rounding down when dividing.</p>  <p><i>This tile has an id of $(480/16)*8 + (32/16)$ $= 240 + 2 = 242$</i></p> <p>Do not add 384 to compensate for autotiles.</p>
EventID	<p>The ID of the event in <code>SECRET_BASE_DECOR_MAP</code> to copy. Do not zero pad this ID (56 not 056). Copies the event exactly as it is created, except for the id.</p>
TileSize	<p>The Width and Height of this decoration. Can be combined with <code>EventID</code>, and if so, the event will be in the bottom right tile of the overall decoration. Defaults to [1,1]</p>
PlacingPerms	One of Floor , Wall , Decor , or Board . Determines valid locations for placing.

Property	Explanation
	<ul style="list-style-type: none"> Floor requires all tiles to be passable. Wall can only be placed on SECRET_BASE_WALL_TAG terrain tags. Decor can be placed on mats or desks if SECRET_BASE_DECOR_ANYWHERE is false, or any passable tile in addition to mats and desks, if true. Board can be placed over holes, but otherwise requires passable tiles. <p>Defaults to Floor.</p>

TileOffset and EventID can be used together, in which case, the Event's Graphic will be used if it has one while placing. The event will be in the bottom right tile of the overall decoration.

Decorations pull their icons from Graphics\Pictures\SecretBases\Icons, falling back to Graphics\Items\000 if one can't be found.

Step 7: Compile and Test

That's all the changes necessary for the script to work. At this point, you just need to compile the game and check out your secret bases.

You can perform a full compile by holding CTRL with the Game window focused before all the text finishes appearing. You can compile just the plugins by holding SHIFT in the same manner.

Useful Variables and Methods

\$secret_bag

A new bag for Secret Bases. An instance of class SecretBag, it has a number of methods that can be used directly if you need to add or remove items silently.

Method	Explanation
\$secret_bag.has?(decoration)	Returns true if the player has a decoration with this id and false otherwise. This decoration may currently be placed.
\$secret_bag.can_add?(decoration)	Returns true if the player has room to add this decoration and false otherwise. Will always return true if the pocket this decoration would go into is infinite.
\$secret_bag.add(decoration)	Gives the player this decoration. Returns true if successfully added, and false otherwise.

Method	Explanation
<code>\$secret_bag.can_remove?(decoration)</code>	Returns true if it is possible to remove this decoration and false otherwise. Placed decorations are not eligible to be removed.
<code>\$secret_bag.remove(decoration)</code>	Removes 1 copy of this decoration from the bag. Returns true if successfully removed, and false otherwise. Placed decorations are not eligible to be removed.

\$PokemonGlobal.secret_base_rank

The rank of this player, starting at 0. There is no code to automatically change the rank. The developer is free to change it under whichever conditions they feel is appropriate. Only impacts the usage of Secret Base Skills at other secret bases.

\$stats.moved_secret_base_count

A count of every time the Player packs up their base, either from their Secret Base PC or outside their new base.

pbSecretBaseMart(stock, speech=nil, cansell=false)

A Mart for Secret Base Decorations. Use like regular Marts, except for the last variable, as Secret Base Marts default to not allowing you to sell, compared to regular Marts, which default to allowing you to sell.

pbReceiveDecoration(decoration)

Gives the player the decoration with messages for the name of the decoration and that it was sent to the PC. Returns true if it was successfully added, and false otherwise.

pbGetPlayerBaseLocation(mapname=-1, mapid=-1)

Sets the passed game variables to the map name and map id respectively of the player's secret base. Also returns an array of the same information. The map id is -1 if the player does not have a base.

Record Mixing and the Cable Club

This script supports Record Mixing, allowing the transfer of data between players connected online. It will send all bases on this save, and receive and save as many of the bases as possible.

At other bases, players can perform several actions.

- Interact with the Secret Base PC
 - Register bases, preventing them from being overwritten by other bases when record mixing
 - View the Registry, allowing the player to check the rough location of registered bases, and unregister them remotely.
- Speak with the Base Owner
 - Battle the owner once daily with their team at the time of Record Mixing.
 - Use a Secret Base Skill, which can be performed once or twice daily depending on rank.
 - Ask them to move out, deleting the base from the game.

The Base owner will not perform daily actions the day they record mixed if they were a newly added base. Updates to a previously existing base will keep whichever daily action status they had previously. Daily actions refresh at midnight.



This script uses `Time.now` to set the daily timer. This means that it will refresh on the system's time, and not the game's time, if you have a script that aliases `pbGetTimeNow`.

Secret Base Skills

In `[001] New GameData\004_SecretBaseSkill.rb`, all the Secret Base Skills are defined.

A Secret Base Skill consists of 3 parts, a symbolic ID, a player facing Name, and a Usage Proc.

The ID can be any name you want, so feel free to make it descriptive.

The Name is the string shown in the selection menu in the base.

The Usage Proc is a proc that does `next true` if the skill is successfully used, and `next false` otherwise.

You can reference the predefined ones for creating new ones.