

# 介绍

这个文档描述了 SCUM 中的战利品定制特性。战利品定制是通过创建和修改各种 JSON 文件来完成的。熟悉 JSON 基础知识是先决条件，因此如果您从未听说过 JSON 或者只是需要复习一下知识，这里有一些有用的链接：

- <https://en.wikipedia.org/wiki/JSON>
- [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)
- <https://www.json.org/>

大多数与战利品相关的 JSON 文件可以通过各种管理命令从游戏中导出，而其中一些必须手动创建。找到 JSON 文件的根路径是：

- 多人游戏服务器的 < Server > \ SCUM \ Saved \ Config \ windows Server \ Loot。
- % local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot for single player。

JSON 文件在游戏启动时加载，任何后续的更改都可以通过 # reloadlootcustomizationsandresetspowners 命令重新加载，以快速测试修改。后面将详细描述提到的命令以及导出、创建和修改与 loot 相关的 JSON 文件的过程。

在深入研究命令和战利品修改之前，我们将会介绍一些概念和先决条件，这些对于有效的战利品定制是必不可少的。

如果您想立即开始，请跳到导出部分。

# 稀薄

稀有性在战利品定制中被广泛使用。它决定了从对象集合中选择某个对象的概率。对象是一个抽象术语，可以表示任何东西。你现在可以把一个物体想象成一个物品。集合中的每个对象都被赋予一个稀有值，该值可以是下列值之一：

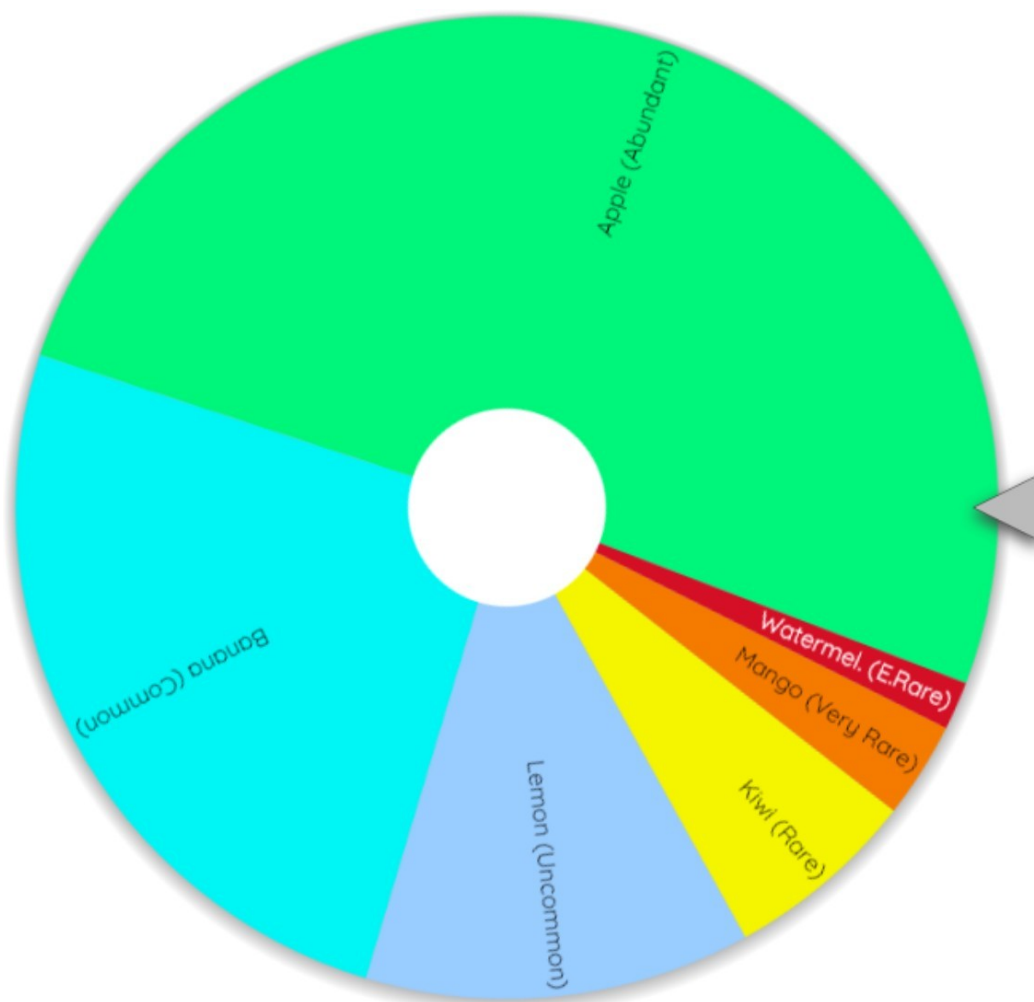
- 丰富:被选中的可能性是极其罕见的 32 倍
- 常见:被选中的可能性是极其罕见的 16 倍
- 不常见:被选中的可能性是极其罕见的 8 倍
- 罕见:被选中的可能性是极其罕见的 4 倍
- 非常罕见:被选中的可能性是极其罕见的 2 倍
- 极其罕见:和其他极其罕见的情况一样有可能被选中集合中的对象

例如，假设您有以下一组对象(在本例中，对象是项目)：

{苹果(丰富)、香蕉(常见)、柠檬(不常见)、猕猴桃(罕见)、芒果(非常罕见)、西瓜(极其罕见)}

苹果被选中的可能性是香蕉的 2 倍，柠檬的 4 倍，猕猴桃的 8 倍，芒果的 16 倍，西瓜的 32 倍。另一方面，香蕉被选中的可能性比苹果低 2 倍，比柠檬高 2 倍，比猕猴桃高 4 倍，比芒果高 8 倍，比西瓜高 16 倍。与剩余水果相比，选择其他水果的可能性可以以类似的方式推断。

另一种观察稀有物的方法是通过轮盘赌，物体占据的面积决定了它被选中的可能性。对于上面的例子，轮盘看起来像这样：

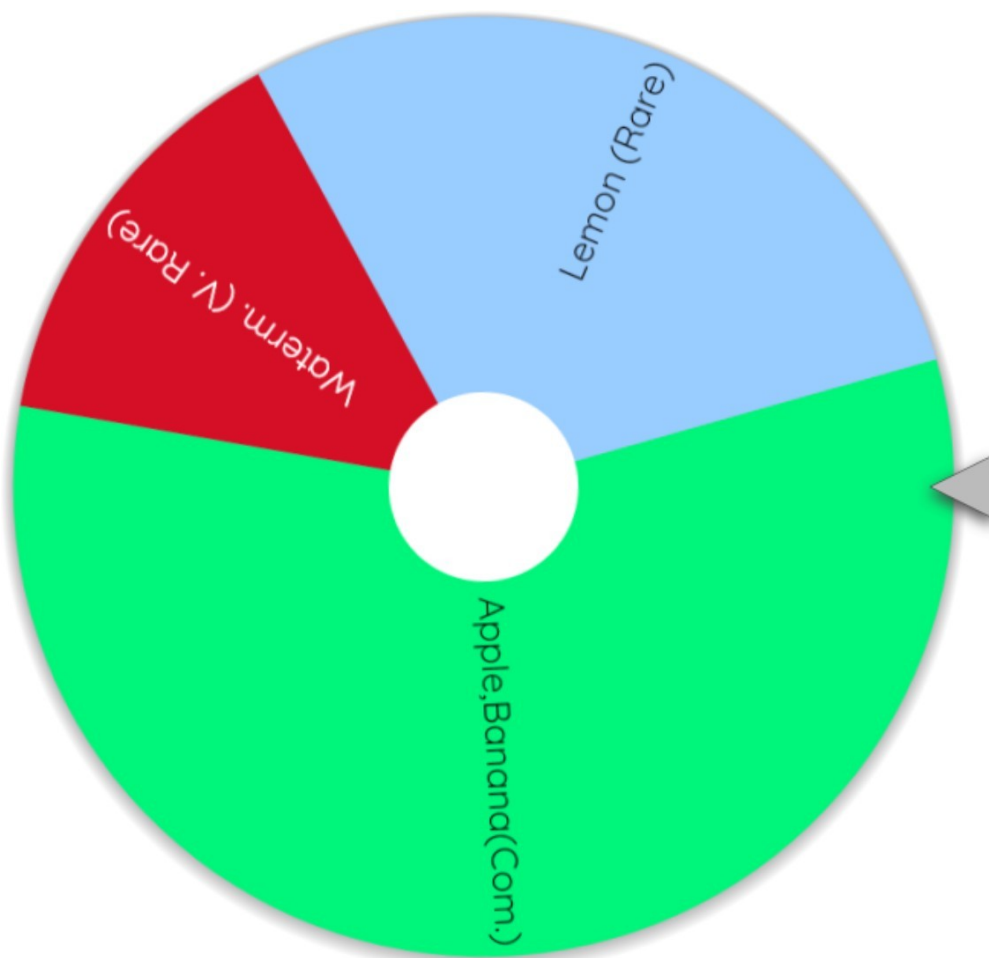


如你所见，苹果占地面积最大，因此最有可能被选中，而西瓜占地面积最小，因此最不可能被选中。其他水果介于两者之间。

让我们看另一个例子：

{苹果|普通，香蕉|普通，柠檬|嫩，西瓜|非常嫩}

这一次，一些稀有的水果“失踪”了，我们有两个同样稀有的水果。具有相同稀有性的对象被分组在一起。为了确定该组中选定的对象，对所有具有相同稀有度的对象再掷一次骰子，机会均等。让我们看看上面例子中的轮盘赌：



在这种情况下，苹果或香蕉被选中的可能性是柠檬的 4 倍，是西瓜的 8 倍。在上面的轮盘赌游戏中，首先选择苹果或香蕉，然后再掷一次骰子来选择苹果或香蕉，每个人都有 50% 的机会被选中。

了解对象选择如何使用稀有性是很重要的，因为它在文档的其余部分用于项目选择、节点选择和子重置选择。如果你不知道节点和上界是什么意思，也不用担心。它们将在下面的章节中解释。

# 拾取树节点

## 概观

在解释什么是战利品树节点之前，让我们先了解一下它们的动机。假设我们有一个可以生成以下项目的栏：

- 啤酒
- 苦艾酒
- 炸薯条
- 榛子
- 1H\_厨房刀具

我们必须以某种方式给物品分配概率，因为榛子和啤酒有同样的概率产卵是没有意义的。至少不是在酒吧里。我们已经有了指定概率的稀有物，所以让我们试试：

- 啤酒(丰富)
- 苦艾酒(非常罕见)
- 筹码(不常见)
- 榛子(罕见)
- 1H\_KitchenKnife(极其罕见)

这种方法的问题是，这些概率在一家酒吧有意义，但在其他地方不一定有意义。例如，如果我们需要在住宅厨房房产卵，1H 厨房刀比榛子或苦艾酒更有可能被发现。此外，如果我们要处理大量的项目，稀有性就不能提供足够的粒度。一个解决方案可以是使用概率而不是稀有的数字，但这对于调整和维护来说可能会很麻烦。

我们通过在树中对项目及其稀有性进行分类来解决这个问题。每个树节点都有名字和稀有度。父节点代表一包共享一些共同特征的项目。例如，父节点可以是一个栏，其中包含在栏中有意义的项目。此外，Bar 节点可以包含其他包，如饮料、食物和武器，其中包含对这些包有意义的项目。让我们将上面的 5 个项目放入一棵树中：

- 酒吧
  - 饮料
    - 啤酒
    - 苦艾酒
  - 食物
    - 炸薯条
    - 榛子
  - 武器
    - 1H\_厨房刀具

因此，第一步是将项目组织成一棵树。下一步是分配稀有物品。每个节点的稀有性是相对于其他兄弟节点的稀有性而言的。既然是酒吧，饮料应该比食物更常见，比武器更常见。此外，啤酒应该比苦艾酒更常见，薯片应该比榛子更常见。我们可以这样分配稀有度：

- 酒吧
  - 饮料(丰富)
    - 啤酒(丰富)
    - 苦艾酒(罕见)
  - 食物(不常见)
    - 薯片(丰富)
    - 榛子(普通)
  - 武器(非常罕见)
    - 1H\_厨房刀具

注意，Bar 和 1H\_KitchenKnife 节点没有指定稀有性。酒吧，因为它是一个根和 1H\_KitchenKnife，因为它没有兄弟姐妹，所以它是唯一的选择武器节点。

剩下的问题是如何从像上面这样的树中选择一个项目。答案是从左到右或者用树的术语来说，从根节点到叶节点。在上面的例子中，Bar 节点是根节点，所以选择从那里开始。接下来要选择的是 Bar 的子节点。酒吧节点有三个子节点：饮料(丰富)、食物(不常见)和武器(非常罕见)。子节点的选择是根据稀有性进行的，如稀有性部分所述。假设选择了一个食物节点。现在，该过程继续进行，直到到达叶节点。食物节点有两个子节点：薯片(丰富)和榛子(普通)。根据稀有度进行选择，假设选择了芯片节点。Chips 节点是叶节点，选择在此停止。玩家将会得到一个筹码。

同样的过程也适用于最大的树。选择从根节点开始，根据稀有度选择子节点，直到到达叶节点。叶节点总是项目。包含子节点的节点总是一袋袋的项目。

我们称这种类型的树为 loot 树，这种树中的节点为 loot 树节点(或简称为节点)。

## 节点 id

为了在某个地方使用节点，它们必须以某种方式被引用。例如，假设我们在酒吧中有三个位置：

- 包含吧台厨房工具的抽屉
- 包含各种饮料的冰箱
- 可以放饮料或食物的吧台

我们将使用上一节中的节点来指定哪些项目可以由哪个位置生成：

- 抽屉获得武器节点
- 冰箱获得饮料节点
- 桌子上有饮料和食物

在上面的例子中，我们使用节点名来指定哪个节点为哪个位置提供物品。这对于例子中简单的树来说是很好的，但问题是游戏中实际使用的战利品树很大，节点名不必是唯一的。例如，我们可以在酒吧和警察局都有一个名为“武器”的节点。我们必须有一个更精确的方法来指定哪些节点被引用。这就是节点 id 发挥作用的地方。每个节点都有一个唯一标识它的 ID。特定节点的节点 ID 是通过将祖先节点的名称添加到该节点名称前面，同时用点符号分隔所有名称来形成的。现在，让我们使用 id 而不是原始名称将节点分配到上面的位置：

- 抽屉里有吧台。武器节点
- 冰箱里有酒吧。饮料节点
- 桌子变吧台了。饮料和酒吧。食物节点

查看节点 id 的另一种方式是将它们视为文件系统中的路径。根节点和内部节点(包)可以是文件夹，叶节点(项目)可以是文件。唯一显著的区别是路径段的分隔符是点符号，而不是斜线符号。

## JSON 规范

使用 JSON 对象指定节点。每个节点都是一个 JSON 对象，具有以下两个属性：

名字	类型	描述
名字	线	节点的名称。在兄弟姐妹中必须是唯一的。
稀薄	线	节点的稀有性。可以是中列出的值之一 <a href="#">稀有品区。</a>

包含其他节点(包)的节点有一个附加属性：

名字	类型	描述
儿童	对象数组	此节点的子节点。

表示项目的节点(叶节点)可以有一个附加属性：

名字	类型	描述
变化	字符串数组	一些项目表示相同的逻辑项目，但是具有不同的“皮肤”。这个例子是同一件不同颜色的衣服。颜色变化的项目可以在项目节点的变化数组中指定。每当选择具有变体的项目节点时，由该节点表示的项目或其变体之一将以相等的机会被选择。

这是 JSON 在上一节中查找树的方式：

```
{
  【名称】：【酒吧】
  【儿童】：[
    {
      【名称】：“饮料”、“稀有度”：“丰富度”、“儿童”：[
        {
          “名称”：“啤酒”、“稀有度”：“丰富度”
        },
        {
          【名称】：“苦艾酒”，“稀有”：“稀有”
        }
      ]
    },
    {
      “名称”：“食物”，“稀有度”：“不常见”，“儿童”：[
        {
          【名称】：“芯片”，“稀有度”：“丰富度”
        },
        {
          【名称】：“榛子”，“稀有度”：“普通”
        }
      ]
    }
  ]
}
```



```

    ],
    {
        "名称": "武器", "稀有度": "非常稀有", "儿童": [
            {
                【名称】: "1H_厨房刀具"
            }
        ]
    }
]
}
}

```

## 导出内置节点

到目前为止，我们一直在使用一个简单的战利品树来说明一些基本概念。现在是时候导出游戏使用的实际战利品树了。为此，使用了两个命令：`#ExportDefaultLootTree` 和 `#ExportCurrentLootTree`。 `# ExportCurrentLootTree` 将在定制部分解释。

### # ExportDefaultLootTree

在默认状态下输出战利品树。默认状态是定制战利品之前的状态，也是游戏出厂时的状态。在你执行这个命令后，你会在下面的文件夹中得到战利品树：

- 多人游戏服务器的 `< Server > \ SCUM \ Saved \ Config \ windows Server \ Loot \ Nodes \ Default`。
- `% local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot \ Nodes \ Default for single player`。

当导出到单个文件时，默认的 loot 树太大而无法管理，因此它被导出到多个 JSON 文件，其中每个文件包含 `ItemLootTreeNodes` 节点的每个直接子节点的 JSON。`ItemLootTreeNodes` 是游戏中使用的所有节点的根节点。这是输出文件夹的外观：

htame	修改日期	类型	大小
空军基地编号 5.10 023。21:49 JSDN 文件 6 KB			
巨头	5.102023. 21:49	JSDN 文件	16 KB
Barn.json	5.10。 21:49	JSDN 文件	23 KB
浴室	5.10 023.21:49	JSDN 文件	16 KB
海滩	5.10 023。 21:49	JSDN 文件	3 KB
汽车修理厂	5.10。 21:49	JSDN 文件	22 KB
城堡	5.10 023.21:49	JSDN 文件	12 KB
挖空	5.10 023。 21:49	JSDN 文件	51 KB
中国商店	5.10 023.21:49	JSDN 文件	5 KB
钙矿物	5.10 023.21:49	JSDN 文件	21 KB
保管处	5.10 023。 21:49	JSDN 文件	2 KB
饮料商店	5.10 023.21:49	JSDN 文件	21 KB
农场	5.10 023.21:49	JSDN 文件	23 KB
消防站	5.10。 21:49	JSDN 文件	4 KB
食品店	5.10 023.21:49	JSDN 文件	8 KB
车库	5.10 023.21:49	JSDN 文件	2 KB
煤气，煤气	5.10。 21:49	JSDN 文件	4 KB
医院 n	5.10 023.21:49	JSDN 文件	16 KB
• 亨廷顿铁塔	5.10。 21:49	JSDN 文件	38 KB
	5.10 023.21:49	JSDN 文件	23 KB
	5.10 023.21:49	JSDN 文件	21 KB
ICrtchen n	5.10。 21:49	JSDN 文件	80 KB
起居室	5.10 023.21:49	JSDN 文件	13 KB
砖石建筑	5.10 023。 21:49	JSDN 文件	3 KB
军事	5.10 023.21:49	JSDN 文件	16 KB
天文台	5.10。 21:49	JSDN 文件	32 KB
办公室+办公室	5.10 023.21:49	JSDN 文件	22 KB
药房	5.10 023.21:49	JSDN 文件	64 KB
PoF+ce n	5.10。 21:49	JSDN 文件	49 KB
采石场	5.10 023.21:49	JSDN 文件	18 KB
萨克林	5.10 023.21:49	JSDN 文件	21 KB
学校	5.10。 21:49	JSDN 文件	36 KB
^	5.102023. 21:49	JSDN 文件	22 KB
垃圾	5.10。 21:49	JSDN 文件	22 KB
车间 n	5.10 023.21:49	JSDN 文件	4 KB
第二轮世界大战	5.102023. 21:49	JSDN 文件	17 KB
	5.10。 21:49	JSDN 文件	百万字节
	5.10 023.21:49	JSDN 文件	20 KB

你可以检查生成的默认战利品树 [here](#)。

在你导出默认的战利品树后，你还没有做任何定制。你就有了一个对分析和修改有用的起点。

## 用户化

本节将指导你定制游戏中使用的战利品树。我们将一步一步地向你展示如何修改和测试游戏中某个物体在被检查后掉落。请记住，游戏对象不一定需要使用节点来指定它们放下哪些物品。它们可以直接引用项目，而根本不使用节点。但是因为我们在解释节点，所以我们会讨论这种情况。指定游戏对象丢弃哪些物品的不同方法在“生成者预设”部分有所解释。

我们现在将完全自定义下图中红色垃圾箱的内容。



出于演示的目的，这个想法是让红色垃圾桶放下饮料和食物。饮料应该比食物更容易掉落。选择饮料时，我们希望放下水或啤酒。啤酒应该很少掉落。当选择食物时，我们希望苹果和香蕉有均等的机会掉落。

为了完成上面的工作，我们需要根据上面的描述制作一个战利品树 JSON。下面是这个例子的完整 JSON loot 树：

```
{
  " Name": "ItemLootTreeNodes ", "
  Children": [
    {
      " Name": "Trash ", "
      ChildrenMergeMode": "Replace ", "
      Children": [
        {
          【名称】：“饮料”、“稀
          有度”：“丰富度”、“儿
          童”:[
            {
              【名称】：“水
              _051”，“稀有
              度”：“丰富度”
            },
            {
              【名称】：“啤酒”，
              “稀有”：“稀有”
            }
          ]
        },
        {
          【名称】：“食物”，
          “稀有”：“常
          见”、“儿童”:[
            {
              【名称】：“苹果”
            },
            {
              【名称】：“香蕉”
            }
          ]
        }
      ]
    }
  ]
}
```

[Here](#) 也是 GitHub 链接。

上面有一个 JSON 属性在 JSON 规范一节中没有提到，那就是 bag 上的 ChildrenMergeMode 属性。它所做的是指示战利品定制系统如何合并两个有相同 ID 的袋子的孩子。在上面的战利品树中，自定义节点是一个 ID 为 ItemLootTreeNodes.Trash 的袋子。这个 ID 的袋子已经存在于默认的战利品树中，与我们正在制作的袋子完全不同。当行李具有相同的 ID 时，Loot 系统有两个选项：

- 用自定义包中的子包完全替换默认包中的子包。
- 合并子包，以便生成的包更新同名的子包并添加新的子包。

第一种行为是通过将 ChildrenMergeMode 属性添加到冲突的行李并将其值设置为 Replace 来实现的。第二种行为是通过完全省略 ChildrenMergeMode 属性或将其值设置为 UpdateOrAdd 来实现的。我们为自定义包选择了替换为合并模式，因为我们不关心默认 ItemLootTreeNodes 中的内容。垃圾袋。

为了进一步帮助您理解合并是如何工作的以及合并的最终结果是什么，您可以使用 # ExportCurrentLootTree 命令。它的工作方式与 # ExportDefaultLootTree 相同，但有两个重要区别：

- 战利品树被导出到一个名为 Current 的文件夹中，而不是默认的。
- 结果是一个实际的战利品树，将用于评估哪些项目产卵，这是默认的战利品树和任何自定义的战利品树合并的结果。如果你没有任何定制的战利品树，当前的战利品树和默认的战利品树是一样的。

所以现在我们有了一个定制的战利品树，我们需要让红色的垃圾桶放下水瓶、啤酒、苹果和香蕉。为此，请按照下列步骤操作：

1. 开始游戏。我们建议使用 singleplayer 进行节点实验。
2. 使用以下命令传送你的角色：`#Teleport -493223 -423063 611`。红色垃圾桶应该在你身边。
3. 使用 `# ExportDefaultLootTree` 命令导出默认战利品树。游戏会告诉你树被导出到了哪里。在单人游戏中，这应该是：  
`% local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot \ Nodes \ Default`
4. 红色垃圾桶从 ItemLootTreeNodes 中丢弃项目。默认为垃圾袋。你将在后面学习如何确定哪个游戏物体掉了什么，它是否使用节点，固定的一组物品或其他方式来指定战利品的结果。现在，只要记住红色的垃圾箱可以从 ItemLootTreeNodes 中删除任何项目。垃圾袋。打开 Trash.json 并检查它。我们不会使用这个文件，只是浏览一下，感受一下节点是如何构造的，以及哪些项目可以从红色的垃圾箱中删除。
5. 因为我们将修改节点并测试我们得到的内容，所以每次搜索时，红色的垃圾箱都会丢弃一些项目，这将是非常有用的。我们不想浪费时间等待产卵者冷却，特别是如果游戏对象没有下降。我们要做的第一件事是增加游戏物体掉落物品的概率。[\[计\] 下载](#)

[GeneralZonesModifiers.json](#) 文件并将其放在

% local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot 文件夹。这个文件会增加地图上所有检查产卵者掉落物品的概率。概率将增加 100 倍，所以基本上，所有检查产卵将有 100% 的机会下降的东西。GeneralZonesModifiers.json 部分对此进行了详细解释。就像现在一样使用一般的区域修改器。

6. 因为你现在已经在战利品中添加了一个新的自定义(普通区域修改器)

目录，你需要告诉游戏重新加载它。

执行# reloadlootcustomizationsandresetspowners 命令来重新加载所有战利品定制。执行命令后，游戏将增加检查产卵者掉落物品的概率 100 倍，并重置所有检查产卵者的冷却时间。

7. 尝试搜索红色垃圾箱，然后重复连续执行# reloadlootcustomizationsandresetspowners 命令。如果你这样做了，你会发现你可以反复搜索，并从红色的垃圾桶里找到一些东西。这使您能够快速测试丢弃的内容。

8. 在中创建覆盖文件夹

% local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot \ Nodes 文件夹。游戏从覆盖文件夹中读取所有的战利品树自定义。

9. 在 Override 文件夹中创建 MyTrash.json 文件，并复制粘贴自定义 ItemLootTreeNodes。垃圾袋 JSON 以前制作成的文件。或者，下载 [this](#) 文件到你的覆盖文件夹。
10. 使用# reloadlootcustomizationsandresetspowners 来应用和测试 loot 树定制。
11. 尝试不同的节点结构，稀有和项目。
12. 完成后不要忘记删除 GeneralZonesModifiers.json 文件，除非你希望在常规的单人游戏中以 100% 的概率产生所有东西。

这应该是你现在需要知道的关于节点的一切。稍后，您将学习如何从自定义生成器预设中引用节点。

## 常见陷阱

在 JSON 文件中犯一个小的语法错误，然后想知道为什么 loot 定制不工作是很常见的事情。当战利品定制没有按预期工作时，我们建议检查游戏日志中的错误。例如，如果我们在 MyTrash.json 中犯了一个错误，并将逗号放在了错误的位置，那么输出日志将包含如下内容：

LogItemSpawningDataRegistry: 错误: 无法解析 “E:/Projects/SCUM/Main/SCUM/Saved/Config/Windows/Loot/Nodes/Override/my trash . JSON”。我们同意这不是非常描述性的，但至少你知道在哪个文件中寻找错误。

除了语法错误之外，与战利品相关的游戏日志还包含了其他关于战利品定制的有用信息。与 Loot 相关的类别有 LogItemSpawning, LogItemLootTree 和 LogItemSpawningDataRegistry，所以你应该可以用这些短语过滤游戏日志。



# 出口

我们已经导出了下面写的所有内容。你还会找到所有的区域和一些城镇，这样你就可以在单人游戏中练习修改，而不必在多人游戏中按地点导出。您可以从这里下载：

<https://github.com/CheeseKingu/Loot-Modifiers/blob/main/Loot.7z>

## # ExportDefaultItemSpawningParameters

这个命令将会创建一个 .json 文件，其中包含了所有物品的参数，这些物品都是由战利品系统产生的。未包含的项目示例是手工制作的项目。

- 在多人游戏中输入# ExportDefaultItemSpawningParameters 后，您将导出位于以下位置的“parameters . JSON”：Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Items \ Default
- 在 Singleplayer 中键入# ExportDefaultItemSpawningParameters 后，您将导出“Parameters.json”位于此处：  
% local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot \ Items \ Default
- 我们将以 M82A1 为例来解释这些线条的含义：

```
{
  "Id": "Weapon_M82A1",
  "IsDisabledForSpawning": false,
  "AllowedLocations": [
    "Coastal",
    "Continental",
    "Mountain"
  ],
  "CooldownPerSquadMemberMin": 0,
  "CooldownPerSquadMemberMax": 0,
  "Variations": [
    "Weapon_M82A1_Black",
    "Weapon_M82A1_Desert",
    "Weapon_M82A1_Snow"
  ],
  "ShouldOverrideInitialAndRandomUsage": false,
  "InitialUsageOverride": 0,
  "RandomUsageOverrideUsage": 0
},
```

- “isdisableforspawning”:false-该物品可以被繁殖，如果你设置“true”该物品将不会被繁殖。
- “非婚生地址”：  
海岸-只能在地图的海岸区域产卵的物品。大陆-只能在地图的大陆区域繁殖的物品。  
山区-只会在地图的山区产卵的物品。
- “CooldownPerSquadMemberMin”: 0 -如果物品被你队伍中的任何人抢劫，整个队伍都找不到这里设定的最低数量(以小时为单位)的物品。

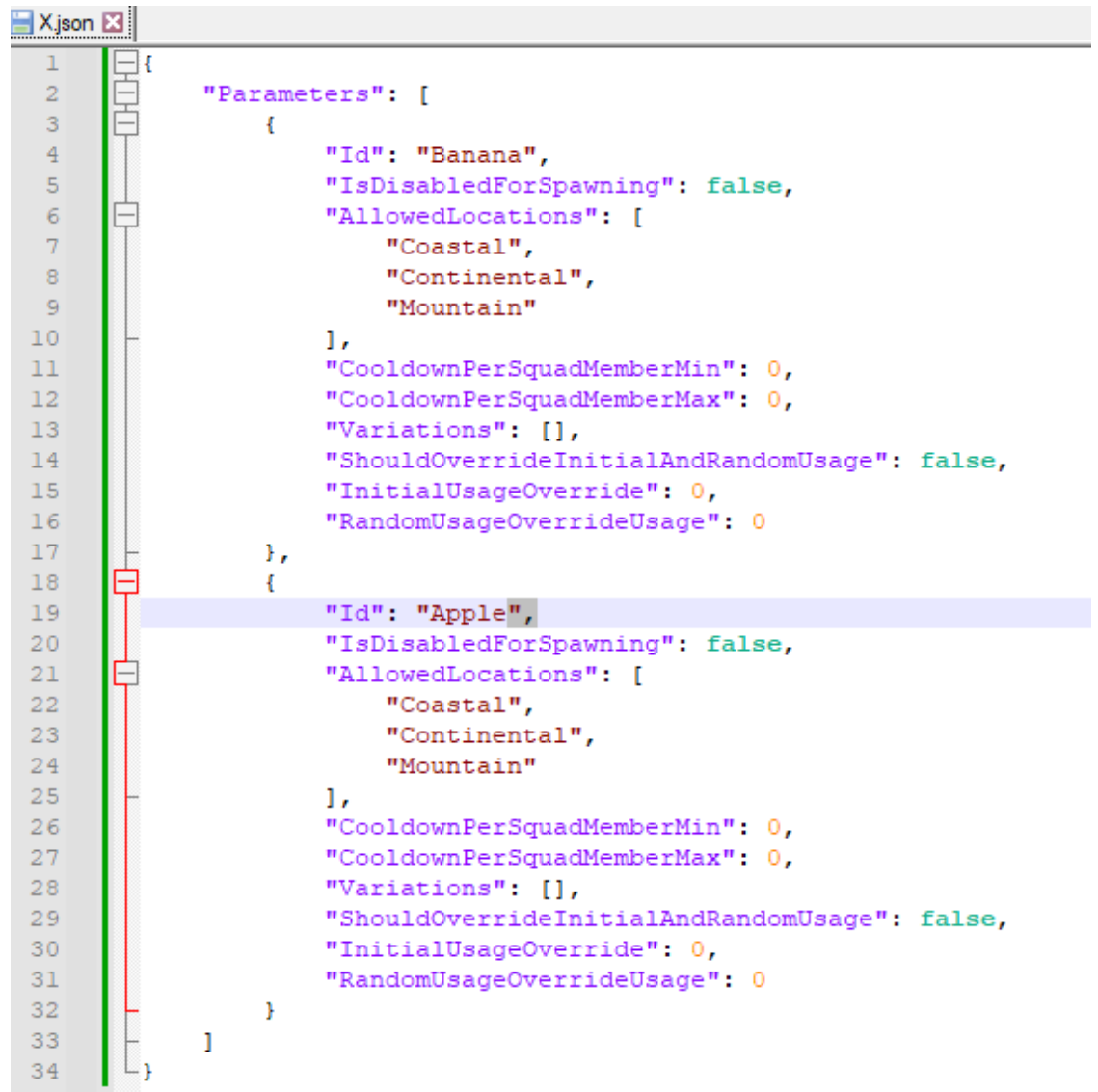
- “CooldownPerSquadMemberMax”: 0 -如果物品被你队伍中的任何人抢劫，整个队伍都找不到这里设定的最大数量的物品(以小时为单位)。
- 如果是这样设定的话，某个物品的冷却时间将会是 1 到 4 个小时。
- “变奏”:[“武器\_ M82A1 \_黑色”、“武器\_ m82a 1 \_沙漠”、“武器\_ m82a 1 \_雪”]-如你所见这些都是 m82a 1 放在这里的皮肤变奏。如果你把 Weapon\_M82A1 放在任何 spawner 预设的节点或项目设置中，它也能产生它的变体。
- “ShouldOverrideInitialAndRandomUsage”:false-它不会使用 spawner 预设中的用法设置，而是使用该文件中的“InitialUsageOverride”和“RandomUsageOverrideUsage”。
- “InitialUsageOverride”: 0 -该值以百分比表示，如果您将它设置为 80，它将删除一个项目最大使用量的 80%。
- “RandomUsageOverrideUsage”: 0 -该值以百分比表示，如果您将其设置为 20，将从最大数量中删除 0-20%。
- 例如:我们的煤气罐有 40 种用途。我们设置了“InitialUsageOverride”:80 和“RandomUsageOverrideUsage”:20。如果“RandomUsageOverrideUsage”:20 选了 10，那么总共 90%将从 40 次使用中移除。在这种情况下，我们的瓦斯罐将产生 4/40 的使用。

## 覆盖 Parameters.json

- 为了修改您的“Parameters.json”，您需要创建“Override”文件夹
  1. 对于多人游戏，进入 Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Items  
并创建“覆盖”文件夹。
  2. 对于单人游戏，请在此处创建:  
% local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot \ Items \
  3. 你可以创造很多.json文件，并在Override文件夹中随意命名它们，这纯粹是为了便于组织。
  4. 您可以在中放置任意数量的项.json文件，如何组织它们取决于你自己。这种文件夹的示例:<https://github.com/CheeseKingu/Loot-Modifiers/blob/main/Items.7z>
  5. 在该文件夹中，您可以看到我们已经创建了默认和覆盖。



6. 在 Override 文件夹中，我们创建了 2 个 json 文件，并根据自己的喜好命名。在 json 文件中，我们覆盖了几个项目。



```
1 {
2     "Parameters": [
3         {
4             "Id": "Banana",
5             "IsDisabledForSpawning": false,
6             "AllowedLocations": [
7                 "Coastal",
8                 "Continental",
9                 "Mountain"
10            ],
11            "CooldownPerSquadMemberMin": 0,
12            "CooldownPerSquadMemberMax": 0,
13            "Variations": [],
14            "ShouldOverrideInitialAndRandomUsage": false,
15            "InitialUsageOverride": 0,
16            "RandomUsageOverrideUsage": 0
17        },
18        {
19            "Id": "Apple",
20            "IsDisabledForSpawning": false,
21            "AllowedLocations": [
22                "Coastal",
23                "Continental",
24                "Mountain"
25            ],
26            "CooldownPerSquadMemberMin": 0,
27            "CooldownPerSquadMemberMax": 0,
28            "Variations": [],
29            "ShouldOverrideInitialAndRandomUsage": false,
30            "InitialUsageOverride": 0,
31            "RandomUsageOverrideUsage": 0
32        }
33    ]
34 }
```

## # ExportDefaultItemSpawnerPresets:

这个命令将导出我们在游戏中使用的所有默认的产卵器预设。

- 首先，在多人游戏中输入# ExportDefaultItemSpawnerPresets，这将在您的配置中创建“Loot”文件夹，位置如下:Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Spawners \ Presets \ Default
- 如果您在Singleplayer中# ExportDefaultItemSpawnerPresets，它将保存在此处:  
% local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot \ Spawners \ Presets \ Default
- 在默认文件夹中，您会发现所有导出的 spawner 预设。你不能创建新的，你只能改变现有的。
- 名称中带有“检查”的预设被分配给你必须“搜索”才能掉落战利品的对象，例如，建筑物-厨房-检查\_橱柜被分配给你在房子中搜索的橱柜。
- 名称中带有“世界”的预设被分配给在附近产卵的产卵者。例如，Buildings-Garage-Residential-World \_ Shelf 被指定给住宅区车库中的货架，并放置煤气罐或汽车电池等物品。

## # exportitemspawnerpresetsinzone






该命令将导出某个区域中使用的所有 spawner 预设。

- 只能用地图上的矩形或使用扇区名称来指定位置。
- 这个命令不会生成一些风景预置，农场，角色和货物掉落的预置，这些都需要手动添加，在后面的步骤中会解释。
- 所有的区域输出应该在多人游戏中完成，在单人游戏中，你的囚犯周围有一定的距离，所以你不能有效地输出。

按扇区名称导出:

- 如果你在多人游戏中输入# ExportItemSpawnerPresetsInZone A2。这将创建以下文件夹:Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Spawners \ Presets \ Override \ A2
- 如果在Singleplayer中键入# ExportItemSpawnerPresetsInZone A2。这将创建以下文件夹:  
% local appdata % \ SCUM \ Saved \ Config \ windows editor \ Loot \ Spawners \ Presets \ Override \ A2

- 在“A2”文件夹中，你会发现 A2 扇区中使用的所有的 spawner 预置。
- 在该文件夹的底部，您会找到“Zones.json”：

	Street-Residential-World_Rabbitary.json	10/2/2023 11:59 AM	JSON File
	Water-Shipwrecks-Examine_Crate.json	10/2/2023 11:59 AM	JSON File
	Water-Shipwrecks-Examine_Crate_Green...	10/2/2023 11:59 AM	JSON File
	Water-Shipwrecks-Examine_Crate_Green...	10/2/2023 11:59 AM	JSON File
	Zones.json	10/2/2023 11:59 AM	JSON File

- 该文件包含矩形的“左上”和“右下”位置，在本例中是 A2 扇区：

```

1  [
2    {
3      "Zones": [
4        {
5          "TopLeft": "X=9600.000 Y=-295198.375",
6          "BottomRight": "X=-295198.375 Y=-599996.750"
7        }
8      ]
9    }
10 ]

```

- “区域”文件确定了给定文件夹中所有 spawner 预设的位置边界；下一段会更有意义。

## 按矩形位置导出：

- 打开任何管理助手程序或打开 [https://scum-map.com/en/interactive\\_map](https://scum-map.com/en/interactive_map)
- 找到您想要修改的位置，例如我们要修改 Tisno：



- 如果我们要键入# ExportItemSpawnerPresetsInZone A2, 我们将获得整个 A2 的 spawner 预设, 我们只想在这里指定 Tisno。
- 此命令使用的格式是# ExportItemSpawnerPresetsInZone X Y X Y Name。
- 因此, 我们将输入左上角位置的 X Y, 然后输入右下角位置的 X Y, 这个名称完全是为了便于组织。指定的名称将是将要创建的文件夹的名称。
- 右键单击矩形的左上角位置, 然后按复制 TP 位置。你会得到这个值:# Teleport-15613.1299-472342.2348 0(X =-15613.1299, Y=-472342.2348)。
- 将位置保存在某处然后复制矩形的右下角位置, 就会得到这个值:# Teleport-83853.2402-516124.1469 0(X =-83853.2402, Y=-516124.1469)。
- 现在要使用这个命令, 我们需要从第一个位置和第二个位置获取 X, Y, 并把它放在命令中。出于组织目的, 我们还将键入 Tisno。
- 因此, 键入# ExportItemSpawnerPresetsInZone-15613.1299-472342.2348 -83853.2402 -516124.1469 Tisno。它已经为给定位置导出了 130 个预设

Exported 130 spawner preset(s).

- 我们现在已经创建了名为“Tisno”的文件夹, 其位置如下:Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Spawners \ Presets \ Override \ Tisno。
- 在该文件夹中, 我们可以找到在该位置使用的所有 spawner 预设, 我们可以看到该命令创建了适当的“Zones.json”文件。

## 区域. json

- “Zones.json”决定了特定文件夹中的 spawner 预设的边界。
  - 如果某个文件夹中没有区域文件, 它将使用父文件夹中的区域(因此子区域文件将覆盖父文件夹的区域文件)。
  - 如果整个“覆盖”文件夹中没有包含任何区域文件, 那么它将使用全局位置(预设将用于整个地图)。
1. 我们将使用我们已经导出的示例, 我们将在 Tisno 端口导出另一个小矩形(我们已经导出了 A2 扇区和 Tisno 本身)。
  2. 因此, 我们将使用与导出位置相同的步骤, 即使用端口本身的左上角和右下角位置。
  3. 左上角位置:# Teleport-52954.4182-493783.9695 0
  4. 右下角位置:# Teleport-77080.9405-510602.7629 0

5. 现在我们将 X Y X Y 和 TisnoPort 放在命令本身中：

```
# exportitemspawnerpresetsinzone-52954.4182-493783.9695-77080.9405  
-510602.7629 提斯诺波特
```

6. 这是我们现在在地图上的位置：



7. 现在，我们在 A2 文件夹中有了“Zones”文件，在“Tisno”文件夹中也有了它，在“TisnoPort”文件夹中也有了它。

8. 这些位置按从小到大的顺序排列。

9. 因此，在这种情况下，我们对 A2 的配置将针对 A2 中的所有内容进行设置，不包括 Tisno 和 TisnoPort。

10. 在 Tisno 文件夹中完成的配置将优先于“A2”文件夹中的 spawner 预设。

11. 对于 TisnoPort 也是一样，它将优先考虑“Tisno”中的 spawner 预设  
spawner 预设。

## 修改较小的位置

- 如果你想的话，你可以一个房子一个房子的修改，但是请注意管理帮助程序上的位置可能不准确。
- 如果你想一个房子一个房子地修改，你应该选择游戏中最好由两个玩家完成的位置(一个放在左上角，一个放在右下角)。
- 当管理员到达他们想要的位置时，他们应该键入#Location。

- 在你获得游戏中的位置后，你应该使用生成的 X Y X Y 位置将它们转移到管理命令中。

## 重叠区域

- 如果你有如下所示的情况，红色矩形将有自己的战利品预设。
- 但是由于设置位置较小，红色矩形的右上部分将被绿色矩形覆盖。
- 因此，重叠矩形上的所有 spawner 预设将优先于绿色矩形文件夹。
- 例如:如果你设置红色矩形中的石头掉落一把武士刀，绿色矩形中的石头掉落一把 MK18，那么重叠矩形中的每块石头都会掉落一把

MK18。





## GeneralZonesModifiers.json

- 如果我们想在地图上选择某个矩形区域或整个区域来修改所选区域的全局战利品，我们可以使用这个工具。
- 您必须在根目录“Loot”中手动创建“GeneralZonesModifiers.json”  
文件夹位于:Server \ SCUM \ Saved \ Config \ windows Server \ Loot
- 如何使用它的示例，复制粘贴链接：

<https://github.com/CheeseKing/Loot-Modifiers/blob/main/GeneralZoneModifiers>



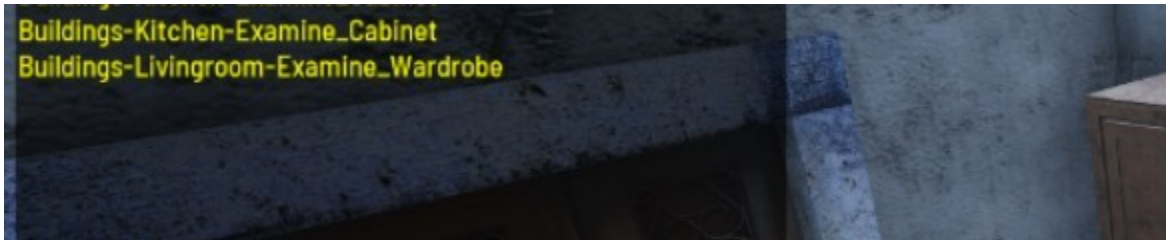
```
1 {
2   "Modifiers": [
3     {
4       "Zones": [
5         {
6           "Name": "FishFactory",
7           "TopLeft": "X=-327211.8142 Y=-346843.5702",
8           "BottomRight": "X=-363868.8911 Y=-383500.647"
9         }
10      ],
11      "SpawnerProbabilityMultiplier": 100,
12      "ExamineSpawnerProbabilityMultiplier": 100,
13      "ExamineSpawnerQuantityMultiplier": 5
14    },
15    {
16      "Zones": [
17        {
18          "Sector": "A1"
19        }
20      ],
21      "ExamineSpawnerQuantityMultiplier": 5
22    }
23  ]
24 }
```

1. 我们创建了“GeneralZonesModifiers.json”，并添加了鱼工厂的“左上”和“右下”位置。
2. “名称”行严格用于组织目的。
3. 这将在全球范围内增加我们在下面指定的特定区域的战利品。
4. 你也可以像上面的图片一样增加整个区域的战利品。
5. 这些基本上是乘以你的“服务器设置.ini”你所选择的区域的战利品设置，也是你输入的数量。
6. 因此，如果在“ServerSettings.ini”中设置为0.5，我们将整个A1扇区的ExamineSpawnerQuantity设置为5。它将被设置为2.5 ( $5 * 0.5 = 2.5$ )。
7. 我们还将整个FishFactory设置为如图所示的值。

## # setshouldprintecancerspawnerpresets

这个命令将打印出你从哪个物体中掠夺了哪个产卵器预置。

- 要找出哪个检查预设被分配给游戏中的哪个对象，您需要键入# setshouldprintecancerspawnerpresets true。搜索一个对象后，指定的 spawner 预置将被打印到您的聊天和日志中。
- 您可以使用这几行来过滤日志，以查找您使用这一行搜索的所有内容:检查 spawner 预设。
- SCUM.log 和 Loot.log 中打印的行示例:log item spashing:Examine spawner 预置:Buildings-Office-Police-Examine \_ File \_ Cabinet。



## 单人游戏

要在 Singleplayer 中进行测试，您所做的所有搜索都将打印到您的“SCUM.log”中，该文件位于:C:\ Users \ % username % \ AppData \ Local \ SCUM \ Saved \ Logs

## 多光盘播放器

为了在多人游戏中进行测试，你所做的所有搜索都会被打印到你的“Loot.log”，它位于:Server \ SCUM \ Saved \ save files \ Logs \ loot . log

## # reloadlootcustomizationsandresetspowers

重新载入你在配置/战利品中定制的所有战利品。同时重置检查产卵者，这样他们不用等待冷却时间就可以掷出要产卵的物品(如果他们最近被搜索过的话)。附近的产卵器不会重置，所以你需要等待冷却时间来测试它们。或者，你可以在单人游戏中测试，并在每次修改战利品后创建一个新角色。



如果您希望快速测试您所做的定制，此命令非常有用。一个常见的工作流程可能是：

1. 定制你选择的战利品(不要忘记保存修改后的文件)。
2. 执行# ReloadLootCustomizationsAndResetSpawners 命令。
3. 通过搜索受步骤 1 中的修改影响的容器来测试战利品定制。

## 所有者预设

产卵器预设决定了你将在世界上或在搜索物体时找到哪些战利品。您搜索的每个对象都有一个指定给它的“生成者”预设，每个可搜索的对象也是如此。

### 产卵器预设注释：

- 名称中带有“检查”的预设被分配给你必须“搜索”才能掉落战利品的对象，例如，建筑物-厨房-检查\_橱柜被分配给你在房子中搜索的橱柜。
- 名称中带有“世界”的预设被分配给在附近产卵的产卵者。例如：货架上的煤气罐、洗衣机上的衣服等。
- 只有在“覆盖”文件夹中完成的更改才会修改 spawner 预设的更改，您提取的其他所有内容都是它们的默认值。
- 如果您仅将 Spawner 预设添加到“覆盖”文件夹，这些更改将是全局的，因为您没有使用“Zones.json”指定位置。
- 一旦修改了“生成者”预设，地图上已经存在的项目将不会被替换。
- 修改某个 spawner 预设后，您需要等待该项目过期并等待新项目生成，或者您可以使用 # reloadlootcustomizationsandresetspawners 来重新加载可搜索的对象。这不会影响邻近区域或“世界”区域。
- 最好的方法是提取你在多人游戏中需要的所有东西，然后通过粘贴到 C:\ Users \ % username % \ AppData \ Local \ SCUM \ Saved \ Config \ windows editor 将整个战利品文件夹转移到 Singleplayer
- 在单人游戏中，你可以创建一个新角色，你周围的生物也会被重新载入。

## Spawner 预设设置

- 我们将以建筑物-工厂-采石场-世界\_生产线为例来描述产卵器预设设置所指的内容：



```
1 {
2   "Nodes": [
3     {
4       "Rarity": "Uncommon",
5       "Ids": [
6         "ItemLootTreeNodes.Quarry.Crafting",
7         "ItemLootTreeNodes.Quarry.Tools.Heavy",
8         "ItemLootTreeNodes.Quarry.WorkClothes.Torso",
9         "ItemLootTreeNodes.Quarry.WorkClothes.Legs",
10        "ItemLootTreeNodes.Quarry.WorkClothes.Feet",
11        "ItemLootTreeNodes.Quarry.Other.Ammo",
12        "ItemLootTreeNodes.Quarry.Other.Explosives",
13        "ItemLootTreeNodes.Quarry.Other.Flares",
14        "ItemLootTreeNodes.Quarry.Other.Money",
15        "ItemLootTreeNodes.Quarry.Other.Backpacks.Belt_Pistol_Holster",
16        "ItemLootTreeNodes.Quarry.Other.Backpacks.Hiking.Other",
17        "ItemLootTreeNodes.Quarry.Other.Backpacks.Plain",
18        "ItemLootTreeNodes.Quarry.Other.Backpacks.School",
19        "ItemLootTreeNodes.Quarry.Other.Backpacks.WaistBags"
20      ],
21    },
22  ],
23  "Probability": 25,
24  "QuantityMin": 1,
25  "QuantityMax": 1,
26  "AllowDuplicates": false,
27  "ShouldFilterItemsByZone": true,
28  "InitialDamage": 5,
29  "RandomDamage": 35,
30  "InitialUsage": 5,
31  "RandomUsage": 35
32 }
```

## 可能性

- “概率” :25, - 25%的物品掉落概率乘以你的 ServerSettings.ini 和区域修饰符。
- 如果你想要 100%的下降机会, 你可以简单地删除整个“概率”线
- 注意, 如果你设置概率为 100, 它仍然会与 ServerSettings.ini 和 zone 修饰符相乘, 你的掉落几率不会是 100%

## 量

- “quantity min” :1, -将从 spawner 预设中丢弃的最小项目数。
- “quantity max” :1, -将从 spawner 预设中丢弃的最大项目数。

- 如果你输入“最小数量”:4和“最大数量”:9,产卵器预设将从4到9中选择一个数字,并丢弃相应数量的项目。

## 复制

- “allow duplicates”:false, -它不会给你两个相同的项目
- 即使产卵器预置选择给你7个项目,但其中3个项目是重复的,你只会得到4个项目。
- “allow duplicates”:true-产卵器预设可以多次产卵相同的项目。

## 损害

- “初始伤害”:5, -该物品应该产生100%的耐久性,但产卵者预设将对该物品造成5%的耐久性伤害,因此你将获得95%的耐久性
- “RandomDamage”:35, -从0到35中选择一个数字,造成相当于最大持续时间%的伤害。
- 在这种情况下,我们有“初始伤害”:5和“随机伤害”:35选择了25,我们的项目将产生70%的持久性( $100 - 5 - 25 = 70$ )

## 使用

- “InitialUsage”:5, -移除一个物品最大使用次数的5%,如果该物品有20次使用,它将移除1次使用。
- “RandomUsage”:35, -从0到35中选择一个数字,并造成相当于最大耐久性的%的伤害。
- 在这种情况下,我们有“initial usage”:5和“random usage”:35选择了15,我们的项目将在16/20使用时产生( $20 - 20\% = 20 - 4 = 16$ )。

## ShouldFilterItemsByZone

- 这是指在Parameters.json中设置的“沿海”、“大陆”和“山区”区域位置。
- “ShouldFilterItemsByZone”:true, -表示它将使用Parameters.json中设置的区域位置,如果将它设置为false,它可以在任何地方繁殖。  
(例如:在节点中有10个项目,并且在大陆区域的对象上设置了“产卵器”预设。在AllowedLocations中没有Continental的每个项目都不会在那里生成)。
- 您可以使用以下命令导出parameters . JSON:#  
ExportDefaultItemSpawningParameters 它将位于 Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Items \ Default
- 请注意,如果您想要编辑Parameters.json,您需要将项目添加到在“Override”文件夹中创建的文件中,这在Overriding Parameters.json一节中有解释。该位置将是:Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Items \ Override
- 海岸-只能在地图的海岸区域产卵的物品。

- 大陆-只能在地图的大陆区域繁殖的物品。
- 山区-只会在地图的山区产卵的物品。
- 举个例子，你永远不会在海边找到圣诞老人夹克，因为那里没有沿海地区：

```
"Id": "Santa_Jacket",
"IsDisabledForSpawning": false,
"AllowedLocations": [
    "Continental",
    "Mountain"
],
"CooldownPerSquadMemberMin": 0,
"CooldownPerSquadMemberMax": 0,
"Variations": [],
"ShouldOverrideInitialAndRandomUsage": false,
"InitialUsageOverride": 0,
"RandomUsageOverrideUsage": 0
```

- 你也永远不会在大陆或沿海地区找到战术夹克，因为它只有山：

```
{
  "Id": "Tactical_Jacket_01_04",
  "IsDisabledForSpawning": false,
  "AllowedLocations": [
    "Mountain"
  ],
  "CooldownPerSquadMemberMin": 0,
  "CooldownPerSquadMemberMax": 0,
  "Variations": [
```

## 组织生成者预设：

- 我们将修改我们已经在上面的步骤中提取的 spawner 预设。
1. 所以现在我们已经提取了 A2、Tisno 和 TisnoPort 中使用的所有 spawner 预置。

2. 由于我们的“Override”文件夹中有这三个提取的内容，我们将按优先级将它们放入文件夹中，以便更好地组织，因为它们都位于 A2 扇区。

Name	Date modified	Type	Size
A2	10/2/2023 12:27 PM	File folder	
Tisno	10/2/2023 12:23 PM	File folder	
TisnoPort	10/2/2023 2:41 PM	File folder	

3. 由于“TisnoPort”将覆盖“Tisno” spanger 预设，而“Tisno”将覆盖“A2” spanger 预设，因此我们将按照 Zones.json 中所述的顺序排列它们。
4. server \ SCUM \ Saved \ Config \ windows server \ Loot \ Spawners \ Presets \ Override \ A2 \ Tisno \ tisno port.

> Server > SCUM > Saved > Config > WindowsServer > Loot > Spawners > Presets > Override > A2 > Tisno > TisnoPort
--

## 修改生成器预设：

- 有 4 种不同的方法可以在特定的道具预置中修改战利品：固定道具，物品，节点和子预置。
- 我们将一步一步地解释如何使用其中的每一个，以及它们的含义和用法。

## 固定项目

当您希望生成列表中的所有项目时，可以使用 FixedItems。

- FixedItems 忽略除 InitialDamage、RandomDamage、InitialUsage、RandomUsage 和 PostSpawnActions 之外的所有 spawner 预设设置。
- 使用“FixedItems”复制粘贴预设的 spawner 链接：<https://github.com/CheeseKing/Loot-Modifiers/blob/main/FixedItems>

1. 打开我们已经在“Override”文件夹中提取的任何 spawner 预设，我们已经在位于以下位置的“TisnoPort”文件夹中选择了 Street-Residential-Examine \_ Trash \_ Container \_ red . JSON:

server \ SCUM \ Saved \ Config \ windows server \ Loot \ Spawners \ Presets \  
Override \ A2 \ Ti Sn0 \ TisnoPort

```
1 {  
2   "Nodes": [  
3     {  
4       "Rarity": "Uncommon",  
5       "Ids": [  
6         "ItemLootTreeNodes.Trash"  
7       ]  
8     }  
9   ],  
10  "Probability": 20,  
11  "QuantityMin": 1,  
12  "QuantityMax": 1,  
13  "AllowDuplicates": false,  
14  "ShouldFilterItemsByZone": true,  
15  "InitialDamage": 50,  
16  "RandomDamage": 40,  
17  "InitialUsage": 50,  
18  "RandomUsage": 40,  
19  "PostSpawnActions": [  
20    "SetAmmoAmount_SmallStash",  
21    "SetCashAmount_SmallStash",  
22    "SetClothesDirtiness_DirtyClothes"  
23  ]  
24 }
```

2. 这个 spawner 预置里面有一个特定的节点，可以放下很多东西，但是我们想要一个特定的项目或者多个项目。
3. 我们将从

[https://github.com/cheese kingu/Loot-Modifiers/blob/main/fixed\\_items](https://github.com/cheese kingu/Loot-Modifiers/blob/main/fixed_items)

```
1 {  
2   "FixedItems": [  
3     "Weapon_M82A1",  
4     "Weapon_MK18",  
5     "2H_Katana"  
6   ],  
7   "QuantityMin": 1,  
8   "QuantityMax": 2,  
9   "AllowDuplicates": true,  
10  "ShouldFilterItemsByZone": false,  
11  "InitialDamage": 0,  
12  "RandomDamage": 0,  
13  "InitialUsage": 0,  
14  "RandomUsage": 0  
15 }
```

4. 我们已经给产卵器预置添加了三个项目。为了立即对此进行测试，您可以使用以下命令：`# reloadlootcustomizationsandresetspowers`。现在，当我们在提斯诺波特掠夺红色容器时，你将得到如下所示的物品：



## 项目

- 当你想添加一些物品时，你也需要给它们分配一个稀有性
- 拷贝粘贴带有“项目”的预置的链接：<https://github.com/CheeseKingU/Loot-Modifiers/blob/main/Items>
- 内含“物品”的产卵器预设取决于它们的稀有度、概率、最小数量和最大数量

1. 我们将以上面链接中的产卵器预设为例。



2. 我们将把它添加到我们已经在上面的步骤中提取的一个 spawner 预设中。
3. 去...  
server \ SCUM \ Saved \ Config \ windows server \ Loot \ Spawners \ Presets \ Override \ A2 \ Ti Sn0 并选择任何 spawner 预设。
4. 我们会选择风景-树叶-检查\_芦荟\_芦荟。
5. 复制上面链接中的代码，并替换  
风景-树叶-检查\_芦荟\_芦荟。  
-现在我们在 Tisno 地区掠夺的每一个芦荟都会有这个产卵器预置：



```
1 {
2   "Items": [
3     {
4       "Rarity": "Abundant",
5       "Id": "Weapon_M82A1"
6     },
7     {
8       "Rarity": "Common",
9       "Id": "1H_Hatchet"
10    },
11    {
12      "Rarity": "Uncommon",
13      "Id": "2H_Katana"
14    }
15  ],
16  "Probability": 20,
17  "QuantityMin": 1,
18  "QuantityMax": 2,
19  "AllowDuplicates": true,
20  "ShouldFilterItemsByZone": false,
21  "InitialDamage": 0,
22  "RandomDamage": 0,
23  "InitialUsage": 0,
24  "RandomUsage": 0
25 }
```

6. 当你在选定的区域搜索芦荟后，首先你有机会获得由概率决定的战利品。
7. 如果您将获得任何项目，数量取决于您在“数量”中输入的内容  
“QuantityMax”，在这种情况下，您将获得1到2个项目。



8. 稀有程度较低的物品会有较高的掉落几率，如稀有程度中所述。



## 子重置

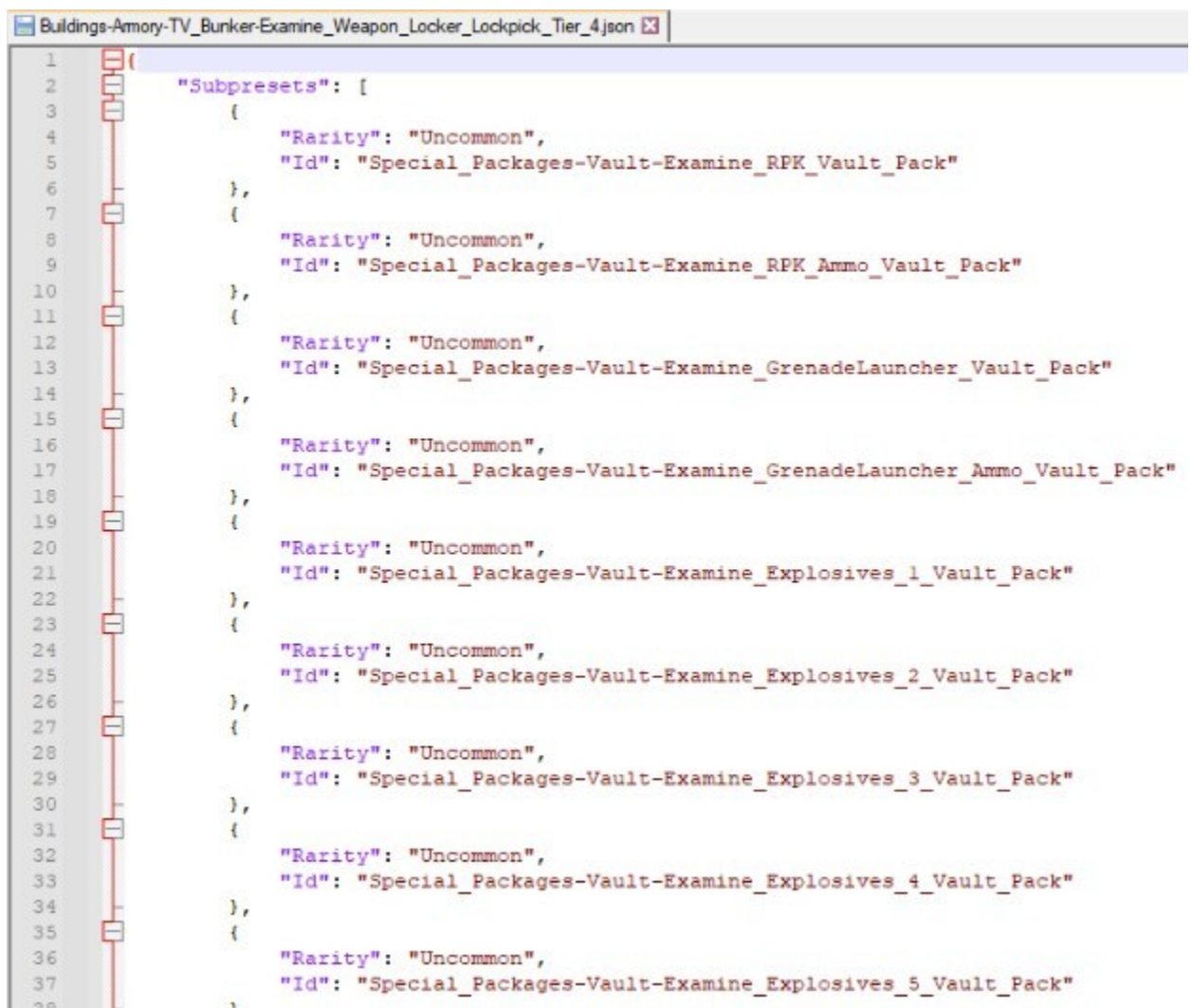
- 如果您想要在某个产卵器预设中有多个带有任何种类产卵预设的预设，使用子预设是最方便的。在我们的例子中，我们在子重置中使用了 FixedItems。
- 如果子重置没有指定稀有性，默认情况下它们是不常见的。如果你想给它们分配稀有度来增加或减少机会，你应该按照下面的描述去做。

- 复制带有“子预置”的预置的粘贴链接：

<https://github.com/CheeseKingu/Loot-Modifiers/blob/main/Subpresets>

### 1. 我们要用

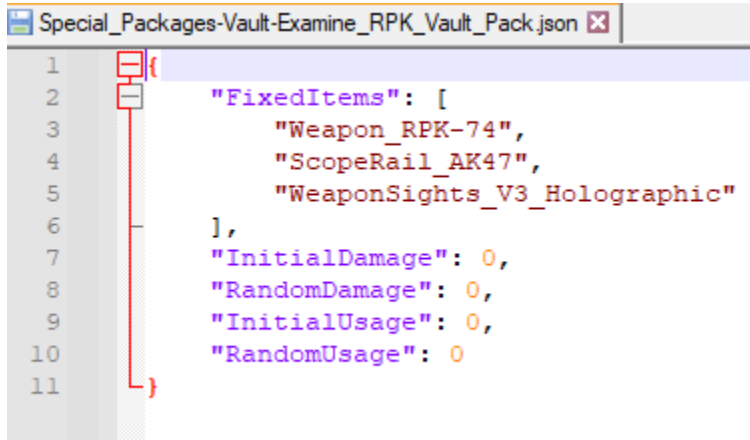
建筑-军械库-电视\_掩体-检查\_武器\_锁柜\_开锁\_第4层为例



```
1  {
2      "Subpresets": [
3          {
4              "Rarity": "Uncommon",
5              "Id": "Special_Packages-Vault-Examine_RPK_Vault_Pack"
6          },
7          {
8              "Rarity": "Uncommon",
9              "Id": "Special_Packages-Vault-Examine_RPK_Ammo_Vault_Pack"
10         },
11         {
12             "Rarity": "Uncommon",
13             "Id": "Special_Packages-Vault-Examine_GrenadeLauncher_Vault_Pack"
14         },
15         {
16             "Rarity": "Uncommon",
17             "Id": "Special_Packages-Vault-Examine_GrenadeLauncher_Ammo_Vault_Pack"
18         },
19         {
20             "Rarity": "Uncommon",
21             "Id": "Special_Packages-Vault-Examine_Explosives_1_Vault_Pack"
22         },
23         {
24             "Rarity": "Uncommon",
25             "Id": "Special_Packages-Vault-Examine_Explosives_2_Vault_Pack"
26         },
27         {
28             "Rarity": "Uncommon",
29             "Id": "Special_Packages-Vault-Examine_Explosives_3_Vault_Pack"
30         },
31         {
32             "Rarity": "Uncommon",
33             "Id": "Special_Packages-Vault-Examine_Explosives_4_Vault_Pack"
34         },
35         {
36             "Rarity": "Uncommon",
37             "Id": "Special_Packages-Vault-Examine_Explosives_5_Vault_Pack"
38         }
39     ]
40 }
```

- ### 2. 我们在这里使用“Subpresets”，因为我们想要“FixedItems”的多个预置卵

3. 它们都一样稀有，所以它们都有同样的机会产卵。
4. 如果我们想要更改“子重置”或者想要查看它们包含的内容，我们需要在 Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Spawners \ Presets \ Default 中找到它们
5. 我们将采取特殊\_包-保险库-检查\_ RPK \_保险库\_包，并打开它在“默认”文件夹



6. 如果它将产生这个“Subpreset”，我们将得到包含在其中的“FixedItems”。
7. 要更改它们包含的内容，您可以简单地打开一个子重置并更改项目。

## 节点

- 关于如何完全使用节点的链接:拾取树节点
- 我们将提供两种方法来说明如何在你的 spawner 预设中使用它们。
- 为您的产卵器预设复制粘贴节点链接:  
<https://github.com/CheeseKing/Loot-Modifiers/blob/main/Nodes>
- 由稀有性分隔的多个节点:  
<https://github.com/CheeseKing/Loot-Modifiers/blob/main/MultipleNodes>

## 按稀有程度分组

- 您可以按稀有度对节点进行分组，当您的产卵器预设滚动以确定您将获得哪个节点时，您可以将多个节点添加到相同的稀有度中，例如：

```
{
  "Nodes": [
    {
      "Rarity": "Uncommon",
      "Ids": [
        "ItemLootTreeNodes.Workshop.Tools",
        "ItemLootTreeNodes.Workshop.WorkClothes.Feet",
        "ItemLootTreeNodes.Workshop.WorkClothes.Torso",
        "ItemLootTreeNodes.Workshop.WorkClothes.Legs",
        "ItemLootTreeNodes.Workshop.Crafting",
        "ItemLootTreeNodes.Workshop.LightsFire"
      ]
    },
    {
      "Rarity": "Rare",
      "Ids": [
        "ItemLootTreeNodes.Trash"
      ]
    }
  ],
}
```

- 您将有更大的机会将节点分组到“不常见”中稀有性比用“稀有”分开的节点稀有性高。
- 当一个稀有度中有多个节点时，里面的节点仍然有它们自己的稀有度，我们在分组中指定的稀有度只是决定了分组节点的稀有度。
- 在这种情况下，我们有：

-车间。“不常见”的工具，

```
  "Name": "tools",
  "Rarity": "Uncommon",
  "Children": [
```

-车间。“普通”的工作服

```
  "Name": "WorkClothes",
  "Rarity": "Common",
  "PostSpawnActions": [
    "SetClothesDirtiness_DirtyClothes"
  ],
```

-车间。制作“普通”的工艺品

```
"Name": "Crafting",  
"Rarity": "Common",  
"Children": [  
  {
```

-车间。“非常罕见”的火光

```
"Name": "LightsFire",  
"Rarity": "VeryRare",  
"Children": [  
  .
```

- 在我们的 spawner 预设选择了这组节点后，它仍然会检查里面提到的节点的稀有性，因此根据它们的稀有性，某些项目的机会会有所不同。
- 为了给某些节点分配期望的稀有度，请检查下面的例子。

## 按稀有程度分类

- 如上文所示，您可以按稀有程度来分隔节点，但下面是一个包含大量节点的示例：

```
"Nodes": [
  {
    "Rarity": "Uncommon",
    "Ids": [ "ItemLootTreeNodes.Workshop.Tools" ]
  },
  {
    "Rarity": "Common",
    "Ids": [ "ItemLootTreeNodes.Workshop.WorkClothes.Feet" ]
  },
  {
    "Rarity": "Rare",
    "Ids": [ "ItemLootTreeNodes.Workshop.WorkClothes.Torso" ]
  },
  {
    "Rarity": "VeryRare",
    "Ids": [ "ItemLootTreeNodes.Workshop.WorkClothes.Legs" ]
  },
  {
    "Rarity": "Uncommon",
    "Ids": [ "ItemLootTreeNodes.Workshop.Crafting" ]
  },
  {
    "Rarity": "Uncommon",
    "Ids": [ "ItemLootTreeNodes.Workshop.LightsFire" ]
  }
]
```

- 产卵者预置将掷骰子来决定从列表中选择哪个节点，稀有度越低，获得的几率越高。
- 在 spawner 预设选择你将在这里得到哪个节点后，你只能得到包含在提到的节点中的项目。

## 结合

- 可以用多种方法修改生成器预设。
- 最好的例子就是我们在游戏中已经有的，我们将开放 `landscape-Examine _ ground rocks _ snow` . JSON 位于此处：



\\Loot\\Spawners\\Presets\\Default



```
1 {
2   "Items": [
3     {
4       "Rarity": "Abundant",
5       "Id": "Stone_Small"
6     },
7     {
8       "Rarity": "Common",
9       "Id": "Stone"
10    },
11    {
12      "Rarity": "Uncommon",
13      "Id": "Snowballs_01"
14    },
15    {
16      "Rarity": "VeryRare",
17      "Id": "Snowballs_02"
18    }
19  ],
20  "FixedItems": [
21    "Stone_Small",
22    "Snowballs_01"
23  ],
24  "QuantityMin": 1,
25  "QuantityMax": 2,
26  "AllowDuplicates": true,
27  "ShouldFilterItemsByZone": false,
28  "InitialDamage": 0,
29  "RandomDamage": 0,
30  "InitialUsage": 0,
31  "RandomUsage": 0
32 }
```

- 这个 spawner 预置说你会一直得到“石头\_小”，“雪球\_01”如“FixedItems”中所说。如“最小数量”和“最大数量”中所述，您还将从“项目”部分获得 1 到 2 个项目：
- 您可以将任意数量的方法添加到任何 spawner 预置中，  
将“FixedItems”与“Items”/“sub presets”/Nodes”结合使用最为方便。

## 风景、木偶、货物和农业空投：

- 当你导出特定地点的产卵器预设时，它不会给你木偶，货物，农场和哨兵掉落产卵器预设，也不会给你一些风景产卵器预设，如岩石
  - 为了修改这些 spawner 预设，您需要将它们从  
\\Loot\Spawners\Presets\Default 文件夹，并将它们添加到您提取的位置(如果您将它们添加到覆盖，它们将全局更改)
  - 请注意，有很多这样的产卵器预置。举个例子：有 53 个木偶产卵器预置，如果你想让你的木偶什么都不掉或者掉某个东西。你需要改变他们所有人
1. 转到此位置：  
server \ SCUM \ Saved \ Config \ windows server \ Loot \ Spawners \ Presets \ Default
  2. 我们将在这里进行一些预设：  
景观-检查\_基岩  
特殊\_包裹-货物\_掉落-检查\_阿斯瓦尔\_货物掉落人物-木偶-军事-  
检查\_SK\_军事\_僵尸\_03 农业-检查\_卷心菜
  3. 为了修改它们，您需要将整个预设复制粘贴到您想要的区域，在这种情况下，我们已经提取了 A2 sector Server \ SCUM \ Saved \ Config \ windows Server \ Loot \ Spawners \ Presets \ Override \ A2
  4. 我们将把它们放在 A2 扇区文件夹中，并且我们将使用我们在“FixedItems”中为这些预置中的每一个链接的代码  
<https://github.com/CheeseKing/Loot-Modifiers/blob/main/FixedItems>
  5. 现在我们已经用想要的“固定物品”替换了它们，我们在 A2 区搜索的每一块石头都会掉落一个 M82A1，一个 MK18 和一个武士刀
  6. 如果你搜索一个军事\_僵尸\_03，它也会掉落那些物品
  7. 如果你找到了 ASVal\_Cargodrop 预置，你也会得到那 3 个物品
  8. 你种植的每一棵卷心菜在成熟时都会掉落这三样东西，所以你基本上可以种植任何你放进去的东西
  9. 你可以使用任何种类的战利品修改器给这些预设，如上面的  
[修改生成器预设](#)



## 繁殖后操作

- 如果我们想修改将要掉落物品，那么在“Post spawn actions”行中会添加Post spawn 动作
- 使用多个“PostSpawnActions”的产卵器预设的链接：<https://github.com/CheeseKing/Loot-Modifiers/blob/main/PostSpawnActions>

## 生成后操作列表：

- 如果该物品是钥匙卡，指定它可以打开最近的地堡。
- 如果物品是弹药，将弹药数量设置为口径的 50-100% (例如:cal\_22 最大数量是 20，它将是 10-20/20)。
- 如果物品是弹药，将弹药数量设置为口径的 0-35% (例如:cal\_22 最大数量是 20，它将是 0-7/20)。
- SetCashAmount\_BigStash -如果项目是现金，将其值设置为 200-500。
- SetCashAmount\_MediumStash -如果物品是现金，将其值设置为 50-200。
- SetCashAmount\_SmallStash -如果项目是现金，将其值设置为 1-100。
- 如果物品是衣服，设置肮脏度为 93-96%
- 如果物品是衣服，设置脏度为 60-85%
- 如果物品是衣服，设置脏度为 0-20%
- SetUsage\_Max -所有有使用次数的物品将产生 0 次使用次数，例如水瓶将产生 0/5 次使用次数，Pet 瓶产生 0/20 次使用次数，等等。

种子生成后操作的示例：

1. 我们将以街道-住宅-检查\_汽车\_残骸\_废料为例：



```
1 {
2   "Nodes": [
3     {
4       "Rarity": "Uncommon",
5       "Ids": [
6         "ItemLootTreeNodes.Barn.Tools",
7         "ItemLootTreeNodes.Barn.Crafting",
8         "ItemLootTreeNodes.Barn.LightsFire",
9         "ItemLootTreeNodes.Barn.WorkClothes"
10      ]
11    }
12  ],
13  "Probability": 15,
14  "QuantityMin": 1,
15  "QuantityMax": 2,
16  "AllowDuplicates": false,
17  "ShouldFilterItemsByZone": true,
18  "InitialDamage": 0,
19  "RandomDamage": 50,
20  "InitialUsage": 0,
21  "RandomUsage": 50,
22  "PostSpawnActions": [
23    "SetClothesDirtyiness_ResidentialClothes"
24  ]
25 }
```

2. 这个 spawner 预置说，如果掉落物品是衣服，这个物品的脏度将被设置为 0-20% 脏度，如上所述。

3. 在该生成器预设中使用多个“PostSpawnActions”的另一个示例：  
建筑-车库-住宅-检查\_衣柜\_储物柜：

```
Buildings-Garage-Residential-Examine_Wardrobe_Locker.json
1  {
2      "Nodes": [
3          {
4              "Rarity": "Uncommon",
5              "Ids": [
6                  "ItemLootTreeNodes.Garage.WorkClothes",
7                  "ItemLootTreeNodes.Garage.Other.Ammo",
8                  "ItemLootTreeNodes.Garage.Other.Backpacks.WaistBags",
9                  "ItemLootTreeNodes.Garage.Other.Backpacks.School",
10                 "ItemLootTreeNodes.Garage.Other.Backpacks.Plain"
11             ]
12         }
13     ],
14     "Probability": 20,
15     "QuantityMin": 1,
16     "QuantityMax": 1,
17     "AllowDuplicates": false,
18     "ShouldFilterItemsByZone": true,
19     "InitialDamage": 10,
20     "RandomDamage": 30,
21     "InitialUsage": 1,
22     "RandomUsage": 30,
23     "PostSpawnActions": [
24         "SetAmmoAmount_SmallStash",
25         "SetCashAmount_SmallStash",
26         "SetClothesDirtiness_DirtyClothes"
27     ]
28 }
```

4. 如果物品是弹药，设置弹药数量为口径的 0-35%。  
5. 如果物品是现金，将其值设置为 1-100。  
6. 如果物品是衣服，设置脏度为 60-85%

# 放弃

本文档中的第三方链接可能会将您引向与我们无关的第三方网站。我们不负责检查或评估内容或准确性，我们不保证也不会对任何第三方材料或网站或第三方的任何其他材料、产品或服务承担任何责任。

对于与购买或使用商品、服务、资源、内容或与任何第三方网站相关的任何其他交易相关的任何伤害或损害，我们概不负责。请仔细阅读第三方的政策和惯例，并确保在参与任何交易之前理解这些政策和惯例。有关第三方产品的投诉、索赔、疑虑或问题应提交给第三方。