

库存 版本 1.9.0  
资产 向导 -



## 引言

资产清单是您的终极资产伴侣: 在当前项目之外快速搜索资产。在你没有导入的情况下购买或下载的资产中找到内容, 只需点击一下就可以找到单个文件。

消除寻找声音文件、纹理或模型的耗时任务, 你知道你已经购买了, 但是它们隐藏在你的许多资产商店购买的其中之一。资产清单提供了你拥有的所有资产的完整列表, 包括它们的内容。



# 目录

---

引言 1

特点 3

安装 5

开始 6

搜索面板 7

包装面板 12 选项卡

报告面板 16

设置面板 18

资产预览 23

高级功能 24

配置 25

技术细节 26

限制 26

常见问题 27

第三方 29

建议 30

支持联系人 31





# 特点

该工具旨在提供您一直想要的资产商店和包管理器。有效地查找资产。搜索你购买的商品。包括你从 Humble Bundle 或其他地方下载的自定义软件包。以及注册表中的软件包。通过标签管理一切。执行批量操作。一次性导入多个资产和包。识别项目中使用过的资产。使用特定资产标准进行搜索。还有更多。

## 强大的搜索

浏览并且在你购买的资产商店软件包中找到任何东西，而不需要导入。快速预览音频文件。使用资产类型、标签、图像尺寸、音频长度、颜色等缩小搜索范围。排除你不想看到的项目。保存和回收搜索。

## 简单的设置

立即开箱工作。许多可选的视图和配置选项。无 hassle 索引。开始，停止和恢复在任何时候。使用 unity 2019.4 及以上版本。Windows, Mac & Linux 操作系统。闪电-快速索引和搜索。

## 智能进口

只导入你需要的东西，而不是整个包。自动确定资产依赖性，以进口复杂的预制件和材料。节省空间，减少项目中的混乱。一次导入多个软件包。自动将导入的资产存储在一个特定的子文件夹中，并保持 Assets 根清洁。

## 来源很多

完整的资产库：自动索引资产商店采购。触发器下载丢失的资产。处理注册表中的软件包。支持自定义文件夹搜索从其他地方下载的 unity 包。索引文件夹和 Zip 文件包含任意的媒体文件，如 3d 模型，音频库，纹理和更多。自动生成预览。



## 组织起来

自动从资产商店导入标签。使用附加标签来有效地对资产进行分组。将标签分配给包或者包内的单个文件。按标签分配颜色和组。排除不需要的项目。执行批量操作。一次性导入多个资产和包。基于 Package2Folder，允许将包导入到自定义的子文件夹中。

## 反向查找

快速识别项目中使用过的资产和包。

## 持续更新和支持

接受定期更新，优化和新特性。超快的支持。在一个伟大的不和谐社区讨论想法。

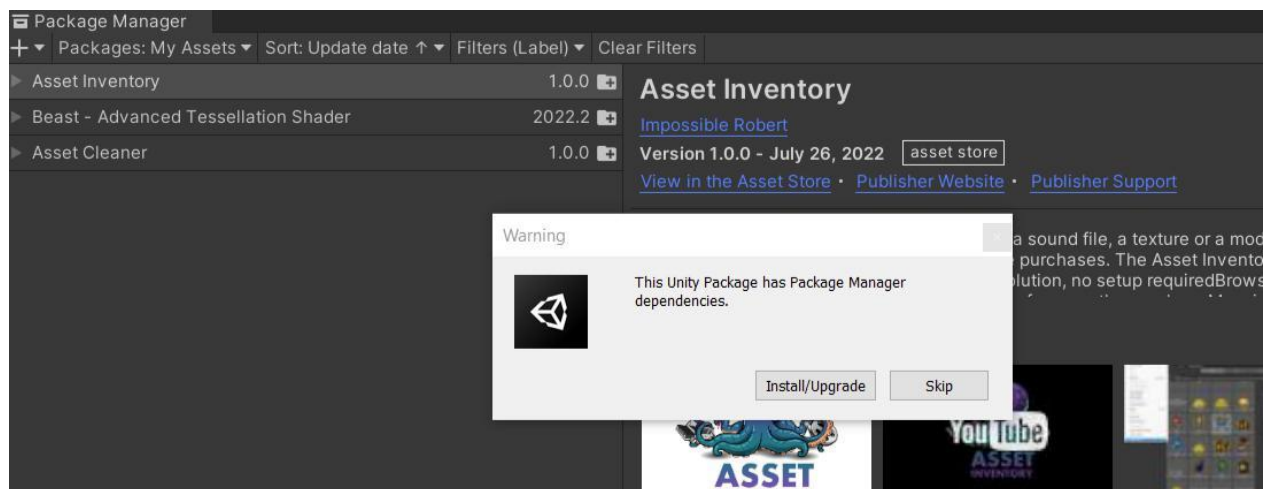


# 安装

强烈建议在安装新版本之前删除任何以前版本的资产清单。这样做的最佳方法是在启动 unity 之前删除完整的 Asset Inventory 文件夹，因为如果 unity 处于活动状态，有时库不能被删除。

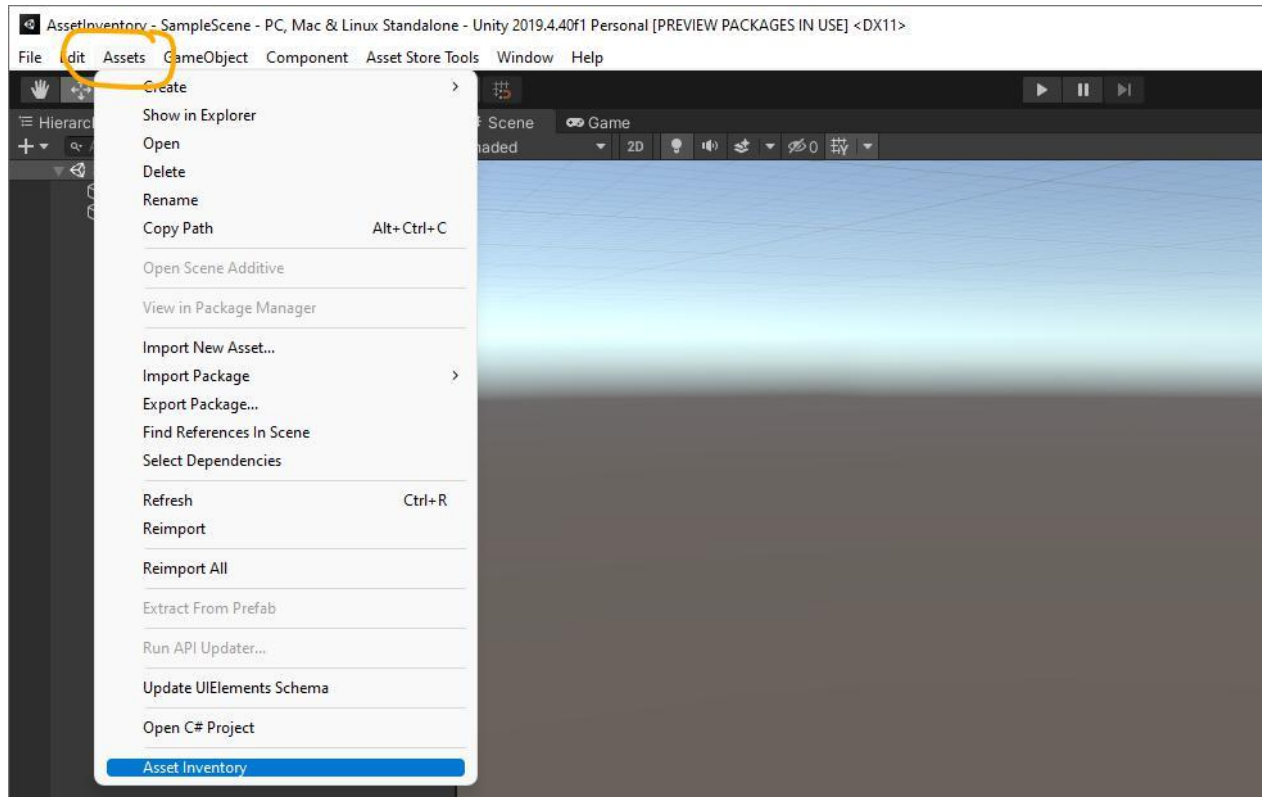
Pro 提示: 在一个新的空 unity 项目中运行 Asset Inventory，用于初始索引，并使用最新的可用 Alpha/Beta。索引结果将被存储在一个中心位置，以便在任何其他项目中重用。这将为您提供最高的索引性能，因为不需要刷新其他资产/脚本，而且 unity 的新版本通常会为对象和声音导入器带来巨大的性能改进(在某些情况下是 10 倍)。

通过 unity Package Manager 安装包。当被问及是否安装附加依赖项时，选择 Install/Upgrade，否则该资产将无法工作。



# 开始了

通过“资产”菜单打开资产清单。



不需要手动设置。唯一需求的是在提取和浏览过程中存储预览图像和临时文件的相当大的磁盘空间。

一旦工具打开，按下大的开始索引按钮，等待结果进来。当索引完成时，删除已经积累了大量临时不需要的数据的项目 Library 文件夹是有益的。

无论何时购买新包或向文件夹添加新资产，请确保在“设置”下按一次“更新”按钮，以便更新索引指数。

**专业提示 1:** 激活设置下的下载资产索引期权选项，以索引您的完整库，无需首先手动下载资产。

**Pro Tip 2:** UI 只显示核心功能性(简单模式)。你可以通过按住 CTRL 键来访问许多额外的选项。



# 搜索面板

一旦资产被索引，就可以在搜索中找到它。搜索将自动开始时，键入。默认情况下，输入的词都需要匹配搜索结果，但不一定是相同的顺序。如果确切的短语应该匹配，那么搜索的前缀是 ~ 。

例子:

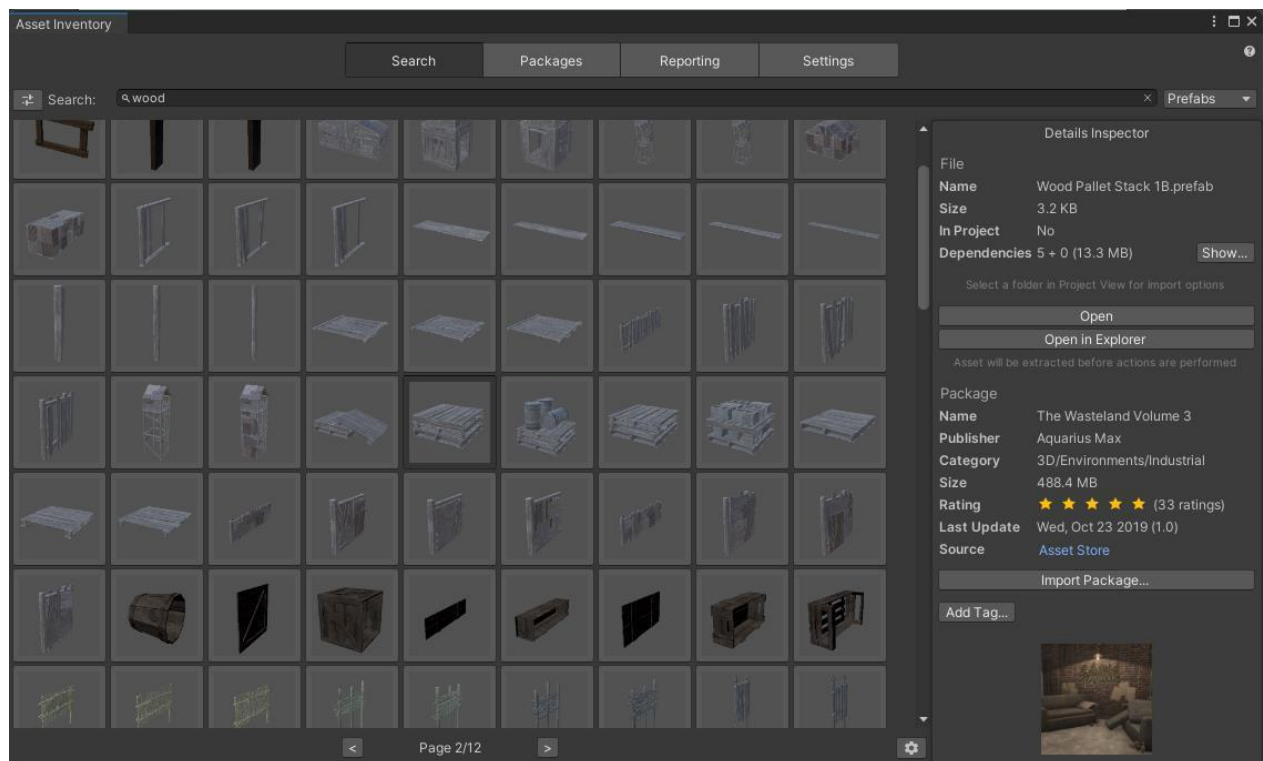
*"汽车内饰" 将返回 "CarBlueInterior.fbx" 和 "InteriorCarDes.png" 等结果*

*" ~ Car interior" 不会返回以上结果， 只会返回 " Car interior. fbx" 这样的结果*

*" car + interior-fbx" 将返回匹配 " car" 和 " interior" 但不匹配 " fbx" 的结果*

还有一个可用的专家搜索(以 " =" 开头的搜索)，稍后将介绍。

可以选择其他过滤器进一步缩小结果范围。一旦一个资产被选中，关于文件和文件所包含的包的详细信息就会显示在右边。





选择一个音频文件将自动播放它。这样很容易快速预览音频文件。

单击搜索前面的 **Filter** 图标将打开过滤器栏。它包含额外的过滤器和媒体特定的属性来过滤特定的图像尺寸或音频长度。如果结果比输入的大或小，使用每个值前面的按钮可以切换。过滤器下拉菜单只会显示可用的值。

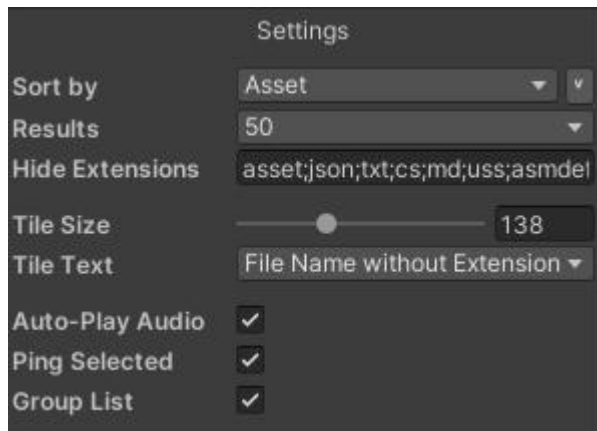
在注册表软件包的情况下，只有最新版本的软件包将被索引。



筛选条还包含“已保存搜索”部分。这样可以保留当前的搜索过滤器，以后可以轻松地召回这些过滤器。搜索结果不会被持久化，而是在数据库中实时显示。

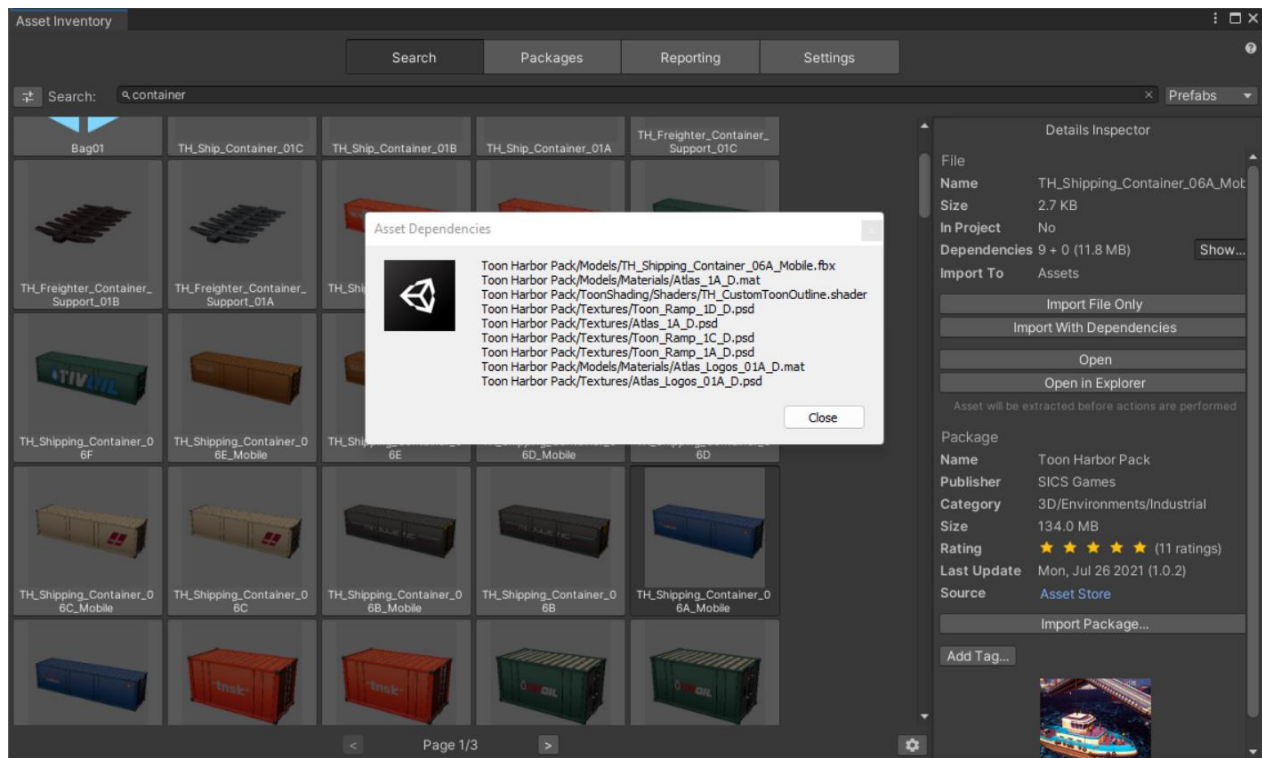


使用右下角的设置图标可以打开额外的视图设置。指定瓷砖大小，在瓷砖上显示哪些文本，以及是否应该在选择时自动定位资源。



### 进口资产

在 unity 项目视图中选择一个文件夹会在搜索中选择一个资产时弹出导入选项。资产可以单独导入，也可以与所有检测到的依赖项一起导入。后者会自动创建一个带有资产名称的子文件夹，并在其中存储所有文件。



如果存在脚本依赖项，也可以导入这些依赖项。由于脚本中不可预见的依赖关系，这通常只适用于自包含的脚本。否则，可能会有编译错误。

单击依赖项信息后面的 Show 按钮将显示依赖项信息的详细信息，列出所有文件、它们的大小以及它们是否已经在项目中。

也可以通过双击某个项或将其拖动到 Project 窗口中来触发 Import。

## 专家搜索

几乎可以使用 sqlite3 的全部特性集进行搜索。这个模式在以“=”开始搜索时被激活。之后，数据库的字段和条件可以被陈述。

### 示例

`= Asset. PackageSize > 0 和 AssetFile. Type = “wav”`

参考下面的表格来获得可用字段的概述。



#### Asset

Id	integer
AssetSource	integer
Location	varchar
ForeignId	integer
PackageSize	integer
PreviewImage	varchar
SafeName	varchar
DisplayName	varchar
SafePublisher	varchar
DisplayPublisher	varchar
SafeCategory	varchar
DisplayCategory	varchar
Slug	varchar
Revision	integer
Description	varchar
KeyFeatures	varchar
CompatibilityInfo	varchar
SupportedUnityVersions	varchar
Keywords	varchar
Version	varchar
LastRelease	bigint
AssetRating	varchar
RatingCount	integer
MainImage	varchar
Requirements	varchar
ReleaseNotes	varchar
ETag	varchar
CurrentState	integer
OfficialState	varchar
IsHidden	integer
Exclude	integer
LastOnlineRefresh	bigint
License	varchar
LicenseLocation	varchar
IsLatestVersion	integer
PreferredVersion	varchar
Registry	varchar
Repository	varchar
PackageSource	integer
MainImageSmall	varchar
MainImageIcon	varchar
OriginalLocation	varchar
OriginalLocationKey	varchar
LatestVersion	varchar
CurrentSubState	integer

#### AssetFile

Id	integer
AssetId	integer
Guid	varchar
Path	varchar
SourcePath	varchar
Type	varchar
PreviewFile	varchar
Size	integer
Width	integer
Height	integer
Length	float
FileName	varchar
PreviewState	integer
DominantColor	varchar
DominantColorGroup	varchar

#### Tag

Id	integer
Name	varchar
Color	varchar
FromAssetStore	integer

#### TagAssignment

Id	integer
TagId	integer
TagTarget	integer
TargetId	integer

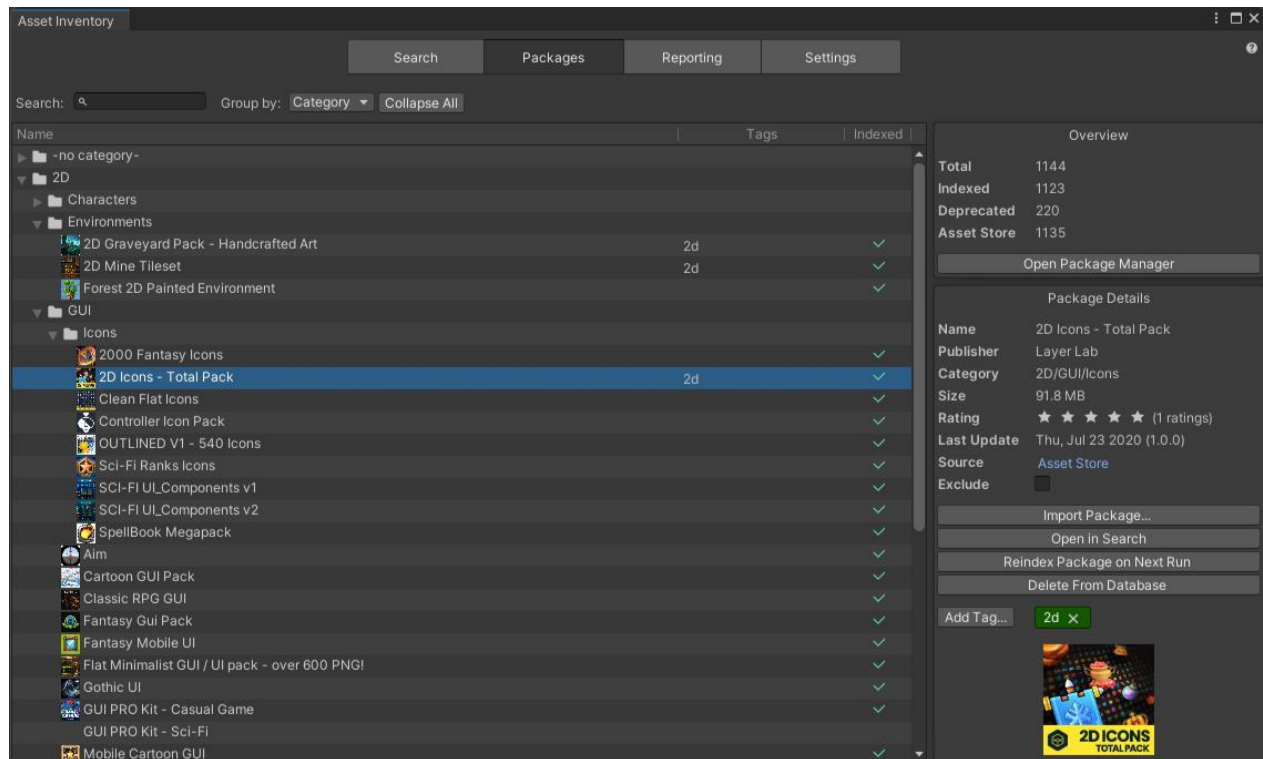


# 包装面板

软件包概览将显示所有找到的软件包的列表。这些可以来自多个来源:

- 会自动检索 assetstore 购买的列表
- 本地 unity 资产缓存被扫描
- 扫描本地 unity 包缓存(又称注册表包)
- 如果指定, 其他文件夹也会被扫描

该列表将指出其中哪些已被索引。要指数更多的软件包, 下载到缓存。Display 可以是一个简单的列表, 也可以按类别、发布者、状态或标签进行分组。



选择一个软件包后, 会出现多个选项。可以导入它。或者, 它也可以从索引中删除, 以便在下一次运行时触发“驯鹿”。这对于不正确的索引文件很有用。软件包也可以从数据库中完全删除, 这在清除本地缓存中过期的资产后很有用。双击任何软件包都会显示搜索中的内容。



根据包类型的不同, 可以看到其他选项:

#### 注册表软件包

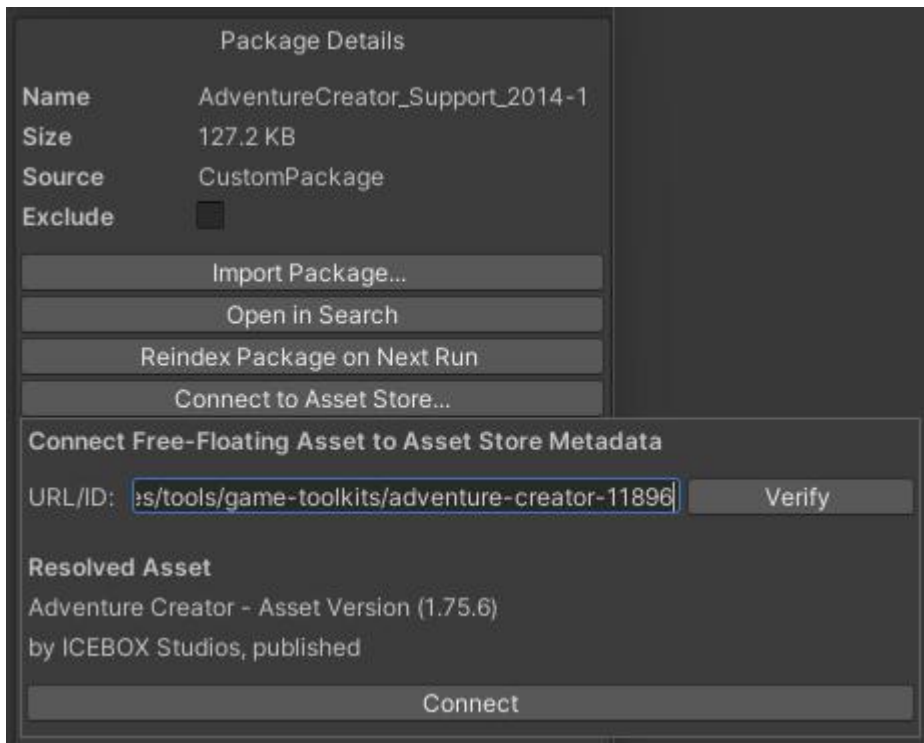
支持设置首选版本, 该版本将用作导入时的默认版本。

#### 自定义软件包

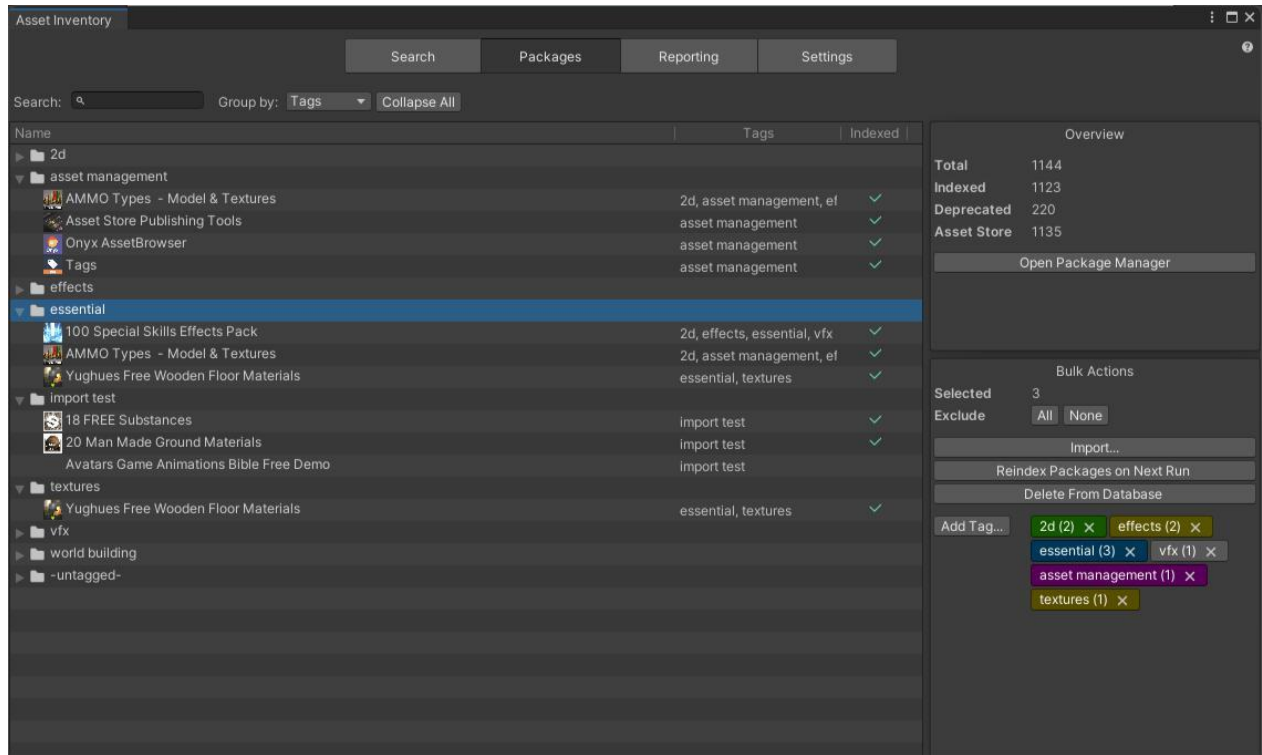
支持从 `assetstore` 连接到元数据。这将载入名称、描述、评级等, 尤其是在从其他来源(如 Humble Bundle)下载软件包到

仍然有所有可用的元数据。

支持再次从 `assetstore` 断开连接



标签将从资产商店导入，以防有任何设置在那里。此外，本地标签也可以在这里添加和删除(它们还没有同步到 **Asset Store**)。当选择树中的多个项目时，可以批量编辑标签。



添加标记时，可以通过小齿轮打开 Tag 管理窗口。在那里，标签可以被创建，着色，重命名和删除。

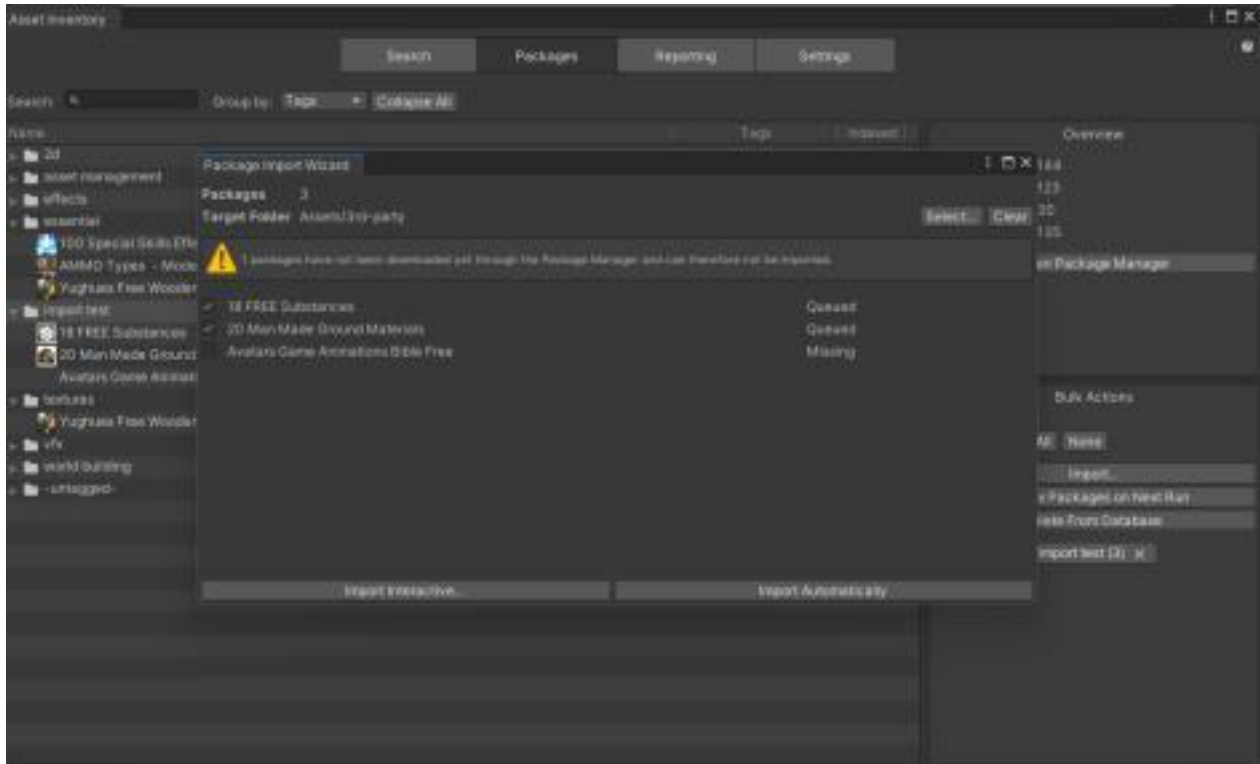
如果包不应该出现在搜索结果中，也不应该被索引，那么可以在包的详细信息中使用 Exclude 切换。

Bulk 选择还将显示压缩资产在下载之前的总大小，以便您可以估计临时下载所有用于索引的资产是否仍然适合硬盘驱动器。





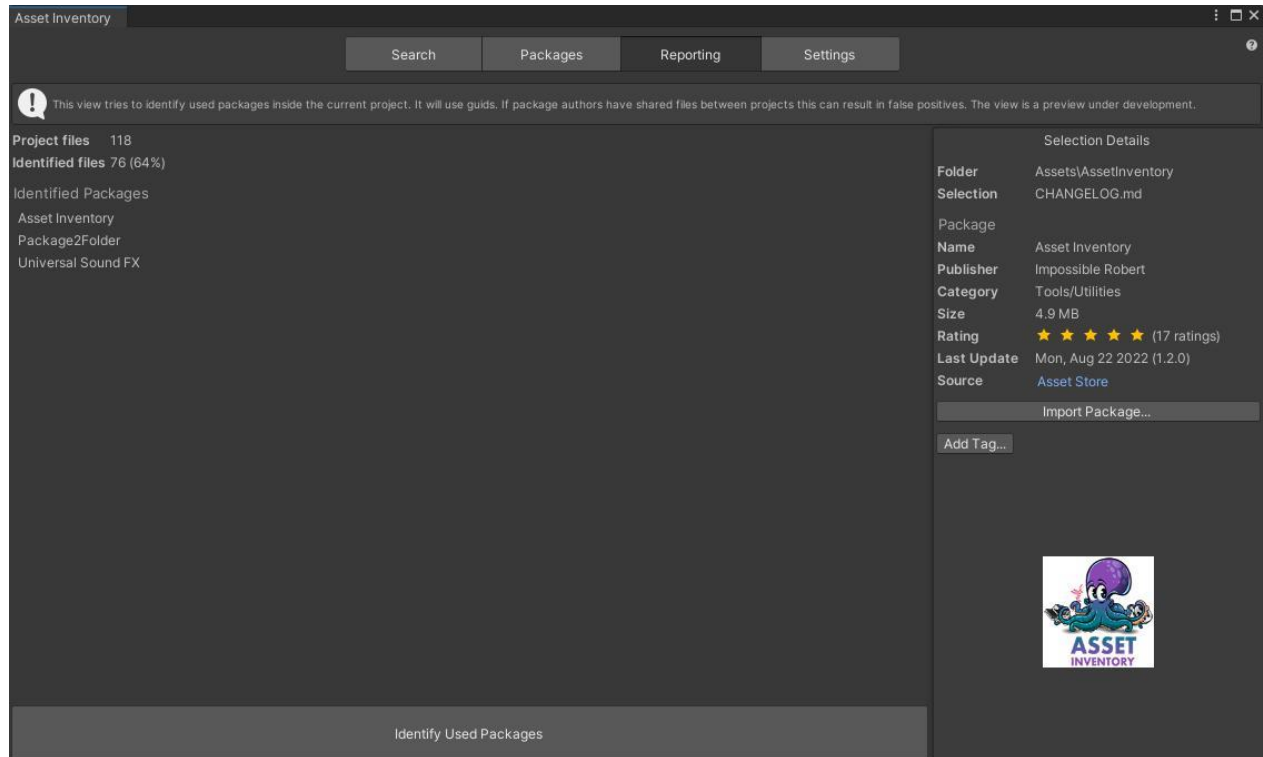
在选择多个包时，可以一个接一个地、批量地导入包。交互式 and 自动模式都可用。此外，还可以选择一个自定义的根文件夹，在该文件夹下可以导入资产。如果 **/Assets** 根目录应该保持整洁和有序，这一点尤其有用。来自注册中心的软件包将被添加到项目清单中。如果注册表包需要一个自定义范围的注册表，这也将自动添加。





# “报告”面板

这个模块将扫描整个项目，并尝试识别正在使用的包。它还支持导出数据。目前在预览阶段，这将演变成一个合规报告，以检查项目中的所有许可证是否都被考虑在内。



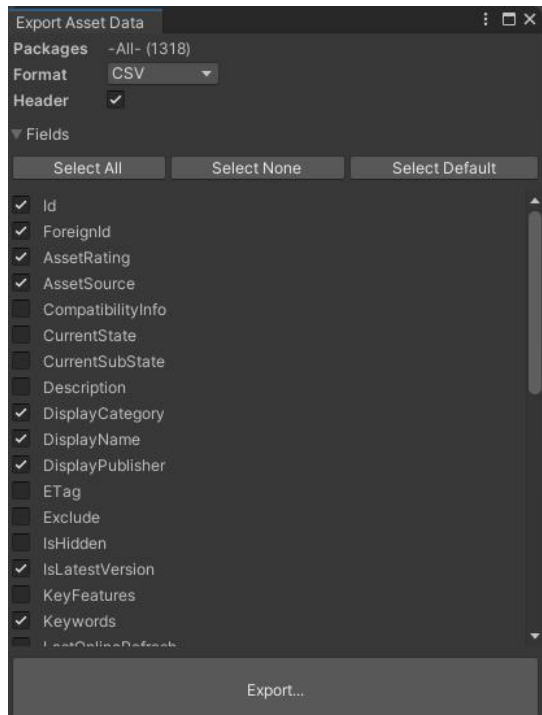
当在 unity 项目视图中选择一个资产时，报告面板将尝试识别相关的包。

## 数据导出

如果您想使用自己的分析或表示方式，可以将数据库内容导出为其他格式。目前支持的是：

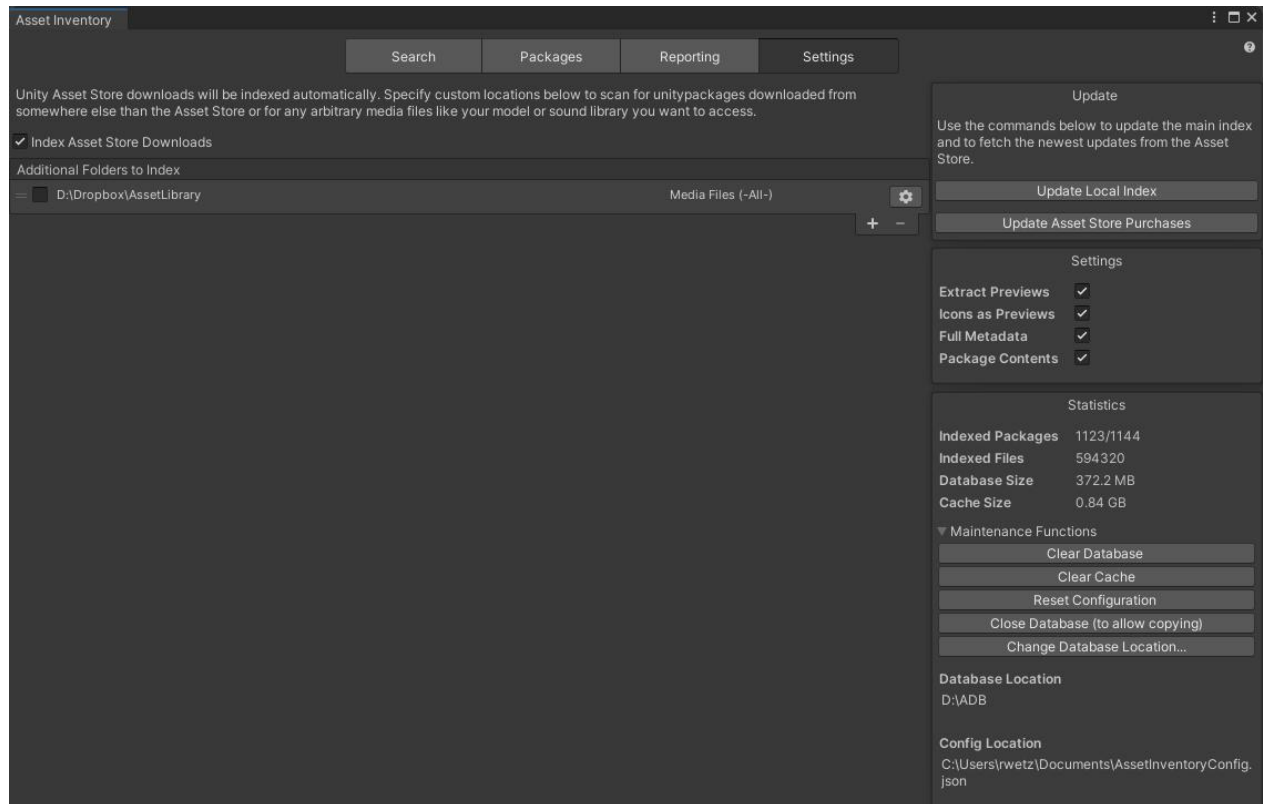
CSV: 可以很容易地在 Excel 和其他兼容软件中处理





# “设置”面板

使用此部分来配置应该索引的内容。索引可以在任何时候启动和停止，并且会在上次停止的地方再次启动。



资产商店的下载和包默认情况下是激活的。有两个选项可用:

指数所有已经下载到本地缓存。

这意味着您只需要在 **Package Manager** 窗口中单击 **Download** 按钮，或者在 **Asset Inventory** 中右键单击您拥有的每个资产(不需要导入它们)，就可以将它们用于索引(也可以进行批量选择)。

自动下载资产进行索引，然后再删除它们。

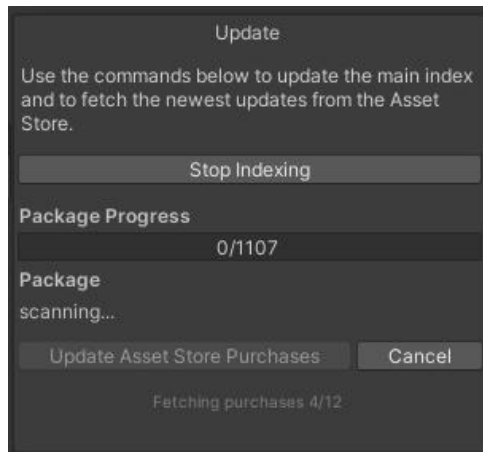
使用这个选项，您将索引所有您拥有的东西，而不需要下载所有的东西，节省了大量的磁盘空间，代价是由于所有必要的下载可能需要很长的索引期权。



在正常的更新周期中，以下操作按以下顺序进行:

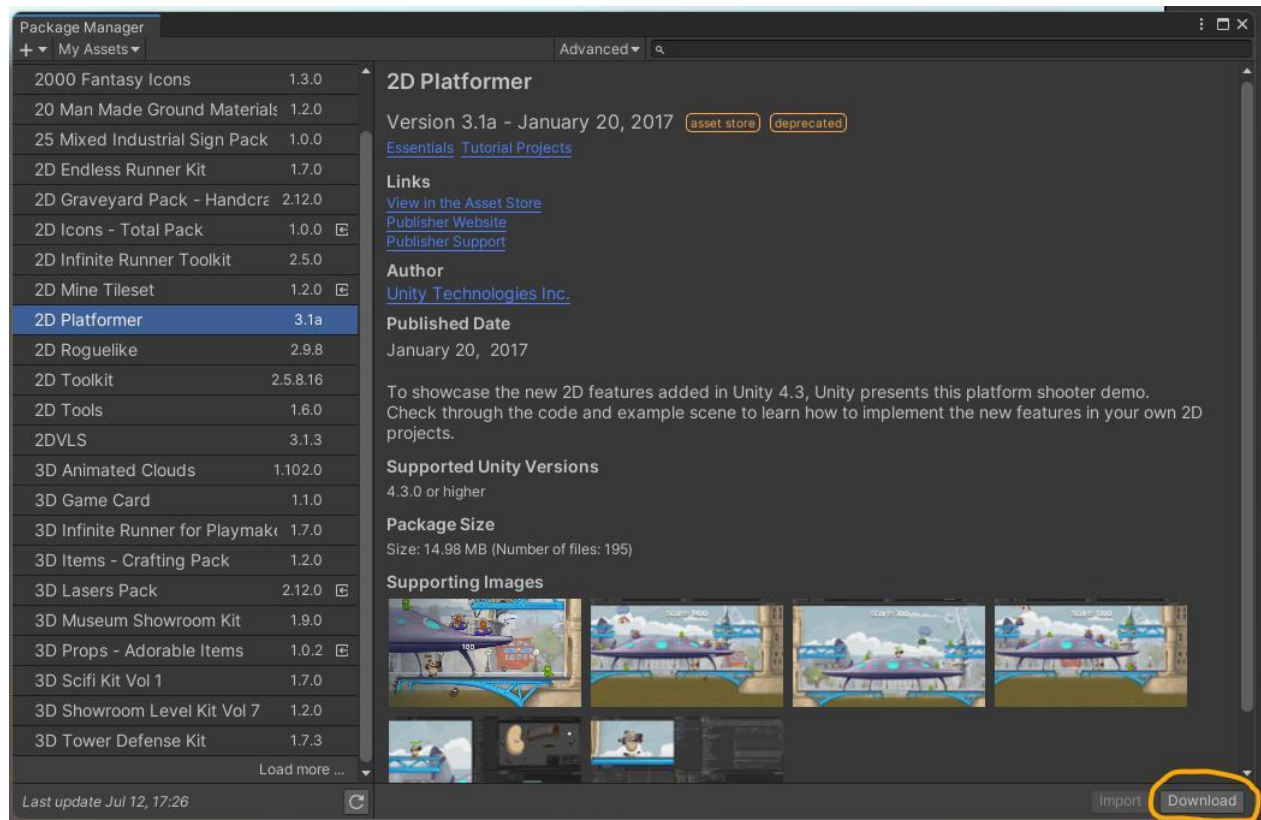
- 取出所有在资产商店购买的物品
- 获取所有购买和接收更新的资产的详细信息
- 并行扫描新的本地 assetstore 包和更新元数据
- 对注册表包也这样做
- 索引新的或更改的包的内容的指数
- 如果指定，索引任何额外的文件夹指数
- 为颜色搜索提取新资产的颜色
- 执行备份活动

所有的活动都是增量的，这意味着您可以在任何时候中断并重新启动这些活动，而且它们大多数都会继续它们停止的地方。



提示: 从 unity 版本 2022.1 开始，Package Manager 支持批量下载资源。一次性下载所有需要的资源。大容量下载也可以在 unity 2020 及以上版本的资产库存中找到。此外，你可以激活设置下的下载资产索引期权下载资产动态只是为了索引。之后它们会被再次删除以节省磁盘空间。





如果希望对从其他来源下载的 unity 包进行索引，只需将文件夹添加到附加文件夹列表中即可。如果一个文件夹以“ Asset Store-5”结尾。它将被认为是一个包含发布者/类别/资产布局的 unity 兼容结构。这是在 unity 2022.1 及更高版本中定义自定义资产缓存目录的最佳解决方案。如果设置了环境变量 `ASSETSTORE_cache_path`，它将覆盖默认的包缓存位置。

当从附加文件夹列表选择一个条目时，可以在右侧指定更多的选项。这样就可以选择搜索什么类型的文件：

Unity 包: 在文件夹中找到任何.unitypack 文件，并对内容进行索引

媒体文件: 允许指数所有其他种类，自由浮动的资产，如图像，模型，音频库等。

Zip 档案: 将提取档案的飞行和索引所有内容的指数

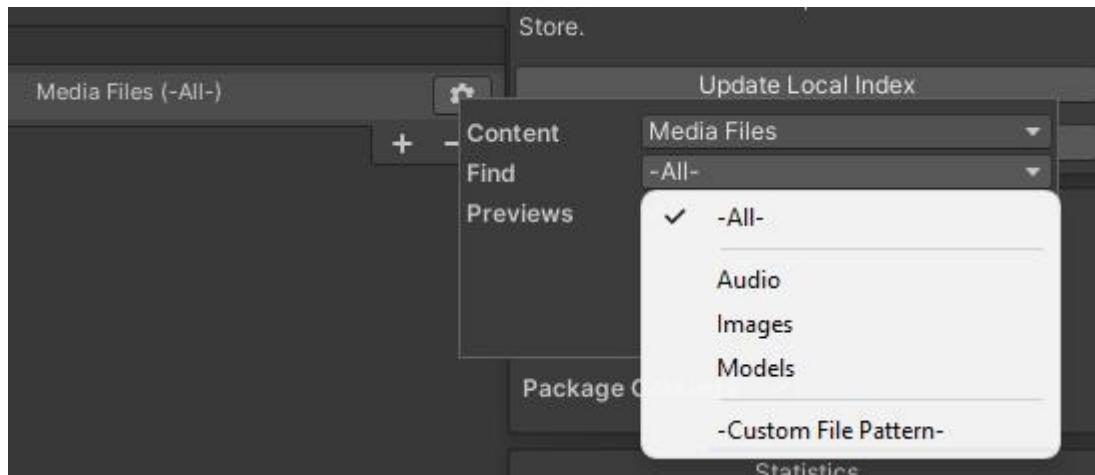
使用额外的媒体文件夹可以让 Asset Inventory 充当资产管理的首选地点，并且可以从单个一致的 UI 快速查找驱动器上的任何可用资产。媒体文件将没有连接的资产，但其他行为是相同的。如果



他们有一个。元文件旁边，他们的向导也将存储。处理其他文件夹的顺序可以通过在列表中向上或向下拖动来改变。停用一个附加文件夹不会从索引中移除它的指数。

一旦一个资产被索引，如果你只是想搜索它，它不需要保留在你的硬盘上。只有在导入的时候它才是可用的。

预览文件生成将通过临时将每个文件复制到当前 unity 项目中，让 unity 创建一个预览并再次删除它来实现。这个过程需要一点时间，但是可以让 Asset Inventory 显示预览图像，以及更多图像文件的额外元数据，而不仅仅是 png 和 jpg (例如 gif)。



## 备份

该工具支持将包复制到永久备份的专用位置。通过这种方式，您不会受到 unity 的软件包弃用的影响，因为在这种情况下，您可能无法再下载资产的旧版本(但是对于您来说可以工作或兼容)。

备份工具可以保留包的多个版本，并自动处理生命周期、版本比较等。



## 维修

### 维护功能也位于设置面板:

建议偶尔跑跑步

重新创建预览: 将尝试为资产文件创建预览图像, 这些文件到目前为止还没有或被标记为要重新创建。这可能需要相当长的时间, 特别是对于复杂的预制件。

删除孤立的预览图像: 删除磁盘上的剩余图像, 其中没有相应的资产不再存在的指数。

移除重复的媒体索引: 解决文件已被索引多次的情况, 例如, 当把媒体文件放在一个资产下, 文件夹已被索引一次, 没有这样的连接。

优化数据库: 压缩和重组数据库, 以获得更高的性能和更少的空间需求, 应该在更大的索引活动之后完成

### 无害

Close 数据库: 允许备份或复制文件

更改数据库位置: 为数据库设置一个不同的位置, 移动当前位置, 使用另一个现有位置或创建一个新位置

清除缓存: 删除临时文件, 节省空间, 可以做到没有任何副作用

### 潜在的破坏性

Clear 数据库: 删除完整的数据库, 重新开始

重置配置: 将一切设置为最初的样子, 删除已经完成的任何额外配置



# 资产预览

搜索结果将显示资产的预览。这些预览可以来自三个来源:

包含在包中，并在构建资产时创建

这是由资产存储工具完成的，也是为什么声音文件的预览可以根据所使用的 unity 版本看起来不同的原因。

在导入期间由资产库存创建

对于没有依赖关系的原子资产，例如声音文件、纹理或模型，这种情况会自动发生。

按需资产清单

这个模式也可以为预制板和材料创建预览，但是需要更长的时间，因为所有的依赖都需要首先实现。

可以从搜索视图内部重新创建单个预览、包的所有预览或对完整数据库中缺少的预览执行重新创建。

预览由 unity 编辑器创建。这个工作良好，但是被限制为 128 x 128 像素。在设置中有一个高级的选项，可以产生更大的预览期权，但是会降低质量。此外，VFX 和 2d 预制件不能以这种方式预览。一个自定义机制正在为未来的版本进行中。

在创建预览时，可能会出现一个名为 `_terrainautoupgrade` 的新文件夹作为副作用出现在 Assets 下面。

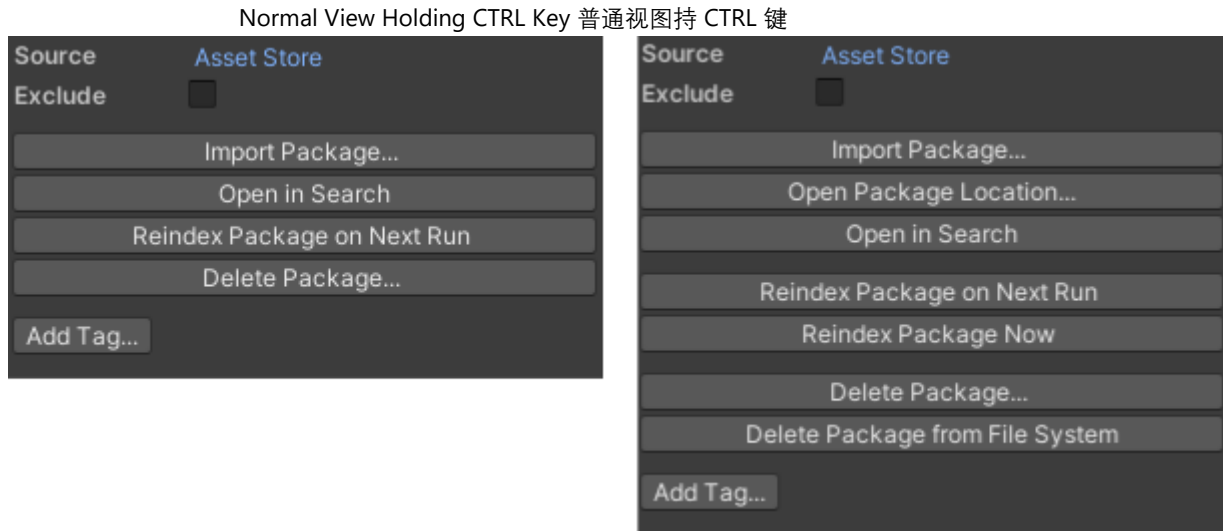




# 高级功能

## 专家职能

许多很少使用的高级功能最初都是隐藏起来的，以免弄乱用户界面。  
按住 CTRL 键可以使它们可见。



## 自动化及空气污染指数

这个工具可以通过脚本来控制。这是通过 AssetInventory 类的静态方法实现的。在与工具交互之前一定要调用 Init ()。



# 配置

设置将自动保存在一个名为 `AssetInventoryConfig.json` 的文件中。默认情况下，这会在当前登录用户的 `Documents` 文件夹中。这样，这个工具可以独立于 `unity` 项目工作，而且资产搜索在任何地方都可以以相同的方式使用。有两种选择：

可以强制工具使用特定于项目的配置。为此，将 `AssetInventoryConfig.json` 文件复制到项目的任何地方，然后重新启动 `unity`。检测到的位置将显示在 `Maintenance Functions` 下的设置面板上。

使用环境变量“`ASSETINVENTORY_config_path`”可以指定一个自定义位置(只是文件夹名称)。

某些高级设置只能在。而且不能通过用户界面访问。这方面的一个例子是搜索开始的延迟。你也可以随意调整这些值。

## 高级配置

为了不使 `UI` 混乱，一些很少使用的设置将直接在配置文件中设置。它是格式化的 `JSON` 格式，可以在任何文本编辑器中打开。下面的属性只出现在那里：

**自动隐藏设置(bool) :** 标记是否自动关闭视图设置或仅当再次单击设置按钮

**缓存文件夹(字符串) :** 缓存应该位于的位置，例如“`f:/AI/Cache`”。默认是数据库下的“`Extracted`”。

**hueRange (float) :** 扩大颜色搜索的度数

**maxResultsLimit (int) :** 即使选择了 `-all`，也要显示的最大结果的硬限制

**searchDelay (float) :** 输入一个字符后的几秒钟，等待触发搜索



# 技术细节

该指数将存储在 SQLite 数据库中。这个数据库将在第一次使用时自动创建。它在你的文档/资产目录中。它的大小应该相当小，但是随着索引资产的数量而增加。作为一个经验法则，假设每 30k 文件有 20mb 的数据库 + 200Mb 的预览图像。

在 AssetInventory 目录中，将创建另外三个文件夹：

**预览:** 包含资产目录的所有图片

**解压缩:** 包含临时解压缩文件。如果需要，你可以随时安全地删除解压缩文件夹。

**备份:** 充当包备份的默认位置

有些操作需要 unity 来执行工作，例如创建预览。对于这样的任务，文件夹 \_assetinventorypreviewstemp 会在很短的时间内出现在 Assets 下面，然后自动再次被删除。

## 限制

只能在使用 text/yaml 串行化的包中计算依赖项。一些传统的包仍然使用二进制串行化。这些包在大多数情况下也可以工作，除非 unity 不能解决它们，例如因为缺少预置的脚本。

导入脚本是试验性的，如果脚本需要运行其他脚本或程序集定义文件，可能会产生控制台错误。

运行游戏视图最大化的发挥和有资产库存窗口停靠将导致它重新初始化，重置所有搜索设置。



# 常见问题

## 有些预制件没有依赖性

这是由于串行化格式。现在只有串行化为文本的资产才能被正确地解析。

## 一些音频文件在预览中使用不同的颜色

预览颜色取决于资产作者用来上载资产的 unity 版本。Unity 在不同的 unity 版本之间似乎会随着时间的推移而改变颜色。可能有一天会有一个功能，用当前版本的 unity 重新创建预览图像，使它们根据兴趣统一。

## 有些预告片是灰色的，什么都没有

如果在上传资产时，assetstore 工具没有创建正确的预览图像，则可能会出现去中心化的标识符(did)。一个例子是 unity 为 2021 + 发布的 Danger Zone 资产。重新创建预览图像的功能性正在开发中。

## 控制台在索引过程中显示错误

Unity 创建的一些错误不能被拦截，但是可以被安全地忽略。这些是：

**“无法为剪辑”创建 FMOD: : Sound 实例”(FMOD 错误: 不支持文件或音频格式。)**

这通常意味着一个音频剪辑被保存错误的扩展名，例如，一个 wave 作为一个 ogg 由一个资产出版商。使用 Open 按钮启动原生文件播放器，这样通常会正确播放文件。

**“纹理宽度/高度超出范围”**

在这种情况下，unity 不能读取纹理的维度，你也不能看到纹理的维度或者过滤器。否则，它没有任何效果。

**“参数例外: 当重新绘制时，在一个只有 4 个控件的组中获得控件 4 的位置”**



如果在进行索引的同时进行了大量的用户界面变化，这种情况就会发生。可以安全地被忽略，没有任何影响。

**“Curl 错误 28: 操作在接收到 0 字节的 30000 毫秒后超时”**

如果 unity 服务器没有及时响应，并且无法检索资产的详细信息，则可能发生去中心化的标识符(did)。这可以通过重新启动 Asset Store 更新过程来解决。

**“联系时无效或过期的 API 代币...”**

如果 unity 在较长时间内没有被使用，就会发生这种情况。用户登录发生在 unity hub，在这种情况下，在线访问的代币已经过期。通过重启 unity 解决。

**“无法从 Assets/\_ AssetInventoryPreviewsTemp/1016771. png 创建资产: 无法读取文件”**

如果 unity 不能为特定资产创建预览，或者这是一个格式错误的 png，就会发生这种情况。可以忽略。

**“TLS 分配器 ALLOC \_ temp \_ thread，基础分配器 ALLOC \_ temp \_ thread 具有未释放的分配，大小为 1261”**

可以在触发下载后发生。到目前为止没有发现任何负面的副作用。最有可能被忽略，似乎是一个 unity 的问题。

**“您正试图从引用缺失脚本的实例’ ...’替换或创建预制体。这是不允许的。请更改脚本或将其从游戏对象中删除。”**

当预制体有脚本依赖关系时，可能会在创建预览图像的过程中发生。可以忽略，因为资产清单自动修复这种情况，但错误不能被抑制。

**“正常映射设置对话框出现”**

当预制体中的纹理配置错误时，可能会在创建预览图像的过程中发生。可以忽略，因为这在预览中是不可见的。



# 第三方

资产清单使用多个第三方库，真正使它发光。

[Package2Folder by code Stage 使用了一个巧妙的想法来拦截 unity 包管理器并调整安装资产的目标文件夹。](#)

[SQLite 是一种将数据存储在一个文件中的好方法，同时提供了简单快速的数据库操作。](#)

[SQLite-Net 是 SQLite 的一个令人惊讶的扩展，在执行 SQL 查询时提供了方便的高级函数。](#)

[Editor 音频 Utils 在非播放模式下也可以播放音频文件。](#)

[Redcode 的 waitextensions 非常有助于将基于 IEnumerator 的协同程序转移到更现代的 c # 异步范例中，并将两者结合使用。](#)

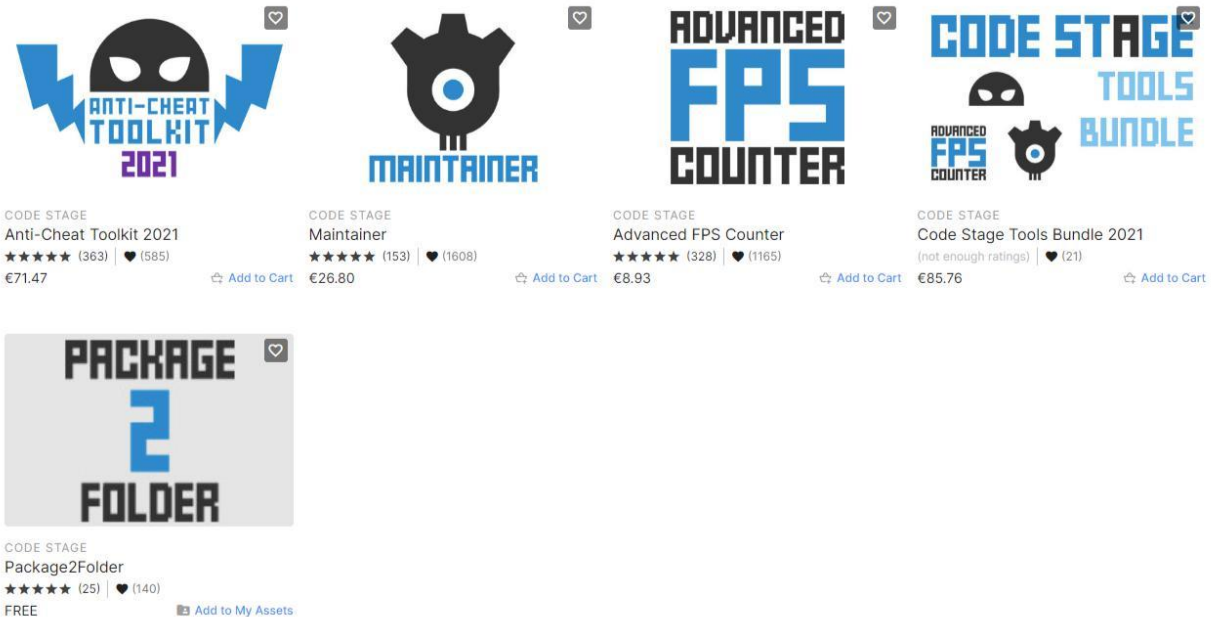


# 建议

我想利用这个空间来突出我认为非常有价值的资产，并与我们支持我们的地方建立友好关系。

## 代码阶段的重要资产

一个高质量的工具集合，以提高您的游戏和 unity 的生产力。



# 支持联系人

网址: <https://www.wetzold.com/tools>

Discord: <https://Discord.gg/uzehzemm4b>

路线图: <https://trello.com/b/SOSDzX3s/asset-inventory>

