ASSET INVENTORY

GUIDE - VERSION 2.6.0

INTRODUCTION

Put your asset workflow on steroids and say good-bye to the Package Manager as you know it! Asset Inventory is your ultimate asset management companion: a lightning-fast search for assets for your current project. Find content in assets you purchased or downloaded without importing and bring single files in with just a click.

Eliminate the time-consuming task of finding a sound file, a texture or a model you know you purchased but which is hidden inside one of your many Asset Store purchases. The Asset Inventory provides a **complete list of all assets you own**, including their content.

CONTENTS

FEATURES

The tool aims to provide **what you always wanted the Asset Store & Package Manager to be**: The best possible Asset Management solution for Unity.

Find assets effectively. Search inside your purchases. Include custom packages you downloaded from Humble Bundle or other places. And packages from a registry. Manage everything through tags. Perform bulk operations. Import multiple assets and packages in one go. Identify used assets inside your projects. Search using asset-specific criteria. And much more.

Powerful Search

Browse & find anything inside your purchased Asset Store packages without importing. Quickly preview audio files. Narrow your search using asset type, tags, image dimensions, audio length, colors and more. Exclude items you don’t want to see. **Save and recall** searches.

Easy Setup

Works immediately out of the box. Lots of optional view & configuration options. Hassle-free indexing. Start, stop & resume at any time. Works with Unity 2019.4 and higher. Windows, Mac & Linux. Lightning-fast indexing and search.

Intelligent Updates

Handle updates from the Asset Store and from the Unity registry in one holistic UI. Define update strategies per package. Instantly see if there are any updates you should consider. Full package manager functionality for registry packages including add/update/remove and compatibility information.

Import & Export

Import only what you need instead of a whole package. Automatically determines asset dependencies to import complex prefabs and materials. Save space & reduce clutter in your project. **Bulk import** multiple packages at once. Automatically store imported assets in a specific sub-folder and keep the Assets root clean. **Export** assets easily for reuse in other contexts.

Many Sources

Your complete asset library: Automatically indexes Asset Store purchases. Triggers download of missing assets. Handles packages from registries. Integrate the Unity Cloud Asset Manager. Add custom folders to search through Unity packages downloaded from other locations. Indexes folders and zip/rar archives containing **arbitrary media files** like 3D models, audio libraries, textures and more. Automatically generates previews.

Organize

Automatically imports labels from the Asset Store. Use additional **tags** to group assets effectively. Assign tags to either packages or individual files inside packages. Assign colors and group by tags. Exclude unwanted items. Browse all sub-packages. Perform bulk operations. Import multiple assets and packages in one go. Builds on Package2Folder to allow importing packages into a custom sub-folder. **Backup** packages automatically to ensure you always have a working version to go back to.

Reverse Lookup

Quickly identify used assets and packages in your project. Ensure license compliance.

Constant Updates & Support

Receive regular updates, optimizations, and new features. Super-fast support. Discuss ideas in a great Discord community.

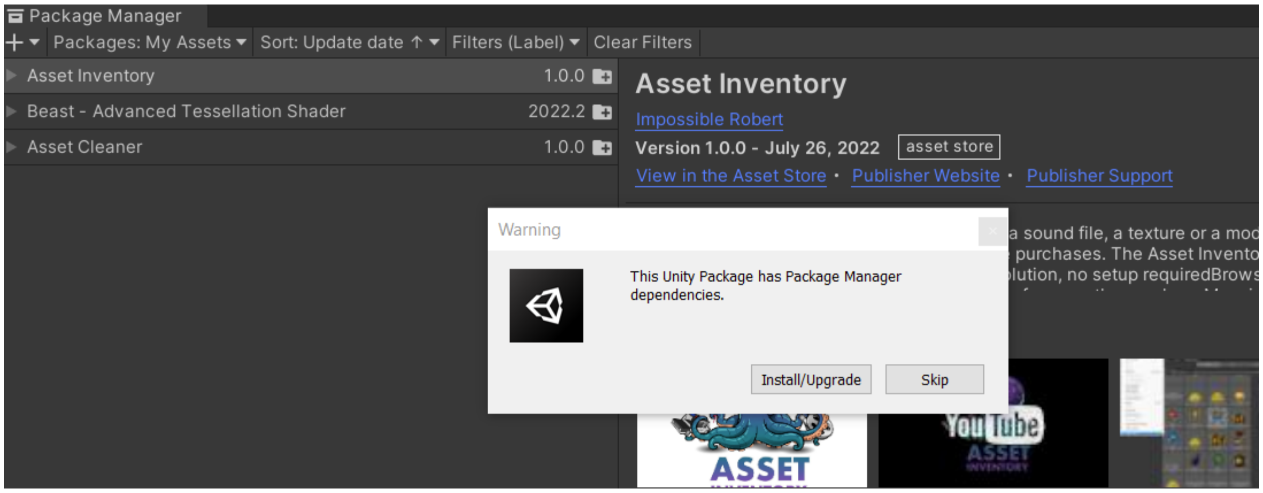
GETTING STARTED

INSTALLATION

It is strongly recommended to remove any previous version of the Asset Inventory before installing a new version. The best way to do so is to close Unity and afterwards delete the full Asset Inventory folder since used libraries cannot be replaced if Unity is active.

Pro Tip: Run the Asset Inventory in a **new empty Unity project** for the **initial indexing** and use the newest available release. The indexing results will be stored in a central location for reuse in any other project. This will give you the highest indexing performance since no other assets/scripts need to be refreshed and new versions of Unity typically bring big performance improvements for object and sound importers (in some cases 10x).

Install the package through the Unity Package Manager. When asked if to install additional dependencies, select **Install/Upgrade**, otherwise the asset will not work.

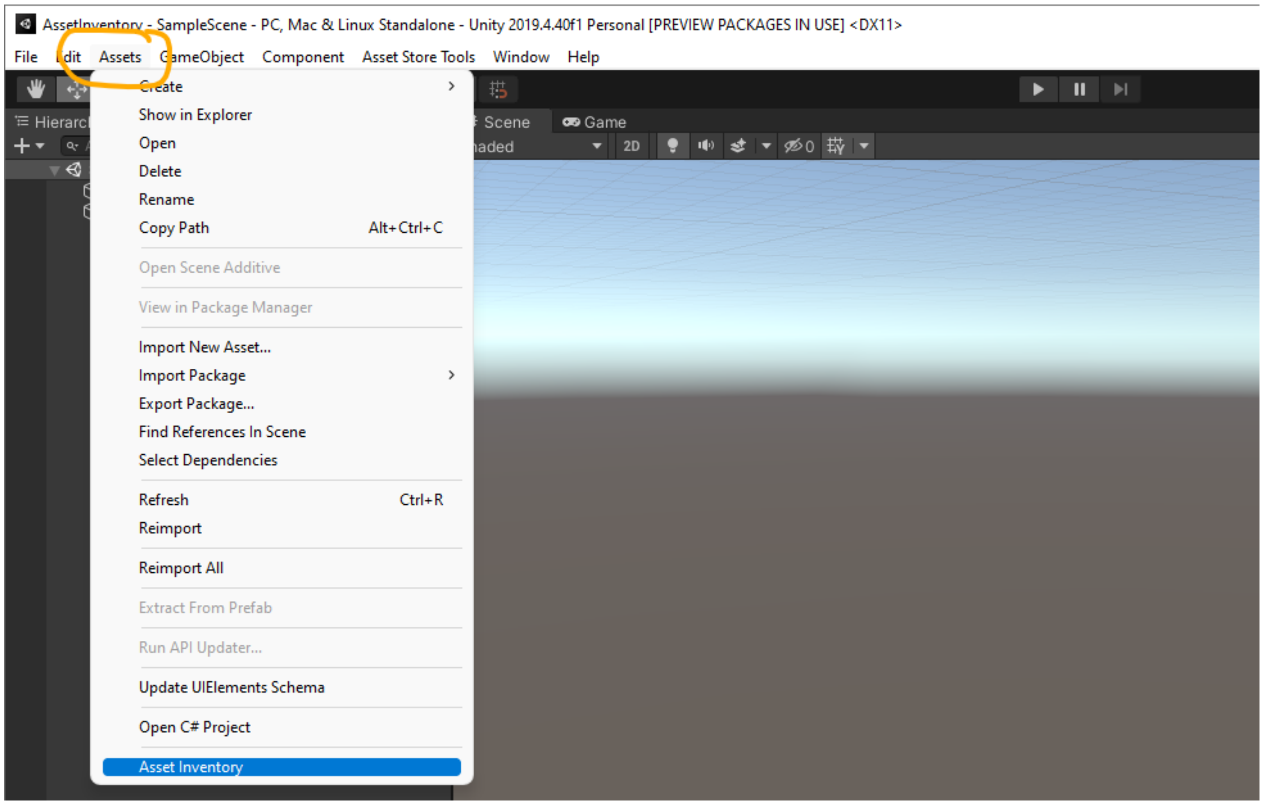


This Unity Package has Package Manager dependencies.

When indexing packages Unity might try to call third party applications like Blender, 3ds Max or Maya. It is recommendable to at least **install Blender** (free) to make sure previews for blend files are calculated correctly.

USAGE

Open the Asset Inventory through the **Assets menu**.



There is **no manual setup** needed. The only requirement is a fair amount of disk space for storing preview images and temporary files during extraction and browsing.

Press **Start Indexing** and wait for results to come in. When indexing is done it can be beneficial to **delete your project Library folder** which has accumulated lots of temporary unneeded data.

Whenever you purchase new packages or add new assets to your folders, make sure to press the "Update" button under **Settings** once so that the index gets updated.

Pro Tip 1: Activate the **Download Assets for Indexing** option under **Settings** to index your complete library without the need to download assets manually first.

Pro Tip 2: The UI displays only the core functionality (easy mode). You can access many **additional options** by holding down the **CTRL key** or pressing the eye icon in the top.

ASSET SEARCH

OVERVIEW

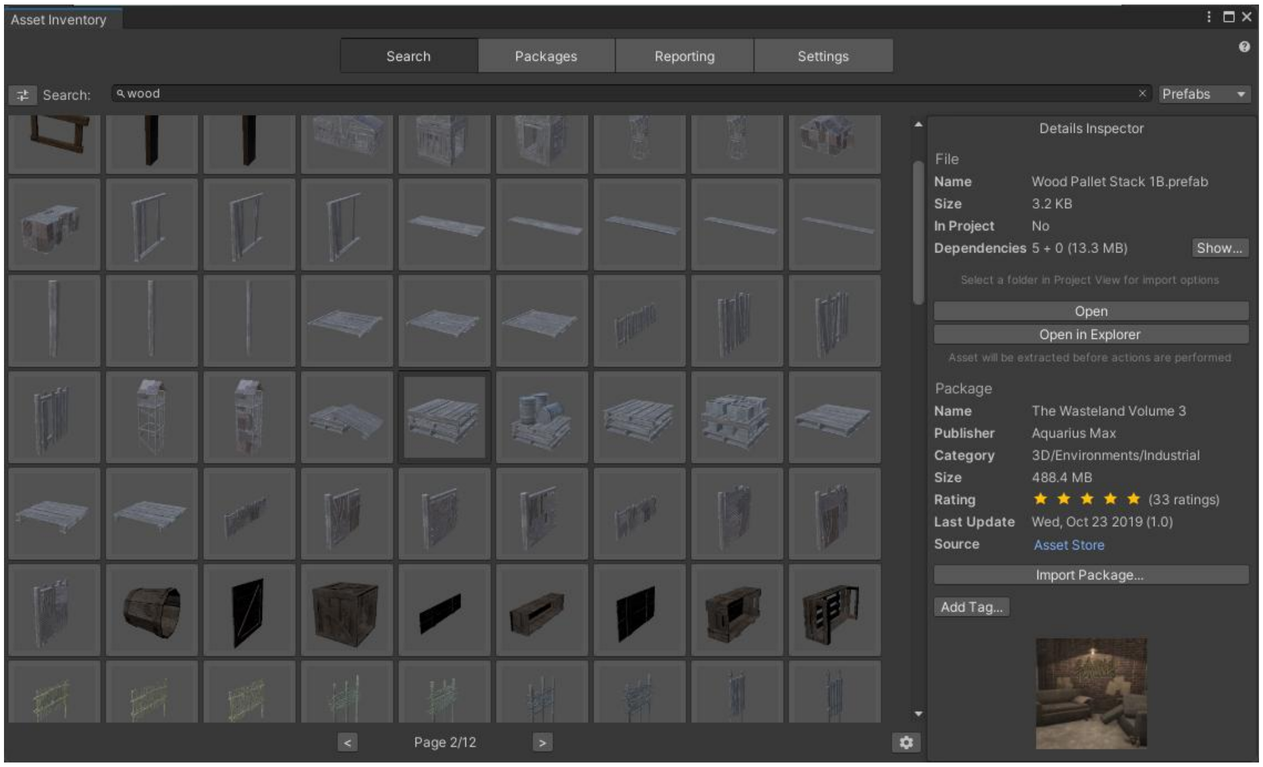
Once an asset is indexed, it can be found in the search. Searching will automatically start when typing. By default, every word entered needs to match the search result but not necessarily in the same order. If the exact phrase should be matched, prefix the search with ~.

Examples:

- *"car interior"* will return results like *"CarBlueInterior.fbx"* and *"InteriorCarDes.png"*
- *"~car interior"* will not return the above but only results like *"Car Interior.fbx"*
- *"car +interior -fbx"* will return results that match *"car"* and *"interior"* but not *"fbx"*

There is also an expert search available (searches starting with "=") which is described later.

Additional filters can be selected to narrow down the results further. Once an asset is selected, details are shown on the right-hand side about the file and the package the file is contained in.



Search: wood

Details Inspector

File

Name	Wood Pallet Stack 1B.prefab
Size	3.2 KB
In Project	No
Dependencies	5 + 0 (13.3 MB)

Select a folder in Project View for import options

Asset will be extracted before actions are performed

Package

Name	The Wasteland Volume 3
Publisher	Aquarius Max
Category	3D/Environments/Industrial
Size	488.4 MB
Rating	★★★★★ (33 ratings)
Last Update	Wed, Oct 23 2019 (1.0)
Source	Asset Store

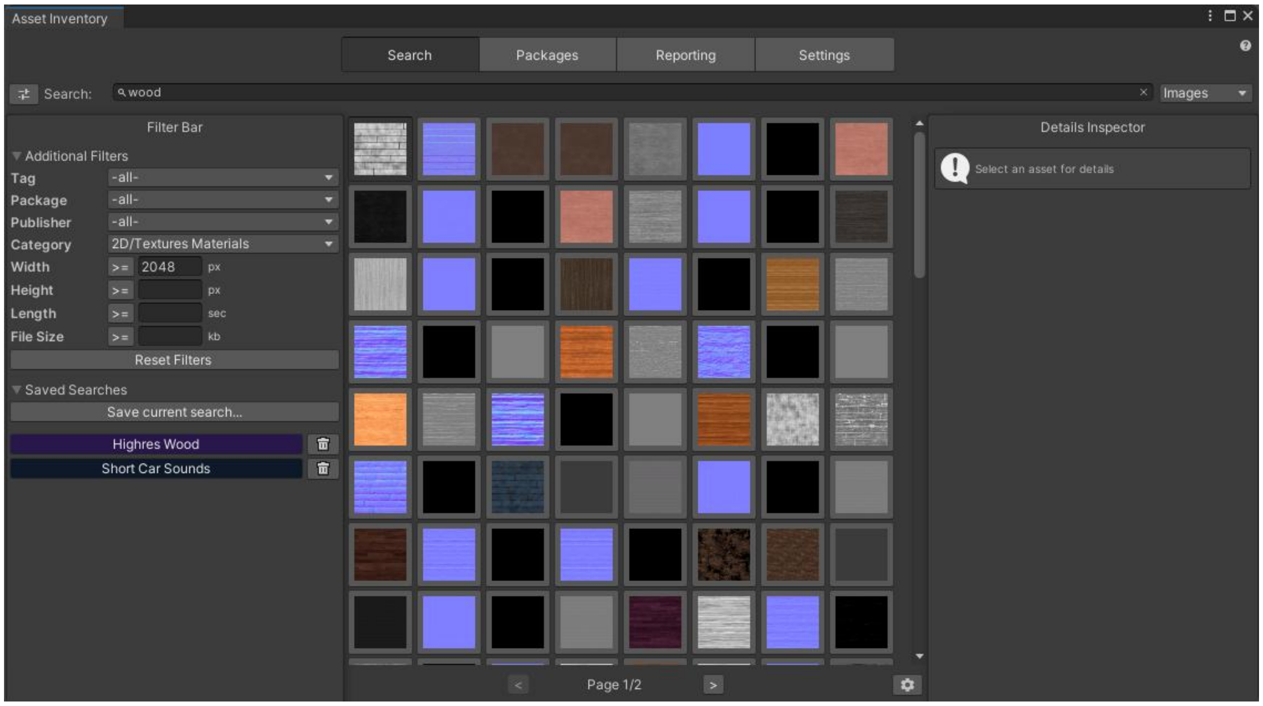
Page 2/12

Selecting an audio file will **automatically play** it. This way it is easy to quickly preview audio files. Initial playback might take a while until the corresponding package is temporarily extracted.

Multi-select is possible by holding down the Shift or Ctrl key. Drag and drop is possible from Unity versions 2021.2+.

ADVANCED FILTERS & SETTINGS

Clicking the **Filter icon** in front of the search will bring up the **filter bar**. It contains additional filters and media-specific properties to filter for specific image dimensions or audio length. Using the buttons in-front of each value toggles if results match that are bigger or smaller than entered. The filter dropdowns will only show values if respective assets are available. If the search type is limited (e.g. to Images), some filters might not show up (e.g. Length) as they are only applicable to other types.



Search: wood

Tag -all-

Package -all-

Publisher -all-

Category 2D/Textures Materials

Width >= 2048 px

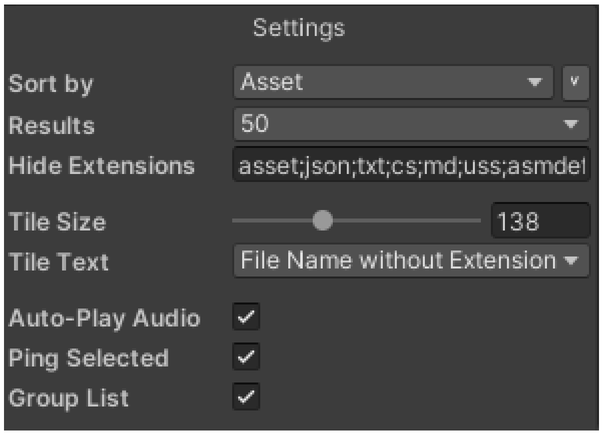
Height >= px

Length >= sec

File Size >= kb

The filter bar also contains the section for **Saved Searches**. This allows to persist the current search filters and later recall these easily. The results are not persisted but instead live from the database.

Using the **settings icon** in the lower right corner will open additional view settings. Specify the tile size, which text to display on the tiles and if assets should be pinged automatically upon selection.

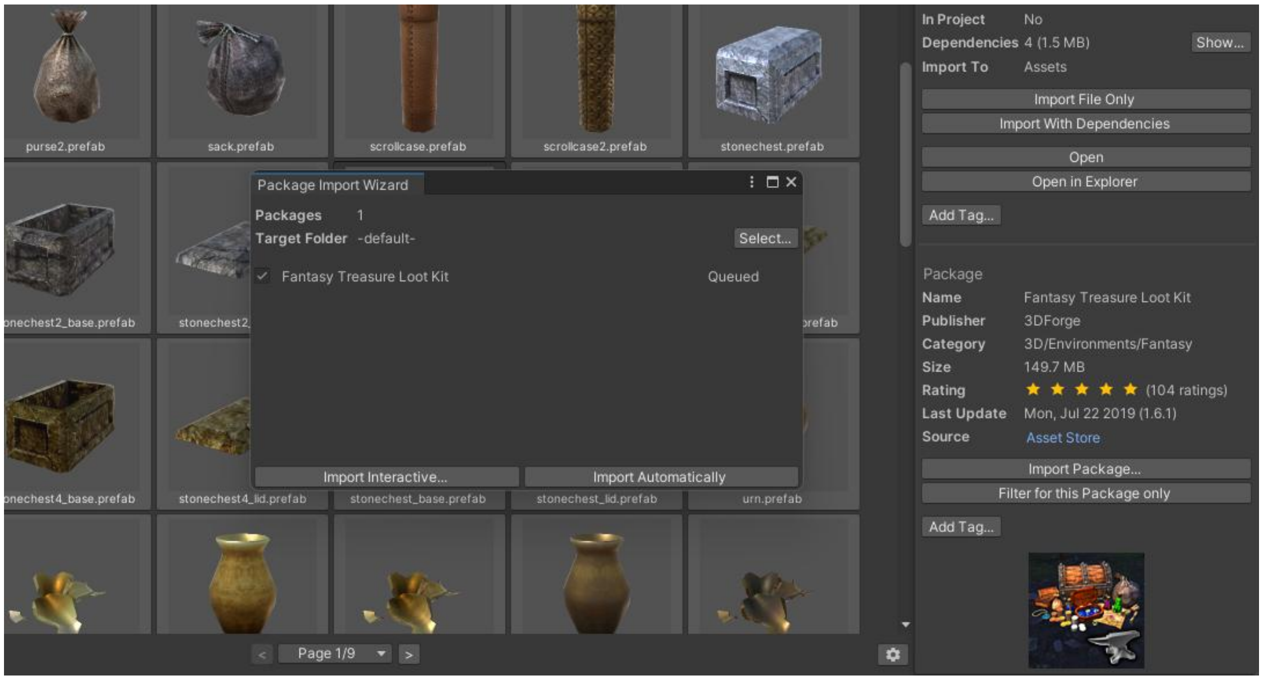


Settings

Sort by	Asset
Results	50
Hide Extensions	asset;json;txt;cs;md;uss;asmdef
Tile Size	138
Tile Text	File Name without Extension
Auto-Play Audio	✓
Ping Selected	✓
Group List	✓

IMPORTING ASSETS

Selecting a folder in the Unity **Project View** will bring up **import** options when selecting an asset in the search. Assets can be imported individually or with all detected dependencies. The latter will automatically create a sub-folder with the asset name and store all files in there.



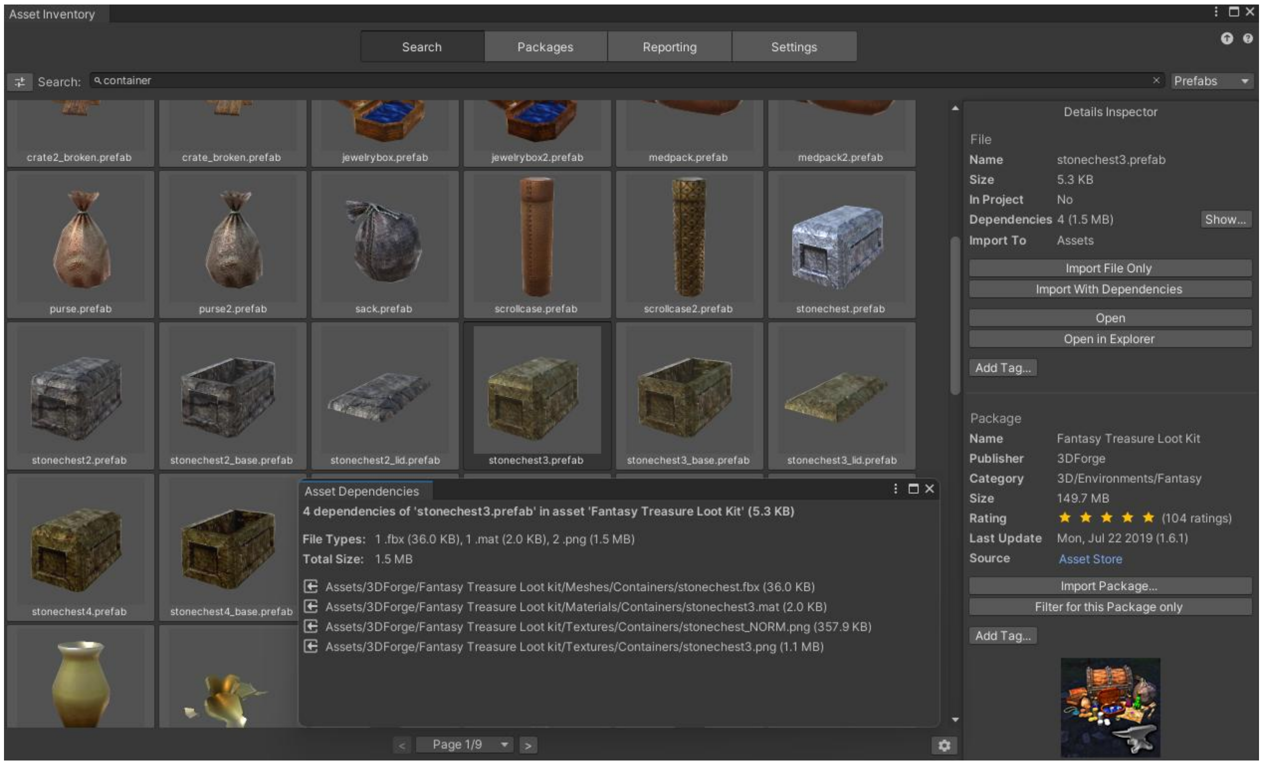
Packages 1
Target Folder -default-
✓ Fantasy Treasure Loot Kit Queued

In Project No
Dependencies 4 (1.5 MB)
Import To Assets

Package
Name Fantasy Treasure Loot Kit
Publisher 3DForge
Category 3D/Environments/Fantasy
Size 149.7 MB
Rating ★★★★★ (104 ratings)
Last Update Mon, Jul 22 2019 (1.6.1)
Source Asset Store

In case there are **script dependencies** these can also be imported. Due to unforeseen dependencies in scripts, that will typically only work for scripts that are self-contained. Otherwise, there might be compilation errors.

Clicking the *Show* button behind the dependency information will bring up the dependency information details, listing all files, their size and if they are already in the project or not.



Details Inspector

File

Name	stonechest3.prefab
Size	5.3 KB
In Project	No
Dependencies	4 (1.5 MB)
Import To	Assets

Package

Name	Fantasy Treasure Loot Kit
Publisher	3DForge
Category	3D/Environments/Fantasy
Size	149.7 MB
Rating	★★★★★ (104 ratings)
Last Update	Mon, Jul 22 2019 (1.6.1)
Source	Asset Store

Asset Dependencies

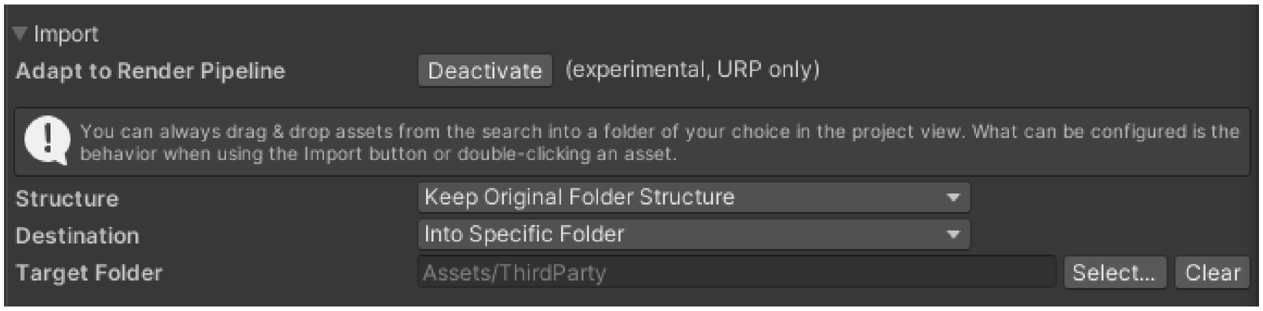
4 dependencies of 'stonechest3.prefab' in asset 'Fantasy Treasure Loot Kit' (5.3 KB)

File Types: 1 .fbx (36.0 KB), 1 .mat (2.0 KB), 2 .png (1.5 MB)
Total Size: 1.5 MB

```
Assets/3DForge/Fantasy Treasure Loot kit/Meshes/Containers/stonechest.fbx (36.0 KB)
Assets/3DForge/Fantasy Treasure Loot kit/Materials/Containers/stonechest3.mat (2.0 KB)
Assets/3DForge/Fantasy Treasure Loot kit/Textures/Containers/stonechest_NORM.png (357.9 KB)
Assets/3DForge/Fantasy Treasure Loot kit/Textures/Containers/stonechest3.png (1.1 MB)
```

Import can also be triggered by **double-clicking** on an item (if activated in Search settings) or by **dragging** it into the Project Window. Once an item is imported into the project it can also be dragged from the search into any other window, e.g. the scene or fields accepting the dragged object type.

When working in a **URP** project, materials can appear with the pink error shader due to incompatibility. The typical solution is to run the **Render Pipeline Converter** which will adjust the imported materials to fit the URP materials. If activated under *Settings* (available in URP projects with URP 14+), the tool will **automatically trigger this conversion**, making it very convenient to import assets. Since there is no programmatic way to influence which materials are converted, this setting will convert all project materials (typically not a problem) and will incur a minor speed penalty. It will also work when recreating previews.



Import

Adapt to Render Pipeline (experimental, URP only)

You can always drag & drop assets from the search into a folder of your choice in the project view. What can be configured is the behavior when using the Import button or double-clicking an asset.

Structure	Keep Original Folder Structure
Destination	Into Specific Folder
Target Folder	Assets/ThirdParty

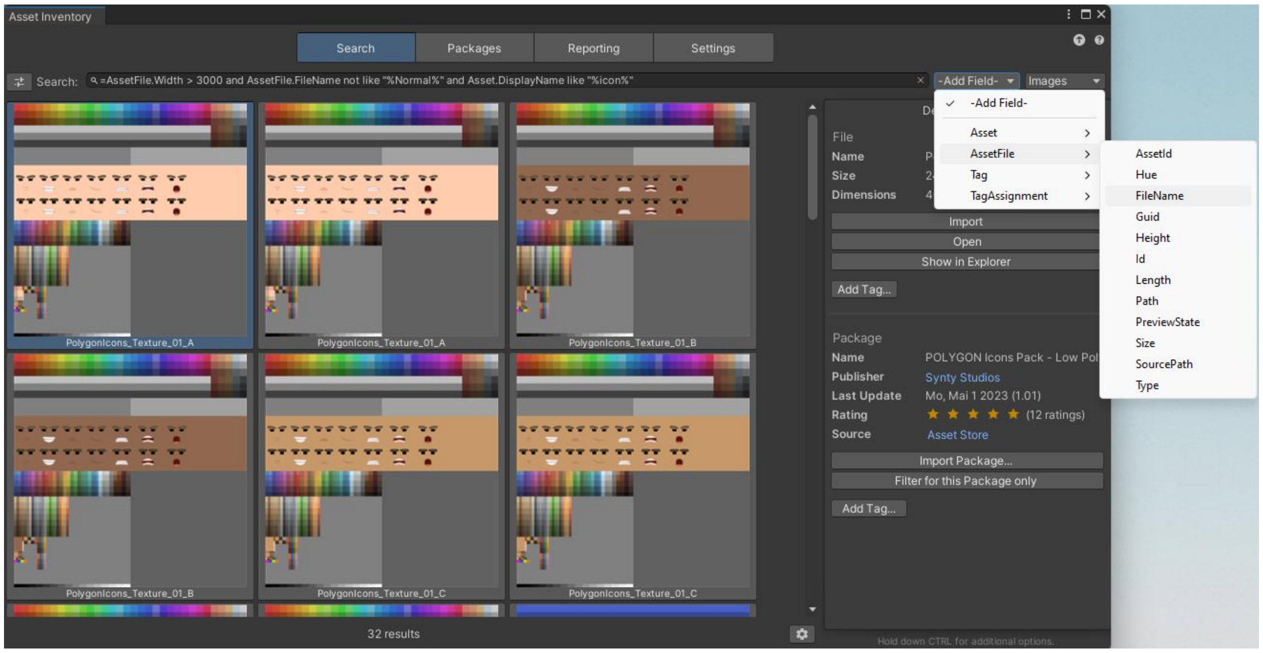
EXPERT SEARCH

It is possible to use nearly the full feature set of SQLite 3 to search. This mode is activated when starting the search with "=". Afterwards the database fields and conditions can be stated.

Examples

```
=Asset.PackageSize > 0 and AssetFile.Type=”wav”
```

```
=AssetFile.Width > 3000 and AssetFile.FileName not like "%Normal%" and Asset.DisplayName like "%icon%"
```



Search: =AssetFile.Width > 3000 and AssetFile.FileName not like "%Normal%" and Asset.DisplayName like "%icon%"

Useable fields can conveniently be picked from the dropdown behind the search field.

PACKAGES

OVERVIEW

The packages overview will show a list of all packages, indexed or detected in the current project. These can come from multiple sources:

Automatic

- Asset Store purchases
- Registry packages
- Unity Asset Manager
- The local Unity asset cache
- The local Unity package cache

Additional folders stated under *Settings*

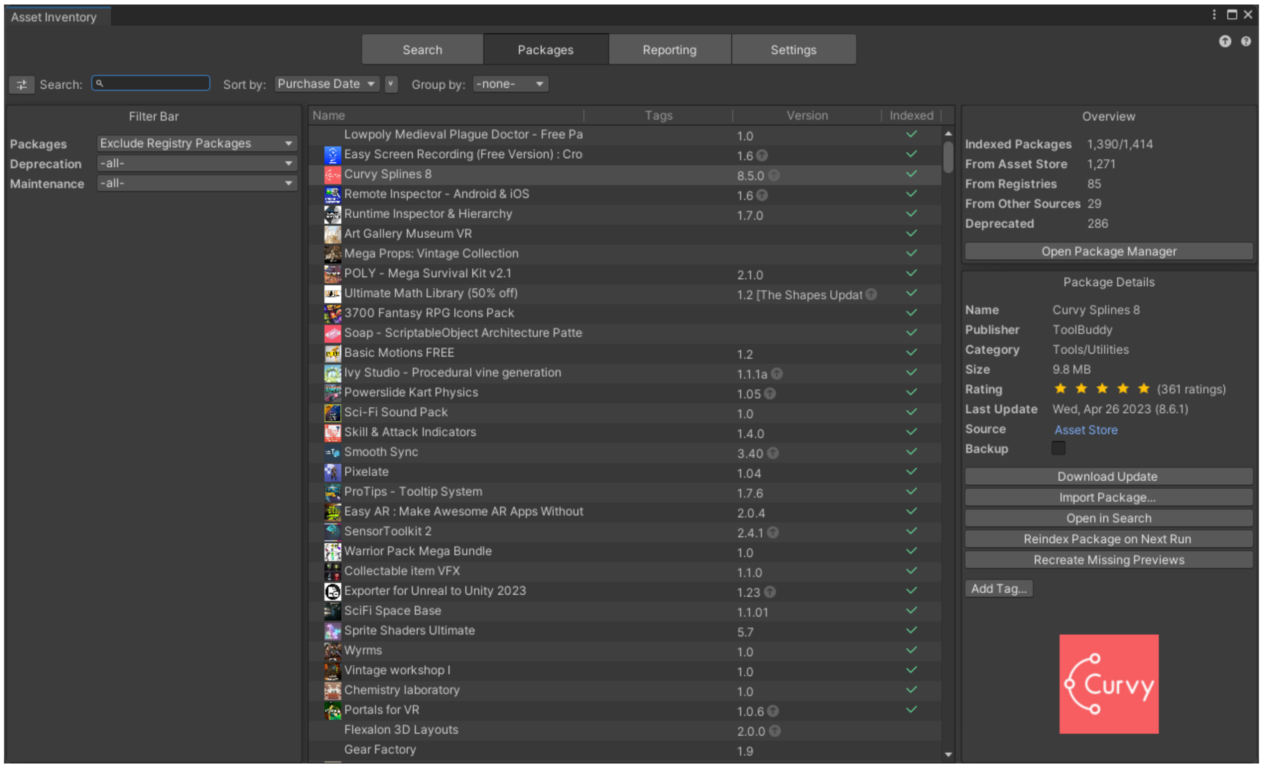
- Unity packages from other sources (e.g. Synty)
- Local asset packages under development (via package.json)
- Local media libraries (sound files, textures, models…)
- Unity projects
- Archives (zip, rar & 7z contents)

The list will indicate which of these were already indexed. To index more packages, download them into the cache or add more additional folders. Display is possible as a plain list or grouped by categories, publishers, state or tags.

In the case of registry packages, only the latest version of the package will be indexed.

If a package contains **sub-packages** (also recursively), they will be shown directly under the package in a tree structure.

Double-clicking any package will show the contents in the search. After selecting a package, multiple options will appear. It is possible to import it. Alternatively, it can also be removed from the index to trigger a reindexing on the next run. This can be useful for incorrectly indexed files. Packages can also be completely removed from the database which can be useful after cleaning up outdated assets from the local cache.



Search: Sort by: Purchase Date Group by: -none-

Packages	Exclude Registry Packages
Deprecation	-all-
Maintenance	-all-

Name	Tags	Version	Indexed
Lowpoly Medieval Plague Doctor - Free Pa		1.0	✓
Easy Screen Recording (Free Version) : Cro		1.6	✓
Curvy Splines 8		8.5.0	✓
Remote Inspector - Android & iOS		1.6	✓
Runtime Inspector & Hierarchy		1.7.0	✓
Art Gallery Museum VR			✓
Mega Props: Vintage Collection			✓
POLY - Mega Survival Kit v2.1		2.1.0	✓
Ultimate Math Library (50% off)		1.2 [The Shapes Updat	✓
3700 Fantasy RPG Icons Pack			✓
Soap - ScriptableObject Architecture Patte			✓
Basic Motions FREE		1.2	✓
Ivy Studio - Procedural vine generation		1.1.1a	✓
Powerslide Kart Physics		1.05	✓
Sci-Fi Sound Pack		1.0	✓
Skill & Attack Indicators		1.4.0	✓
Smooth Sync		3.40	✓
Pixelate		1.04	✓
ProTips - Tooltip System		1.7.6	✓
Easy AR : Make Awesome AR Apps Without		2.0.4	✓
SensorToolkit 2		2.4.1	✓
Warrior Pack Mega Bundle		1.0	✓
Collectable item VFX		1.1.0	✓
Exporter for Unreal to Unity 2023		1.23	✓
SciFi Space Base		1.1.01	✓
Sprite Shaders Ultimate		5.7	✓
Wyrms		1.0	✓
Vintage workshop I		1.0	✓
Chemistry laboratory		1.0	✓
Portals for VR		1.0.6	✓
Flexalon 3D Layouts		2.0.0	
Gear Factory		1.9	

Indexed Packages	1,390/1,414
From Asset Store	1,271
From Registries	85
From Other Sources	29
Deprecated	286

Name	Curvy Splines 8
Publisher	ToolBuddy
Category	Tools/Utilities
Size	9.8 MB
Rating	★★★★★ (361 ratings)
Last Update	Wed, Apr 26 2023 (8.6.1)
Source	Asset Store
Backup	☐

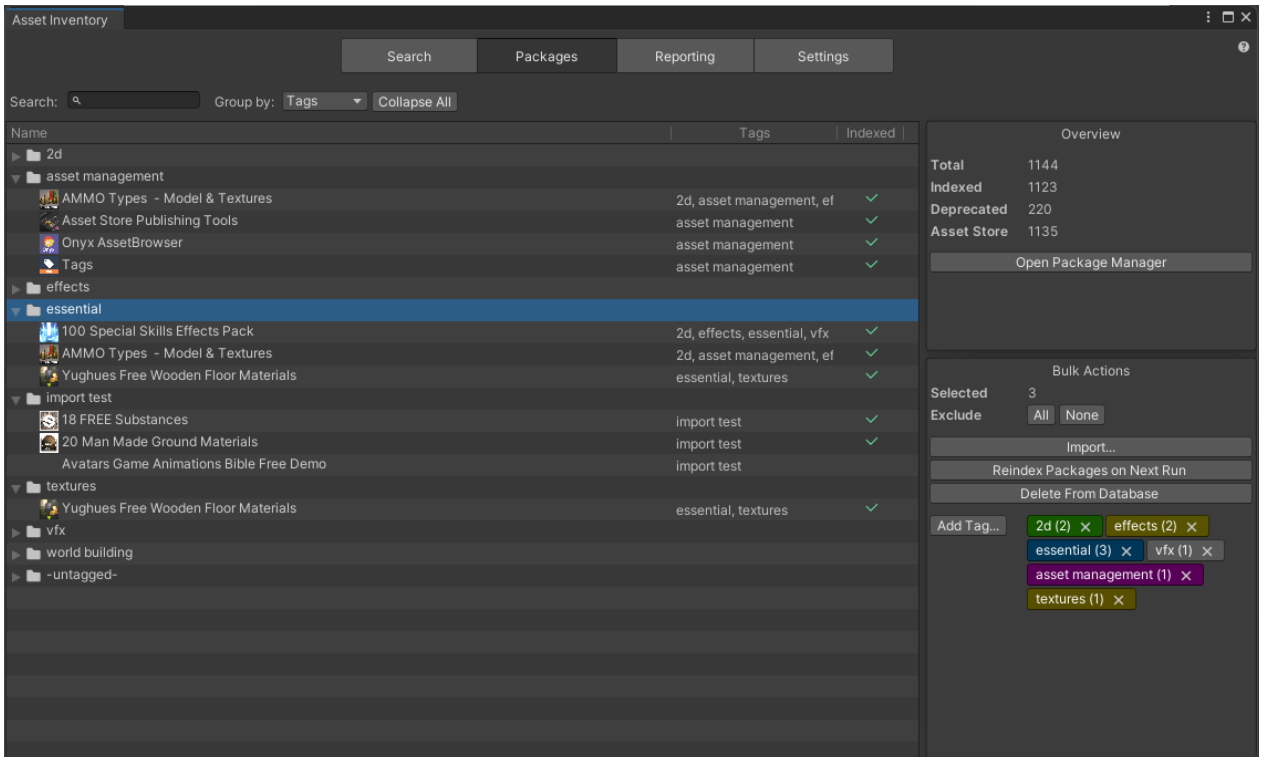
Depending on the package type additional options become visible:

- Registry Packages
  - Support for managing the installed version and setting an **update strategy** (recommended, latest stable compatible, latest compatible, manually).
- Development Packages
  - Use directly via **file link**
- Custom Packages
  - Support for **connecting to metadata** from the Asset Store. This will load the name, description, rating etc. and is especially useful when having downloaded packages from alternative sources like the Humble Bundle to still have all metadata available. See details in Custom Metadata section below.
  - Support for disconnecting from the Asset Store again

Using the buttons under the package list, the display can be switched between list and grid style.

TAGGING

Tags will be imported from the Asset Store in case any are set there. In addition, local tags can be added and removed here as well (they are not synchronized back to the Asset Store yet). Bulk editing of tags is possible when selecting multiple items in the tree.



Name	Tags	Indexed
2d		
asset management		
AMMO Types - Model & Textures	2d, asset management, ef	✓
Asset Store Publishing Tools	asset management	✓
Onyx AssetBrowser	asset management	✓
Tags	asset management	✓
effects		
essential		
100 Special Skills Effects Pack	2d, effects, essential, vfx	✓
AMMO Types - Model & Textures	2d, asset management, ef	✓
Yughues Free Wooden Floor Materials	essential, textures	✓
import test		
18 FREE Substances	import test	✓
20 Man Made Ground Materials	import test	✓
Avatars Game Animations Bible Free Demo	import test	
textures		
Yughues Free Wooden Floor Materials	essential, textures	✓
vfx		
world building		
-untagged-		

Overview

Total	1144
Indexed	1123
Deprecated	220
Asset Store	1135

Selected	3
Exclude	All None

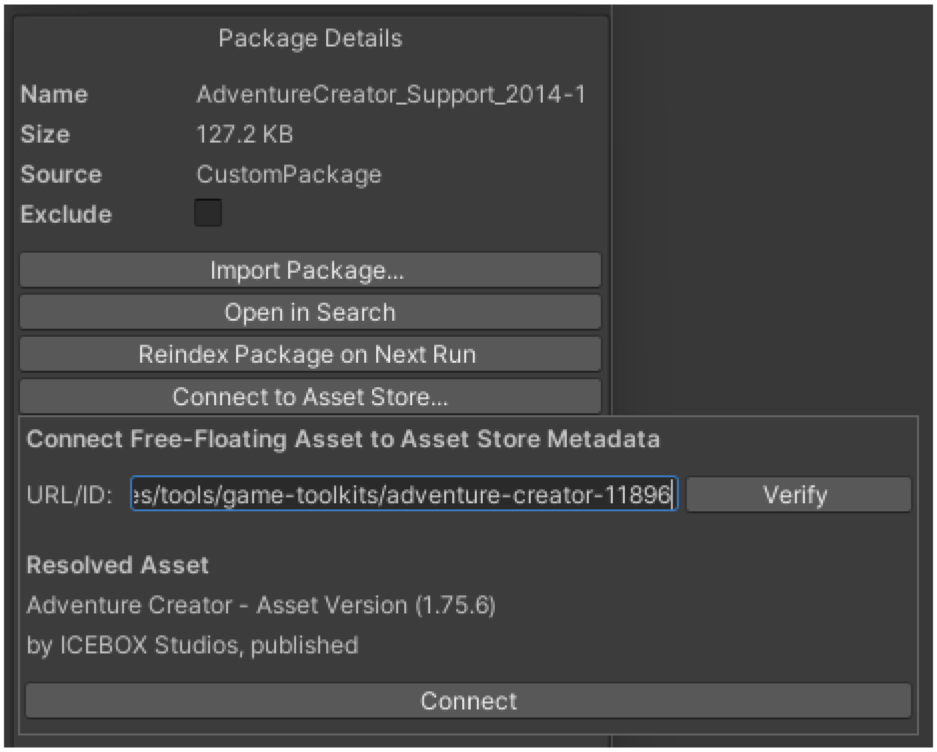
When adding tags, the **Tag Management** window can be opened through the small cog wheel. There, tags can be created, colored, renamed and deleted. The size of the tag selection window can be changed under *Advanced* settings.

CUSTOM METADATA

Package metadata like name, publisher, category etc. is extracted directly from the package if the information is contained in there. If it is missing there are three options.

If the **package exists on the Asset Store**, the package can be linked to the entry there and the metadata will be loaded automatically and also regularly to fetch updates to it.

Copy/paste the URL from the Asset Store website of the asset you want to link, *Verify* it is detected correctly and click *Connect*.


Package Details

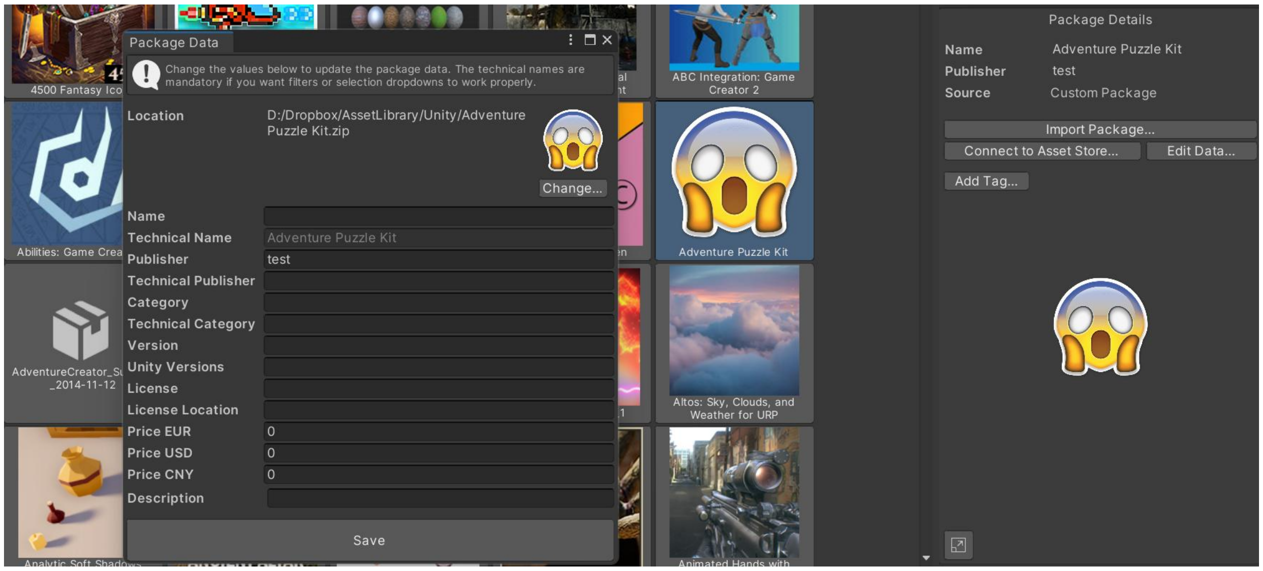
Name AdventureCreator_Support_2014-1
Size 127.2 KB
Source CustomPackage
Exclude

Connect Free-Floating Asset to Asset Store Metadata

URL/ID: :s/tools/game-toolkits/adventure-creator-11896

Resolved Asset
Adventure Creator - Asset Version (1.75.6)
by ICEBOX Studios, published

Metadata can also be entered **manually** using the "Edit Data..." command visible for packages that are not linked to the Asset Store. This way also custom preview images can be set for archives and custom packages.

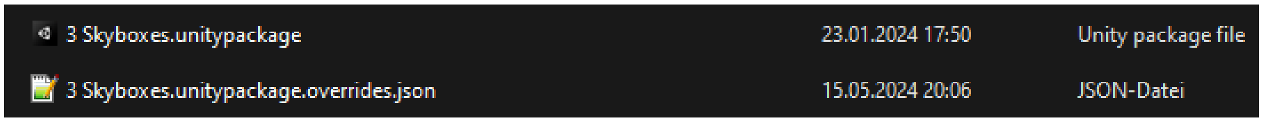

Package Data
Change the values below to update the package data. The technical names are mandatory if you want filters or selection dropdowns to work properly.
Change...
Save
Package Details
Import Package...
Connect to Asset Store...
Edit Data...
Add Tag...

Location	D:/Dropbox/AssetLibrary/Unity/Adventure Puzzle Kit.zip
Name	
Technical Name	Adventure Puzzle Kit
Publisher	test
Technical Publisher	
Category	
Technical Category	
Version	
Unity Versions	
License	
License Location	
Price EUR	0
Price USD	0
Price CNY	0
Description	

Name	Adventure Puzzle Kit
Publisher	test
Source	Custom Package

The last option is to create an **Override Json file** containing metadata. This needs to be named like the package file with the suffix *".overrides.json"*. This overrides file can contain any data a package can hold and also a list of tags in addition. If defined in there, this data will override any existing data read from the Asset Store or the header file.

The advantage of this method is that such files can be put in a central location, so everybody who will index the location has the same metadata. This ensures for example the common use of tags and categories if working in a team.

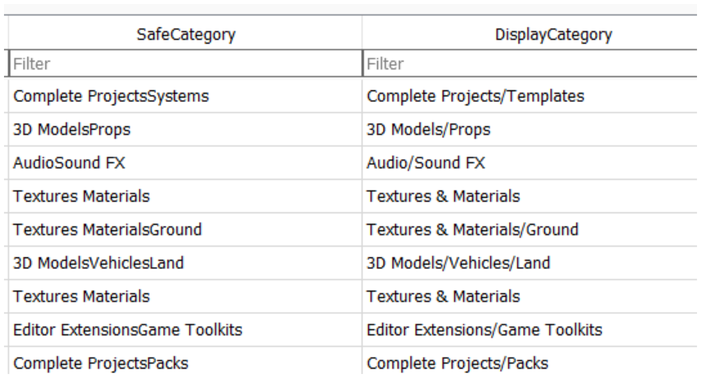


3 Skyboxes.unitypackage	23.01.2024 17:50	Unity package file
3 Skyboxes.unitypackage.overrides.json	15.05.2024 20:06	JSON-Datei

```json
{
    "displayName": "Three Skyboxes",
    "tags": ["sky", "limit"]
}
```

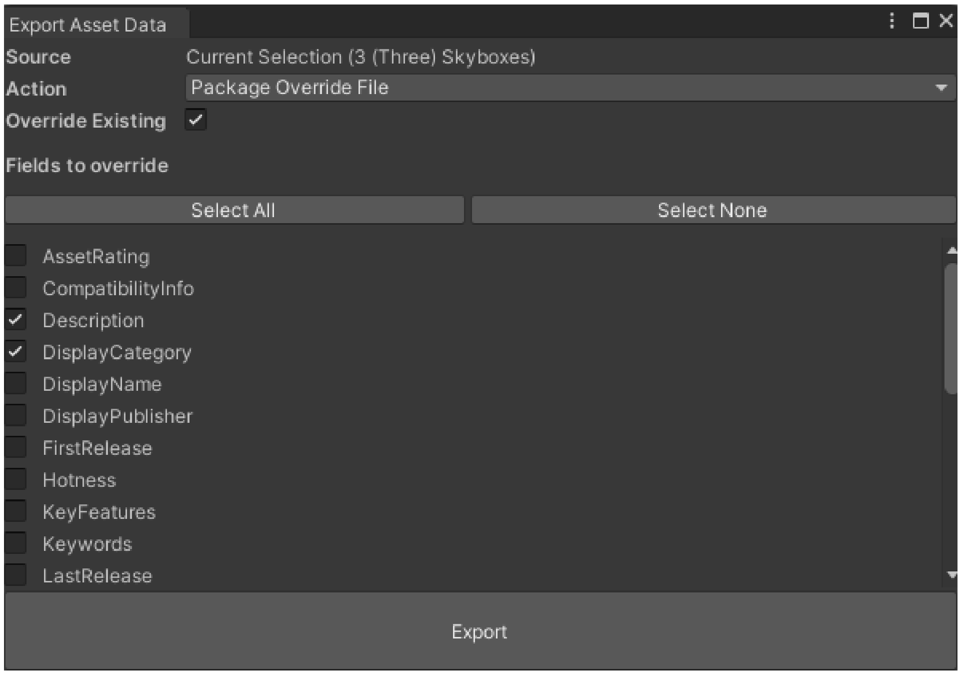
Supported fields: *displayName, displayCategory, safeCategory, displayPublisher, safePublisher, publisherId, slug, revision, description, keyFeatures, compatibilityInfo, supportedUnityVersions, keywords, version, latestVersion, license, licenseLocation, purchaseDate, firstRelease, lastRelease, assetRating, ratingCount, hotness, priceEur, priceUsd, priceCny, requirements, releaseNotes, registry, repository, officialState, tags*

Consider that filtering and grouping works on the *Safe* versions of category and publisher so these should also be set when changing these values. *Safe* versions are named in a way that they could also be used in the file system, without special characters like *&* or */*.



SafeCategory	DisplayCategory
Filter	Filter
Complete ProjectsSystems	Complete Projects/Templates
3D ModelsProps	3D Models/Props
AudioSound FX	Audio/Sound FX
Textures Materials	Textures & Materials
Textures MaterialsGround	Textures & Materials/Ground
3D ModelsVehiclesLand	3D Models/Vehicles/Land
Textures Materials	Textures & Materials
Editor ExtensionsGame Toolkits	Editor Extensions/Game Toolkits
Complete ProjectsPacks	Complete Projects/Packs

To make it easier to create package override files, the package export supports to create these conveniently for an individual or also a list of packages.



Export Asset Data

Source: Current Selection (3 (Three) Skyboxes)

Action: Package Override File

Override Existing: ✓

Fields to override

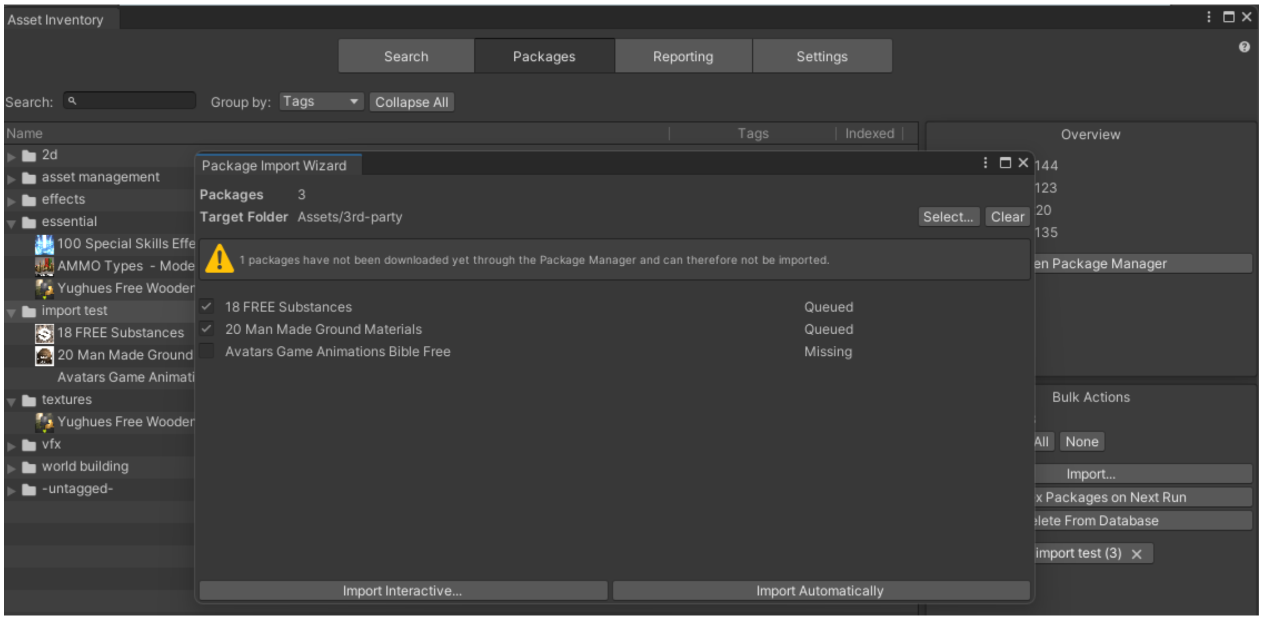
Select All | Select None

- [ ] AssetRating
- [ ] CompatibilityInfo
- [x] Description
- [x] DisplayCategory
- [ ] DisplayName
- [ ] DisplayPublisher
- [ ] FirstRelease
- [ ] Hotness
- [ ] KeyFeatures
- [ ] Keywords
- [ ] LastRelease

Export

IMPORT & BULK IMPORT

Importing packages can happen one-by-one and in bulk while multiple packages are selected. Both interactive and automatic mode is available. In addition, a **custom root folder** under which assets should be imported can be selected. This is especially helpful if the */Assets* root should be kept clean and organized. Packages from registries will be added to the project manifest. If registry packages need a custom scoped registry this will also be added automatically.



Packages 3
Target Folder Assets/3rd-party

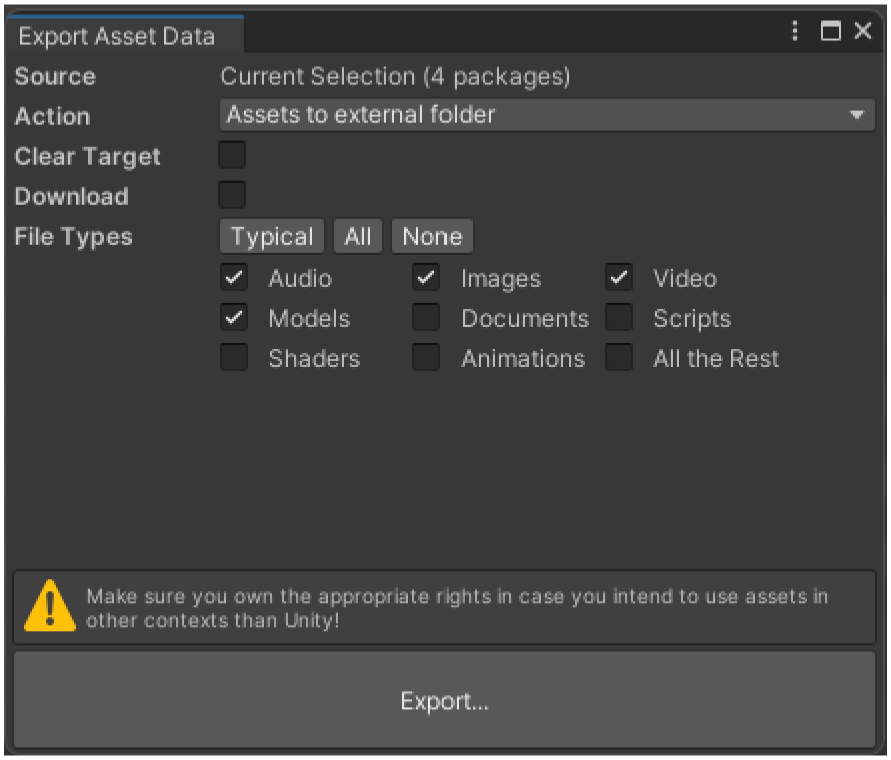
1 packages have not been downloaded yet through the Package Manager and can therefore not be imported.

✓ 18 FREE Substances	Queued
✓ 20 Man Made Ground Materials	Queued
Avatars Game Animations Bible Free	Missing

Importing an archive will extract the archive into the target folder.

EXPORT

Exporting packages can also be done one-by-one or in bulk. The export option (hold down CTRL key) will bring up an export wizard from where the file types to export can be selected. This way it is easy to reuse assets in other contexts.


Export Asset Data

Source: Current Selection (4 packages)

Action: Assets to external folder

Clear Target: ☐

Download: ☐

File Types: Typical | All | None

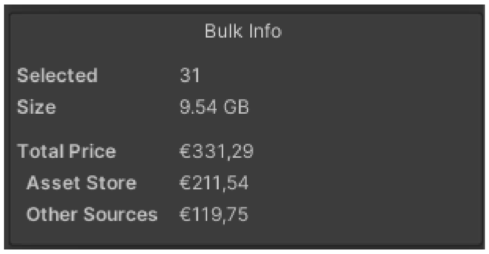
✓ Audio	✓ Images	✓ Video
✓ Models	☐ Documents	☐ Scripts
☐ Shaders	☐ Animations	☐ All the Rest

Make sure you own the appropriate rights in case you intend to use assets in other contexts than Unity!

ADVANCED

In case packages should not appear in the search results and not be indexed, the **Exclude** toggle can be used in the package details. Excluded items will disappear from the list and can be reactivated using the "*Excluded*" maintenance view.

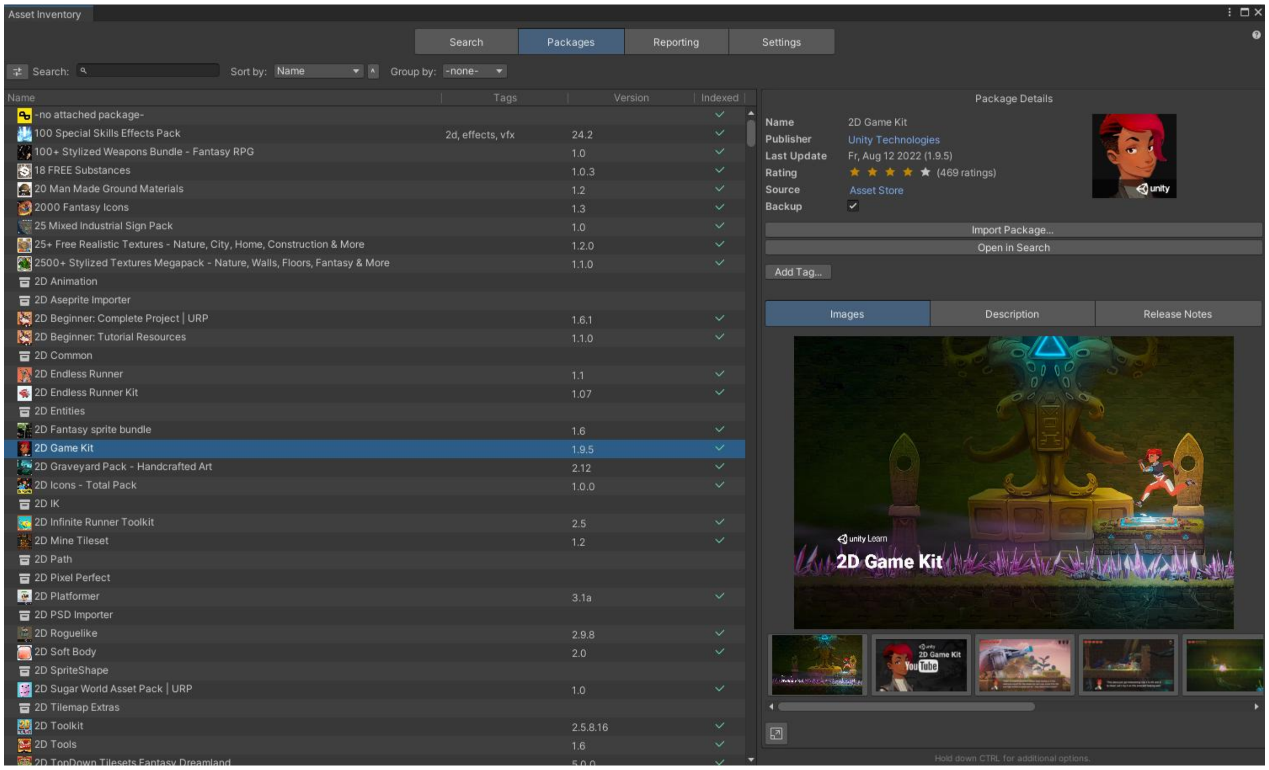
Bulk selection will show the total size of the compressed assets before they are downloaded so that you can estimate if temporarily downloading all for indexing would still fit on the hard drive. If available also costs are shown (current, non-discounted).



Bulk Info

Selected	31
Size	9.54 GB
Total Price	€331,29
Asset Store	€211,54
Other Sources	€119,75

Clicking the icon in the lower right corner will toggle the details view into an **expanded view** twice as big. Depending on availability it will now also display media, description, release notes and dependencies (for registry packages).



Search: Sort by: Name Group by: -none-

Name	Tags	Version	Indexed
-no attached package-			✓
100 Special Skills Effects Pack	2d, effects, vfx	24.2	✓
100+ Stylized Weapons Bundle - Fantasy RPG		1.0	✓
18 FREE Substances		1.0.3	✓
20 Man Made Ground Materials		1.2	✓
2000 Fantasy Icons		1.3	✓
25 Mixed Industrial Sign Pack		1.0	✓
25+ Free Realistic Textures - Nature, City, Home, Construction & More		1.2.0	✓
2500+ Stylized Textures Megapack - Nature, Walls, Floors, Fantasy & More		1.1.0	✓
2D Animation			
2D Aseprite Importer			
2D Beginner: Complete Project | URP		1.6.1	✓
2D Beginner: Tutorial Resources		1.1.0	✓
2D Common			
2D Endless Runner		1.1	✓
2D Endless Runner Kit		1.07	✓
2D Entities			
2D Fantasy sprite bundle		1.6	✓
2D Game Kit		1.9.5	✓
2D Graveyard Pack - Handcrafted Art		2.12	✓
2D Icons - Total Pack		1.0.0	✓
2D IK			
2D Infinite Runner Toolkit		2.5	✓
2D Mine Tileset		1.2	✓
2D Path			
2D Pixel Perfect			
2D Platformer		3.1a	✓
2D PSD Importer			
2D Roguelike		2.9.8	✓
2D Soft Body		2.0	✓
2D SpriteShape			
2D Sugar World Asset Pack | URP		1.0	✓
2D Tilemap Extras			
2D Toolkit		2.5.8.16	✓
2D Tools		1.6	✓
2D TopDown Tilesets Fantasy Dreamland		5.0.0	✓

Package Details

Name	2D Game Kit
Publisher	Unity Technologies
Last Update	Fr, Aug 12 2022 (1.9.5)
Rating	★★★★★ (469 ratings)
Source	Asset Store
Backup	✓

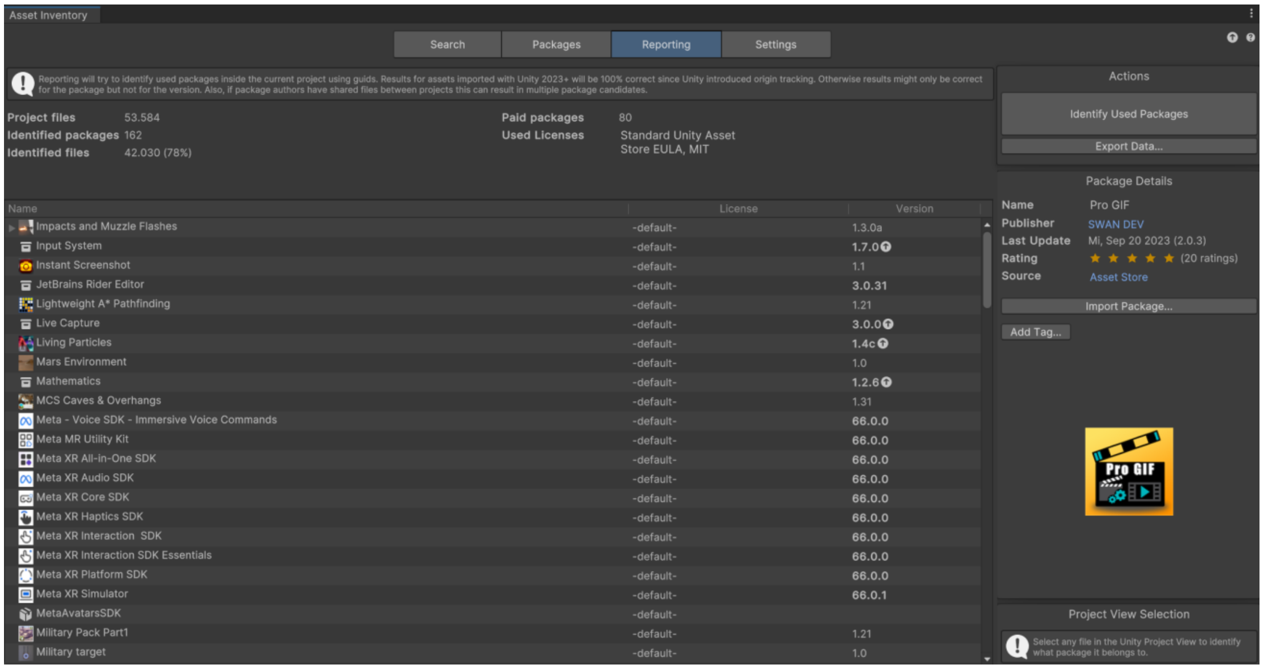
Hold down CTRL for additional options.

Holding CTRL will reveal much additional package data and also many actions.

REPORTING

OVERVIEW

This module will scan the complete project and try to identify packages that were being used. It also supports exporting your data and listing used licenses.



Reporting will try to identify used packages inside the current project using guids. Results for assets imported with Unity 2023+ will be 100% correct since Unity introduced origin tracking. Otherwise results might only be correct for the package but not for the version. Also, if package authors have shared files between projects this can result in multiple package candidates.

Project files	53.584	**Paid packages**	80
Identified packages	162	**Used Licenses**	Standard Unity Asset Store EULA, MIT
Identified files	42.030 (78%)		

Name	License	Version
Impacts and Muzzle Flashes	-default-	1.3.0a
Input System	-default-	**1.7.0**
Instant Screenshot	-default-	1.1
JetBrains Rider Editor	-default-	**3.0.31**
Lightweight A* Pathfinding	-default-	1.21
Live Capture	-default-	**3.0.0**
Living Particles	-default-	**1.4c**
Mars Environment	-default-	1.0
Mathematics	-default-	**1.2.6**
MCS Caves & Overhangs	-default-	1.31
Meta - Voice SDK - Immersive Voice Commands	-default-	**66.0.0**
Meta MR Utility Kit	-default-	**66.0.0**
Meta XR All-in-One SDK	-default-	**66.0.0**
Meta XR Audio SDK	-default-	**66.0.0**
Meta XR Core SDK	-default-	**66.0.0**
Meta XR Haptics SDK	-default-	**66.0.0**
Meta XR Interaction SDK	-default-	**66.0.0**
Meta XR Interaction SDK Essentials	-default-	**66.0.0**
Meta XR Platform SDK	-default-	**66.0.0**
Meta XR Simulator	-default-	**66.0.1**
MetaAvatarsSDK	-default-	
Military Pack Part1	-default-	1.21
Military target	-default-	1.0

Name	Pro GIF
Publisher	SWAN DEV
Last Update	Mi, Sep 20 2023 (2.0.3)
Rating	★★★★★ (20 ratings)
Source	Asset Store

Select any file in the Unity Project View to identify what package it belongs to.

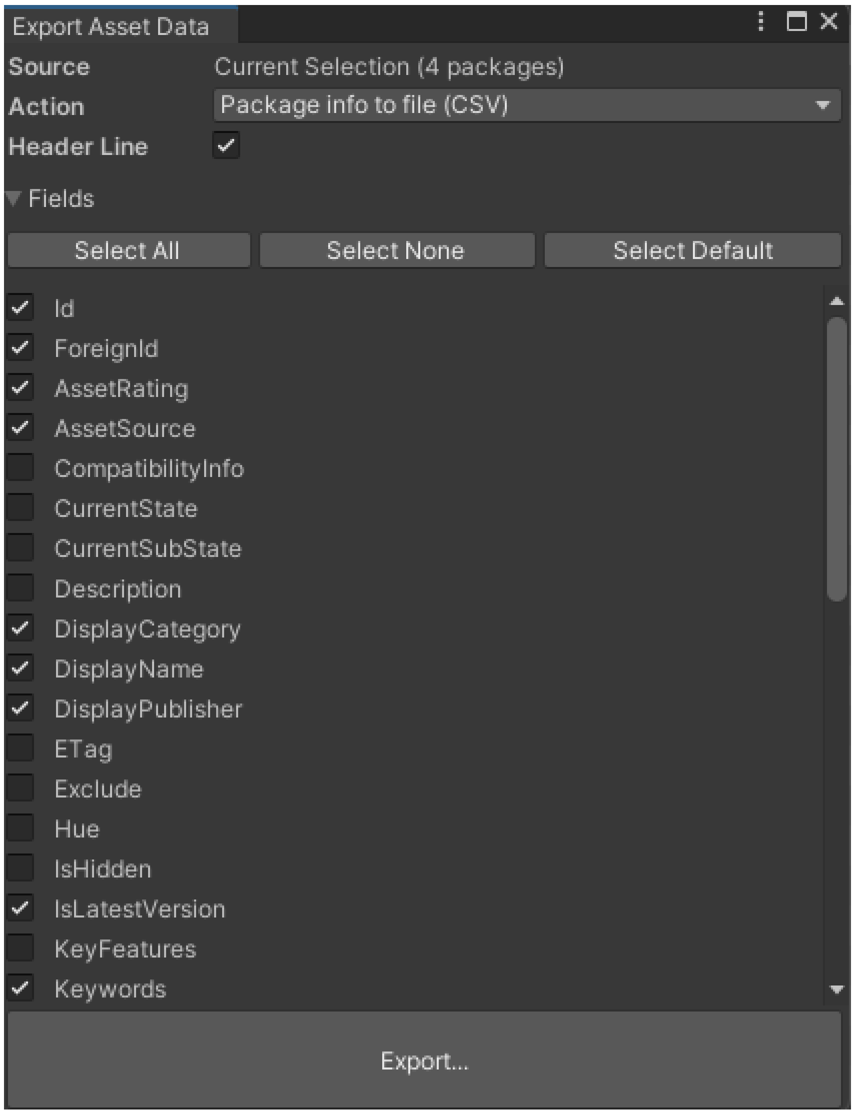
Packages that could be identified with 100% **confidence**, where also the version is guaranteed to be correct, will be shown with a **bold** version column. All registry packages automatically fall into this category. From Unity 2023+ also local assets are typically identified correctly if imported through Unity 2023+, since it will store the asset origin in the meta data files allowing to pinpoint the exact version correctly.

When selecting an asset in the Unity *Project View*, the reporting tab will try to identify the associated package in the lower right info box.

DATA EXPORT

In case you want to use your own analysis or presentation style it is possible to export the database contents into other formats. Currently supported is:

- CSV: can easily be handled in Excel and other compatible software
- Assets: see section above in the Packages chapter
- Licenses: will list all packages with a non-default license in MD format
- Overrides: metadata which can be used to override package-specific metadata in team scenarios (e.g. to enforce common tags & categories)


Export Asset Data

Source: Current Selection (4 packages)
Action: Package info to file (CSV)
Header Line: ✓

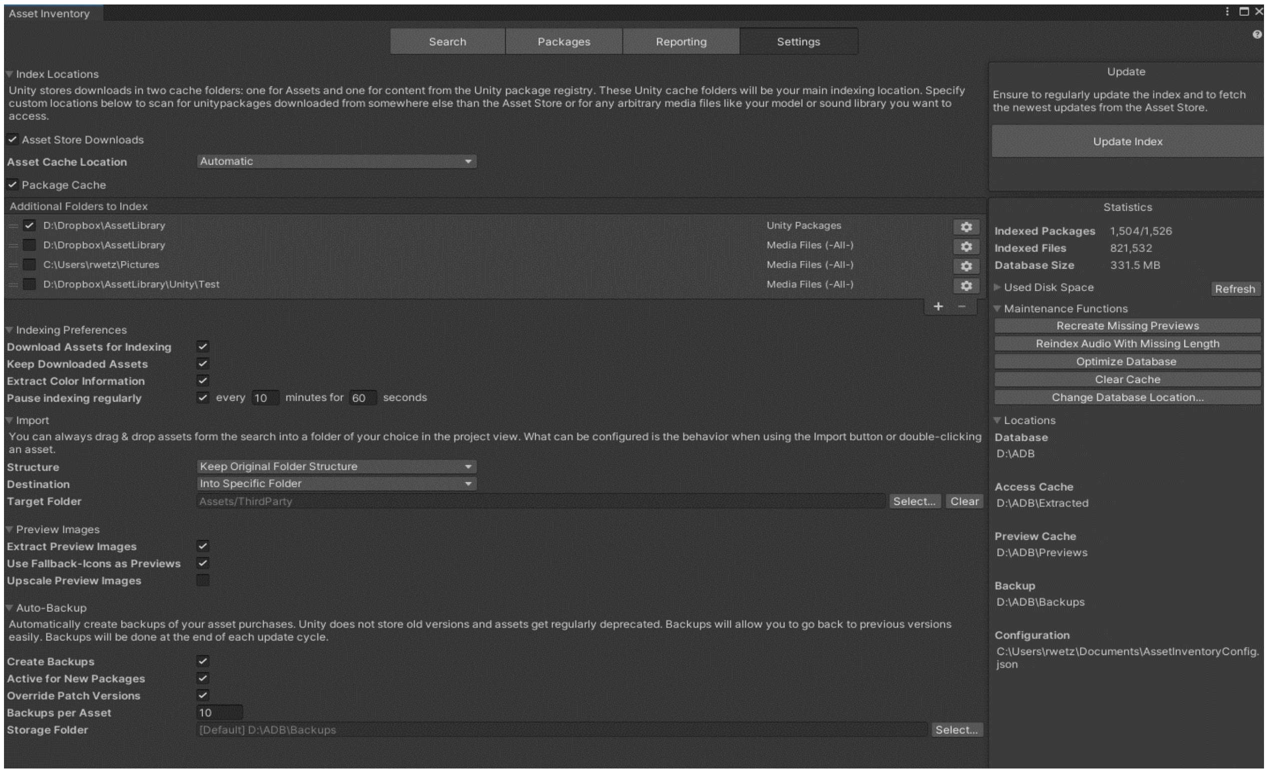
Fields

- [x] Id
- [x] ForeignId
- [x] AssetRating
- [x] AssetSource
- [ ] CompatibilityInfo
- [ ] CurrentState
- [ ] CurrentSubState
- [ ] Description
- [x] DisplayCategory
- [x] DisplayName
- [x] DisplayPublisher
- [ ] ETag
- [ ] Exclude
- [ ] Hue
- [ ] IsHidden
- [x] IsLatestVersion
- [ ] KeyFeatures
- [x] Keywords

SETTINGS

OVERVIEW

Use this section to configure what should be indexed. Indexing can be started and stopped at any time and will pick off again where it last stopped.



Index Locations

Unity stores downloads in two cache folders: one for Assets and one for content from the Unity package registry. These Unity cache folders will be your main indexing location. Specify custom locations below to scan for unitypackages downloaded from somewhere else than the Asset Store or for any arbitrary media files like your model or sound library you want to access.

✓ Asset Store Downloads

Asset Cache Location: Automatic

✓ Package Cache

Additional Folders to Index

✓	D:\Dropbox\AssetLibrary	Unity Packages
	D:\Dropbox\AssetLibrary	Media Files (-All-)
	C:\Users\rwetz\Pictures	Media Files (-All-)
	D:\Dropbox\AssetLibrary\Unity\Test	Media Files (-All-)

Indexing Preferences

Download Assets for Indexing	✓
Keep Downloaded Assets	✓
Extract Color Information	✓
Pause indexing regularly	✓ every 10 minutes for 60 seconds

Import

You can always drag & drop assets form the search into a folder of your choice in the project view. What can be configured is the behavior when using the Import button or double-clicking an asset.

Structure	Keep Original Folder Structure
Destination	Into Specific Folder
Target Folder	Assets/ThirdParty

Preview Images

Extract Preview Images	✓
Use Fallback-Icons as Previews	✓
Upscale Preview Images	

Auto-Backup

Automatically create backups of your asset purchases. Unity does not store old versions and assets get regularly deprecated. Backups will allow you to go back to previous versions easily. Backups will be done at the end of each update cycle.

Create Backups	✓
Active for New Packages	✓
Override Patch Versions	✓
Backups per Asset	10
Storage Folder	[Default] D:\ADB\Backups

Update

Ensure to regularly update the index and to fetch the newest updates from the Asset Store.

Update Index

Statistics

Indexed Packages	1,504/1,526
Indexed Files	821,532
Database Size	331.5 MB

Used Disk Space

Maintenance Functions

Recreate Missing Previews
Reindex Audio With Missing Length
Optimize Database
Clear Cache
Change Database Location...

Locations

Database
D:\ADB

Access Cache
D:\ADB\Extracted

Preview Cache
D:\ADB\Previews

Backup
D:\ADB\Backups

Configuration
C:\Users\rwetz\Documents\AssetInventoryConfig.json

Asset Store downloads and Packages are activated by default. There are two options available:

Index everything that is already downloaded into the local cache.

- That means you only need to click the **Download** button in the **Package Manager** window or right in the Asset Inventory for each asset you own (without importing them) to make them available for the index (bulk selection is also possible).
- If you have specified a **custom asset cache directory** in Unity 2022.1 or higher, it will typically be auto-detected. Press CTRL to see the resolved path and verify. You can manually override the location by switching the Asset Cache Location to *Custom*. If the environment variable *ASSETSTORE_CACHE_PATH* is set it will override the default asset cache location.

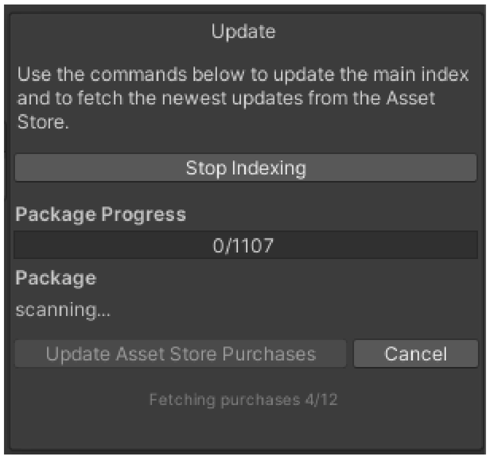
Automatically download purchased assets for indexing and delete them again afterwards.

- With this option you will **index everything you have** without the need to download everything, saving a lot of disk space, at the expense of a potentially very long indexing due to all the necessary downloads.

During a normal update cycle (whenever "Update Index" is clicked) the following actions are done in this order (shown also when clicking "What will happen?"):

- Fetch all Asset Store purchases
- In the background, fetch details for all purchases and for assets that received updates
- In parallel scan for new local Asset Store packages and update metadata
- Do the same for registry packages
- Index the contents of new or changed packages
- Index any additional folders
- Index Unity Asset Manager cloud content
- Extract colors of new assets for color search
- Create AI captions
- Perform backup activities

All activities are incremental, meaning you can **interrupt and restart these at any time** and they will mostly continue where they left off.


Update

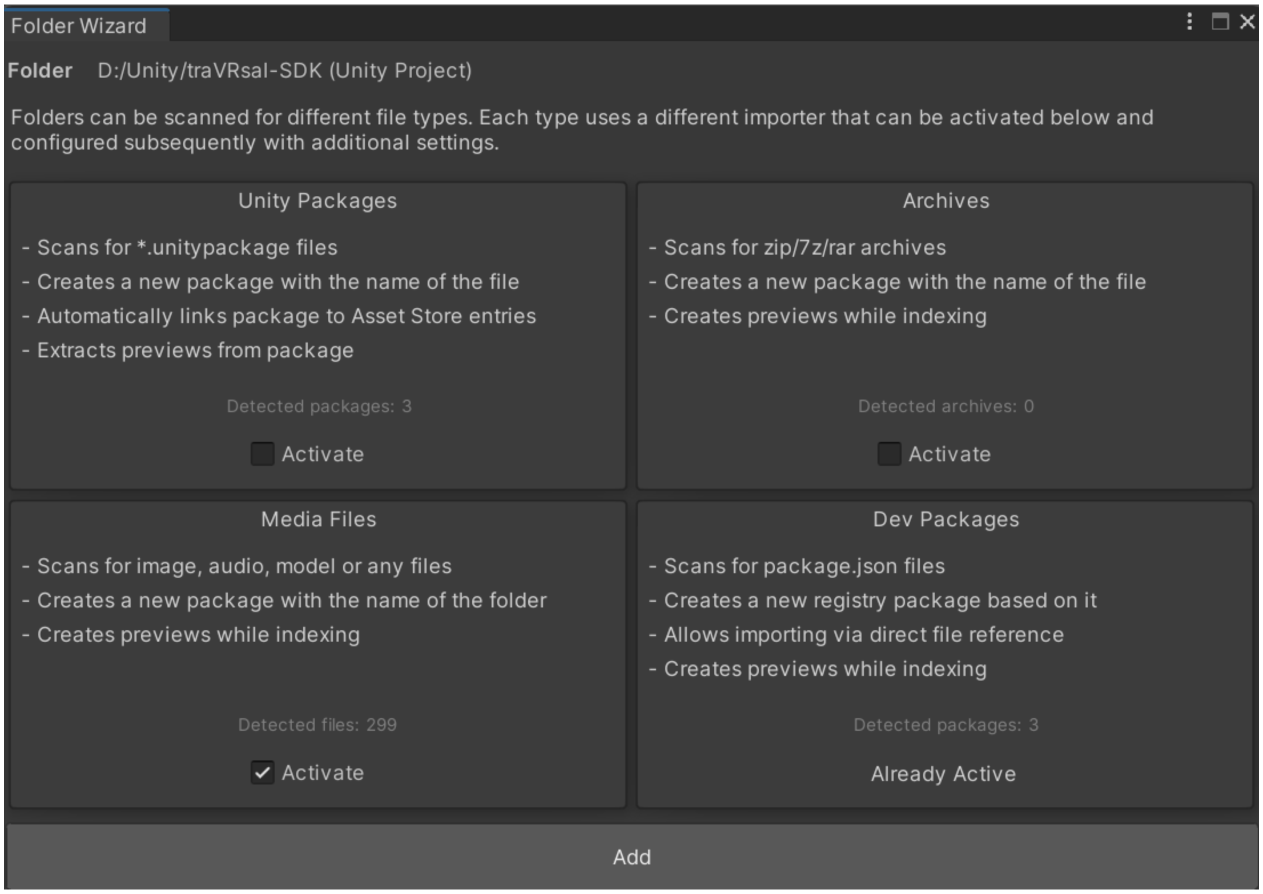
Use the commands below to update the main index and to fetch the newest updates from the Asset Store.

Once an asset is indexed it does not need to remain on your hard drive if you just want to search for it. Only when importing it needs to be available.

LOCAL FOLDERS

If you want to index Unity packages downloaded from **other sources**, simply add the folders to the **additional folders** list. When selecting an entry from the list of additional folders, more options can be specified on the right-hand side. This way it is possible to select what types of files should be searched for:

- **Unity Packages**: Finds any *.unitypackage* files in the folder and indexes the content.
- **Development Packages**: Finds any *package.json* files and treats those as local packages.
- **Media Files**: Allows to index all kinds of other, **free-floating** assets, like images, models, audio libraries etc. Also Unity projects can be indexed this way.
- **Archives**: Will extract archives (zip, rar, 7z) on the fly and index all contents.



Folder D:/Unity/traVRsal-SDK (Unity Project)

Folders can be scanned for different file types. Each type uses a different importer that can be activated below and configured subsequently with additional settings.

Unity Packages
- Scans for *.unitypackage files
- Creates a new package with the name of the file
- Automatically links package to Asset Store entries
- Extracts previews from package

Detected packages: 3

☐ Activate

Archives
- Scans for zip/7z/rar archives
- Creates a new package with the name of the file
- Creates previews while indexing

Detected archives: 0

☐ Activate

Media Files
- Scans for image, audio, model or any files
- Creates a new package with the name of the folder
- Creates previews while indexing

Detected files: 299

☑ Activate

Dev Packages
- Scans for package.json files
- Creates a new registry package based on it
- Allows importing via direct file reference
- Creates previews while indexing

Detected packages: 3

Already Active

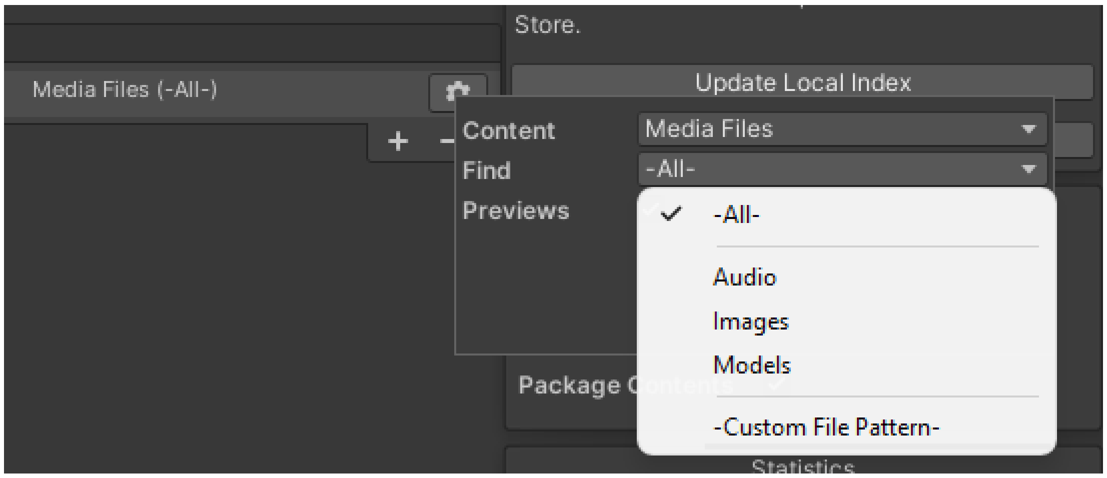
Using additional media folders will let Asset Inventory act as the go-to place for asset management and quickly finding any available asset on your drives from a single, consistent UI. Media files can optionally have no connected asset but otherwise behave the same. If they have a *.meta* file next to them, their guids will also be stored.

The order in which additional folders are processed can be changed by dragging them up or down in the list. Deactivating or removing an additional folder entry will not remove it from the index.

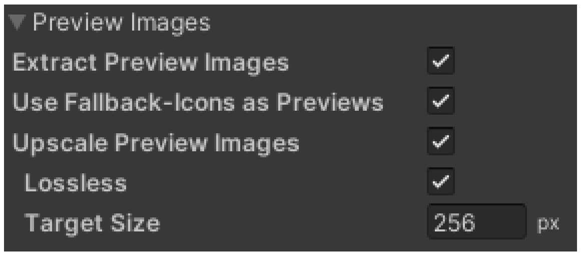
PREVIEWS

Preview file generation will work by temporarily copying each file into the current Unity project, letting Unity create a preview and removing it again. This process will take a bit of time but will allow Asset Inventory to show preview images and also additional meta data for many file types.

Image files will be handled separately for common file types, bypassing Unity, which is much faster and will, if active, automatically scale preview images to a higher resolution already during indexing. This will work for Unity 2021.2 and above.



Preview creation options:



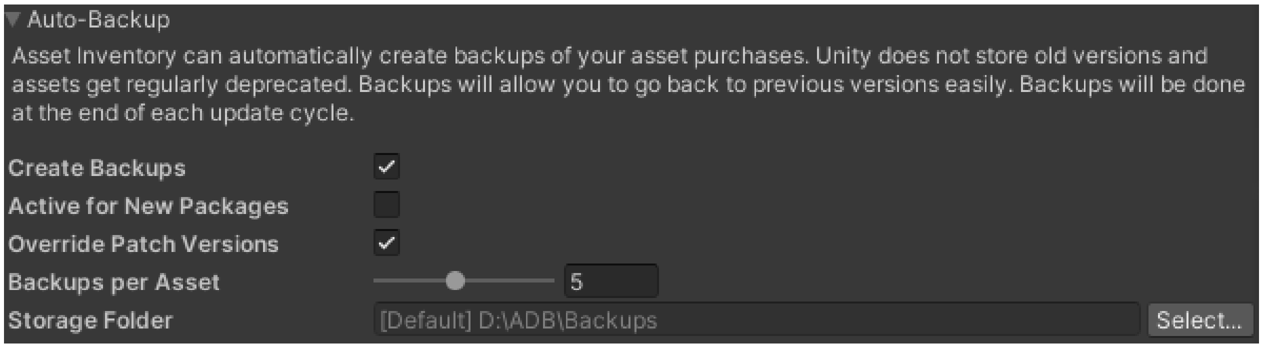
Preview Images

Extract Preview Images	✓
Use Fallback-Icons as Previews	✓
Upscale Preview Images	✓
Lossless	✓
Target Size	256 px

BACKUP

The tool supports copying packages to a dedicated place for permanent backup. This way you are not affected by package deprecation by Unity where it might be impossible to download older (but for you working or compatible) versions of assets anymore.

The backup tool can keep multiple versions of packages and will automatically handle the life cycle, version comparison and more.

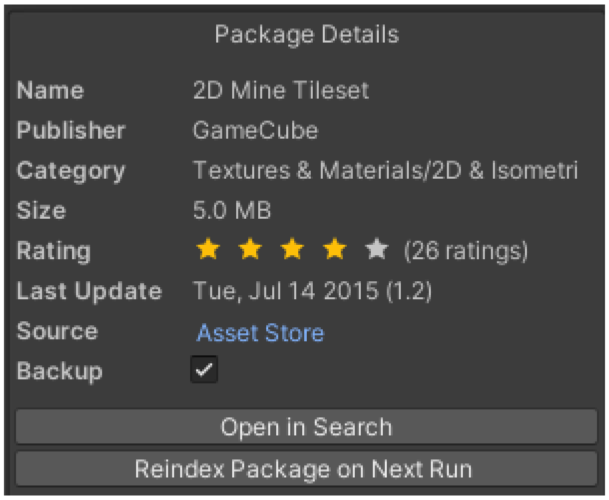


Auto-Backup

Asset Inventory can automatically create backups of your asset purchases. Unity does not store old versions and assets get regularly deprecated. Backups will allow you to go back to previous versions easily. Backups will be done at the end of each update cycle.

Create Backups	✓
Active for New Packages	
Override Patch Versions	✓
Backups per Asset	5
Storage Folder	[Default] D:\ADB\Backups

Once backup is activated it can be set individually per package in the package details and in the next “Update Index” run will back up the selected packages.



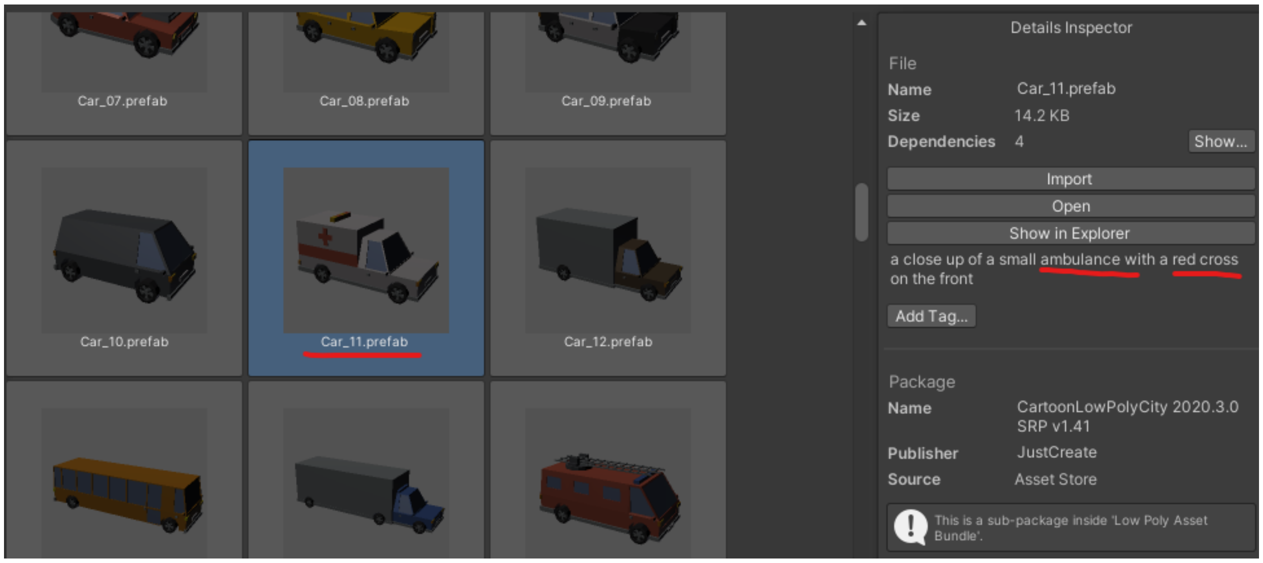
Package Details

Name	2D Mine Tileset
Publisher	GameCube
Category	Textures & Materials/2D & Isometri
Size	5.0 MB
Rating	★ ★ ★ ★ ★ (26 ratings)
Last Update	Tue, Jul 14 2015 (1.2)
Source	Asset Store
Backup	✓

ARTIFICIAL INTELLIGENCE

These are preview features and feedback is highly appreciated.

Using AI it is possible to automatically create captions that describe individual asset files. This way you are also able to find files which are labeled inappropriate or insufficient. See the example below:



File

Name	Car_11.prefab
Size	14.2 KB
Dependencies	4

a close up of a small ambulance with a red cross on the front

Package

Name	CartoonLowPolyCity 2020.3.0 SRP v1.41
Publisher	JustCreate
Source	Asset Store

This is a sub-package inside 'Low Poly Asset Bundle'.

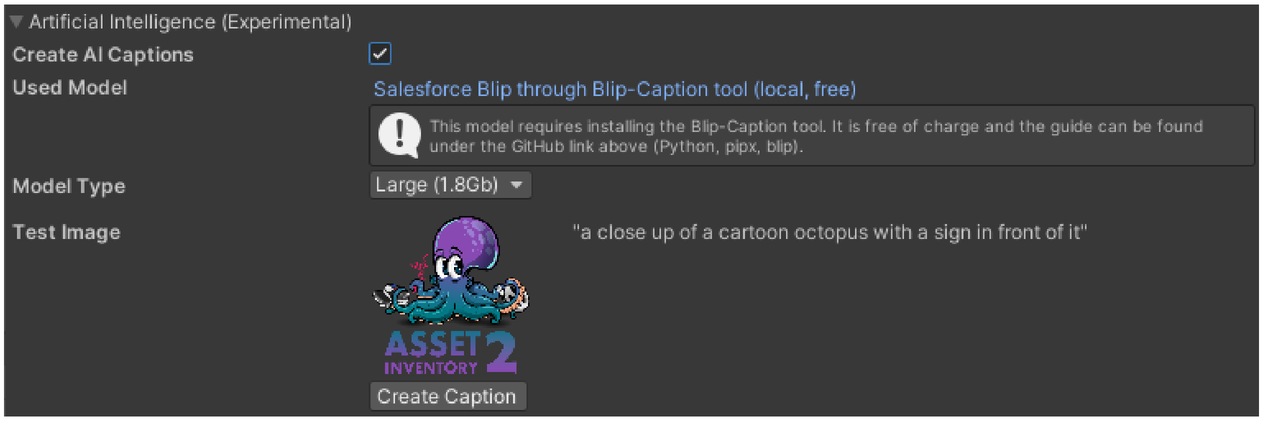
While the asset is simply called *"Car_11"*, the AI has captioned it *"a close up of a small ambulance with a red cross on the front"*.

When searching for *"ambulance"* now and using the AI generated captions during the search will also yield *"Car_11"* as a result, which would have been impossible to find before.

Setup

AI inferencing will happen on your own device right now. This means it does not use any remote server and also does not incur costs for a third party service. It requires a potent PC though as the process is resource intensive. The best free currently available image captioning model is Salesforce Blip, coming in a base and a large version. To use it locally, Python 3, pipx and the blip-caption tool are required to be installed.

On the *Settings* tab the successful installation can be tested. Once activated, the tool will create AI captions during the next Update cycle.



Artificial Intelligence (Experimental)

Create AI Captions ✓

Used Model: Salesforce Blip through Blip-Caption tool (local, free)

This model requires installing the Blip-Caption tool. It is free of charge and the guide can be found under the GitHub link above (Python, pipx, blip).

Model Type: Large (1.8Gb)

Test Image: "a close up of a cartoon octopus with a sign in front of it"

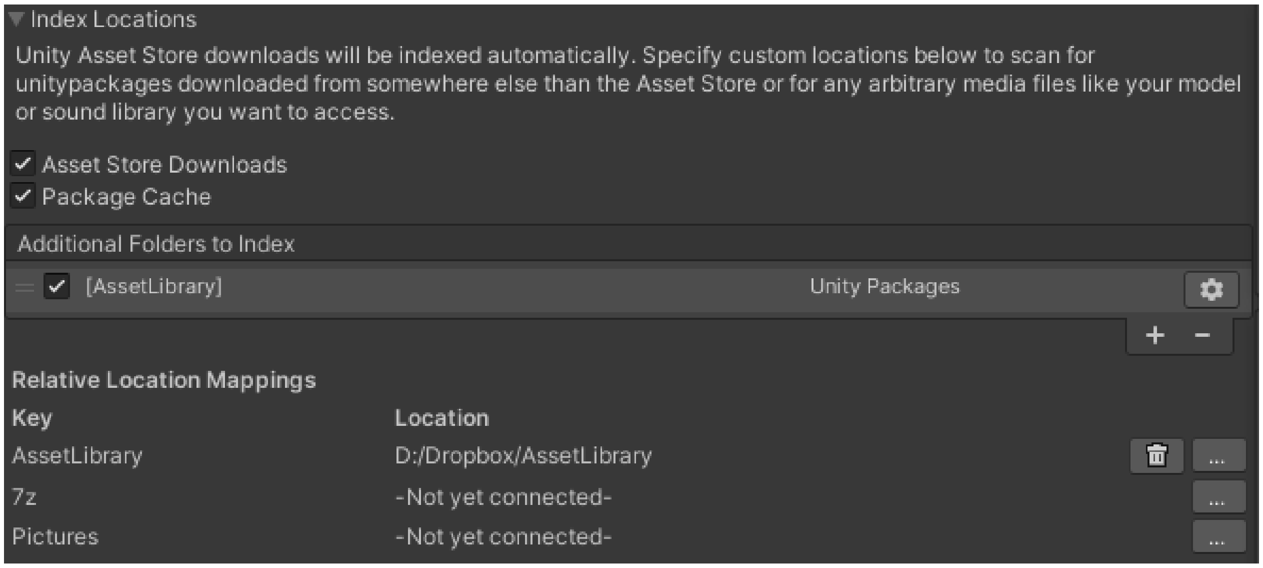
There is support for GPU acceleration but it requires a custom patch of the blip tool to be installed. You also need to make sure to have PyTorch with GPU support as well as a CUDA runtime installed. Instructions can be found on the official website.

Once that is done you can select *CUDA* as the target device for the model, which dramatically speeds up the generation of AI captions.

CROSS-DEVICE USAGE

It is possible to use the same database from multiple devices. This way the indexed assets can be browsed and seen from every device. The easiest way to achieve this is to make the database and all asset folders available through a mounted network drive. If the drives and folders are identical from each device, it will instantly work this way. Also if Asset and Package cache folders differ the database will be compatible since these paths will be stored with *[ac]* and *[pc]* keys (see below). Furthermore, all paths are stored with forward slashes, making the database compatible between different operating systems.

Sometimes drives or folder mappings cannot be kept identical for various reasons. The tool supports this scenario through a mechanism called **Relative Persistence**. Each additional folder entry is replaced with a **key** and all occurrences in the database to this folder are also replaced with this key. On each device the key can then be **mapped** to the required drive and folder. This is a one-time action and once done the tool can be used as always.


Index Locations

Unity Asset Store downloads will be indexed automatically. Specify custom locations below to scan for unitypackages downloaded from somewhere else than the Asset Store or for any arbitrary media files like your model or sound library you want to access.

✓ Asset Store Downloads
✓ Package Cache

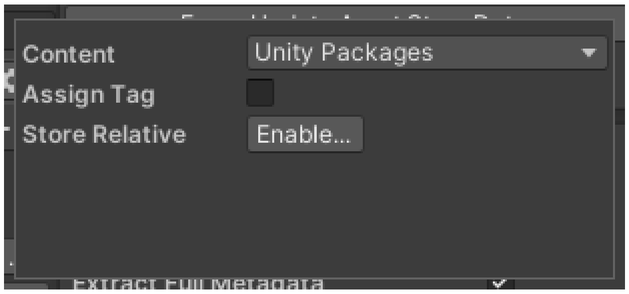
Additional Folders to Index

✓ [AssetLibrary] Unity Packages

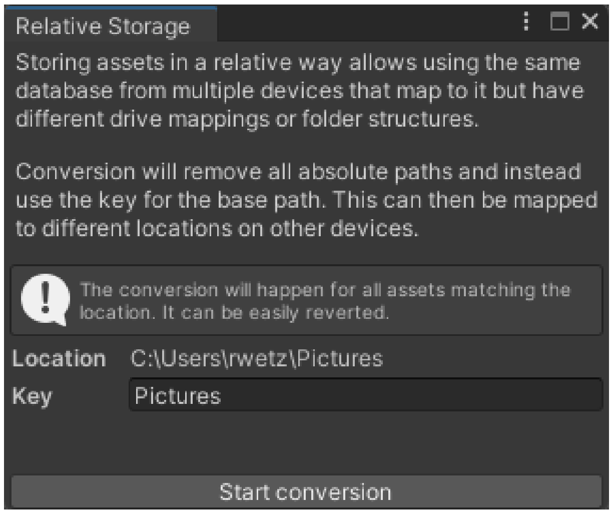
Relative Location Mappings

Key	Location
AssetLibrary	D:/Dropbox/AssetLibrary
7z	-Not yet connected-
Pictures	-Not yet connected-

In order to convert a folder to relative format, open the folder settings, hold down CTRL and select **Enable…**



Content: Unity Packages
Assign Tag
Store Relative: Enable...



Relative Storage

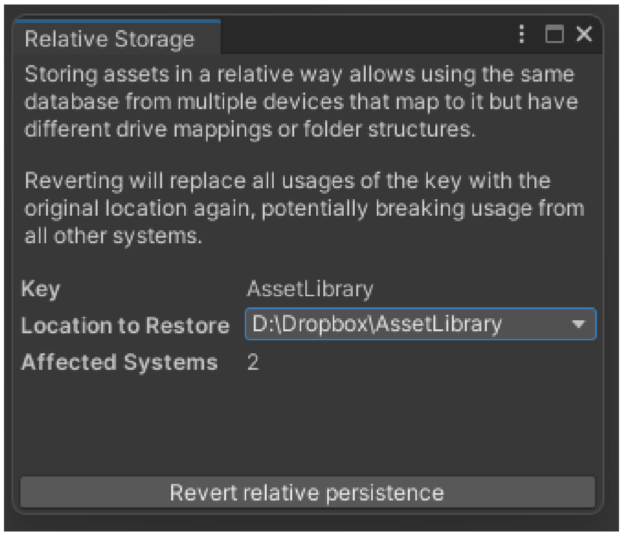
Storing assets in a relative way allows using the same database from multiple devices that map to it but have different drive mappings or folder structures.

Conversion will remove all absolute paths and instead use the key for the base path. This can then be mapped to different locations on other devices.

The conversion will happen for all assets matching the location. It can be easily reverted.

Location: C:\Users\rwetz\Pictures
Key: Pictures

If a folder is already converted it can be **switched back** again through the same mechanism. This will remove all mappings and reorganize the database to contain the full paths again.

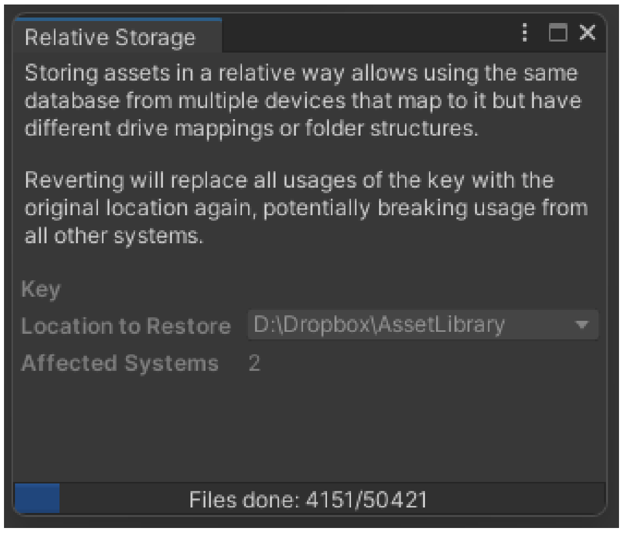


Relative Storage

Storing assets in a relative way allows using the same database from multiple devices that map to it but have different drive mappings or folder structures.

Reverting will replace all usages of the key with the original location again, potentially breaking usage from all other systems.

Key: AssetLibrary
Location to Restore: D:\Dropbox\AssetLibrary
Affected Systems: 2



Relative Storage

Storing assets in a relative way allows using the same database from multiple devices that map to it but have different drive mappings or folder structures.

Reverting will replace all usages of the key with the original location again, potentially breaking usage from all other systems.

Key
Location to Restore: D:\Dropbox\AssetLibrary
Affected Systems: 2

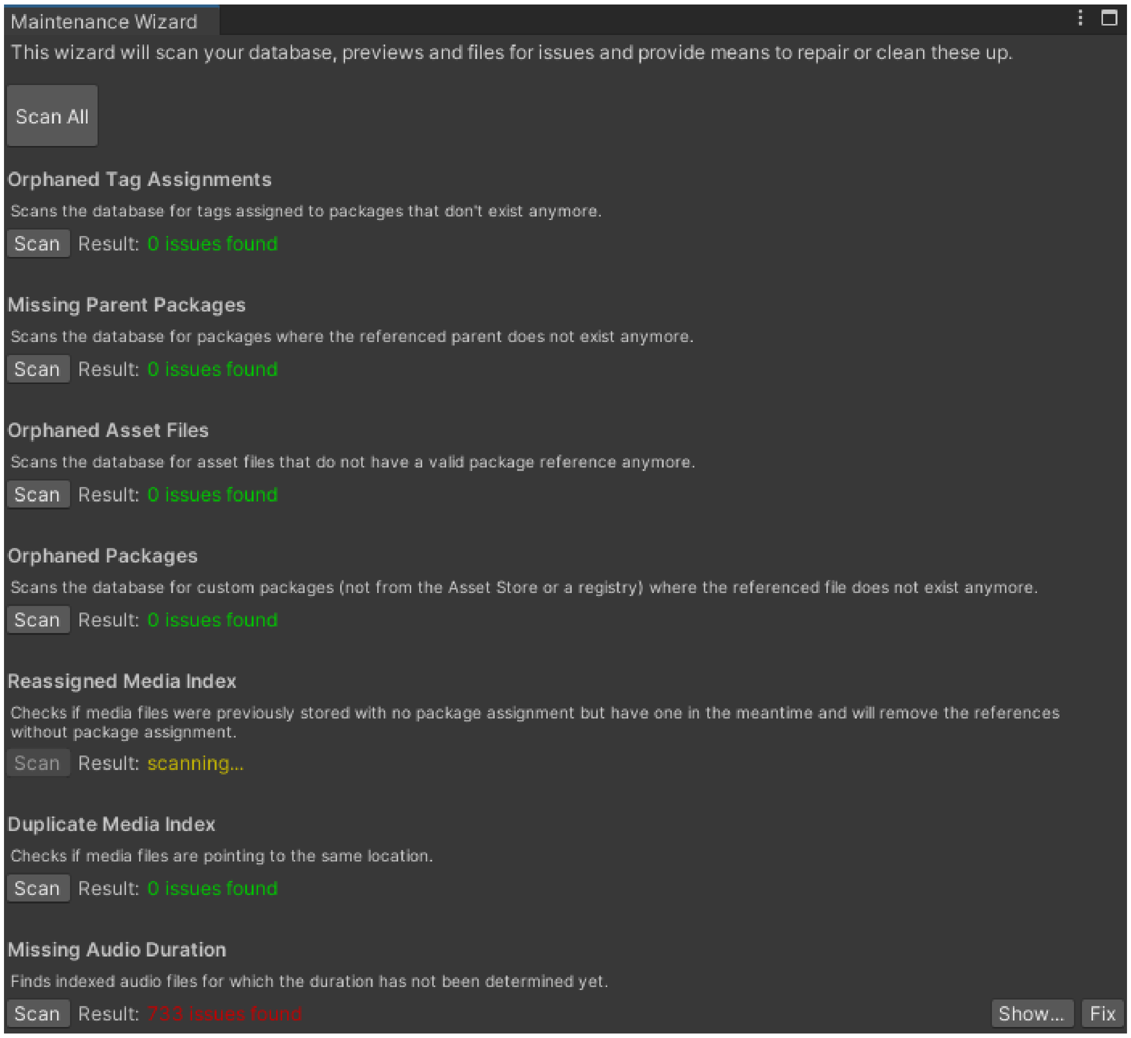
Files done: 4151/50421

MAINTENANCE

Maintenance functions are also located on the Settings tab:

Harmless & recommended to run every now and then

⇨ Maintenance Wizard: will scan for common database and file system inconsistencies and offer automatic solutions



Maintenance Wizard

This wizard will scan your database, previews and files for issues and provide means to repair or clean these up.

Scan All

Orphaned Tag Assignments

Scans the database for tags assigned to packages that don't exist anymore.

Scan Result: 0 issues found

Missing Parent Packages

Scans the database for packages where the referenced parent does not exist anymore.

Scan Result: 0 issues found

Orphaned Asset Files

Scans the database for asset files that do not have a valid package reference anymore.

Scan Result: 0 issues found

Orphaned Packages

Scans the database for custom packages (not from the Asset Store or a registry) where the referenced file does not exist anymore.

Scan Result: 0 issues found

Reassigned Media Index

Checks if media files were previously stored with no package assignment but have one in the meantime and will remove the references without package assignment.

Scan Result: scanning...

Duplicate Media Index

Checks if media files are pointing to the same location.

Scan Result: 0 issues found

Missing Audio Duration

Finds indexed audio files for which the duration has not been determined yet.

Scan Result: 733 issues found Show... Fix

⇨ Recreate Previews: will try to create preview images for asset files which do not have ones so far or were flagged to be recreated. This can take quite some time especially for complex prefabs.

⇨ Optimize Database: compacts and reorganizes the database for more performance and less required space, should be done after bigger indexing activities

Harmless

- Close Database: allow backing up or copying the file
- Change Database Location: set a different location for the database, move the current, use another existing one or create a new one
- Clear Cache: deletes the temporary files to save space, can be done without any side-effects

Potentially destructive

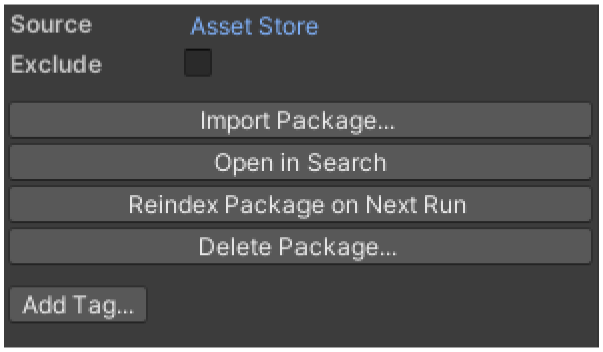
- Clear Database: deletes the complete database to start over
- Reset Configuration: set everything to like it was initially, removing any additional configuration that was done

ADVANCED FEATURES

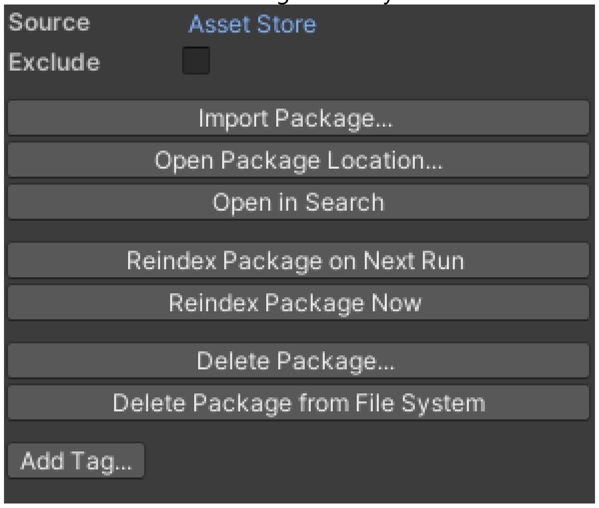
EXPERT FUNCTIONS

Many more seldom used and advanced features are initially hidden to not clutter the UI. They can be made visible by holding down the **CTRL key**.

Normal View



Holding CTRL Key



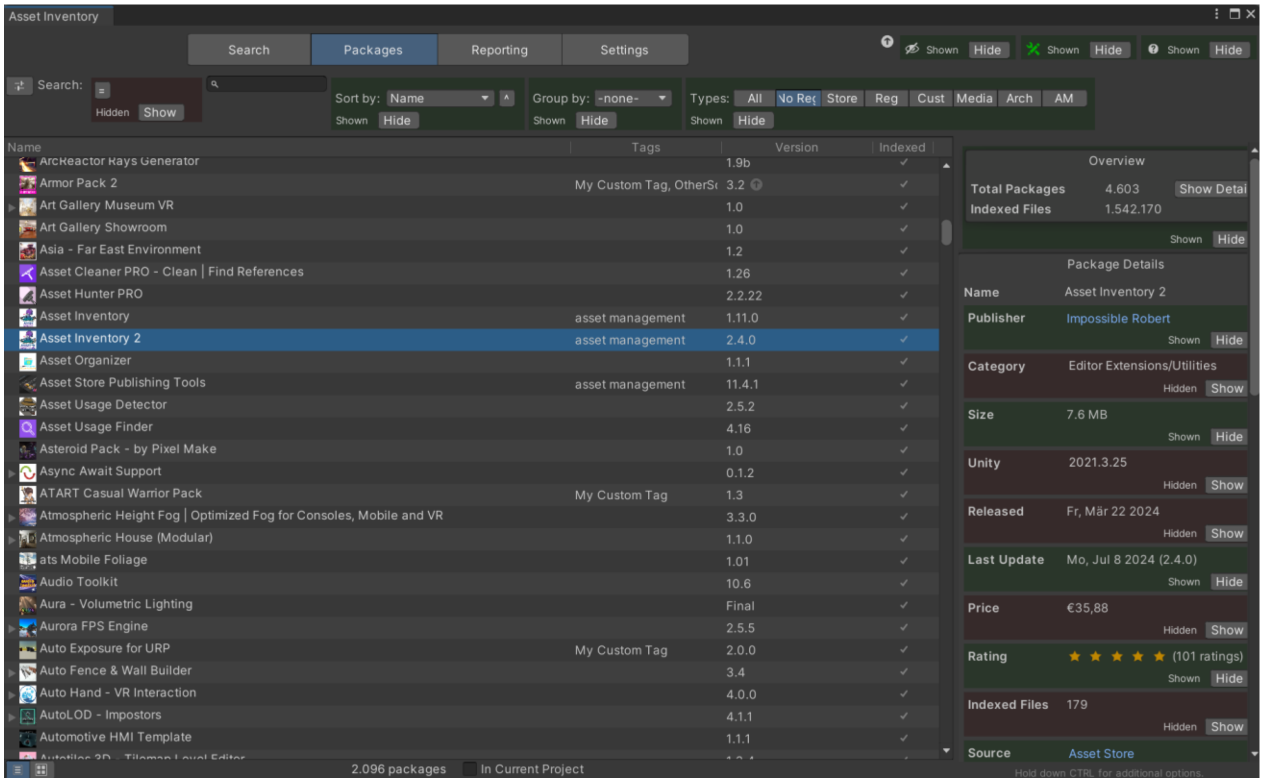
Expert function visibility will not only apply to buttons but also to **additional information** which is not often needed. This ensures the UI is clean for the 90% use-case while also catering to experts on-demand. There are two ways to influence the visibility of the advanced UI:

- Temporarily toggle it on via the eye icon in the upper toolbar or permanently via a setting in the *Advanced* section on the settings tab.
- Customize what should be considered “advanced” (see next section).

UI CUSTOMIZATION

Using the customization icon in the upper toolbar, the UI can be switched to *design-mode* (and back). When active, sections (labels, buttons, info boxes, panels…) that can be customized will be highlighted in green and red and additional actions will be visible. Green sections will always be shown. Red sections will be considered *advanced* and only shown when advanced UI is toggled.

During design-mode, all advanced UI will automatically be shown. Depending on the current selection and context, not all UI elements will be visible (e.g. registry packages will show different actions than Asset Store packages).



Name	Tags	Version	Indexed
ArcReactor Rays Generator		1.9b	✓
Armor Pack 2	My Custom Tag, OtherS	3.2	✓
Art Gallery Museum VR		1.0	✓
Art Gallery Showroom		1.0	✓
Asia - Far East Environment		1.2	✓
Asset Cleaner PRO - Clean | Find References		1.26	✓
Asset Hunter PRO		2.2.22	✓
Asset Inventory	asset management	1.11.0	✓
Asset Inventory 2	asset management	2.4.0	✓
Asset Organizer		1.1.1	✓
Asset Store Publishing Tools	asset management	11.4.1	✓
Asset Usage Detector		2.5.2	✓
Asset Usage Finder		4.16	✓
Asteroid Pack - by Pixel Make		1.0	✓
Async Await Support		0.1.2	✓
ATART Casual Warrior Pack	My Custom Tag	1.3	✓
Atmospheric Height Fog | Optimized Fog for Consoles, Mobile and VR		3.3.0	✓
Atmospheric House (Modular)		1.1.0	✓
ats Mobile Foliage		1.01	✓
Audio Toolkit		10.6	✓
Aura - Volumetric Lighting		Final	✓
Aurora FPS Engine		2.5.5	✓
Auto Exposure for URP	My Custom Tag	2.0.0	✓
Auto Fence & Wall Builder		3.4	✓
Auto Hand - VR Interaction		4.0.0	✓
AutoLOD - Impostors		4.1.1	✓
Automotive HMI Template		1.1.1	✓

Overview

Total Packages	4.603	Show Detai
Indexed Files	1.542.170	

Package Details

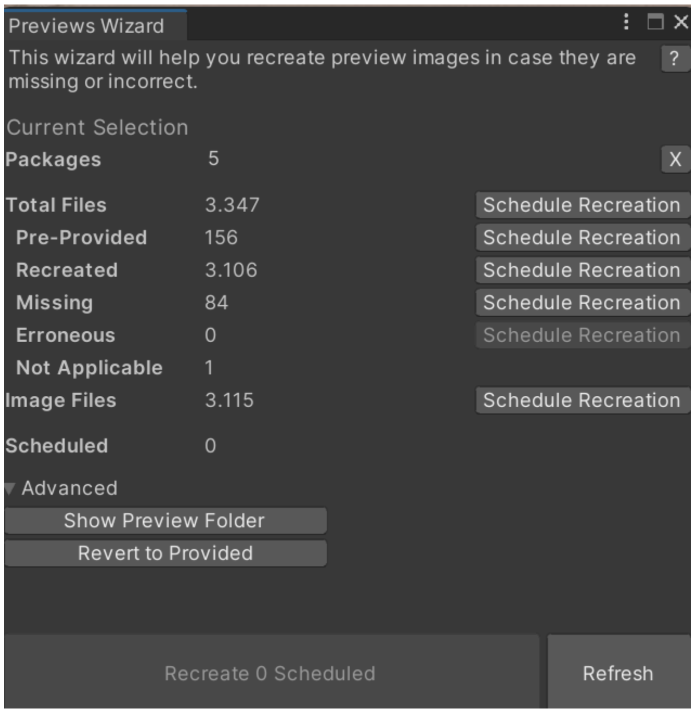
Name	Asset Inventory 2
Publisher	Impossible Robert
Category	Editor Extensions/Utilities
Size	7.6 MB
Unity	2021.3.25
Released	Fr, Mär 22 2024
Last Update	Mo, Jul 8 2024 (2.4.0)
Price	€35,88
Rating	★★★★★ (101 ratings)
Indexed Files	179
Source	Asset Store

ASSET PREVIEWS

The search result will show previews of the assets. These previews can come from three sources:

- Included in the package and created at the time the asset was built (state: pre-provided)
  - This is done by the Asset Store tooling and the reason why e.g. previews for sound files can look different depending on which version of Unity was used.
  - Sometimes these previews can be empty when exported incorrectly.
- Created by Asset Inventory during import (state: recreated)
  - This happens automatically for atomic assets without dependencies, e.g. sound files, textures or models.
- Created by Asset Inventory on-demand (state: recreated)
  - This mode can also create previews for prefabs and materials but will take **much** longer since all dependencies need to be materialized first.

It is possible to recreate a single preview from inside the search view, all previews for a package or perform recreation for missing previews in the complete database. There is a **Preview Wizard** to help with all steps. Previews can also easily be switched between provided and recreated there in case of issues, e.g. incompatible render pipelines.


Previews Wizard

This wizard will help you recreate preview images in case they are missing or incorrect.

Current Selection

Packages	5	X
Total Files	3.347	Schedule Recreation
Pre-Provided	156	Schedule Recreation
Recreated	3.106	Schedule Recreation
Missing	84	Schedule Recreation
Erroneous	0	Schedule Recreation
Not Applicable	1	
Image Files	3.115	Schedule Recreation
Scheduled	0	

Technical Details

Previews are mostly created by the Unity Editor. This works well but is limited to 128 x 128 pixels. An upscale option is available in the *Settings* which produces bigger previews for image files. For all other types this will reduce visual quality (tick the *Lossless* upscale option to only upscale images). VFX and 2D prefabs cannot be previewed this way. A custom mechanism is under way for future releases.

While creating previews it might happen that a new folder called _*TerrainAutoUpgrade* will appear under Assets as a side-effect.

AUTOMATION & API

The tool can be controlled via scripting. This happens through the static methods of the *AssetInventory* class and through a dedicated search API. The existence of the tool can be detected via the automatically added script define *ASSET_INVENTORY*.

To use the powerful search features (through the *ResultPickerUI*), the following code can be used (it is also provided in the *Examples* folder). Optionally a search type and a search string can be supplied to narrow down the search.

Two modes are available: close upon selecting a result tile or explicitly using the *Select* button. Once the user selects something, a callback with the path of the asset inside the project will be invoked. The asset will automatically be materialized.

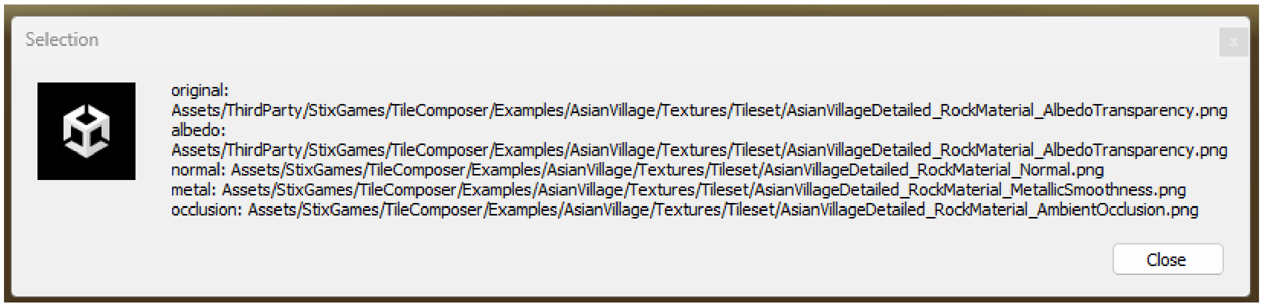
The *ResultPickerUI* offers multiple methods:

Show()

will return the path of the selected asset inside the current project.

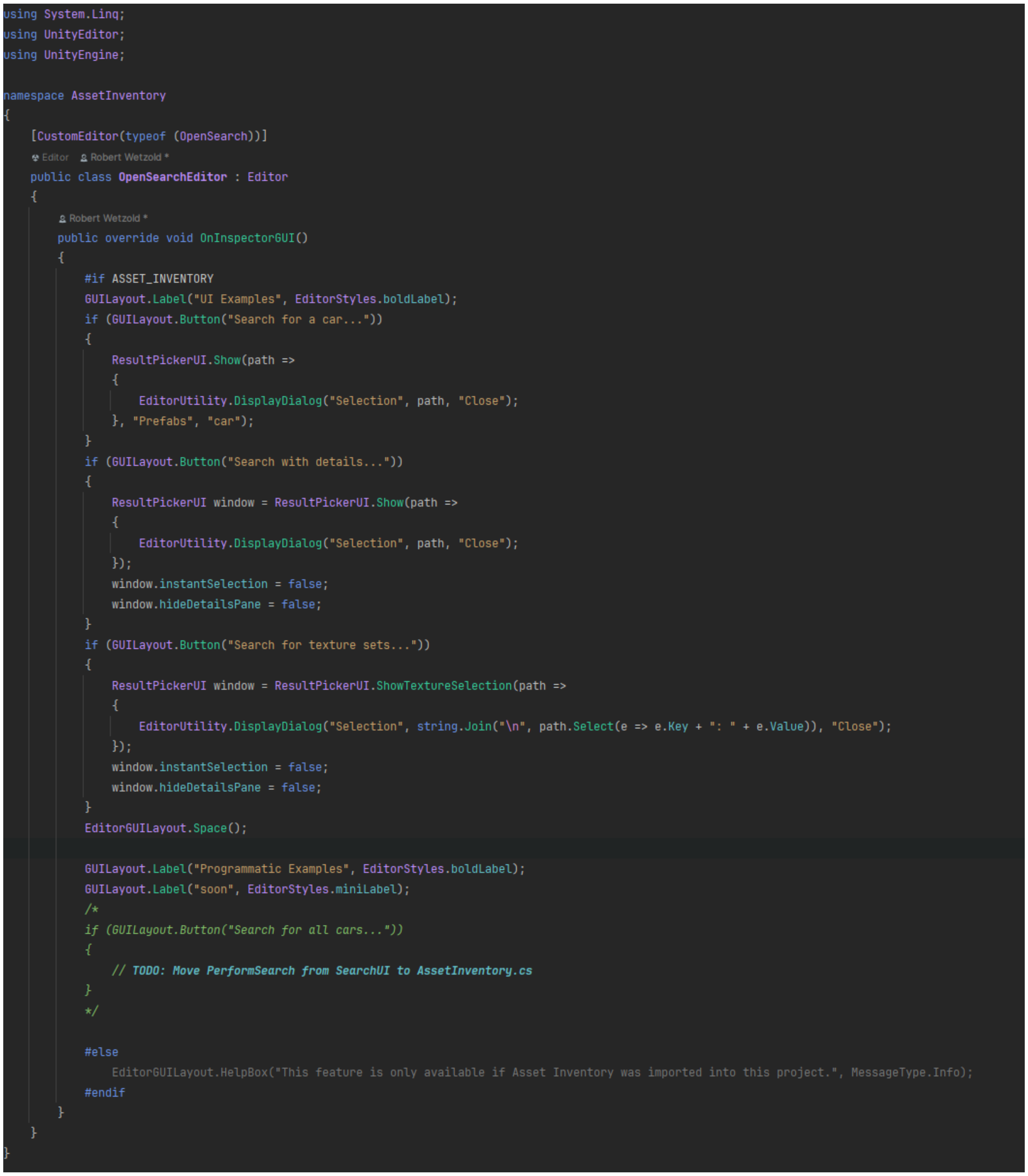
ShowTextureSelection()

Will default to "Images" as search type and return a dictionary of files. The selected entry will be inside "original". The tool will do a heuristic analysis of additional adjacent files that belong to the same texture, like normal maps, metal maps etc. If found, they will also be materialized and returned.



```
original:
Assets/ThirdParty/StixGames/TileComposer/Examples/AsianVillage/Textures/Tileset/AsianVillageDetailed_RockMaterial_AlbedoTransparency.png
albedo:
Assets/ThirdParty/StixGames/TileComposer/Examples/AsianVillage/Textures/Tileset/AsianVillageDetailed_RockMaterial_AlbedoTransparency.png
normal: Assets/StixGames/TileComposer/Examples/AsianVillage/Textures/Tileset/AsianVillageDetailed_RockMaterial_Normal.png
metal: Assets/StixGames/TileComposer/Examples/AsianVillage/Textures/Tileset/AsianVillageDetailed_RockMaterial_MetallicSmoothness.png
occlusion: Assets/StixGames/TileComposer/Examples/AsianVillage/Textures/Tileset/AsianVillageDetailed_RockMaterial_AmbientOcclusion.png
```

Supported Types: albedo, normal, specular, metal, occlusion, displacement, height, emission, reflection, alpha, mask, roughness



```
using System.Linq;
using UnityEditor;
using UnityEngine;

namespace AssetInventory
{
    [CustomEditor(typeof (OpenSearch))]
    Editor   Robert Wetzold *
    public class OpenSearchEditor : Editor
    {
        Robert Wetzold *
        public override void OnInspectorGUI()
        {
            #if ASSET_INVENTORY
            GUILayout.Label("UI Examples", EditorStyles.boldLabel);
            if (GUILayout.Button("Search for a car..."))
            {
                ResultPickerUI.Show(path =>
                {
                    EditorUtility.DisplayDialog("Selection", path, "Close");
                }, "Prefabs", "car");
            }
            if (GUILayout.Button("Search with details..."))
            {
                ResultPickerUI window = ResultPickerUI.Show(path =>
                {
                    EditorUtility.DisplayDialog("Selection", path, "Close");
                });
                window.instantSelection = false;
                window.hideDetailsPane = false;
            }
            if (GUILayout.Button("Search for texture sets..."))
            {
                ResultPickerUI window = ResultPickerUI.ShowTextureSelection(path =>
                {
                    EditorUtility.DisplayDialog("Selection", string.Join("\n", path.Select(e => e.Key + ": " + e.Value)), "Close");
                });
                window.instantSelection = false;
                window.hideDetailsPane = false;
            }
            EditorGUILayout.Space();

            GUILayout.Label("Programmatic Examples", EditorStyles.boldLabel);
            GUILayout.Label("soon", EditorStyles.miniLabel);
            /*
            if (GUILayout.Button("Search for all cars..."))
            {
                // TODO: Move PerformSearch from SearchUI to AssetInventory.cs
            }
            */

            #else
                EditorGUILayout.HelpBox("This feature is only available if Asset Inventory was imported into this project.", MessageType.Info);
            #endif
        }
    }
}
```

DATABASE ACCESS

It is also possible to **connect to the database** directly and perform your own analysis and queries with the collected data. It is in **SQLite** format. A good viewer is for example DB Browser for SQLite.

CONFIGURATION

OVERVIEW

Settings will be saved automatically in a file called *AssetInventoryConfig.json*. This happens by default in the *Documents* folder of the currently logged in user. This way the tool can work independently of the Unity project and the asset search is available in an identical way everywhere. Two alternatives are available:

- It is possible to force the tool to use a project-specific configuration. To do so, copy the *AssetInventoryConfig.json* file anywhere into the project and restart Unity. The detected location will be shown on the **Settings tab** under **Maintenance Functions**.
- Using the environment variable "*ASSETINVENTORY_CONFIG_PATH*" a custom location (just the folder name) can be specified.

Some advanced settings can only be set in the *.json* file and are not otherwise accessible through the UI. An example for this is the delay after which the search will start searching. Feel free to adjust these values as well.

INSTALLING AS A PACKAGE

Especially when managing many Unity projects keeping Asset Inventory up-to-date can mean spending quite some time on manual package updates. An alternative is to install the tool in one Unity project only and then referencing this single installation via package reference in the *manifest.json*. This way all Unity installations always use the same tool version.

The easiest way to set this up is to open the manifests from the *Packages* folder where the tool should be referenced and adding it directly there. The path can either be absolute or relative. Example:

```
"com.wetzold.asset-inventory": "file:../../MySourceProject/Assets/AssetInventory",
```

ADVANCED SETTINGS

In order not to clutter the UI some seldom used settings are to be set directly in the configuration file. It is in formatted JSON format and can be opened in any text editor. The following properties do only appear there:

- **cacheFolder** *(string)*: location where the cache should be located, e.g. “F:/AICache”. Default is with the database under *“Extracted”*.
- **centerTiles** *(bool)*: will center grid views instead of left-aligning within window
- **dbJournalMode** *(string)*: defines the journal mode to be used with the SQLite database. Default is “WAL”. Use “DELETE” for a more conservative method in case of database issues. Other options see link.
- **filterOnlyIfBarVisible** *(bool)*: will only apply additional search filters if the filter bar is expanded (legacy pre-version 2 behavior)
- **hueRange** *(float)*: degrees to widen color search
- **maxResultsLimit** *(int)*: hard limit of maximum results to be shown even if *-all-* is selected
- **mediaHeight** *(int)*: size in pixels for big media image
- **mediaThumbnailWidth** *(int)*: size in pixels for width of media thumbnails
- **mediaThumbnailHeight** *(int)*: size in pixels for height of media thumbnails
- **rowHeightMultiplier** *(float)*: adjustment factor to the height of table rows
- **searchDelay** *(float)*: seconds after typing a character to wait until triggering search
- **showHints** *(bool)*: flag if to show additional usability hints in the UI

THIRD PARTY

USAGE

Asset Inventory uses multiple third-party libraries that really make it shine:

- Package2Folder by Code Stage uses an ingenious idea to intercept the Unity package manager and adjust the target folder where to install assets to.
- SQLite is a great way to store data in a single file while providing easy and fast database operations.
- SQLite-Net is an amazing extension to SQLite offering convenient high-level functions when executing SQL queries.
- Editor Audio Utils does a great job to allow playing audio files also while not in play mode.

INTEGRATION

Assets that already integrate with Asset Inventory:

Gaia from Procedural Worlds

Makes it really easy to pick new prefabs to spawn and textures to use in your procedural content. Access all your assets quickly with just a few clicks. The corresponding selectors will appear automatically once you have both tools imported in your project.

TECHNICAL DETAILS

The index will be stored in an SQLite database. This database will be automatically created upon first usage. It is in your *Documents/AssetInventory* directory. The size should be rather small but increases with the number of indexed assets. As a rule of thumb assume 20Mb database + 200Mb preview images per 30k files (if not upscaled).

Inside the *AssetInventory* directory, three additional folders will be created:

- **Previews**: containing all images for the asset catalog
- **Extracted**: containing temporarily extracted files. You can safely delete the *Extracted* folder at any time if needed.
- **Backup**: acting as the default location for package backups

Some operations require Unity to perform work, e.g. creating previews. For such tasks, the folder *_AssetInventoryPreviewsTemp* will appear under *Assets* for a short time and will be automatically removed again once done.

LIMITATIONS

- Dependencies can only be calculated in packages that use *text/yaml* serialization. A few legacy packages still use *binary* serialization. These packages also work in most cases except if Unity cannot resolve them, e.g. because of missing scripts on prefabs.
- Importing scripts is experimental and can produce console errors if the script requires other scripts or assembly definition files to run.
- Running the game view maximized on play and having the Asset Inventory window docked will cause it to reinitialize, resetting all search settings.

FAQ

SQLITE IS ALREADY IN MY PROJECT (CONFLICT)

The only requirement is that SQLite is available. In this case it is safe to delete the one supplied by Asset Inventory. Close Unity and delete either the full *ThirdParty/SQLite* folder or if your own package doesn't bring *System.Data.SQLite* only delete *ThirdParty/SQLite/Plugins/x64 & x86*.

SOME PREFABS SHOW NO DEPENDENCIES

This is due to the serialization format. Only assets serialized as text can properly be parsed right now.

SOME AUDIO FILES USE DIFFERENT COLORS IN THE PREVIEW THAN OTHERS

Preview colors depend on which Unity version an asset author used to upload his asset. Unity seems to change colors over time between Unity versions. There could be a feature someday to recreate preview images with the current version of Unity to make them all uniform depending on interest.

SOME PREVIEWS ARE GREY AND DON'T SHOW ANYTHING

This can happen if the Asset Store tooling did not create a correct preview image while uploading an asset. An example is the Danger Zone asset from Unity released for 2021+. Functionality to recreate preview images is under development.

CONSOLE SHOWS ERRORS DURING INDEXING

There are a couple of errors Unity creates that cannot be intercepted but can safely be ignored. These are:

Could not determine image dimensions for 'filename': Not enough memory to complete operation [GDI+ status: OutOfMemory]

- Can happen when either the image dimensions are too big (5000px+), it is actually a different format (incorrectly named file) or the image file has a sub-format that Unity cannot read, e.g. special types of *png*.

“Cannot create FMOD::Sound instance for clip “” (FMOD error: Unsupported file or audio format.)”

- It typically means that an audio clip was saved with the wrong extension, e.g., a wave as an ogg by an asset publisher. Use the **Open** button to start the native file player which will typically play the file correctly then.

“TEXTURE HAS OUT OF RANGE WIDTH / HEIGHT”

- In that case Unity cannot read the texture dimensions and you will not be able to see the dimensions of the texture or filter for it. Otherwise, it does not have any effect.

“ArgumentException: Getting control 4's position in a group with only 4 controls when doing repaint”

- This can happen if a lot of UI changes are going on while indexing is happening. Can safely be ignored and is without any effect.

“Curl error 28: Operation timed out after 30000 milliseconds with 0 bytes received”

- This can happen if Unity servers did not respond in time and the details for an asset could not be retrieved. This is solved by starting the Asset Store update process again.

“Invalid or expired API Token when contacting…”

- Happens if Unity was not in use for a longer period. The user login happens in the Unity hub and in that case the token for online access has expired. Solved by restarting Unity.

“Could not create asset from Assets/_AssetInventoryPreviewsTemp/1016771.png: File could not be read”

- Happens if Unity cannot create a preview for specific assets or if this is a malformed png. Can be ignored.

“TLS Allocator ALLOC_TEMP_THREAD, underlying allocator ALLOC_TEMP_THREAD has unfreed allocations, size 1261”

- Can occur after triggering a download. No negative side-effect seen so far. Can most likely be ignored and seems to be a Unity issue.

“You are trying to replace or create a Prefab from the instance '...' that references a missing script. This is not allowed. Please change the script or remove it from the GameObject.”

- Can occur during creating preview images when a prefab has script dependencies. Can be ignored since Asset Inventory fixes this situation automatically but the error cannot be suppressed.

“NormalMap settings dialog comes up”

- Can occur during creating preview images when a texture is misconfigured in a prefab. Can be ignored since this is typically invisible in previews.

SUPPORT

Web:	https://www.wetzold.com/tools
Discord:	https://discord.gg/uzeHzEMM4B
Roadmap:	https://trello.com/b/SOSDzX3s/asset-inventory