Hey there!

Huge thanks for buying **Maintainer**, your attentive Unity3D projects housekeeping assistant!

Currently it contains two modules – **Issues Finder** and **Project Cleaner**.
I'm going to add new modules and improve existing ones further with next updates, stay tuned.

Please note:
Maintainer is a Unity3D Editor Extension plugin, thus you need to obtain one license per seat. Thanks!

Installation and setup

IMPORTANT:

- Always close Maintainer window before updating.
- Always completely remove previous version before updating.

  It will let you avoid different compatibility issues and errors.

As you import plugin, you'll see new menu commands:

- **Tools > Code Stage > Maintainer > Show** (CTRL / CMD + SHIFT + ALT + `)
  Opens Maintainer window at last opened tab.
- **Tools > Code Stage > Maintainer > About**
  Opens Maintainer window at About tab.
- **Tools > Code Stage > Maintainer > Find Issues** (CTRL / CMD + SHIFT + ALT + F)
  Starts issues search and shows Maintainer window with results after search.
- **Tools > Code Stage > Maintainer > Find Garbage** (CTRL / CMD + SHIFT + ALT + G)
  Starts Project Cleaner's garbage search and shows Maintainer window with results after search.

Maintainer stores its settings at the **YourProject/ProjectSettings/MaintainerSettings.asset** file in XML format.
All search results are stored separately from all other settings, at the Temp folder of your project. Unity deletes this folder on exit, so search results will reset on Unity restart.

Plugin features in-depth

Modules

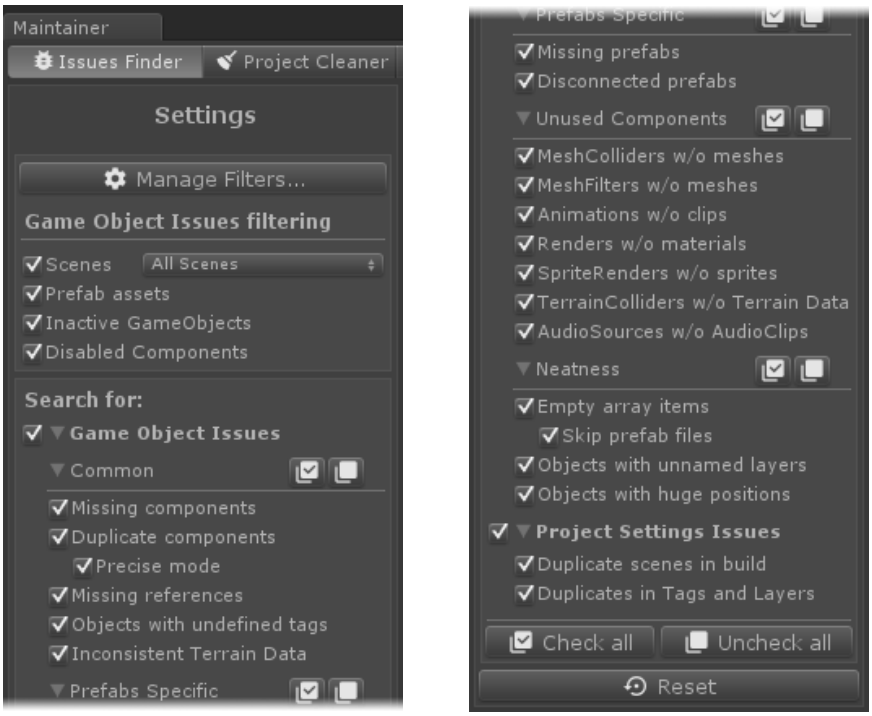
Maintainer consists of few modules: **Issues Finder**, **Project Cleaner BETA**. Below you'll find more details on each of them.

Issues Finder

This module allows to find different issues within your Unity3D project, like missing scripts, unused components, and much, much more (see below). It also able to fix some of these issues in one-by-one or batch mode to let you quickly get rid of them.

You can use this module both from Maintainer window and via menu command / shortcut (see **Installation and setup**).
If you'll use it via menu command / shortcut, it will perform search with latest (or default) settings and will show results in the Maintainer window after that.

If you wish to tune settings before search, open Maintainer window first, go to the **Issues Finder** tab and tune them from there:



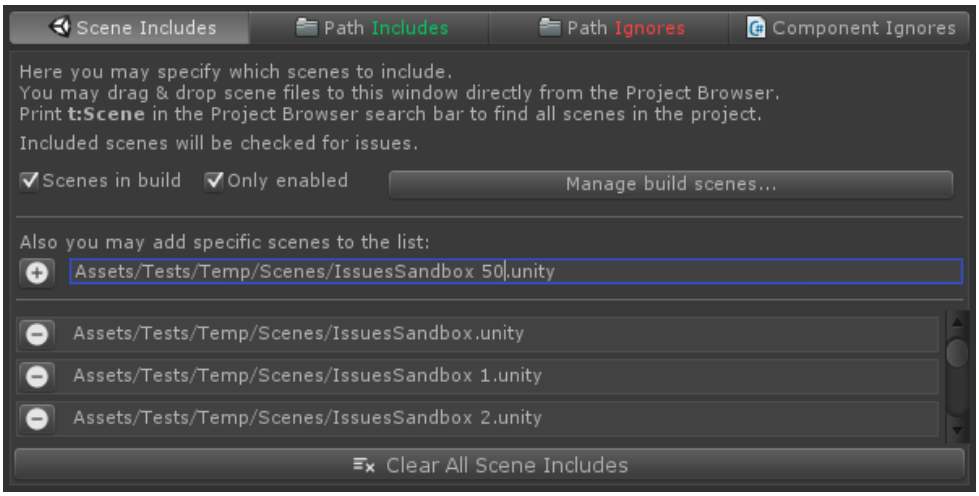
Here you can see different settings available for the Issues Finder module.

Settings are made of two separate groups: different filtering and ignoring settings and "what to look for" settings. Let me just describe them one-by-one, in their exact order.

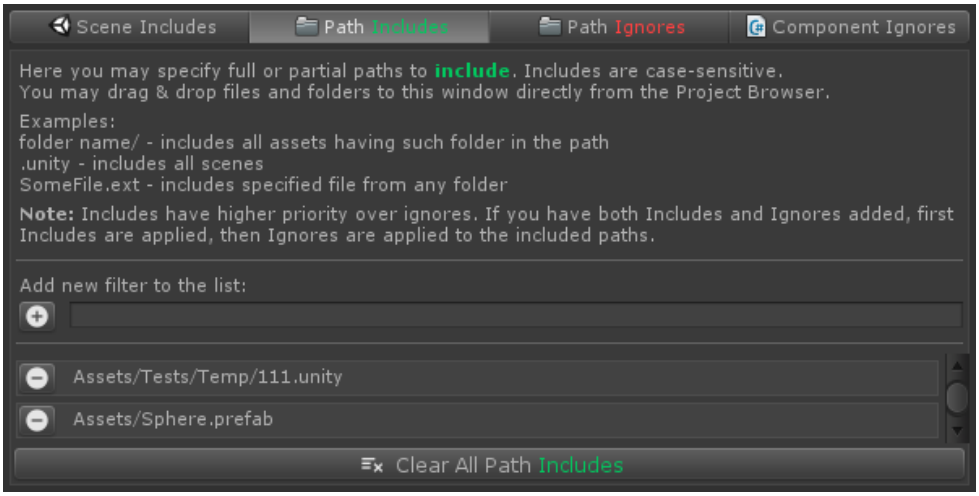
- **Manage Filters…** button. Opens Filters window with 4 tabs – **Scene Includes, Path Includes, Path Ignores** and **Component Ignores**.

  IMPORTANT:
  Please keep an eye on your lists of ignores to make sure they do not have old and not actual items to avoid extra waste of the resources and time during the Issues Search.

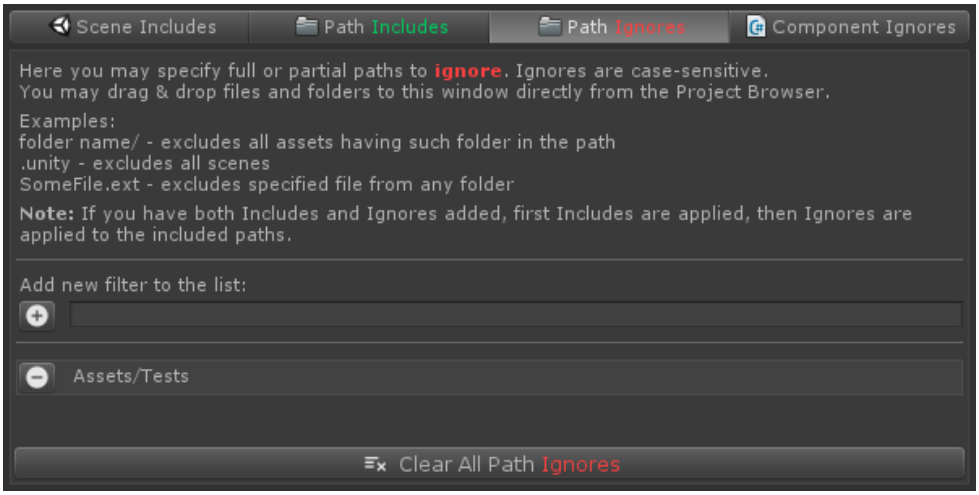
  - **Scene Includes** tab allows to include specific scenes into the issues search. Here you may easily include all scenes added to the Build Settings and specify if you also wish to include disabled items or not. Also you may add specific scenes by dragging them from the Project Browser or by entering their path relative to the project folder.



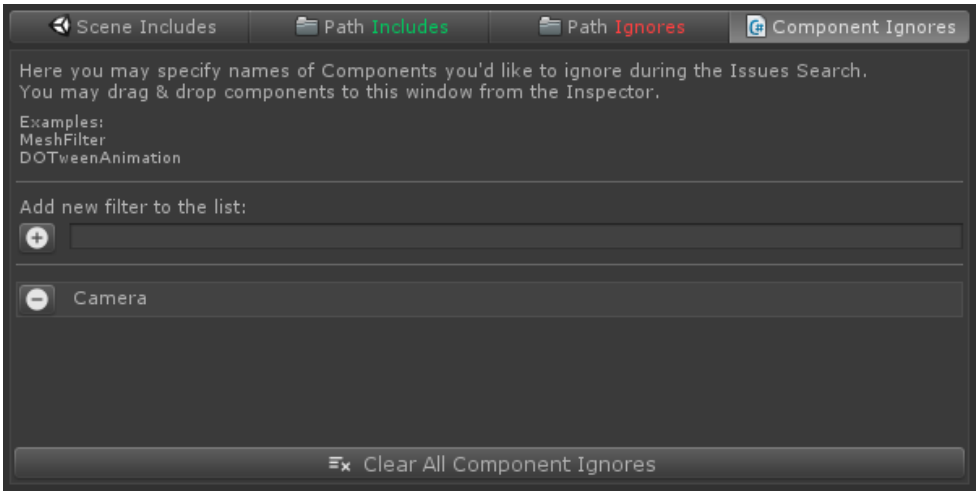
- **Path Includes** tab allows to include specified files, folders or partial paths to the Issues Search.
  You may add paths manually or just drag & drop items from the Project Browser.
  You may include files with specified extension, full or partial paths and names.
  If you have both Includes and Ignores added, first Includes are applied, then Ignores are applied to the included paths.



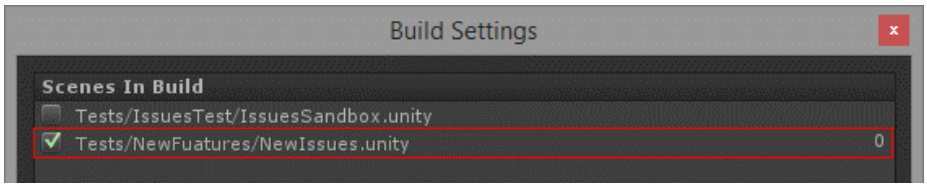
- **Path Ignores** tab allows to exclude specified files, folders or partial paths from the Issues Search.
  You may add paths manually or just drag & drop items from the Project Browser.
  You may exclude files with specified extension, full or partial paths and names.
  If you have both Includes and Ignores added, first Includes are applied, then Ignores are applied to the included paths.



- **Component Ignores** tab allows to exclude components with specified names from the Issues Search.
  You may add names by hand or just drag & drop desired component from inspector to this window.
  In case you'll add component by hands, entered name will be looked in all currently loaded assemblies to make sure you entered it correctly.



- **Scenes**. Should we check scenes at all?
- **Scenes filtering selector:**
    - **All Scenes**. Performs search in all scenes in the project.
    - **Included Scenes.** Checks only those scenes you've included at the Scene Includes filters. By default, all enabled scenes from the "Scenes in build" are included.



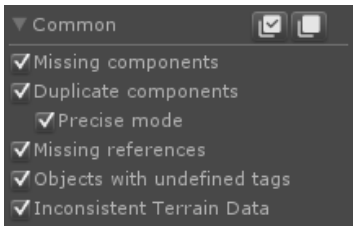
    - **Current Scene Only**. Performs search only in currently opened scene. Multi scene supported, so any additional loaded scenes will be scanned as well. Save scene dialog will not bother you if you have unsaved changes in the scene(s). Path Ignores will not exclude this scene.
- **Prefab assets.** Should we include prefab assets files?

  IMPORTANT:
  In-scene prefabs instances objects nested on 0 or 1 level which have no changed (overriding prefab) properties will be skipped when this option is checked since all such objects will be scanned as prefab assets.
- **Inactive GameObjects.** Should we include any inactive Game Objects?
- **Disabled Components.** Should we include any disabled (unchecked) Components?

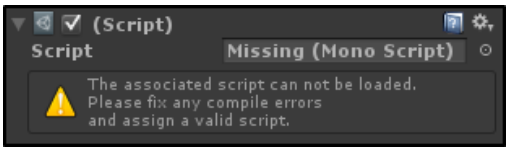
Next we see **"Search for"** settings divided into 2 main sections: **Game Object Issues** and **Project Settings Issues**.

Game Object Issues

- **Common** group. Here we have most serious issues, which can affect your game behavior and stability.

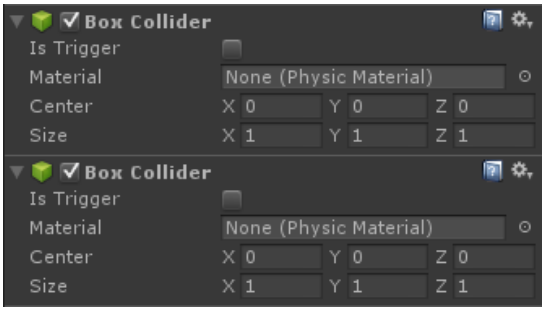


    - **Missing components.** Search for the missing components on the Game Objects. Often happens when you delete some MonoBehaviour script placed on the Game Object from your project.

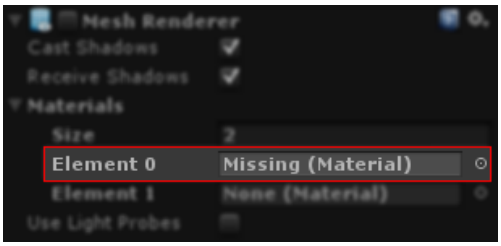


    - **Duplicate components.** Search for the multiple instances of the same component on the same object, optionally taking into account values of the components (***Precise mode***).

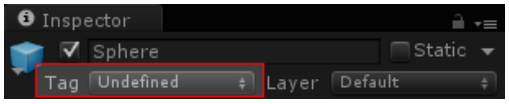
      IMPORTANT:
      Gradient values are not supported for now (so you may get false positives if you have similar components with only differed gradients).



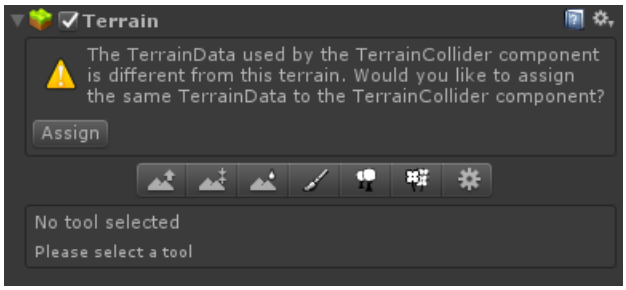
- **Missing references.** Search for any missing references in the serialized fields of the components. Often happens when you remove referenced item from the project.



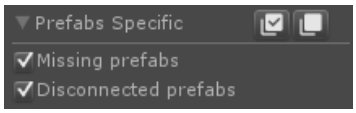
- **Objects with undefined tags.** Search for GameObjects without any tag. May happen if you use some custom tag and remove it from Tag Manager afterwards. It may be dangerous since you'll get exception if you'll try to access tag of such object.



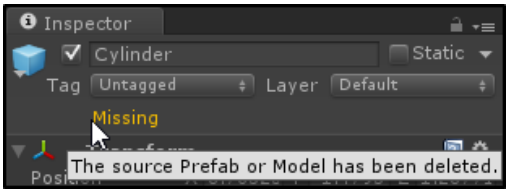
- **Inconsistent Terrain Data.** Search for Game Objects where Terrain and TerrainCollider have different Terrain Data. It may lead to the incorrect collisions with your terrain. Though it may be expected in some cases, e.g. when you wish to make separate smoother collision data to reduce resources usage.



- **Prefabs Specific** group. Contains different issues related to the prefabs.



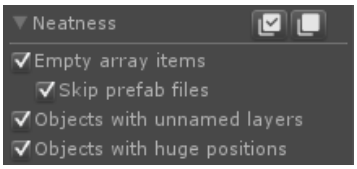
- **Missing prefabs.** Search for instances of prefabs which were removed from project.



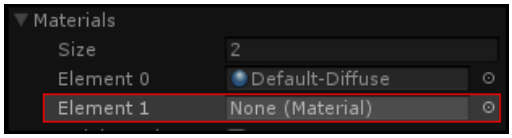
- **Instances of disconnected prefabs.** Search for disconnected prefabs instances. Happens when you delete some nested object from the prefab instance for example.

- **Unused Components** group. Contains different self-explanatory issues about not used by purpose components.

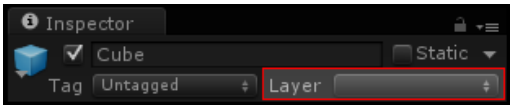
- **Neatness** group. Contains different issues related to the overall orderliness and neatness of the project.



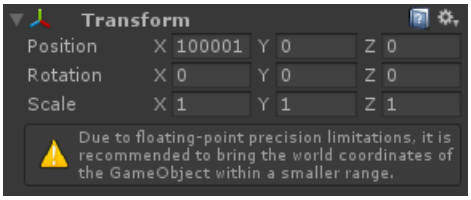
  - **Empty array items.** Look for any unused items in arrays. All prefab files can be ignored using **Skip prefab files** toggle.



  - **Objects with unnamed layers**. Search for GameObjects with unnamed layers. Happens when you use some custom layer and remove it from Tag Manager afterwards.

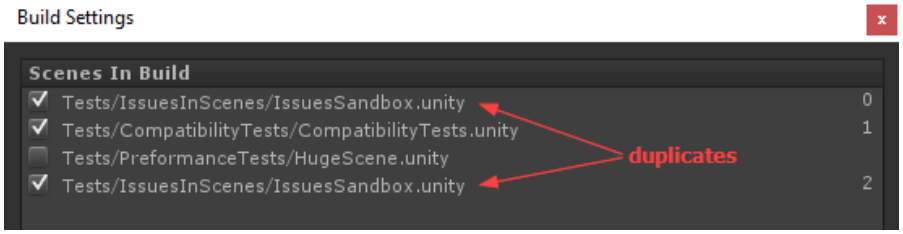


  - **Objects with huge positions.** Search for GameObjects with too big world positions (more than 100k by any axis). Will check both regular objects and UI objects as well.



Project Settings Issues

- **Duplicate scenes in build**. Looks for the duplicate entries at the *Scenes In Build* section of the Build Settings.

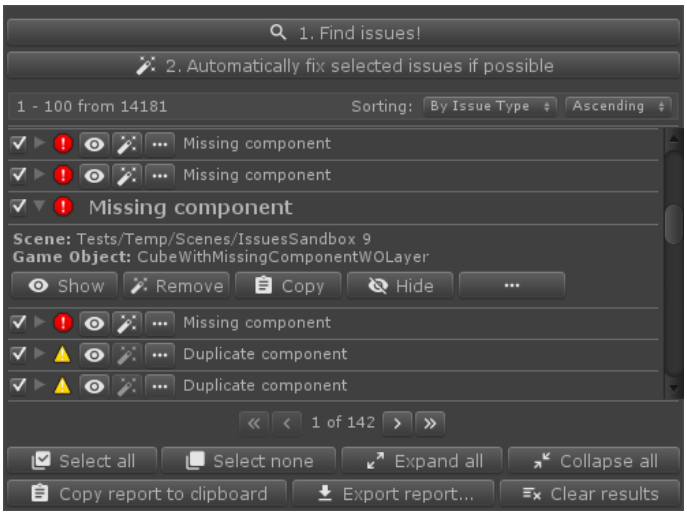


- **Duplicates in Tags and Layers**. Looks for the duplicate entries at the *Tags and Layers* settings of the project.

Use **Check all** and **Uncheck all** buttons to quickly check and uncheck all "**Search for"** options.
Use **Reset** button to quickly reset all **Issues Finder** settings to defaults.

After you've done with settings, just press "**1. Find Issues!**" button or **CTRL / CMD + SHIFT + ALT + F** shortcut to start issues search. If you have any unsaved changes in current scene, you'll be prompted to save them before search starts in case you didn't choose Current Scene Only filtering option.

You'll see search progress indicating current search phase and giving you some insight on what's currently happening. After search you'll see search results in the Maintainer.



Here you can see (from top to the bottom):

- **1. Find issues!** button. Starts the Issues Finder search.
- **2. Automatically fix…** button. Tries to automatically fix selected issues. For more details, see the **Automatic fixes** section below.
- Current page issues indexes and total issues count.
- Sorting options. You may sort by type, severity and path both ascending and descending.
- Found issues list itself (will be covered in more details below).
- Paging controls if there are more than 100 issues found (one page shows 100 issues).
- Some self-describing helper buttons.

Results list consists of separate issues records. Records are collapsed to the compact view by default and look like this:

Compact record view has such elements:

- Selection box. Allows to select items for the auto-fix.
- Expand arrow. Allows to expand a record to the detailed view.
- Severity icon. Shows severity (importance) level. Info, Warning or Critical.
- Optional Prefab icon for issues found in prefab file assets.
- **Show** button (eye icon). Navigates you to the issue as close as possible. Works differently for issues from different sections.

  For Game Object Issues:

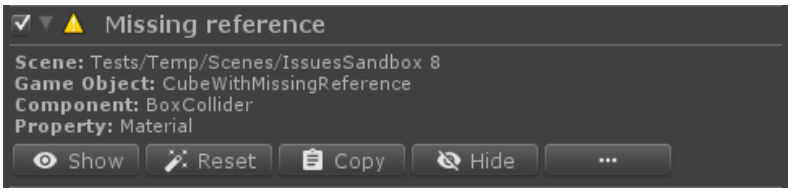
  Selects Game Object with issue in the scene Hierarchy or Project Browser. Opens scene with needed Game Object if necessary and highlights this scene in the Project Browser.
  If target Game Object is nested within Prefab file asset deeper than 2nd level, prefab asset will be highlighted in Project Browser.
  It also tries to collapse all components on the Game Object keeping expanded target component if record has it.

  For Project Settings Issues:

  Opens or highlights desired section of the Project Settings.
- **Fix** button (magic wand icon). Applies suitable auto-fix for the issue, if possible. Over cursor to see the tooltip with description of the applied fix.
- **More** button (dots icon). Shows menu with additional functionality, as quick add to the ignores.
- Issue caption. Usually briefly describes issue. May have some useful info like index of the unnamed layer to let you easily locate and name that layer in Tags Manager.

If you wish to see more details about desired issue, just press at the expand arrow, or click the record itself (except other buttons of course) to expand it to the detailed view.

Detailed view looks like this:



Detailed record view has such elements:

- Selection box. Allows to select items for the auto-fix.
- Collapse arrow. Allows to collapse a record to the compact view.
- Severity icon. Shows severity (importance) level. Info, Warning or Critical.
- Optional Prefab icon for issues found in prefab file assets.
- Issue caption. Usually briefly describes issue. May have some useful info like index of the unnamed layer to let you easily locate and name that layer in Tags Manager.
- Additional info and issue location. Depending on issue kind, can include such levels of precision:

  For Game Object Issues:

  - **Scene** (for objects in scene) / **Prefab** (for objects in prefab file assets) – path to the scene or prefab file.
  - **Game Object** - path to the Game Object in the scene.
  - **Component** (optional) - name of the Component which has an issue.

    IMPORTANT:
    If Game Object has more than 1 of such Components, you'll see the (ID: ***) added to the Component's name. This is either *Local Identifier in File* or *Instance ID* of the target Component. You may use it to easily find desired component using Debug mode of the Inspector.

  - **Property** (optional) - name of the property which has an issue.

  For Project Settings Issues:

  - Where issue was found
  - What exactly wrong there
  - May vary for different kinds of issue

- **Show** button. See the compact view description for details.
- **Fix** button with proper caption, depending on the issue kind. See the compact view description for details.
- **Copy** button. Copies record text to the clipboard.
- **Hide** button. Hides record from the results list. Useful when you fixed an issue and wish to hide appropriate record from the search results.
- **More** button. See the compact view description for details.

If you wish to see less details about expanded issue, just press at the collapse arrow, or click the record itself (except other buttons of course) to collapse it to the compact view.

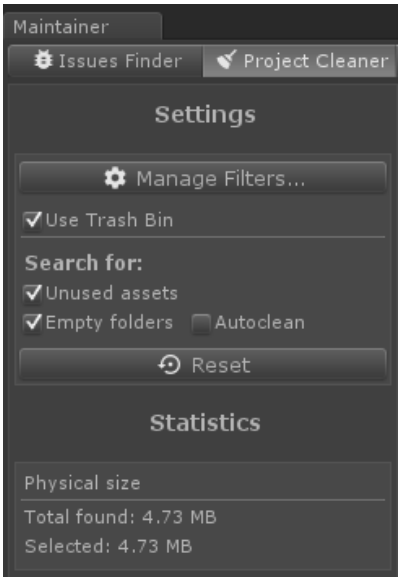
Project Cleaner (BETA)

This module allows to find different garbage, unused stuff in your project and lets you clean it up with few clicks.

IMPORTANT:
This module is at the BETA phase, it has some bugs and false positives. Please always make a backup of your project before using Project Cleaner! I'm not responsible for any data loss due to use of the Project Cleaner!

You can use this module both from Maintainer window and via menu command / shortcut (see **Installation and setup**). If you'll use it via menu command / shortcut, it will perform garbage search with latest (or default) settings and will show results in the Maintainer window after that.

If you wish to tune settings before search, open Maintainer window first, go to the **Project Cleaner** tab and tune them from there:



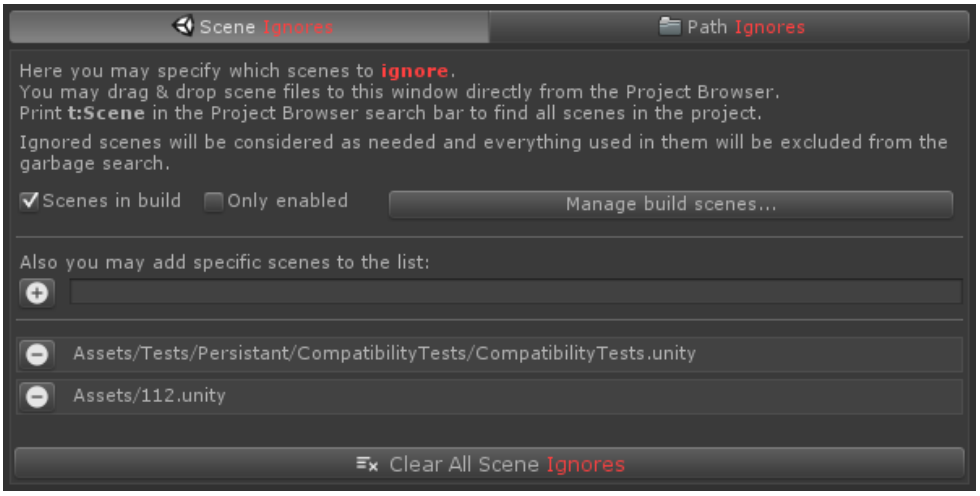
Here you can see settings available for this module.

Settings are made of two separate groups: different filtering and ignoring settings and "Search for" settings.
Let me just describe them one-by-one, in their exact order.

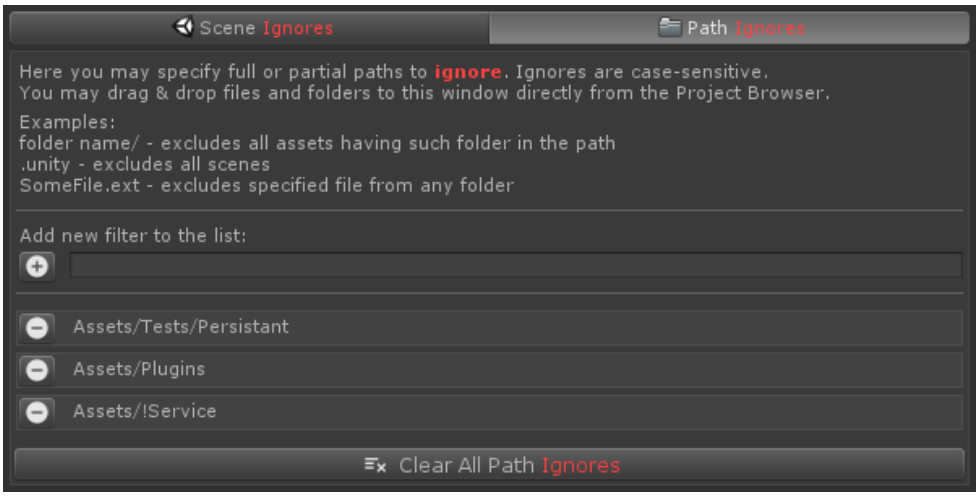
- **Manage Filters…** button. Opens Filters window with 2 tabs – **Scene Ignores and Path Ignores**.

  IMPORTANT:
  Please keep an eye on your lists of ignores to make sure they do not have old and not actual items to avoid extra waste of the resources and time during the garbage search.

  - **Scene Ignores** tab allows to exclude specific scenes from the garbage search. Ignored scenes will be considered needed and everything used in them will be excluded from the garbage search.
    Here you may easily include all scenes added to the Build Settings and specify if you also wish to include disabled items or not. Also you may add specific scenes by dragging them from the Project Browser or by entering their path relative to the project folder.



- **Path Ignores** tab allows to exclude specified files, folders or partial paths from the Garbage Search.
  You may add paths manually or just drag & drop items from the Project Browser.
  You may exclude files with specified extension, full or partial paths and names.



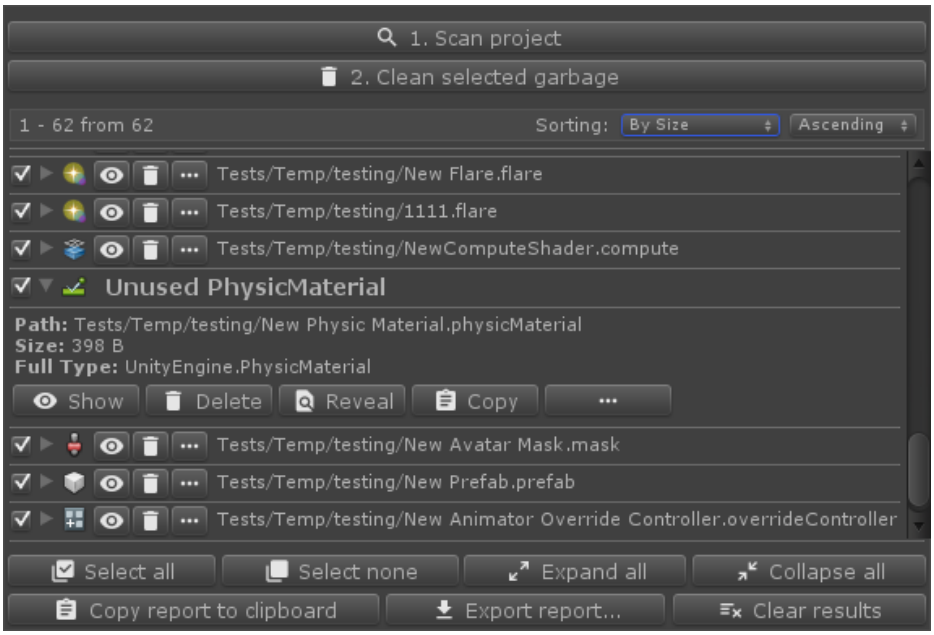
- **Use Trash Bin**. Keep this on checked if you wish to move all garbage to the Trash Bin instead of deleting it permanently. Always recommended setting, it will allow you to recover stuff you deleted by mistake. It may not work in some environments, for example, early Unity builds for Linux ignore this option.
- **Unused Assets**. Check to let Project Cleaner look for unused assets in your project.

  IMPORTANT:
  Assets garbage search doesn't look for the unused scripts, since it's nearly impossible to reliably say if some script is used or not because it might be accessed by name or using Reflection, etc.
  Also, to avoid false positives, Project Cleaner skips some special folders like Plugins and Editor.

- **Empty folders.** Finds empty directories in your project. You also may enable **Autoclean** option to automatically find and delete empty folders on scripts recompile. Useful when you work in big team using GIT or similar VCS which don't sync empty folders. It allows to automatically clean empty folders for all your colleagues as soon as they get the Maintainer with configured settings from the repository.

After you've done with settings, just press the "**1. Scan project**" button or **CTRL / CMD + SHIFT + ALT + G** short key to perform garbage search.

You'll see search progress indicating current search phase and giving you some insight on what's currently happening. After search you'll see search results in the Maintainer.



Here you can see (from top to the bottom):

- **"1. Scan project"** button. Starts the garbage search.
- **"2. Clean selected garbage"** button. Starts the cleanup of the selected items.
- Currently shown items indexes and total items count.

- Sorting options. You may sort by path, size, and type both ascending and descending.
- Found garbage list itself (will be covered in more details below).
- Paging controls if there are more than 100 items found (one page shows 100 items).
- Some self-describing helper buttons.

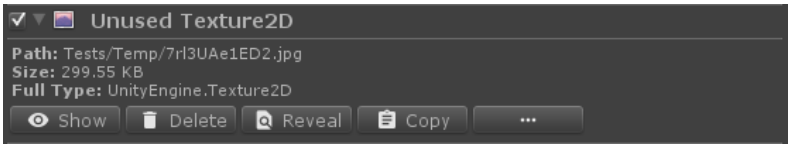
Results list consists of separate garbage records. Records are collapsed to the compact view by default and look like this:

Compact record view has such elements:

- Selection box. Allows to select items for the cleanup. Only checked items will be deleted on the cleanup phase.
- Expand arrow. Allows to expand a record to the detailed view.
- Asset icon. Shows icon similar to what you see at the Project Browser for each asset.
- **Show** button (eye icon). Selects asset at the Project Browser.
- **Delete** button (trash bin icon). Immediately deletes asset.
- **More** button (dots icon). Shows menu with additional functionality, as quick add to the ignores.
- Garbage asset path relative to the Assets folder.

If you wish to see more details about desired garbage, just press at the expand arrow, or click the record itself (except other buttons of course) to expand it to the detailed view.

Detailed view looks like this:



Detailed record view has such elements:

- Selection box. Allows to select items for the cleanup. Only checked items will be deleted on the cleanup phase.
- Collapse arrow. Allows to collapse a record to the compact view.
- Asset icon. Shows icon similar to what you see at the Project Browser for each asset.
- Caption. Usually describes the type of found garbage.
- Garbage asset path relative to the Assets folder.
- Asset size (if it's not an empty folder).
- Full type (if it's not an empty folder).
- **Show** button. See the compact view description for details.
- **Delete** button. See the compact view description for details.
- **Reveal** button. Shows file or folder in the system File Manager (Explorer on Windows, Finder on Mac, etc.).
- **Copy** button. Copies garbage record description to the clipboard.
- **More** button. See the compact view description for details.

If you wish to see less details about expanded garbage item, just press at the collapse arrow, or click the record itself (except other buttons of course) to collapse it to the compact view.

After you observed found garbage, just check desired items and clean them up using the "**2. Clean selected garbage**" button.

Using Maintainer from code

Maintainer has public API for all modules. It allows you to call modules from your code which may be useful when you wish to integrate Maintainer's modules into your build pipeline to make automated issues fixes \ reports or automated garbage cleanup \ reports for example.

See online API docs for details:
http://codestage.ru/unity/maintainer/api

Troubleshooting

- **I have errors in console after importing plugin.**
  - Consider making clean update: delete whole folder with plugin before importing new version into your project.
  - Make sure you have no namespaceless classes in your project which may interfere with some system classes.
  - If errors still there, please contact me, and I'll try to help (see Support contacts below).
- **I can't see issue at the specified object.**
  - Try to enable Debug Inspector mode, some issues may be not visible in Normal Inspector mode.
- **I can't find target component with issue at the specified object because it has lots of such components.**
  - Check if issue record has (ID: ***) next to the Component name. In such case switch inspector to the Debug mode and find desired component looking at the **Local Identifier In File** or **Instance ID** fields.
- **I can't find target property with issue at the specified component both in Normal and Debug inspector modes.**
  - In some rare cases, actual property names may be obscured with PropertyDrawers or Editors. It's not an actual bug. However, if you face such cases too often - just let me know and we'll think how to deal with it.

Compatibility

Plugin works with both Unity Personal and Professional editions.
Plugin works on all platforms (Windows, OS X, Linux).

Plugin takes into account some specific behavior of the third-party plugins and assets to avoid false positives. For example, empty MeshFilters on objects with TextMeshPro or 2DToolkit components will be considered valid.
If you use some third-party assets that make false positives for you because of specific behavior, feel free to report them to me.

Final words

I hope you will find **Maintainer** useful and helpful in your daily work with Unity and it will save you some of your priceless time!

Please, leave your reviews at the Asset Store, it's really important and any user feedback motivates me to continue my work with extra passion. So if you'll have a minute, be a kind person and leave your opinion at the Asset Store! ^^

Also, feel free to drop me bug reports, feature suggestions and other thoughts on the forum or via support contacts!

Maintainer links:

Asset Store | Web Site | Forum | YouTube

Support contacts:
E-mail: focus@codestage.net
Other: codestage.net/contacts

Best wishes,
Dmitriy Yukhanov
Asset Store publisher
codestage.net
@dmitriy_focus

P.S. #0 I wish to thank my family for supporting me in my Unity Asset Store efforts and making me happy every day!
*P.S. #1 I wish to say huge thanks to **Daniele Giardini** (DOTween, HOTools, Goscurry and many other happiness generating things creator) for awesome logos, intensive help and priceless feedback on this plugin!*