TASK INTEGRATION TOOL

Combo: Task Integration Tool

version 0.1 package documentation

Summary

1. Getting Started

Combo is a task aggregator, those tasks run every time a file is created, updated or deleted into a specific file extension and in specific project directory.

Let's see how works for the example Combo package **EnumFromJson**, we create a config like this:

enum from json	
Name:	enum from json
Task Type:	EnumFromJson ▾
Input path: Assets/	Resources
Search Pattern: *.json	
Remove	

This task will run when:
A file with the **.json** extension in the **Assets/Resources** directory was created, updated or deleted.

1.2. Combo Purpose

Combo can be used for tasks like convert files, save file into a specific folder, upload to web, anything what you want to do with those inputs:

- A file **search pattern** (Combo uses the [System.IO.GetFiles](#) method to match this pattern, all rules used in this method are applied to Combo search pattern too), the search pattern is defined by the **IComboTask** implementation of the selected package.

- A **path** (must be inside the Assets/ folder).

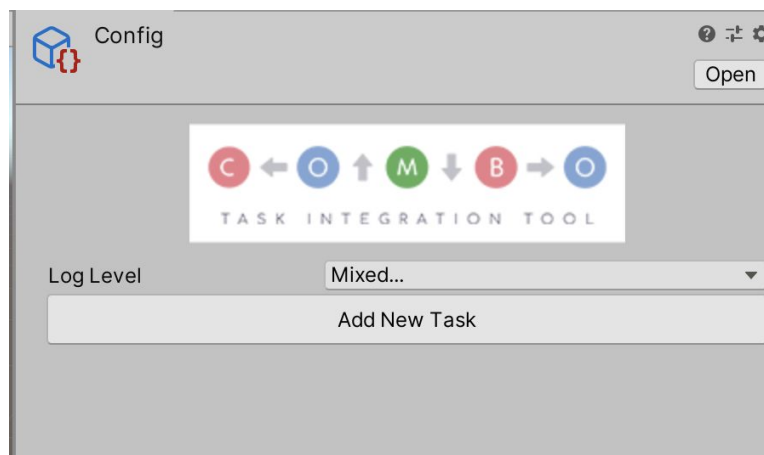- A **Combo package**, you can create your own or use one from community.

TASK INTEGRATION TOOL

2. Combo Usage

Combo on be added to the project automatically creates a singleton ***ComboConfig*** asset file in the path ***Assets/Editor/Combo/Config.asset***, this path can't be changed.
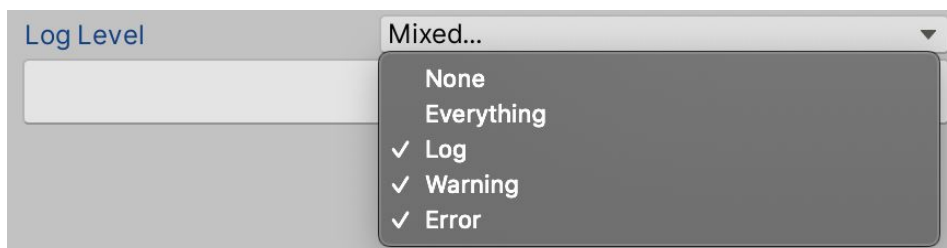
This configuration can be edited in the asset inspector or in the Combo Config window, in ***Tools > Combo Config***.

In the very start will look like the following image, with a log level option and a "Add New Task" button.

2.2. Log Level

The log level is a flag enum, you can choose if the Combo Logger log normal Log, Warnings or Errors, as a flag those options can be exclusive or mixed.
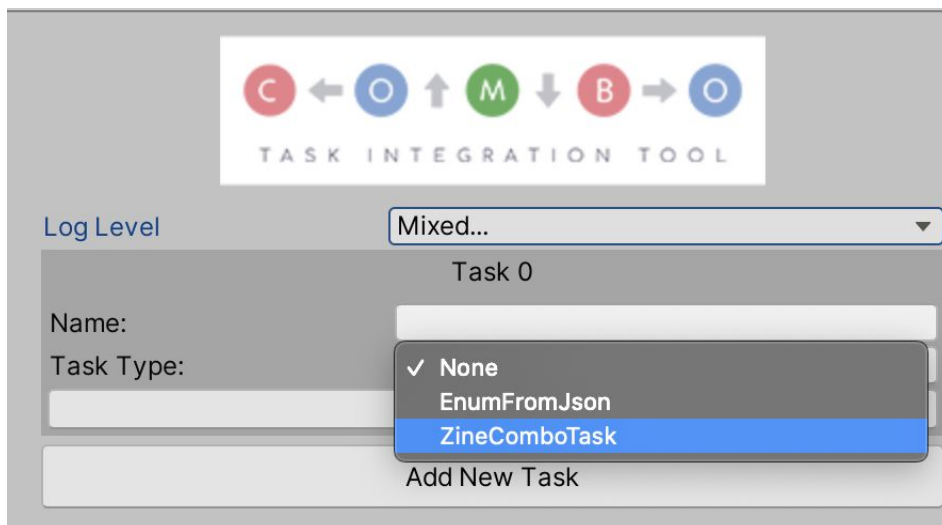
2.3. New Task

On add a new task you have to choose first a Combo package to set the task type, and optionally you can set a name to task to be easy to find later.

Any class or struct with a **ComboTaskAttribute** and with any type of **IComboTask** implementation appears in the list.

3. Create A New Combo Task Package

To create a Combo Task just need to create a class or a struct with a **ComboTaskAttribute** and with any type of **IComboTask**, like the following image:

```
[ComboTask]
public struct EnumFromJson : IComboSingleFileTask
{
```

The **IComboTask** can be implemented with two interfaces, with the **IComboMultipleFilesTask** and with the **IComboSingleFileTask.**

3.1. Interfaces

- **IComboMultipleFilesTask:** Run one time only using all the matching files input.
- **IComboSingleFileTask:** Run one time for each file of the input.

Each interface has a method for create or update event and other a method for the delete event.

Besides that in the interface has the **SearchPattern** to define the task pattern and the **Description** to use in the ComboConfig as a tooltip.

3.2. TaskFileInputData

The representation of the input file passed as parameters for the tasks, it has the following fields:

- **path:** The file relative path.
- **directoryName:** The directory name only without the file name.
- **fileName:** The file name only without the directory or the extension.
- **extension:** The extension, starting with the dot.
- **contents:** The full file contents as string.

4. References Links

Source Code: https://github.com/lavaleak/combo
Library license: https://github.com/lavaleak/combo/blob/master/LICENSE.md