2023

Item List Creator

USER GUIDE

Table des matières

Information :

What is Item List Creator?

Item List Creator is a tool made to create and manage lists of items. Every list is serialized in JSON so you can reuse it at any time. You can create a multitude of lists and manage them separately. This tool allows you to see the content of your lists and edit them quickly so you don't have to modify a real *JSON* file and worry about the structure and rules of this format. The way the *Item* class is made make it easy to save and load data for your players.

Why you made this tool?

When I was working on my own game I wanted to add an inventory system. To make it easy for the future I made it using *Serialized* data and *JSON* format for easy save/load system. But I got tired of modifying my 200 items text files… So I made this little tool to help me work on these huge lists.

What does the package come with?

The *Item List Creator package* include a demo scene, a small inventory system to show you how you can implement lists made with the tool in your game, this guide and the tool itself.

What can I do with the tool?

You can:

- Create/Load lists of items
- Create/Modify/Delete items
- Set IDs for every item automatically
- Check if every ID in your file is unique
- Sort your list by ID/Name
- Export your list as CSV file for readability

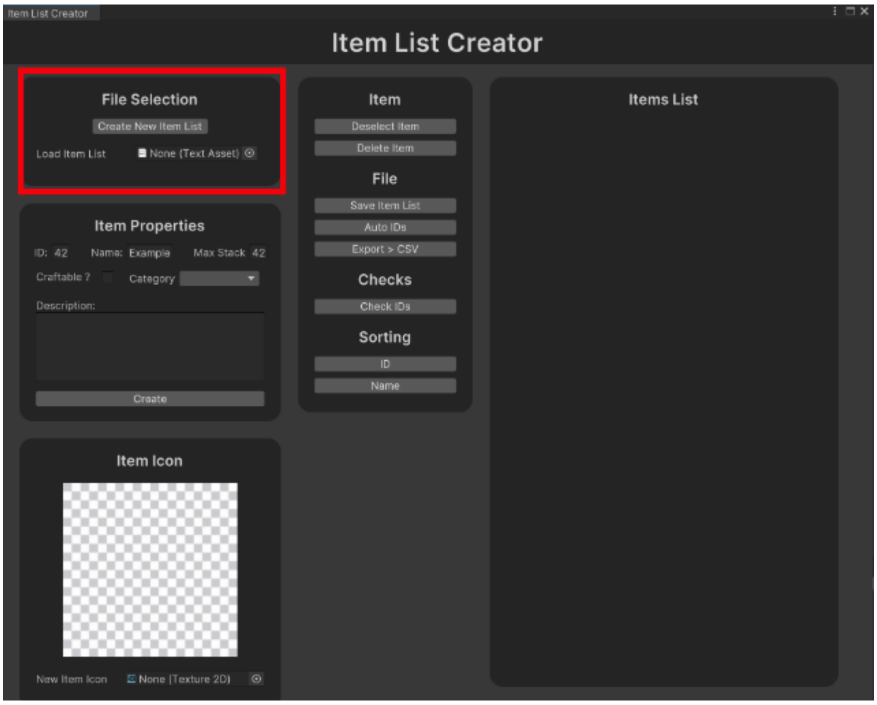
What if my item don't look like yours?

If your item need specific information (like *Rarity* or *drop rate).*You can easily modify the *Item* class and add fields to the UI. The code is heavily commented so you should be able to add your own custom properties. If you need any further help to modify the tool you can send me an email (zartox.support@gmail.com).

How to use Item List Creator

File Selection

Before doing anything you have to either create or load a file. This can be done in the *File Selection* section.



ID: 42 Name: Example Max Stack 42

Create a file:

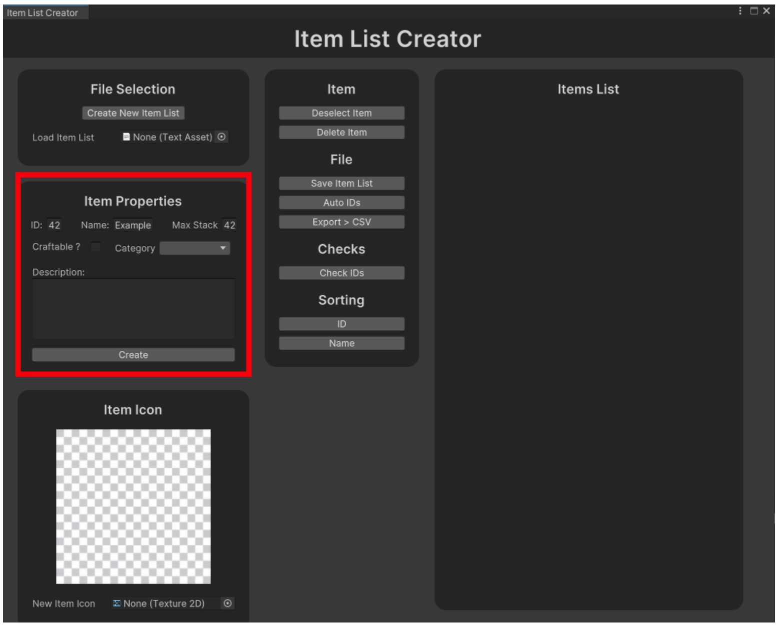
To create a file simply click on *Create New Item List,* you will be asked where you want to save it. Choose a path and a file name. The file will be automatically selected for you and will be opened. You can start working on it directly.

Load a file:

To load a file chose a file in the *file picker* under the *Create New Item List* button. If your file is a txt file it will be opened right away and you can start working. If your file is a text file but is empty the list will be empty. If it contains *JSON* formatted data, the *Items List* will be populated. If your text file user another type of format (or no format/errors) you will receive a warning with the error in the *Console.*

Item Properties

Once you have selected a file to work with you can start to manage your list. In this section we will be talking about the *Item Properties* panel. In this section you can either create or modify an Item.



ID: 42 Name: Example Max Stack 42

Create an Item:

To create an Item you first need to deselect the item you selected (if you selected any). To do so click on the *Deselect Item* button (on the top right of the *Item Properties* panel, under the *Item* section). Then you have to fill every field in the *Item Properties* panel and click create (If you did not fill every field, you will receive a warning in the *Console*. And the item won't be created. When you create an item a warning can appear in the console for 2 possible reason:

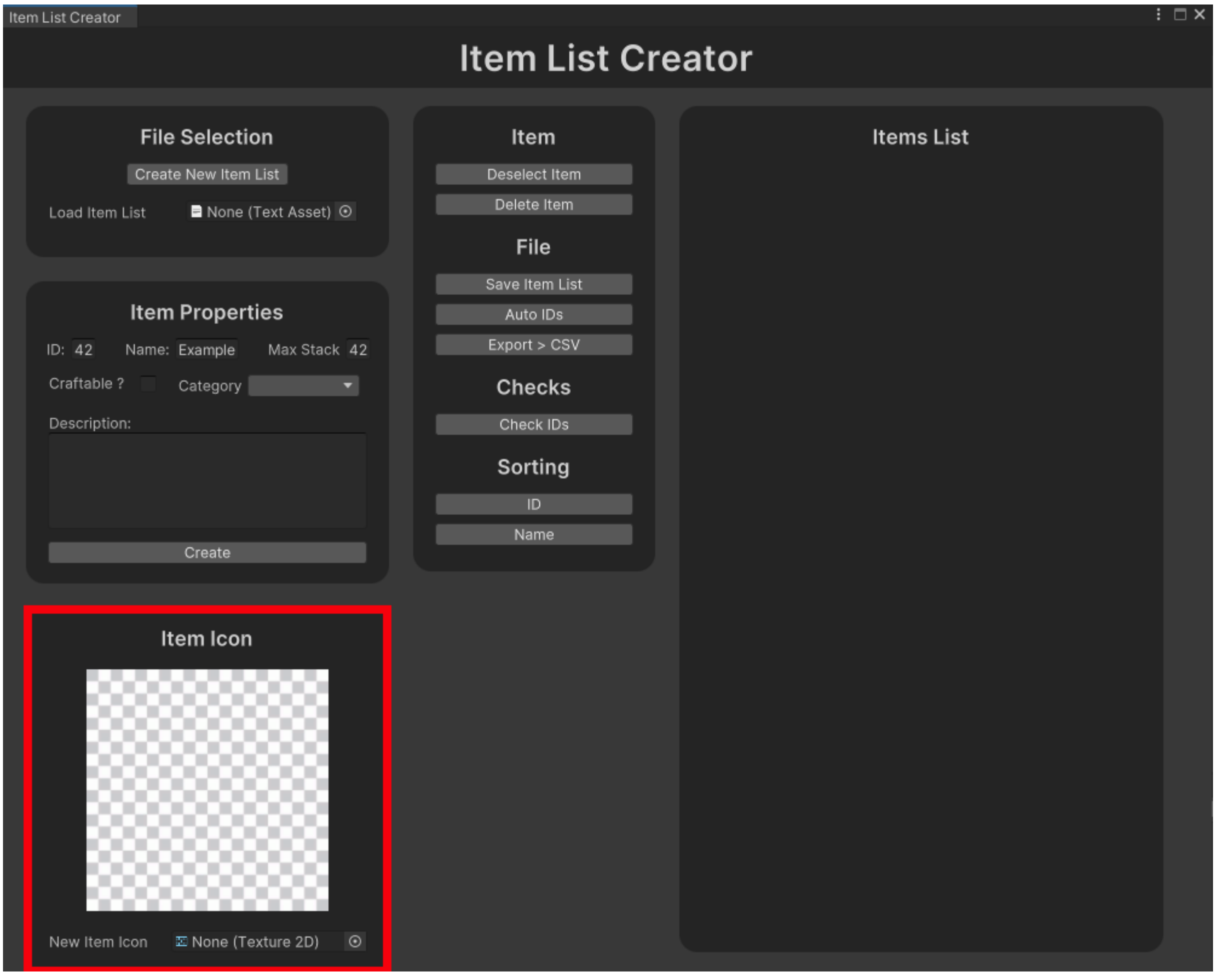
- At least one field is empty.
- The ID you choose is already in use (OK if you plan to use Auto IDs)

Modify an Item:

To modify an existing item you have to first select it in the *Items List.* Once done the *Create* button will change in a *Modify* button. The properties of your item will be shown in the *Item Properties's* fields. You can modify them by changing the values you need, and click on *Modify*.

Modifying an Item’s Icon

To modify an item’s icon you have to first select your Item in the *Items List*. Once done, the icon view is updated to show your Item’s icon. If your item don’t have an icon yet, a predefined texture will show up. To modify it simply pick a *Texture2D* (use Texture2D for your icons so they can be Serialized). In the *New item icon* selector. It will be changed and applied automatically.



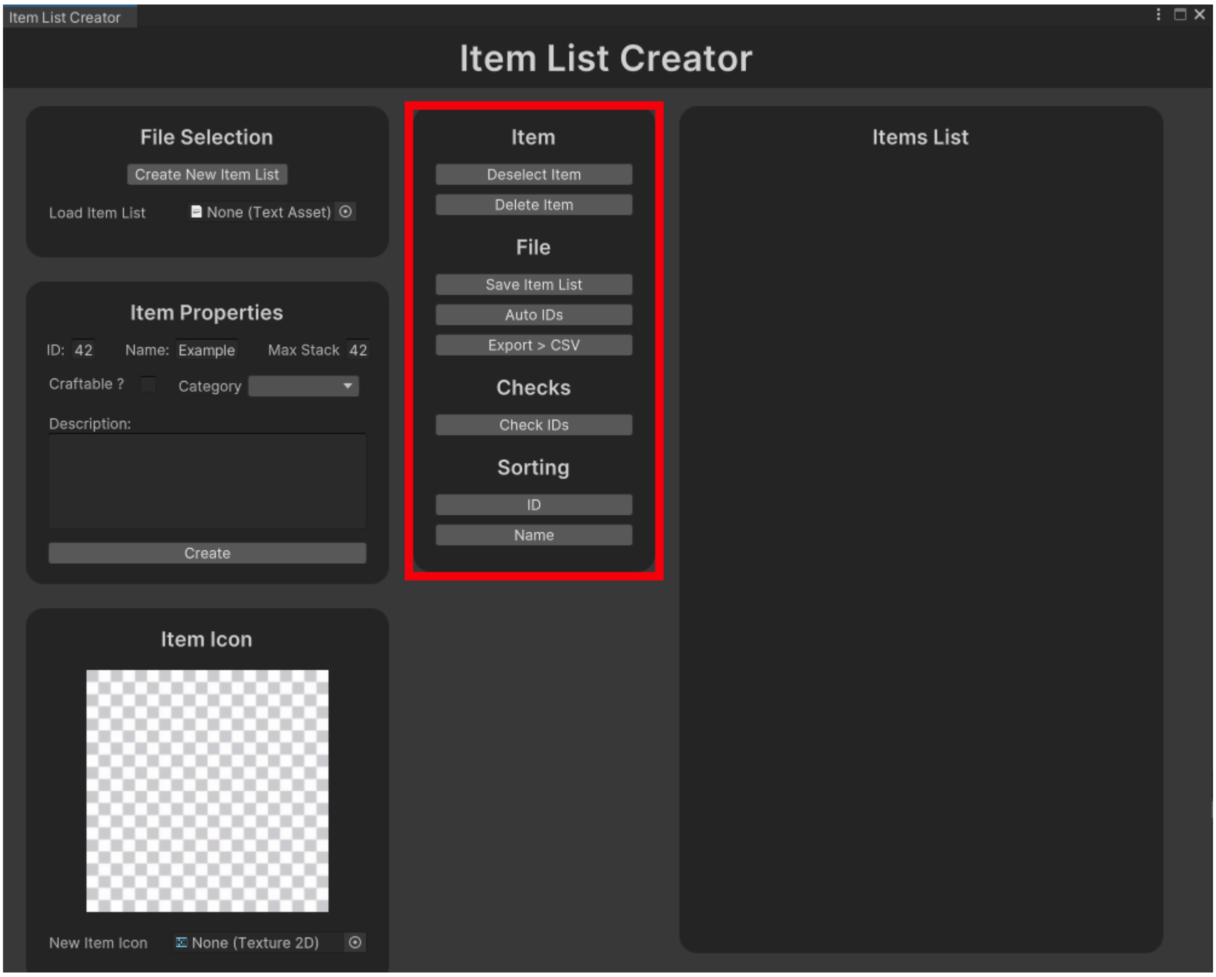
Load Item List: None (Text Asset)

ID: 42 Name: Example Max Stack 42

New Item Icon: None (Texture 2D)

Actions

In the middle of the UI there you can see a panel with different sections. This panel contains all the necessary commands to manage your list. These buttons can be used only if you have opened a file.



Load Item List: None (Text Asset)

ID: 42 Name: Example Max Stack 42

New Item Icon: None (Texture 2D)

Deselect Item:

This button can be used only if you have selected an item. When you press it, your selection will set to empty (this will make the *Create Item* button appear).

Delete Item:

This button can be used only if you have selected an item. When you click on it, the selected item will be erased of the list (and the file once you saved it).

Save Item List:

This button will save your list on the opened file's path. It will save every item in your list and convert it to *JSON*.

Auto IDs:

This functionality allow you to set every ID automatically. Every item's ID will be changed from 0 to your number of item.

Export CSV:

This functionality export your current list to a CSV file (Excel). I made this to enhance the visualization of your list. You can open it in Excel and apply rules, colors, etc…

Check IDs:

Once pressed this button will run a check on every item's ID in your list. The result of this check will appear in the *Console*.

Sort by ID:

Simply sort your list from the lowest ID to the highest.

Sort by name:

Simply sort your list alphabetically.

Demo

In the package you can find a demo scene which contain a really simple inventory for a player. The goal of this scene is to show you how you could store every items in a dictionary where the key is the ID of every item in it. And how you can retrieve the data from your list.

JsonHelper Class

The *JsonHelper* class is really important. This class is used to Serialize and Deserialize data stored in the lists you create with the tool. If you want to retrieve the Items you stored in your list simply use the *FromJson<Item>* Method, it will return an *array* with all your items. If you try to use *JsonUtility* to do it, you won't be able to retrieve your items (which are stored in a *JSON Array*).