



Ultimate PlayerPrefs

Documentation

Introduction

Welcome to Ultimate PlayerPrefs! UPP is an Unity editor extension that provides you a lot of practical functions to manage and oversee your game data stored in Unity's PlayerPrefs. Get rid of the unwieldy PlayerPrefs keys, unsecure data types and other limitations of the bare built-in system.

The PlayerPrefs are Unity's very basic solution for saving simple data of the types string, int and float in a key based data table. Since there is no built-in viewer or editor in Unity to manage these data and we couldn't find any third party extension that meets our needs we decided to develop the Ultimate PlayerPrefs editor.

A major requirement for Ultimate PlayerPrefs was, not to replace the "normal way" how the PlayerPrefs system works but to provide additional functionalities and a much easier and more secure handling.

You can add Ultimate PlayerPrefs to existing projects and decide yourself how deep you want to integrate it into your project. Vice versa you can remove it at any time without harming your data.

How it works

Ultimate PlayerPrefs provides so-called **PlayerPrefItems** (or just **items**) that wrap the original underlying PlayerPrefs entries. An item handles the PlayerPrefs entry completely. It provides inspecting and editing the value, grouping, filtering and searching and much more via the Ultimate PlayerPrefs editor window.

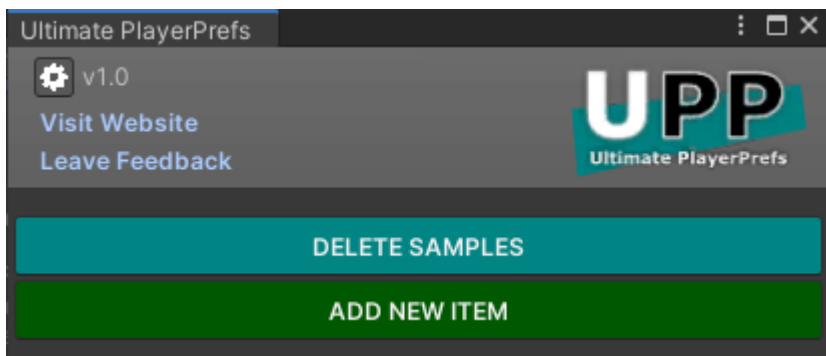
As there exist three basic data types in Unity's PlayerPrefs which are "string", "int" and "float", there are also three basic PlayerPrefItems - one for each data type. But you will find a lot of additional options to store complex data like lists, dictionaries, serializable objects and more in the API.

Get started

Your way **Ultimate PlayerPrefs** starts with importing the Unity package to your project and opening the editor window. You can find it in the unity “Window” menu as “Ultimate PlayerPrefs”.

Adding Items

You can add new UPP items to your project by clicking on “ADD NEW ITEM” at the top of the editor window.



In the “ADD NEW ITEM” dialog you need to insert the name, the data type, a default value and optionally a group the item belongs to (more to groups later).



Delete all sample data before adding your own items. You can also do this later but take care about item name conflicts.

ADD NEW ITEM

Item name:

Type:

Default Value:

Add to group

Group:



The item name has to be unique because it acts as the PlayerPrefs key.

You can also link existing PlayerPrefs keys as UPP items.

Key already exists in PlayerPrefs

Item name	PlayerName
Current Value	Player
Type	String



Take care of the data type when linking existing PlayerPrefs entries!

Click on “Add item” to create the new UPP item.

Data types

PlayerPrefsStringItem

The string item is very versatile but of course basically stores strings.

Beside get and set raw string values you will find several additional function in then **PlayerPrefsStringItem** for storing values like Lists, Dictionaries, Vectors and so on. These values are internally converted to JSON format.



you can store whole serializable objects as JSON strings inside a PlayerPrefsStringItem.

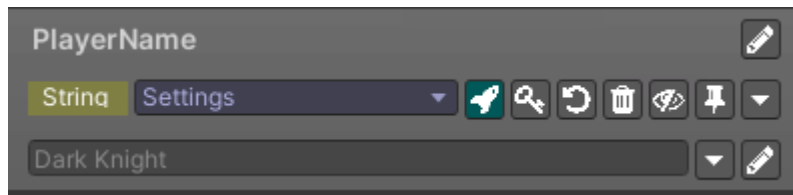
PlayerPrefsIntItem

The int item stores integer values but also provides getter and setter to handle bool values. Internally bool values are stored as 0 or 1.

PlayerPrefsFloatItem

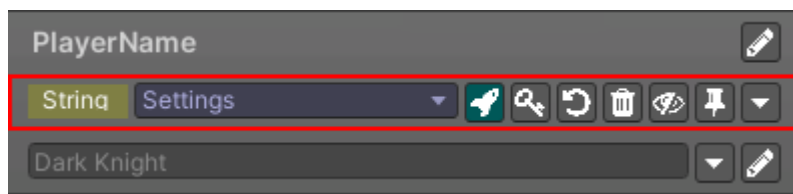
The float item, like the name says, stores floating point values.

The Item box



The item box provides you several functions and an overview over value, type and group membership of an item. Below you get overview about the functions of the item box in detail.

Toolbar



In the toolbar you see the data type, the group the item belongs to and some handy functionalities.



You can assign an item to an existing group, remove it from it's current group or create a new group.



Enable the cache to have faster access to a value. The value is only read once from the PlayerPrefs and is cached inside the PlayerPrefsItem. If the value changes it is synchronized with the PlayerPrefs automatically.



Encrypt a value to prevent players from cheating and changing value in the memory or i.g. in the windows registry. The value is encrypted via AES 256 bit.



If you want to delete an encrypted PlayerPrefsItem but keep the PlayerPrefs entry make sure you disable encryption before. Otherwise you will not be able to restore the value because the encryption key is lost..



The PlayerPrefsItem's value is set to the declared default value.

Delete item

When you delete an item you have the choice to delete the PlayerPrefs entry too or to keep the value in the PlayerPrefs and only delete the PlayerPrefs item.

Hide item

Hides the item in all views. Choose “Show hidden items” from the filter menu to see hidden items again.

Pin item

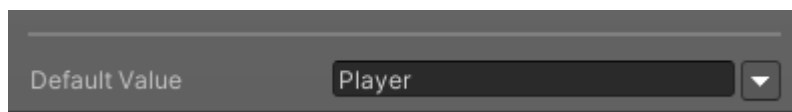
Pinning items is one of the most handy functions. Pinned items are always visible and are displayed on the top of the list.



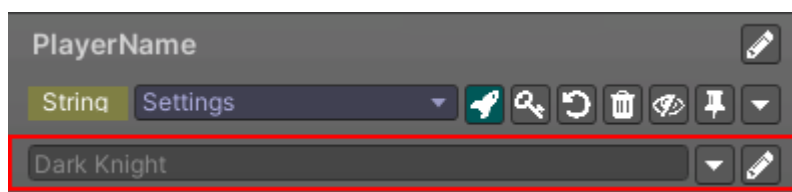
Pinned items are visible regardless of the active filters. In GROUP VIEW all pinned items are grouped inside a special group called “Pinned items”.

Expand item box

Expand the item box to edit the default value of that item. Also you can see the encrypted value of items with enabled encryption.



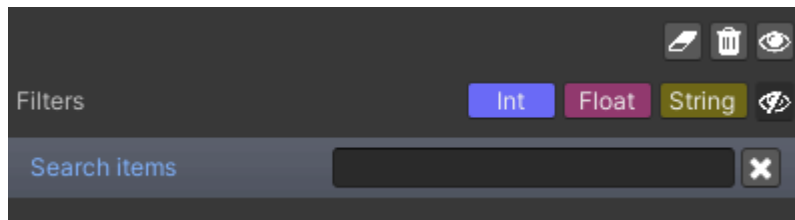
Value



You can see the current value of an item. The view is updated automatically every time you hover or focus the Ultimate PlayerPrefs window.

If you have larger texts you can expand the text field to a text area. Additionally by clicking on the pencil icon you change to the edit mode where you can modify the value even in runtime.

Search, filter and pin items



In the filters panel you will find some useful functions to efficiently find what you are looking for and improve your overview.

Common tools

These functions affect all of your PlayPref items at once. It can be very useful but even you have to confirm your action be careful because you can not undo the results.

Clear PlayerPrefs

All PlayerPref entries are removed from Unity's PlayerPrefs without destroying the PlayerPref items. That means at the end all items are set to their default values.

Delete all items

Deletes all PlayerPref items at once. You are prompted, if you also want to remove the PlayerPrefs entries too or just the items, letting the entries in Unity's PlayerPrefs untouched.

Unhide all items

All hidden items are unhidden and will show up.

Filters



You can show or hide items of specific data types and decide if hidden items are shown or not.

Search items



Maybe one of the most useful functions. If you set a search string only items whose names contain the search string are displayed.

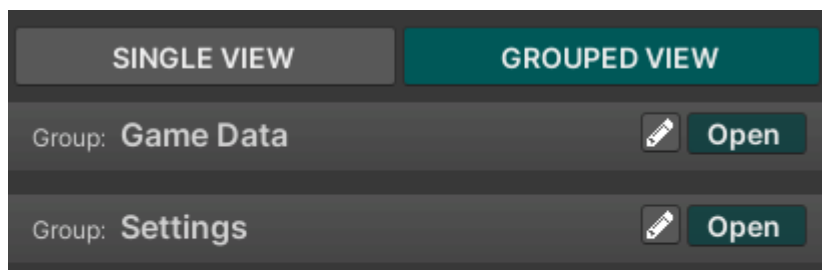
Groups

When you have a lot of data stored in the PlayerPrefs it can be difficult to keep track of all your items. Grouping your items will make it much easier.

Therefore just assign your items to an existing or to a new group via the *item box*. There is no need to declare, create and manage your groups explicitly, since this is done automatically for you.

Switch between SINGLE VIEW and GROUPED VIEW by clicking the tab.

Group boxes



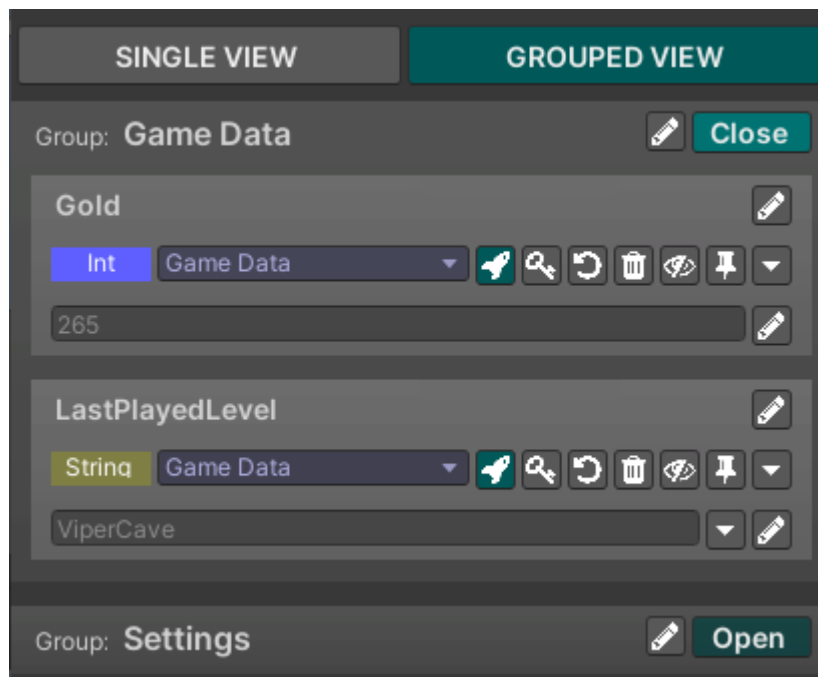
You will find the group's name and two functions in the closed *group box*.

You can expand the groups by clicking the “Open” button at the right to see all items belonging to that group. Here you have the same options to manage your items as in SINGLE VIEW.



Ungrouped items are organized in a special group called “Ungrouped items” that appears at the lower end of your group list.

Expanded group box

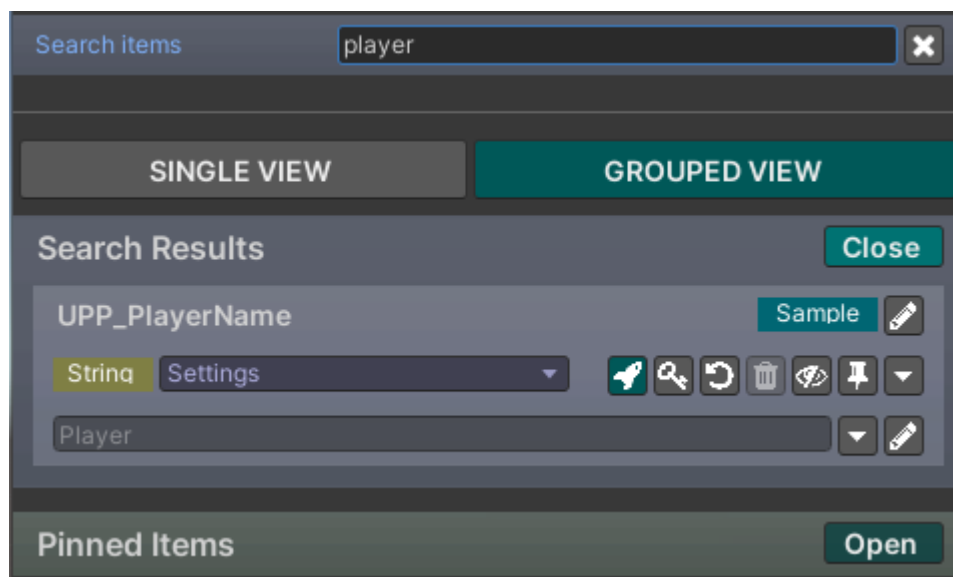


You see all items belonging to that group that pass the active filters.



You can rename groups by clicking on the pencil icon right to the group header.

Special Groups



In GROUP VIEW special groups are automatically created for pinned items as well as for search results.

UI Adapter

For saving you coding time and simplifying storing data gathered by user input or displayed values via UI, Ultimate PlayerPrefs provides so called **UI Adapters**. These powerful components can be added to several Unity UI elements to synchronize data changes automatically.

You just have to add the appropriate adapter script to the UI element's GameObject and link a PlayerPrefsItem to it. You can find a demo scene using UI adapters in the Ultimate PlayerPrefs Sample folder.

UI Adapter components exist for the following UI elements:

- Text
- TextField
- Slider
- Dropdown
- Toggle

Feel free to implement your own adapter components and let us know, if you think, it is a must have for other users too.

Code snippets

Below you will find some examples of using the basic PlayerPrefsItem API.

The first snippet shows you how to read the PlayerName from an PlayerPrefsStringItem and set it into a text field. For sure you also could use the text ui adapter for that purpose.

```
public PlayerPrefsStringItem PlayerNameItem;
public Text PlayerNameText;

void Start() {
    PlayerNameText.text = PlayerNameItem.Value;
}
```

Here we have a simple usage of a PlayerPrefs item called “LastPlayedLevel” to set and read the latest played level.

```
public PlayerPrefsStringItem LastPlayedLevelItem;

public void StartLevel(string levelId) {
    Init(levelId);
    LastPlayedLevelItem.Value = levelId;
}

public RestartLevel() {
    Init(LastPlayedLevelItem.Value);
};
```

Filling a high score list is also a simple task using a wrapper function of the PlayerPrefsStringItem..

```
public PlayerPrefsStringItem HighscoresItem;
public Dictionary<string, int> Highscores;

public void Start() {
    Highscores = HighscoresItem.StringIntDictValue;
```

```
}

public GameOver(string currentLevelId, int score) {
    if(! Highscores.ContainsKey(currentLevelId)
        || Highscores[currentLevelId] < score)
    {
        Highscores[currentLevelId] = score;
        HighscoresItem.StringIntDictValue = Highscores;
    }
};
```



Have a look into the API of the `PlayerPrefStringItem` class to get an overview about all supported data and list types.

Backstage

The PlayerPrefsItem

Everytime a new item is added via Ultimate PlayerPrefs, the engine in the background creates an asset for you, called PlayerPrefsItem. This item is stored in an Unity Resources folder.



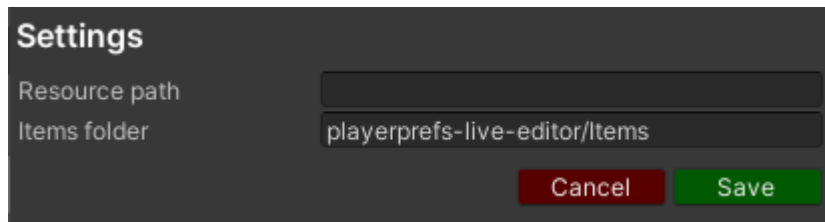
Avoid to create, modify or delete a PlayerPrefsItem directly in the resources. Always use the Ultimate PlayerPrefs window for that.

The item itself does not contain the actual value but all the metadata (like type, default value, encryption key and so on) about it. The value is still stored in Unity's PlayerPrefs register and can be used like before.



If "Caching" is active for an item (which is by default), a copy of the item's value is held inside the item object itself to get a better performance on heavy read operations.

Settings



You can open the settings panel via the small gear icon in the top left corner of the Ultimate PlayerPrefs window. Here you can define where the PlayerPrefs items are stored. Normally you don't need to change the default settings.

The *Resource path* will define which path (relative to Unity's Assets folder) the resource folder is located.

The *Items folder* is a relative path inside the Resources folder

With the default configuration you can find the PlayerPrefs items in the folder *Assets/Resources/ultimate-playerprefs/items*.



If you change the paths, you might have to move your existing PlayerPrefsItems to the new folder.