

EASY COPPA

— User guide —



Index

Easy Coppa User Guide.....	2
Important Links.....	2
Connect with Patoli Studio.....	2
Introduction.....	2
Highlights.....	3
Main Functions.....	3
Using Easy Coppa.....	3
Overview.....	3
Main Menu.....	4
Easy Coppa Settings.....	4
Settings Description.....	5
Using the EC_Canvas prefab.....	6
Scripting API.....	6
Localization.....	7
Testing.....	9
Demo.....	9
FAQ.....	9

Easy Coppa User Guide

This document is the official user guide for the Easy Coppa plugin for Unity by Patoli Studio.

Important Links

- [Support Mail](#)

Connect with Patoli Studio

- [Website](#)
- [Facebook](#)
- [Twitter](#)
- [YouTube](#)

Introduction

Mobile gaming is a big portion of the Videogame industry, and to be able to take part into this branch of game development we need to be ready to comply with the laws and regulation to safely offer our products. One of this regulations is COPPA, the Children’s Online Privacy Protection Act enforced in the USA by the FTC (Federal Trade Commission). This regulation imposes certain requirements on service providers and their products directed to children under 13 years of age. It basically says that service providers should not track user behavior or advertise directly to children in products that might

not be suited for children under 13 years of age. (For more information regarding COPPA you can refer to [this link](#).)

In order to make it easier to developers to comply with COPPA in a quick and easy way, we have decided to create this Unity plugin to save you time to implement an age check so you can focus in what you really want, making great games!

Easy Coppa has been designed to be used in mobile platforms but it should be able to run in most platforms.

Highlights

- Easy to setup and use
- Easy UI customization
- Easy Localization
- Handles Terms of Use and Privacy Policy web visualization
- Underage users rejection.
 - *If your game is not suited for minors but might attract their attention you can reject their use of your game.*

Main Functions

- COPPA Dialog
- Default Terms of Use and Privacy Policy Dialog
- Underage users rejection

Using Easy Coppa

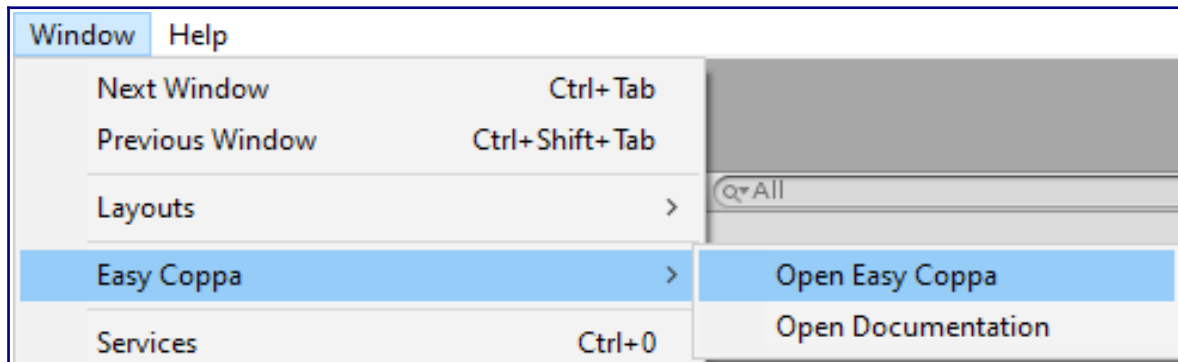
Overview

Using Easy Coppa involves 4 steps:

1. Importing the unity package
2. Setting up the plugin
3. Placing the prefab into the scene
4. Calling the plugin from script

Main Menu

After importing Easy Coppa, there will be a new menu added at *Window > Easy Coppa* from which you can open the configuration settings for the plugin.

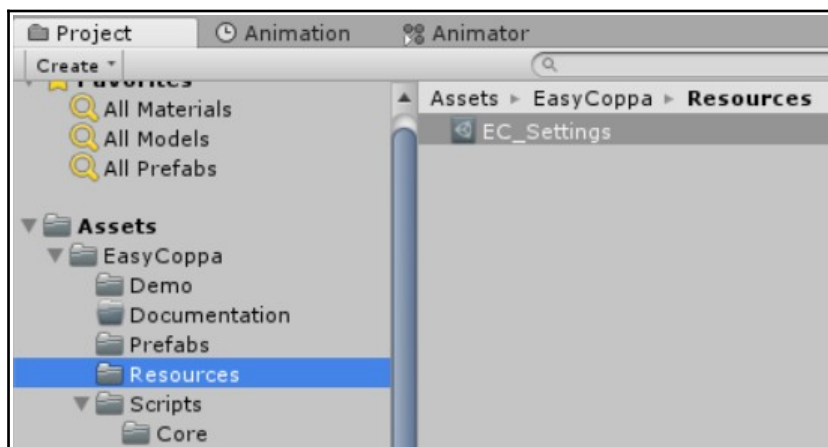


Easy Coppa Settings

This is the global configuration for Easy Coppa. Here you can control the plugin's behavior and set your company's information.



All these settings are stored in the `EC_Settings` object, which is a `ScriptableObject` located at `Assets/EasyCoppa/Resources`. You can access this `EC_Settings` from script.



Settings Description

- **Show Coppa**

If this checkbox is selected, the plugin will show the COPPA dialog when calling *EC_Dialog.ShowDialog()*; Otherwise, the default terms of service dialog will be shown.

- **Reject Minors**

If this checkbox is selected, a dialog stating that there was an initialization issue will be shown if the player enters a birth date under 13 years of age; When the user presses the only button the game will close. If this checkbox is not selected, the dialog will return a *EC_Dialog.EC_DialogResult* stating the player age status (Underage, Not Underage).

- **Block Input**

If this checkbox is selected, the plugin will activate an invisible image that has the Raycast Target option enabled, therefore blocking the user input outside of the dialog.

- **Use Translation**

If this checkbox is selected, all labels in the dialog will be composed from custom assets. More information on how to implement localization can be found in section ***Localization***

- **Translation**

You can load an *EC_Lang* asset to load all the needed translations for the labels in the dialog. More information on how to implement localization can be found in section ***Localization***

- **Date Format**

This setting controls the order of the dropdowns for days, months and years.

- **Terms of Use URL**

This is the URL of your Terms of Use.

- **Privacy Policy URL**

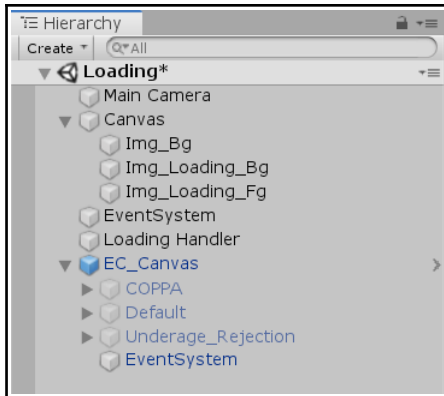
This is the URL of your privacy policy

- **Support Mail**

This is your company's support mail. This is only used for the rejection dialog so the players' parents can contact you for more information on why the game is closing.

Using the EC_Canvas prefab

In order to show the dialog we need to drop the EC_Canvas prefab into the desired scene.



As you can see, the EC_Canvas prefab contains an EventSystem. Unity does not support multiple EventSystems in the scene. Worry not. The EC_Canvas has a method to check if there are multiple EventSystems and if such is the case, it will automatically delete its own EventSystem on runtime.

Once the EC_Canvas is ready on the scene we are ready to call the plugin from a script.

Scripting API

In order to call the plugin from a script you need to add this line at the top of the script.

```
using EasyCoppa;
```

To show the dialog you just need to add this line to your code

```
EC_Dialog.ShowDialog();
```

Remember, the dialog shown is controlled by the EC_Settings's Should Show Coppa property. If it is selected, it will show the Age Check dialog, if not, it will show the default terms of service dialog.

To get the result of the COPPA dialog you need to subscribe to the event by adding this lines to your code

```
private void OnEnable(){
    EC_Dialog.onCoppaDialogCompleted += onCoppaDialogCompleted;
}

private void OnDisable(){
    EC_Dialog.onCoppaDialogCompleted -= onCoppaDialogCompleted;
}

private void onCoppaDialogCompleted(EC_Dialog.EC_DialogResult result){
    switch(result){
        case EC_Dialog.EC_DialogResult.EC_UNDERAGE:
            break;
        case EC_Dialog.EC_DialogResult.EC_NOT_UNDERAGE:
            break;
    }
}
```

```

        case EC_Dialog.EC_DialogResult.EC_REJECTED:
            break;
    }
}

```

Easy Coppa also includes the following events for your convenience:

- `EC_Dialog.onDefaultDialogCompleted`
- `EC_Dialog.onPlayerOpenedPrivacyPolicyURL`
- `EC_Dialog.onPlayerOpenedTermsURL`

And this is how you should subscribe/unsubscribe from them.

```

private void OnEnable(){
    EC_Dialog.onCoppaDialogCompleted += onCoppaDialogCompleted;
    EC_Dialog.onDefaultDialogCompleted += onDefaultDialogCompleted;
    EC_Dialog.onPlayerOpenedPrivacyURL += onPlayerOpenedPrivacyURL;
    EC_Dialog.onPlayerOpenedTermsURL += onPlayerOpenedTermsURL;
}

private void OnDisable(){
    EC_Dialog.onCoppaDialogCompleted -= onCoppaDialogCompleted;
    EC_Dialog.onDefaultDialogCompleted -= onDefaultDialogCompleted;
    EC_Dialog.onPlayerOpenedPrivacyURL -= onPlayerOpenedPrivacyURL;
    EC_Dialog.onPlayerOpenedTermsURL -= onPlayerOpenedTermsURL;
}

```

Now, in order to check the age status of the player you can call the following method

```

EC_Settings.IsPlayerUnderage();

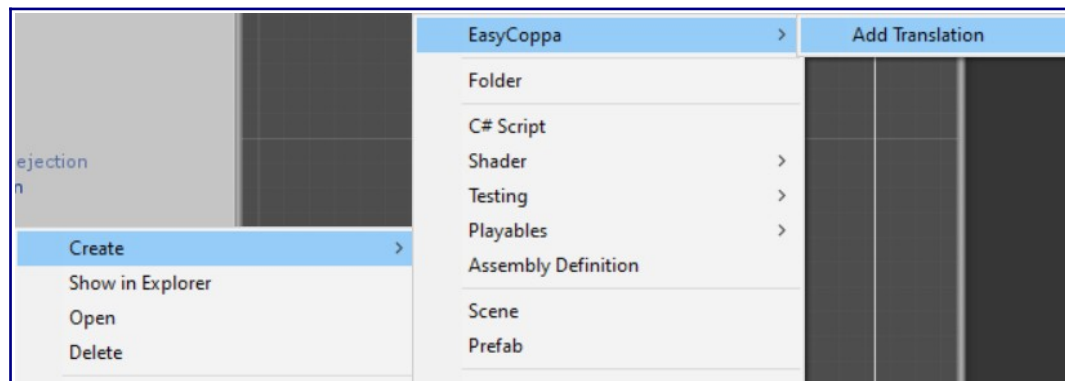
```

Localization

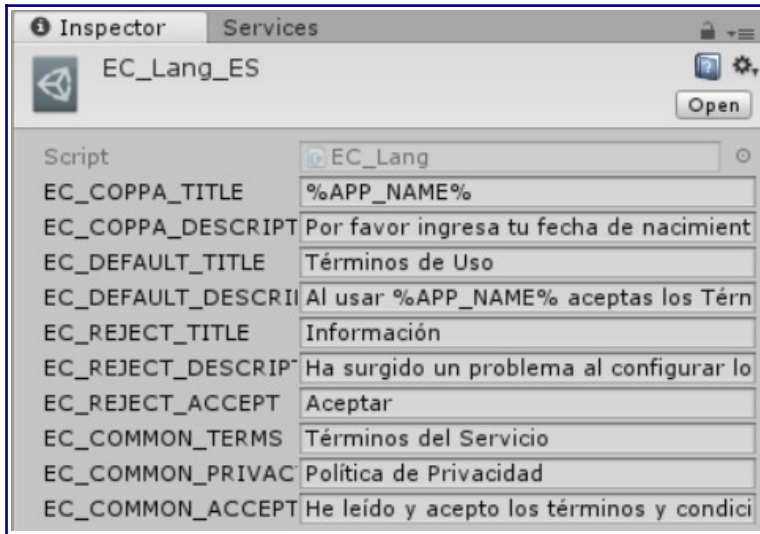
In order to use translations with Easy Coppa make sure to check the Use Translation checkbox.

To create a new translation just right-click the desired folder and select

Create/EasyCoppa/Add Translation



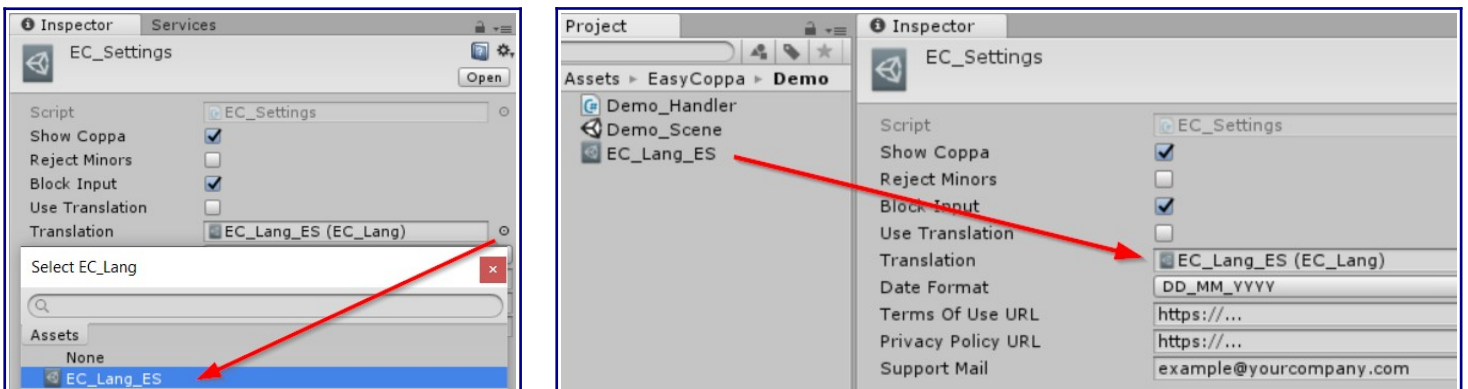
This will create a new EC_Lang asset where you can define your translations for each of the fields.



There are two ways of using this EC_Lang asset for your project.

1. Direct definition on EC_Settings
2. Runtime definition

The quick and easy way is direct definition. This is the best option if you are only localizing your game in one additional language. To do so, you just need to define your EC_Lang asset into EC_Settings either by clicking the little circle at the right and selecting the right EC_Lang asset or by dragging and dropping it into the Translation field.



On the other hand, to define the localization on runtime we need to follow three easy steps

```
//Expose a field for your EC_Lang
[SerializeField] EC_Lang localization;
//Assign it to EC_Settings
EC_Settings.SetTranslation(localization);
//Refresh the dialog's localization
EC_Dialog.RefreshLocalization();
Remember to assign your EC_Lang in the inspector or you'll get a Null Reference Exception
```

Testing

Although Easy Coppa was designed with mobile platforms in mind, it can be 100% tested on the Unity Editor and should work with most platforms.

Demo

You can try the plugin behavior in the demo scene located in *Assets/EasyCoppa/Demo/Demo_Scene*

FAQ

Q: What can I do if I accidentally deleted the EC_Settings asset?

A: Just go to *Window/EasyCoppa/Open Settings*. This menu selects the asset so you can configure it in the inspector. In the case EC_Settings.asset does not exist it creates it.