Console Enhanced

A replacement console for Unity3D
by Mark Currie

Website

unityconsole.com

Support

support@unityconsole.com

About ConsoleE

Console Enhanced is a replacement for the Unity Console. ConsoleE is an editor extension. No part of the console is included in your standalone application. There are two versions of ConsoleE, a free version and a pro version. This document covers both versions, features specific to the pro version are labelled with **[PRO]**.

Showing the Console Window

In the Unity Editor, select Window->Console Enhanced. Alternatively, you can press the hotkey Command+Shift+C on OSX or Control+Shift+C on Windows.

Usage

For the most part, using Console Enhanced is done in the same way as using Unity's built-in console. No code changes are necessary to use the ConsoleE. Just as in the regular console, calls to Debug.Log(), Debug.LogWarning(), and Debug.LogError() will show in ConsoleE.

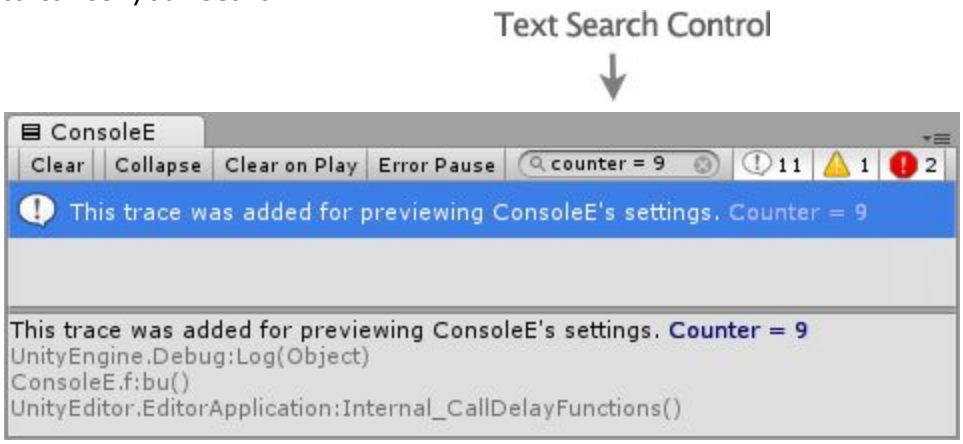
Settings Dialog

To show the Settings Dialog, right-click anywhere on the console window, when nothing is selected. You can also access settings from Unity's Edit->Preferences menu.

Nearly every item in the Settings Dialog has a tooltip. Just hold the mouse cursor over a UI element to see its tooltip.

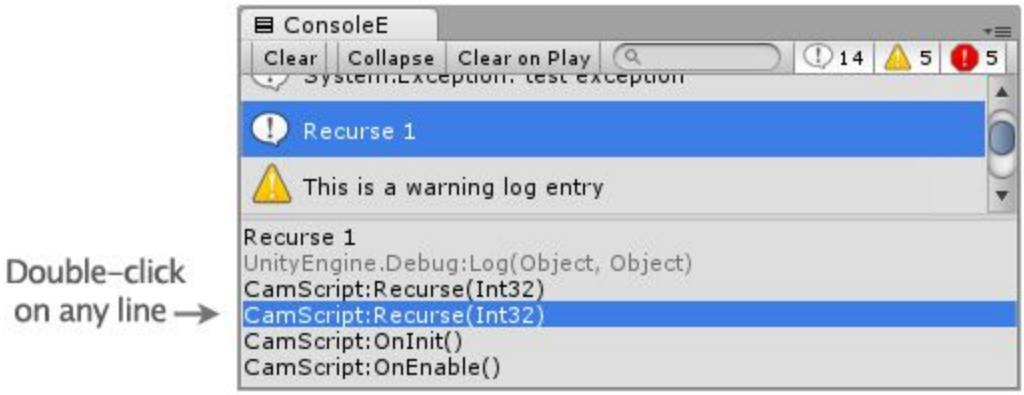
Text Search

ConsoleE has an optimized text search filter. Type some text in the control, and only log entries containing the same text will be shown. Pressing Tab or Ctrl+F will focus the text search control. Press Escape to cancel your search.



Callstack Navigation [PRO]

The pro version of Console Enhanced lets you easily open any row of the callstack. Double-click a row to open it. Using a long press with mouse drag will select a block of text rather than a single row.



Double-clicking this row will open its script file

Open With [PRO]

Right-click on any entry or callstack row to show the *Open With* menu. This feature is useful if you edit your source code in an editor like Visual Studio, and debug your code using MonoDevelop. The list of options found in the *Open With* menu is automatically generated based on your Unity Preferences. It can be configured in the dialog at Edit->Preferences->External Tools->External Script Editor. The *Open With* right-click menu is automatically hidden from view if you only have one external editor option.

Override File Open Behavior [PRO]

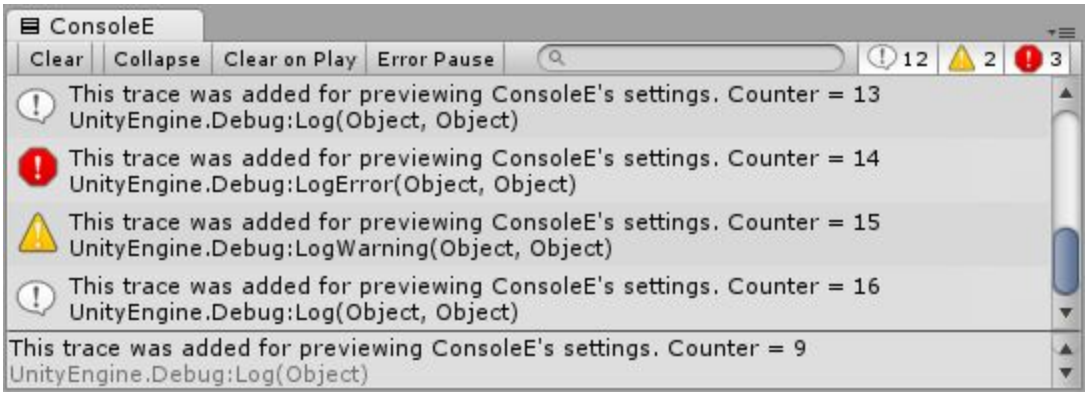
The file open override allows you to override how open files are open when console log entries are clicked. Using this option has no effect on how files are opened from the project window or other parts of the Unity Editor.

Note that on OSX, you may need to prefix your Arg Format String with --args. The Arg Format String is formed using the C# string.Format function. {0} is replaced with the filename and {1} is replaced with the line number.

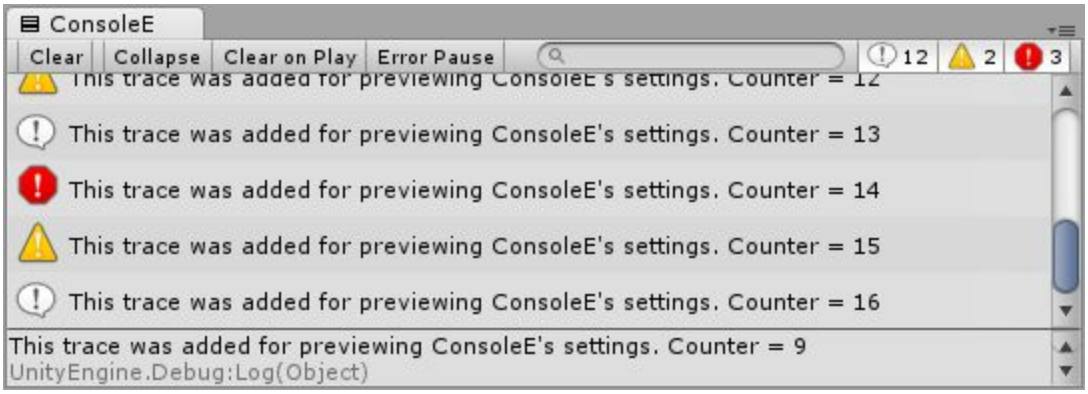
If this feature is enabled, only the primary external editor is overridden. Other editors opened with *Open With* will use the default open behavior.

Hide Second Line

Hiding the second line offers a more concise view of the console. The extra space can be used to fit more rows at once, or can be used to display a larger font.



When unchecked, each entry has two lines



When checked, each entry has just one line

Smooth Auto-scroll

This option smoothens the auto-scrolling that occurs when new entries are added to the console. This helps makes it clear when new log entries are added.

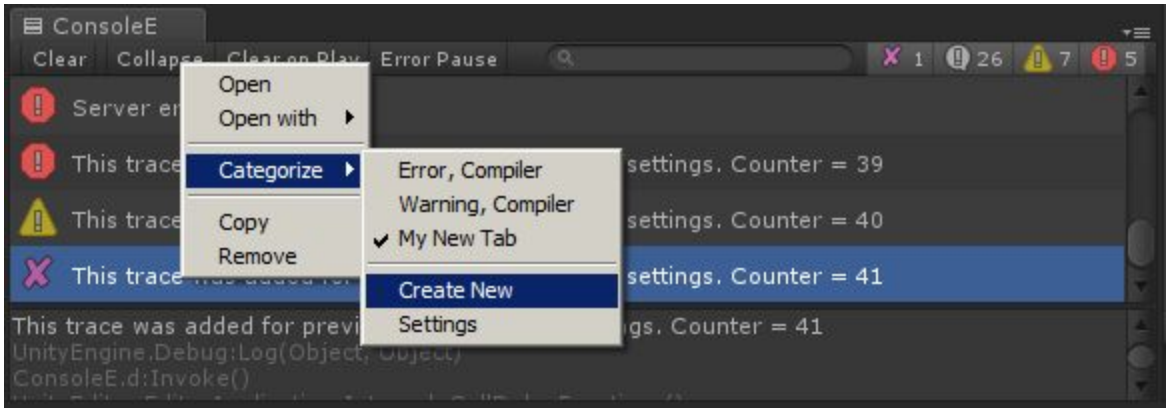
Multiselection

The console supports multiselection. Hold Control while clicking to toggle selection. Hold Shift while clicking to select a bunch of entries at once. Press Control+A to select all entries. Once selected, entries can be copied to the clipboard. Selected entries also can be removed from the console.

End-of-Log Marker

When enabled, the color of separator UI changes to make it clear if you are viewing the last log entry and if auto-scroll is active.

Custom Tabs & Categories [PRO]



The regular console has tabs and categories for Log, Warning, and Error. ConsoleE lets you create your own. To create a custom category, right click on on a log entry or callstack row, and choose *Categorize->Create New*. This will create a new category and ensure the selected item is included in the new category.

Keywords [PRO]

Custom categories tab use keywords to decide which log entries belong to them. Normally, adding keywords to a category is done via the right-click context menu. You can also manage the keyword list in the category's settings dialog.

Enabled Checkbox

Uncheck *Enabled* to complete disable the category. It's useful if you want to temporarily remove a category from the console.

Share with another Tab

Using this option, a category can inherit specific traits of another category. It's essentially a way to create subcategories of another category.

Consume Matches [PRO]

This is a option for custom categories. If checked and if a log entry has a matching keyword, the log entry will not be considered by any lower priority tabs. If not checked, matched results will not only show in this tab but will also show in other matching tabs. Tab order in the tab list decides which tab can consume a result first. Tabs at the bottom of the list are checked first. Tabs can be reordered using drag-and-drop.

Hide Toolbar Tab if Empty

This is a option for all category tabs. When enabled, the small toolbar icon in the top-right of the console will be hidden unless the tab has log entries.

Rows Colors

If Colorize is checked, background colors of rows will be overridden. Colors are overridden on a per-category basis.

Flash Effect

If enabled, a quick flash of color is shown on the background. This is useful for making sure log entries belonging to the category catch your attention.

Icon Editor

Each category has an associated icon. The basis of the icon is a unicode character. Its color, size, outline, and other visual properties can be edited.

Wrapper Functions [PRO]

Sometimes when you click a log entry, you don't want to be taken to Debug.Log(). You'd rather be taken to the method that called it. ConsoleE lets you mark methods in your project as wrapper functions. ConsoleE will then skip these methods as it scans the callstack when considering the best source file to open.

You can manually edit the Wrapper Function list from the Settings Dialog. Alternatively, you can right-click a callstack row, and mark the row as a wrapper.

Hide File Paths [PRO]

When enabled, file paths are hidden from the callstack area. This helps remove clutter.

Hide Wrappers from Second Line [PRO]

When this option is checked, Wrapper Functions are not shown in the second-line display (of any main entry). Instead, the line shown is the one that is opened if the entry is clicked.

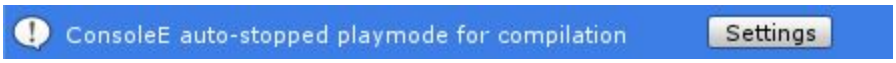
Compile During Play

By default, Unity automatically recompiles script changes, even if done during playmode. This feature is sometimes called *Live Recompile* or *Edit-and-Continue*. Many projects are not compatible with this feature of Unity. ConsoleE offers alternatives for how to handle scripts changes during playmode.

- **Compile on Stop**
  If playmode is active, compilation will be postponed until playmode is stopped.
- **Stop and Compile [PRO]**
  If Unity begins compilation while playmode is active, ConsoleE will automatically stop playmode.
- **Stop, Compile, Restart [PRO]**
  If Unity begins compilation while playmode is active, ConsoleE will automatically stop playmode, and then restart once compilation is complete.

Custom Buttons & Actions

Just like the regular console, ConsoleE supports rich text. Additionally, you can create custom buttons and actions, like this:



The settings button is created using log text in this format:

```
<ConsoleE_Button=ConsoleE.Api.ShowMiscSettingsWindow>Settings</ConsoleE_Button>
```

If the button is clicked, any static method can be called. In this case, the method is ShowMiscSettingsWindow.

It's also possible to call a static method if an log entry is double clicked. It follows a similar format:

```
<ConsoleE_OnClick=ConsoleE.Api.ShowMiscSettingsWindow>
```

If the static method takes a single string parameter, the log text and callstack will be passed to the method.

Activating Pro

ConsoleE runs in free version mode until activated with a pro license. Activation is initiated from the License tab in the Settings Dialog. Enter the unity invoice number in the field provided. Your unity invoice number is included in an email sent by Unity when the asset is ordered. You can also find your invoice number at the Unity website, in the billing transaction history.

Like Unity, the pro version of ConsoleE is licensed per-seat. Each user may use ConsoleE on two computers simultaneously.

Both the free version and pro version use the same binary DLL file. This means you can add the DLL to your repository, and some members of your team can use the free version while others use the pro version.

For teams, it's possible to set up ConsoleE so that the activation field is pre-populated with your team's serial key or unity invoice number. To do this, add a text file with the name *serialkey* (with no file extension) and place file next to ConsoleE.dll in your project. The file should contain a single line of text, your serial key or unity invoice number.

If you have any questions about activation, please email me.