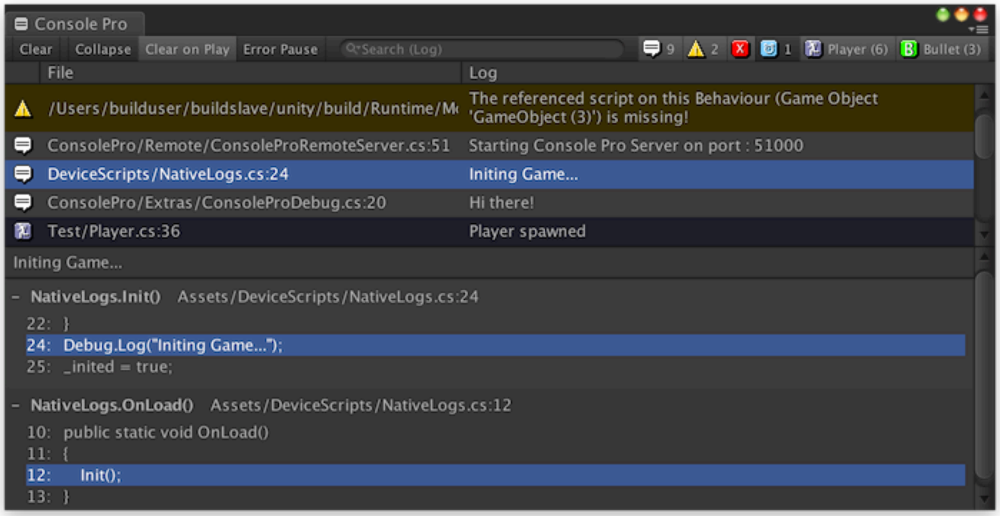
CONSOLE PRO

EDITOR CONSOLE PRO



	File	Log
⚠	/Users/builduser/buildslave/unity/build/Runtime/M	The referenced script on this Behaviour (Game Object 'GameObject (3)') is missing!
	ConsolePro/Remote/ConsoleProRemoteServer.cs:51	Starting Console Pro Server on port : 51000
	DeviceScripts/NativeLogs.cs:24	Initing Game...
	ConsolePro/Extras/ConsoleProDebug.cs:20	Hi there!
	Test/Player.cs:36	Player spawned

```
Initing Game...

- NativeLogs.Init()   Assets/DeviceScripts/NativeLogs.cs:24
  22:  }
  24:  Debug.Log("Initing Game...");
  25:  _inited = true;

- NativeLogs.OnLoad()   Assets/DeviceScripts/NativeLogs.cs:12
  10:  public static void OnLoad()
  11:  {
  12:      Init();
  13:  }
```

GENERAL

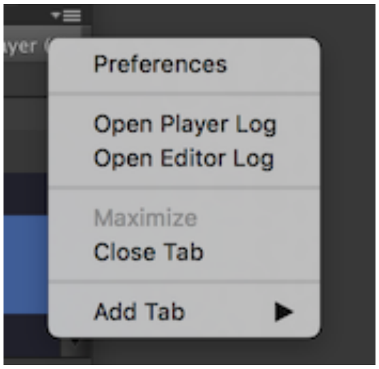
Opening Editor Console Pro

You can open ECP with the hotkey COMMAND+\ or in the menu Window/Console Pro 3

Opening Preferences

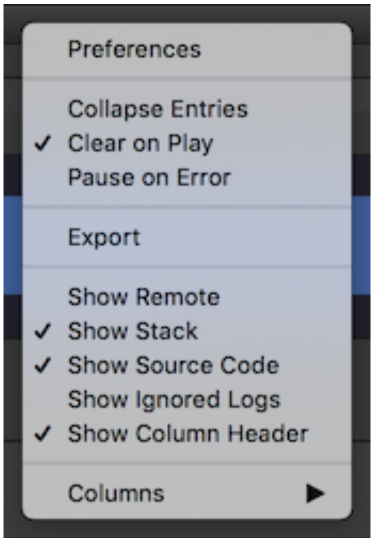
There are a few ways to open preferences:

- Right click on a blank area of the toolbar and click Preferences
- Click the icon in the top right of the window and click Preferences



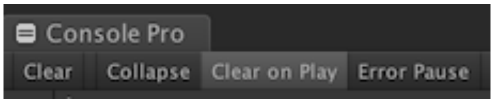
Opening Main Menu

Right click on any blank area of the toolbar to bring up the main menu of ECP.


Preferences
Collapse Entries
✓ Clear on Play
Pause on Error
Export
Show Remote
✓ Show Stack
✓ Show Source Code
Show Ignored Logs
✓ Show Column Header
Columns ▶

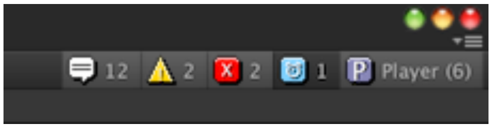
TOOLBAR BUTTONS

Toolbar Buttons



The toolbar button layout can be customized by right clicking on a toolbar button and clicking Customize, or going to Preferences/Toolbar.

Filter Buttons

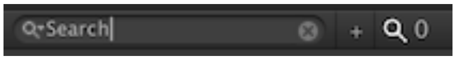


Standard, Custom, and Temporary filters all show on the right side of the toolbar.

- Click a filter to show/hide its matched logs
- Ctrl+Click a filter to hide every filter but that one
- Shift+Click a filter to show all filters
- Right Click a filter to customize it

SEARCHING

Search Field

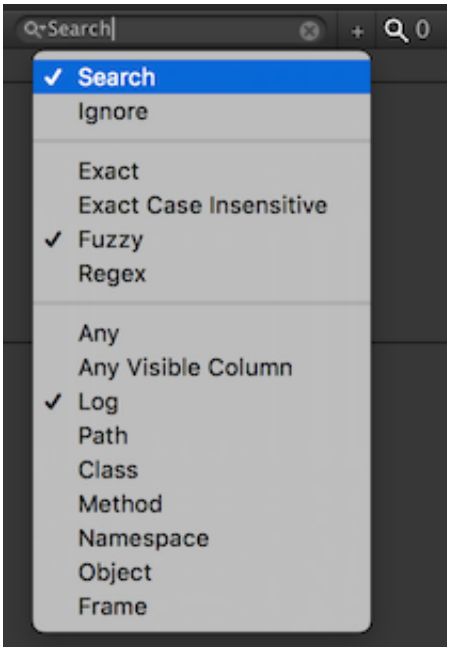


The search field allows you to filter the current visible logs by typing in a search string.

Search Count

The search count shows how many logs matched your search

Search Options

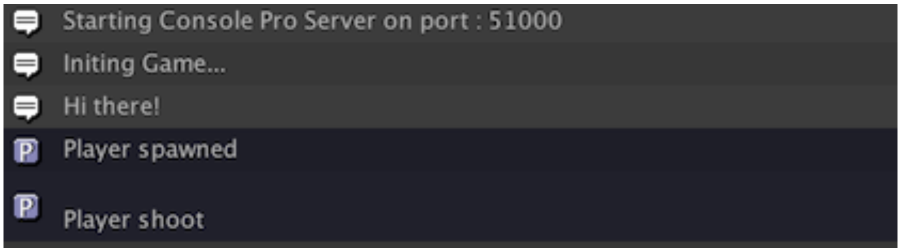


Click the magnifying glass to show search options. They are as follows:

- Search - Only show logs that include this search term
- Ignore - Hide logs that include this search term

- Exact/Exact Case Insensitive - Search term must exactly match
- Fuzzy - More tolerant search, allows some typos
- Regex - Match against regular expression
- The remaining options define what column you are searching in

LOGS



```
Starting Console Pro Server on port : 51000
Initing Game...
Hi there!
Player spawned

Player shoot
```

Colorized
Log backgrounds can be colorized by their matching filter, so they are easier to see. You can disable this in Preferences/Logs.

Word Wrap
Log text can be wrapped which makes the log taller to fit content. You can disable this in Preferences/Logs.

Multiline
Logs will show multiple lines (\n), making the log taller to fit content. When this is disabled logs will show a return character at the end to show that they are multiline. You can disable this in Preferences/Logs.

Show Class And File Under Log Text
This option makes the logs look similar to Unity default logs. You can disable this in Preferences/Logs.

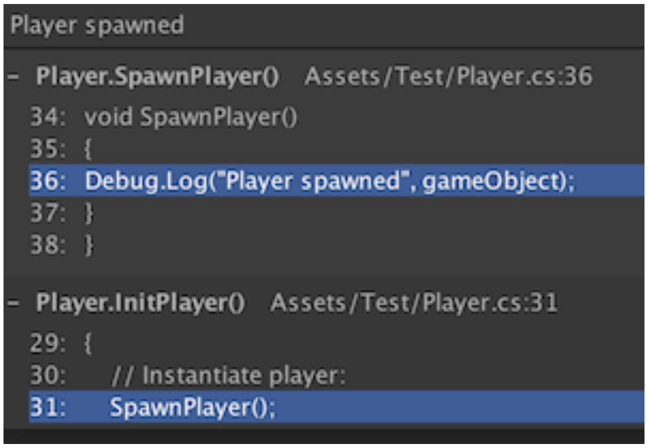
Copy
Right click on a log entry to get various options for copying to the clipboard.

Multi Select
Hold shift or command and click multiple logs to select them all for copy operations.

Ignore

If you would like to always ignore a log, such as "Too many audio sources", you can do this by adding search terms in Preferences/Ignore in the "Ignore Logs" category.

STACK



```
Player spawned
- Player.SpawnPlayer()   Assets/Test/Player.cs:36
    void SpawnPlayer()
    {
    Debug.Log("Player spawned", gameObject);
    }
    }
- Player.InitPlayer()   Assets/Test/Player.cs:31
    {
        // Instantiate player:
        SpawnPlayer();
```

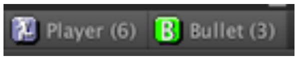
Source Code Preview
ECP will show you a preview of the source code surrounding each line of the stack. You can configure this in Preferences/Source Preview.

Jump To Line
Click on any line in the stack to open your code editor to that line, including source code lines.

Ignore Wrapper Functions
Right click on a stack line and click "Ignore This Stack Entry" to ignore wrapper functions, for example MyDebug.MyLog. ECP will now not show any stack line calling this function, and double clicking the log will skip it as well. This can be configured in Preferences/Ignore. If you would like to ignore an entire class, simply erase the function name from the entry in Ignore Stack Entries (so the above example would just be MyDebug).

Copy
Right click on a stack line and you have various options to copy it to the clipboard, such as with the surrounding source code or the whole stack.

CUSTOM FILTERS



Custom Filters

You can create your own filters, whose buttons show up next to the standard filters (Log, Warning, Error, etc), using search terms. They can be customized in Preferences/Custom Filters. You can quickly make one by typing a search term in the search field and clicking the small "+" button next to it.

Options

The options in Preferences/Custom Filters are as follows:

- Swallow Standard Filter - When enabled, logs matched in this filter will no longer show up in the Log/Error/Warning filters
- Use Colors In Log - When enabled, the filter color will be the background color on colorized logs. Otherwise they will use the standard filter color
- Use Advanced Colors - Overrides the text and background color.
- Show Button In Toolbar - This allows you to colorize and give an icon to log entries but not have a toolbar button
- Hide Button If No Matches - When no logs match this filter, the button will disappear
- Hide Button Label - Don't show a label on this specific button.
- Use Icon In Log - When enabled, this will override the icon in the log entries to be the same as the toolbar icon
- Icon Options - You can setup your own icon for the filter, using colors, pre-made icons, text characters, and custom textures
- Searches - The terms to search for to match a log to this filter

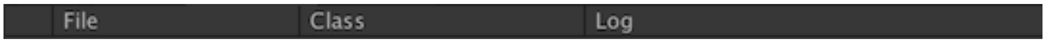
Temporary Filters

Temporary filters are custom filters that are created dynamically when a log with a trigger character occurs.
For example:

```
Debug.Log("#Player# Instantiated")
```

This would create a filter button titled "Player" that only exists while the log exists.
The color is dynamically generated based on the title, and will always be the same as long as the title is the same.
The trigger character and look of them can be customized in Preferences/Temporary Filters.

COLUMNS



Columns
In Preferences/Columns, you can show things like the File, Class, Method, Timestamp of logs in multiple columns. They can be resized by dragging on the separator line of a column.

Column Header
Right click on the column header and click "Hide Column Header" if you don't want to have it visible. It can be reenabled in the Main Menu.

SHARED SETTINGS

Shared Settings

In Preferences/Shared Settings, you can create a file with certain settings that you can share over Source Control. This is useful if multiple people on a project want to share Custom Filters, Ignored Stack Entries, etc.

FONTS AND COLORS

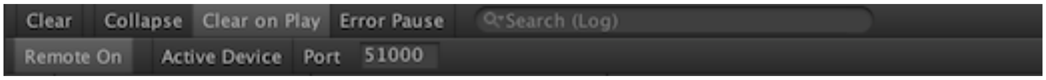
Fonts

In Preferences/Fonts And Colors you can change the font of the logs or stack to any font in your Assets folder.

Colors

In Preferences/Fonts And Colors you can change colors like the alternating log background and the selected log entry.

REMOTE



Remote Mode

Remote Mode allows you to connect to a device running ConsoleProRemoteServer and receive the logs in your editor, much like Unity's Profiler.

Using Remote

To use Remote functionality, follow these steps:

1. Add a GameObject in your scene and add the component ConsoleProRemoteServer to it.
2. Build a to your device with Development Build checked (ConsoleProRemoteServer will only be active in builds with DEBUG defined to prevent Release apps from asking for network permissions).
3. Connect your device to the same network as your computer with Unity. They need to be on the same subnet mask.
4. Start the app, it should tell you "ConsoleProRemoteServer starting on port *" in your device log.
5. Show the Remote Toolbar in Console Pro by right clicking on a blank area of the toolbar and checking Show Remote.
6. Press the button "Remote On" to activate remote mode. If everything is set up correctly, you will now start receiving logs.

Troubleshooting Remote

- Make sure your device is connected to the same network and the same subnet mask as your computer
- Make sure your app is a development build or you have manually defined DEBUG
- Some devices do not support Remote Server yet, such as Windows Phone 8.1
- Check for any logs at the start of your app concerning ConsoleProRemoteServer
- Try a different port, and make sure your firewall isn't blocking the port you are using

CONSOLE PRO API

Console Pro API

Logs can pass in commands to Console Pro through the Console Pro API. To pass in a command, you simply add this to your log:

```
"\nCPAPI:{"cmd":"Command" "ParamName":"ParamValue"}
```

Console Pro Debug Class

ConsoleProDebug.cs contains easy to use functions for API commands.

Watch Command

This will only produce one log entry regardless of how many times it is logged, allowing you to track variables without spam.

Command: *Watch*
Param: *name* - Name of the watch, all entries with this name will replace the previous log

Example:

```
Debug.Log("Player X Position:" + transform.position.x.ToString() + "\nCPAPI:{"cmd":"Watch" "name":"" + "PXPos" + ""}");
```

ConsoleProDebug function:

```
ConsoleProDebug.Watch("Player X Position", transform.position.x);
```

Filter Command

This will end a log to a specific filter regardless of contents

Command: *Filter*
Param: *name* - Name of the filter to send this log to.

Example:

```
Debug.Log("Hi" + "\nCPAPI:{"cmd":"Filter" "name":"" + "CustomFilter" + ""}");
```

ConsoleProDebug function:

```
ConsoleProDebug.LogToFilter("Hi", "CustomFilter");
```

SUPPORT

Support Email
If you have any bugs or feature requests, don't hesitate to email
Support@FlyingWorm.com