

# OML Console Pro User Guide

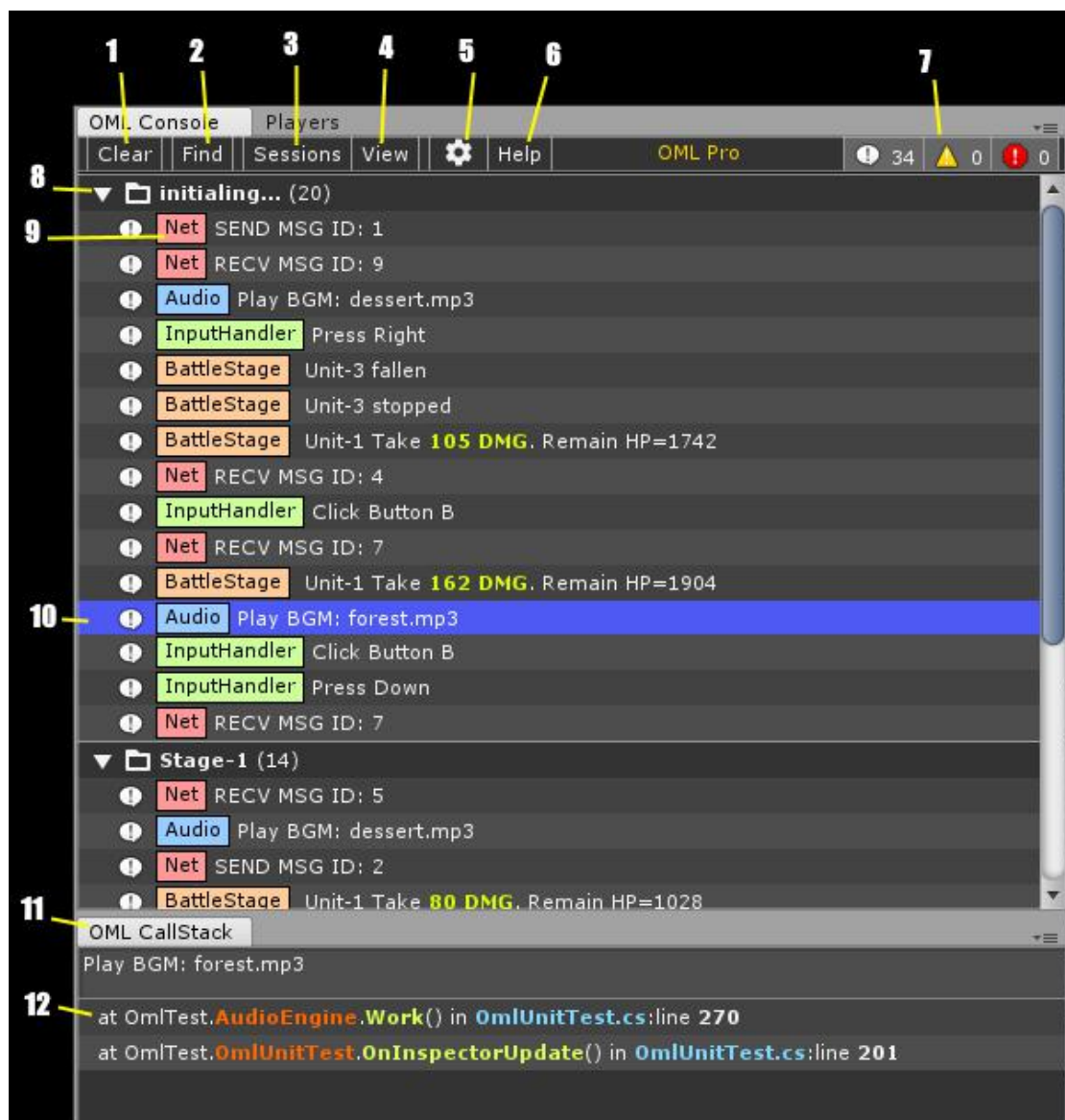
Introduction	1
Console Interface	1
Stack-Trace Interface	3
About Tagging	5
Tag Operations (select, zoom-in, hide)	8
Searching text	10
Log Session	11
Display Setting	14
Advance Filtering (channel and tag group)	15
Log Channel	16
Highlights	17
Other Useful Settings	19
Customize Log Style	22
Remote Logging	22
Watch Data Panel	22
Exporting Logs	23
Thank you	24

## Introduction

OML Console is a professional logging system with advanced filtering features and neat visual design to allow you reading your logs in a much more enjoyable way.

## User Interface Basics

1. Press to clear the current logs.
2. Press to toggle find text control.
3. Press to list sessions without logs for quicker log navigation.
4. Press to toggle a tool bar to adjust console view option.
5. Press to open OML setting panel.
6. Press to show quick info about OML basics.
7. Display current log counts by log type.
8. The begin of a foldable session.
9. Tags assigned to a log.
10. Select log message
11. Call stack window
12. Stack frames of select log



## StackTrace Panel

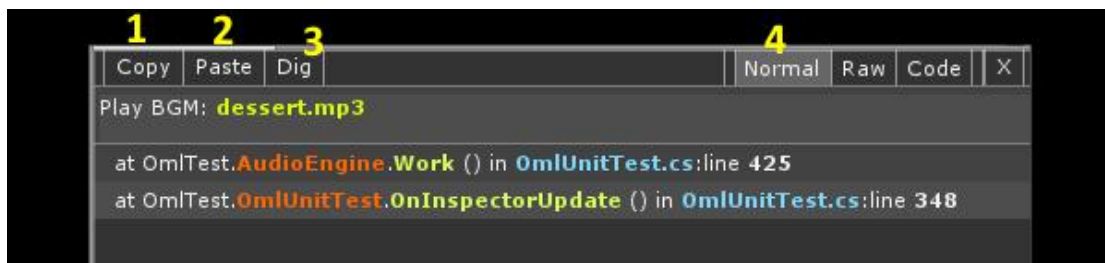
The CallStack show you more info about a log. Double click on a stacktrace to go to the corresponding source file.

To open this panel, click [Tools]->[Oh-My-Log!]->[CallStack] form Unity main menu.



*The default interface look like this.*

You can turn on the CallStack's own menu from view option to have more control.

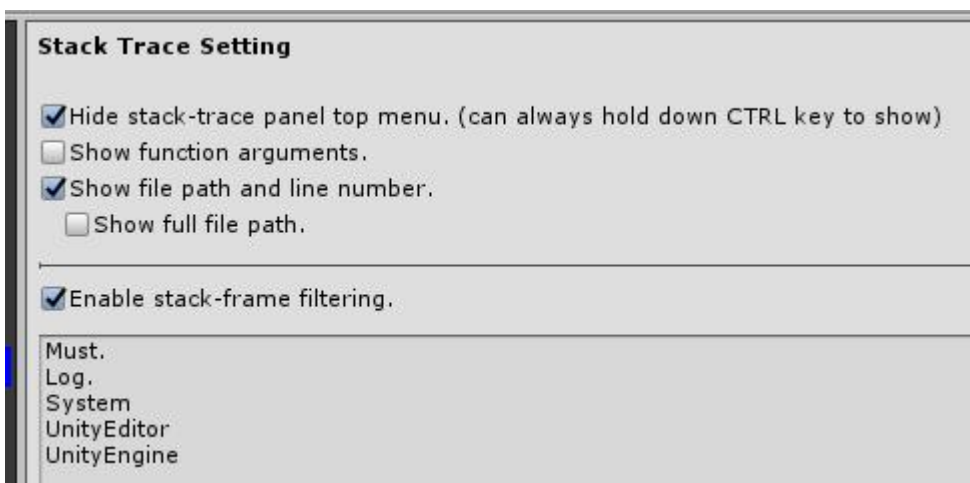


CallStack's own menu with some handy buttons

1. Copy the log text with stacktrace to clipboard.
2. Paste text from clipboard to show in this panel.
3. 'Dig' help you parse stacktrace from message lines.

E. g. running server/client in the same program. Log with stacktrace sent from server to client. Dig can then resolve stacktrace embedded from server log message.

4. Set the display mode between Normal View, Raw View and Code View.



More display setting can be changed to suit your need.

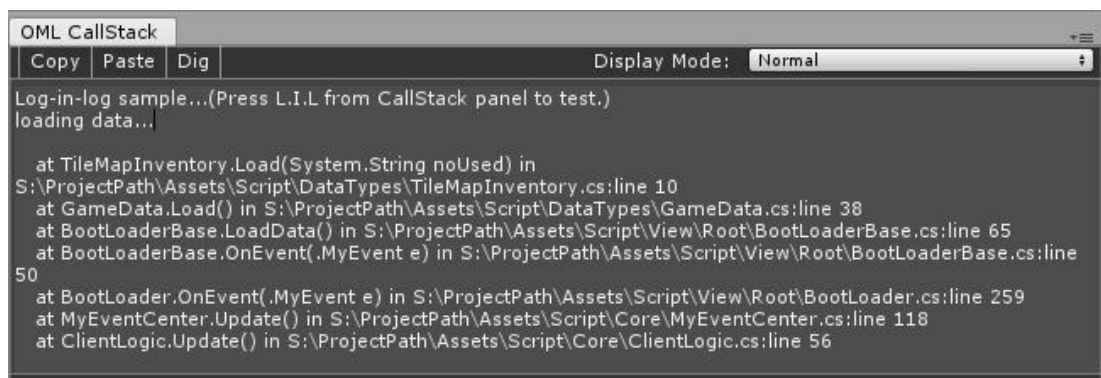
## Example of using Paste and Dig.

Suppose you saved a log to and text file and now want to check that out.  
Here is the log message.

```
Log-in-log sample...(Press L.I.L from CallStack panel to test.)
loading data...

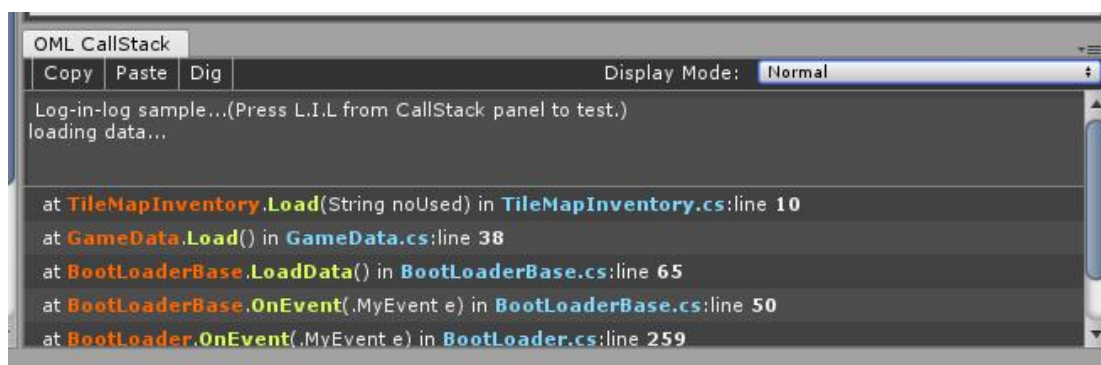
    at TileMapInventory.Load(System.String noUsed) in
S:\ProjectPath\Assets\Script\DataTypes\TileMapInventory.cs:line 10
    at GameData.Load() in S:\ProjectPath\Assets\Script\DataTypes\GameData.cs:line 38
    at BootLoaderBase.LoadData() in S:\ProjectPath\Assets\Script\View\Root\BootLoaderBase.cs:line 65
    at BootLoaderBase.OnEvent(.MyEvent e) in S:\ProjectPath\Assets\Script\View\Root\BootLoaderBase.cs:line 50
    at BootLoader.OnEvent(.MyEvent e) in S:\ProjectPath\Assets\Script\View\Root\BootLoader.cs:line 259
    at MyEventCenter.Update() in S:\ProjectPath\Assets\Script\Core\MyEventCenter.cs:line 118
    at ClientLogic.Update() in S:\ProjectPath\Assets\Script\Core\ClientLogic.cs:line 56
```

You copy the log and press the [Paste] button on OML CallStack panel.



After the “paste” operation. The log is now showing in the CallStack panel.  
Press [Dig] to ask OML to try parse the stacks for you...

\* In recent OML version, when pasted a log , “dig” operation will be auto performed.



As a result, OML parse the stacktrace and beautify it for you to read.

You can quickly go to the source file as usual by double clicking on a stack row (if the log is from the same project).

## About Tagging

Tagging is one of the biggest topic on OML console. Once a log is tagged, you can do amazing things with it. Tagging not only immediately let you know what the message is about, it also help you do group-based filtering as easy as just on key press or click.

There are three ways to tag your log messages. Logs can be tagged by the message itself (embedded), by the caller class or be caller instance.



### 1. Tag by message

Log message can be in the following format to let OML tag it:

[<tag1>,<tag2>,...#<channel>] <log\_message>

where ,<tag2>... and #<channel> are optional if all you want is a simple tag.

Logging with tag examples:

#### 1. a simple log with not tag.

```
Debug.Log("start ABC.ogg");
```

#### 2. tag the log with 'Audio'

```
Debug.Log("[Audio] start ABC.ogg");
```

#### 3. tag the log with 2 tags 'System' and 'Audio'

```
Debug.Log("[System,Audio] start ABC.ogg");
```

#### 4. log to channel 1. (Other channels won't show this log)

```
Debug.Log("[Unit,Data#1] A take 10 DMG.");
```

5. Channels are useful for filtering if some logical units print out similar logs

```
Debug.Log("[Unit,Data#2] B take 5 DMG.", ".");
```

6. Log to channel 3 with out any tag

```
Debug.Log("#3] server is shutting down...");
```

## 2. Tag by caller class

Besides manually tagging a message, you can also let OML auto tag for you and help you keep your log message clean!

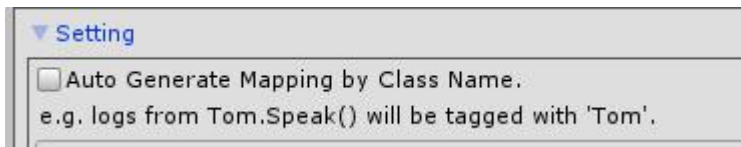
Note: in recent version, an option is added to auto display class label when no tag is present in a message. Class label allow zoom-in mode just like with tag. Unless you need to assign those message to Tag Group, it is not needed to use auto tag anymore for simple filtering.

### How it works:

Each log you normally see is created at some point in your program with **caller** that call the log function. E.g. Debug.Log(), Debug.LogWarning(), etc. We can make use of this info to let OML auto tag your logs for you!

OML keeps a mapping table between the caller class and tag names.

When a log arrived, it check with the caller class name against the mapping table and see if any tag should be given to this log.



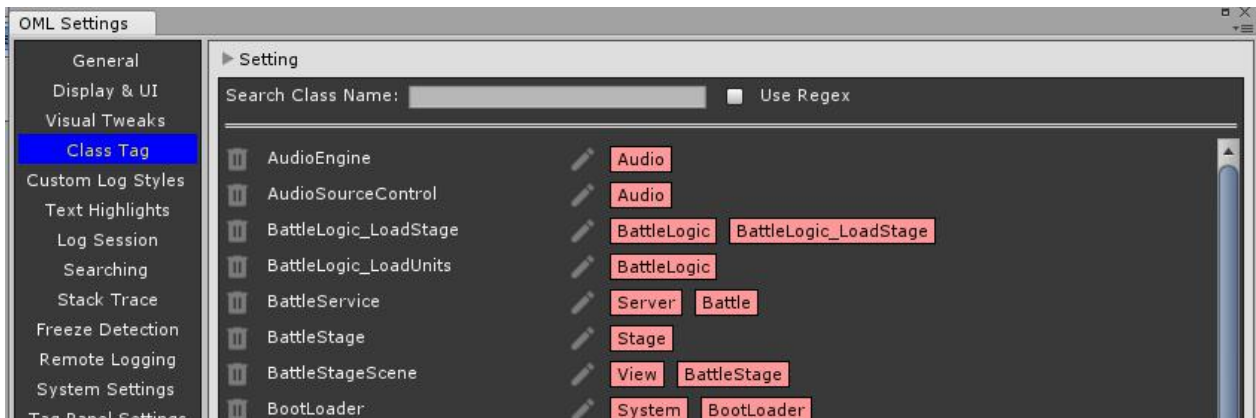
You can enable the “Auto generate Mapping by ClassName” setting to let OML auto insert new mapping for log message. The log will be tagged using class name by default.

If you don’t want to use the original class name and want to customize or group them. You can tag classes by selecting a class label and press **F2**.



This will bring up a UI for creating new class tag.

There is a built-in UI for editing those mappings from setting window.



You can also open `<Project Dir>/oml.class_tags.xml` to group/rename the mappings manually.

**Tips:** don’t forget to press the [Reload] button after you manually edited the mapping data file. If auto reload is not checked.



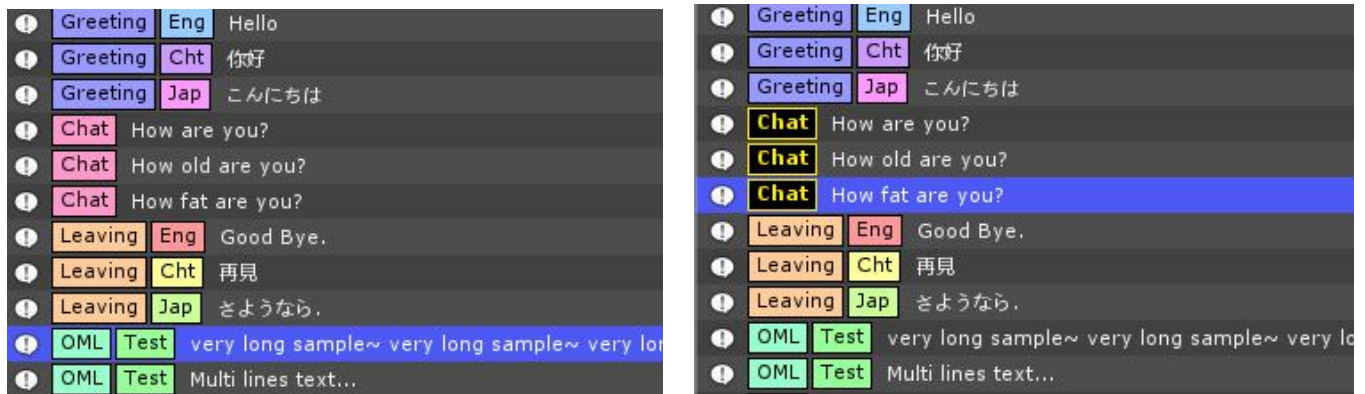
## Tag Operations

Once you have your logs tagged, you can do something with them like filtering in some simple ways.

### Selecting a tag

Just click on a tag to select it. A selected tag appear in yellow black color.

To unselect a tag, simply click on it again.

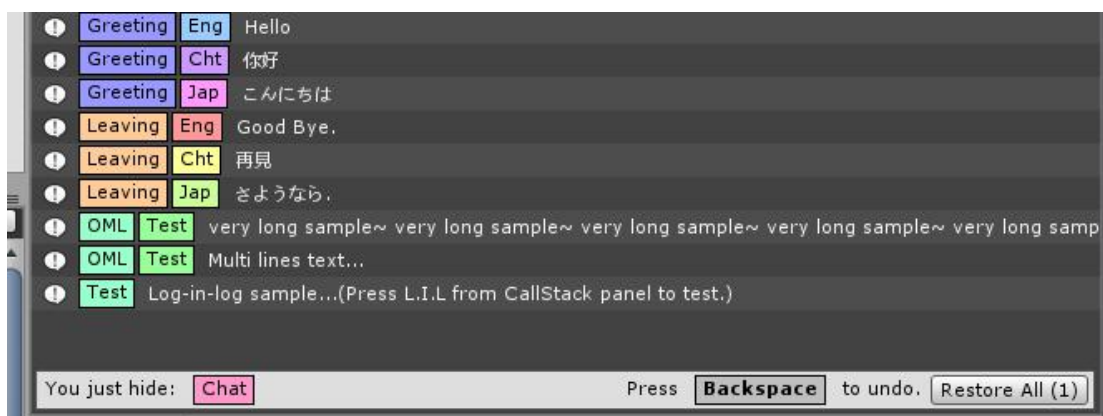


### Hiding logs with tag ( PRO Verion only)

1. Select a tag
2. Press [DEL] button

A hint will be shown at the bottom about your hidden tag. (can be turned off)

To restore the tag you last hide, press the [BackSpace] button from your keyboard once. You can also click on the [Restore All] on the hint UI to unhide tags you DEL'ed,



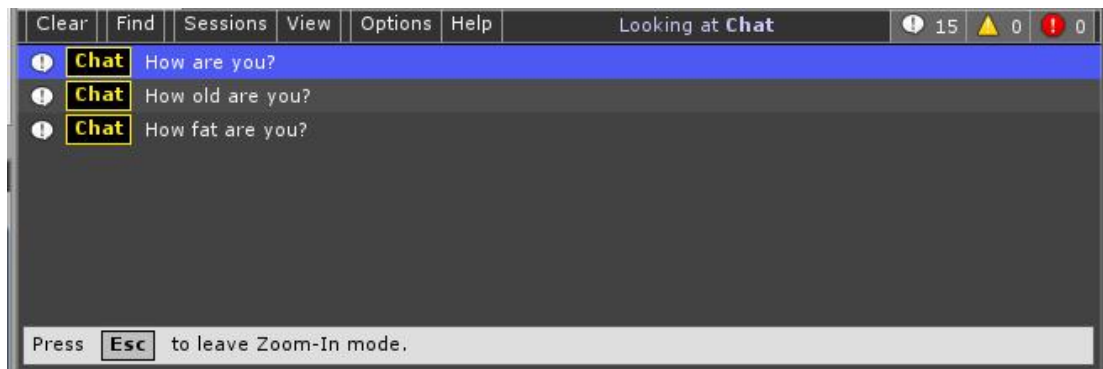


### Focus on a single tag. (a.k.a. Zoom-In mode) ( PRO Verion only)

If you want to show only logs with a certain tag, you can zoom-in to the tag.  
All other logs without that tag will be hidden.

1. Select the tag you want to focus on.
2. Press [Enter] key.

Press [Esc] key to exit Zoom-In mode.



I zoomed in to [Chat] in this example.

Shortcut at the bottom can be turned off in Option panel.

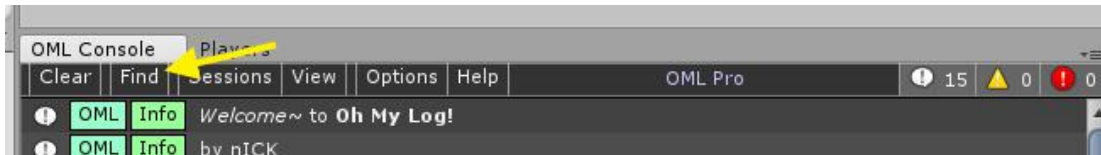
Top menu will also hint you about the current Zoom-In.

E.g. "Looking at Chat" from screenshot.

## Searching the log messages

Searching text form log messages is easy. To start searching, you can either

1. click on the [Find] button on top menu.
2. **Simply start typing** when the console has focus on it.

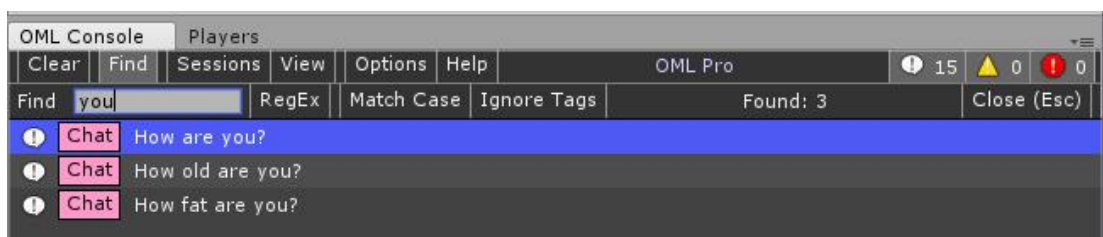


The [Find] button is there just for completeness. You don't need to use it but just click anywhere and start typing.

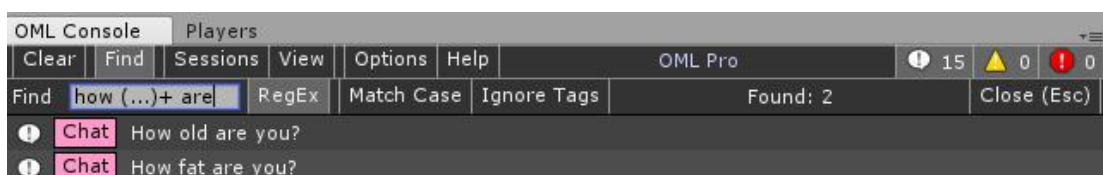
## Finder Interface



1. Search target. Text or Tag
2. Input box for search text or Regular Expression pattern.
3. Enable/disable Regular Expression search.
4. Toggle whether searching should be case sensitive
5. Toggle filter mode which will hide irrelevant logs.
6. Close the Finder. You can simply press ESC which is more direct.



*Simple text search*



*Regular Expression search*

## Log Session ( PRO Verion only)

### What is log session and why?

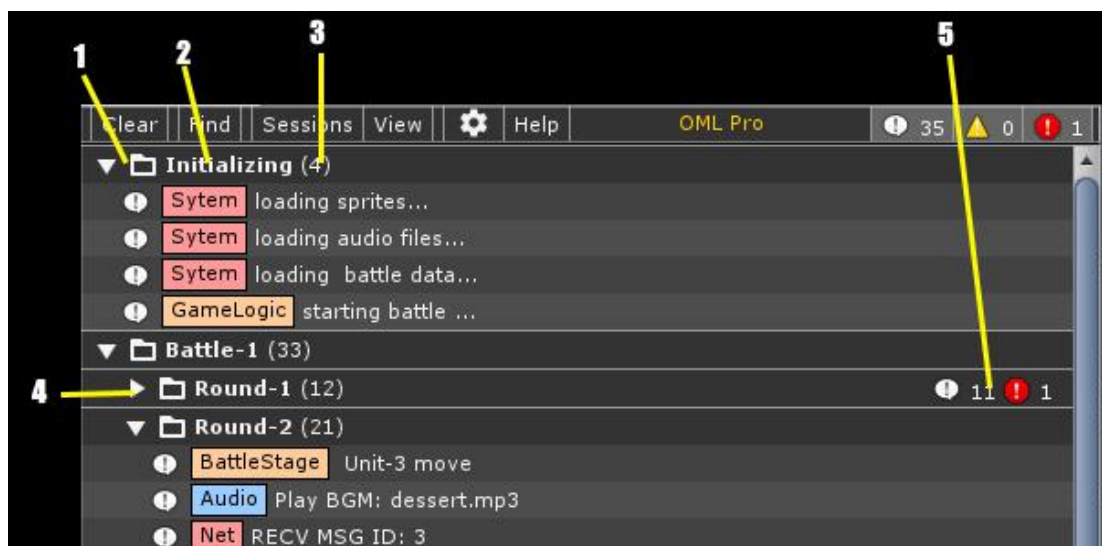
A log session is a chunk of logs created within a time period.

Once logs are under a section, you can fold/unfold it to help you read them.

The session bar between log also stand out and helps you identify important phases in your game.

### Session Interface

1. A session row representing the start of a session
2. Session name
3. Total log count in this session
4. A folded sub-session.
5. Logs count by log type (info, warning, error) in this folded session.

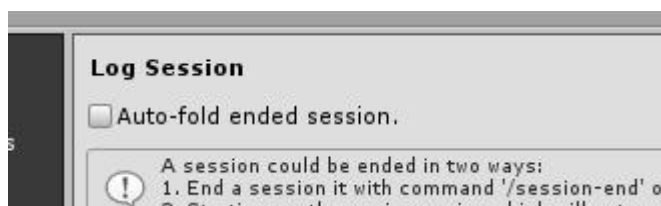


“Round-1” session is folded, logs within it will not be shown, providing a cleaner view.

### Tips:

-Use right-click on a session row to fold/unfold it instead of only selecting it.

-you can check the “Auto-fold ended session” to let OML auto fold a session when it is completed.



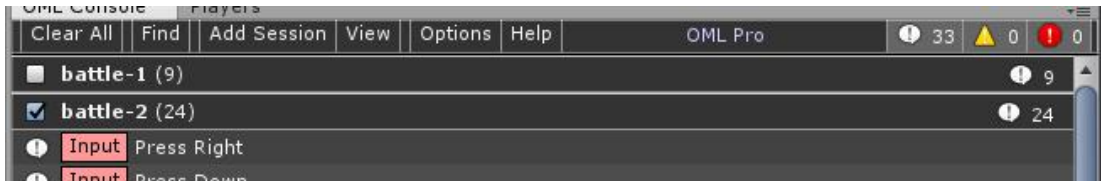
## How to start/stop a session?

To start a session...

### **Option.1** - New session by hand manually

You can start a session whenever you want by

1. Hold Down [Ctrl] to toggle top menu to show alternative buttons.
2. Click [Add Session]



When holding down [Ctrl] key, top menu show [Add Session] instead of the normal [Session] button.

### **Option.2** - New session by code automatically

You can create new sessions automatically by logging special command.

To start a new session or sub-session:

**`/session <NAME>`**

\* <NAME> is optional, When not provide, auto naming will be used.

To start a new **main** session and close all other existing sessions:

**`/main-session`**

To end a session:

**`/session-end <NAME>`**

\* <NAME> is optional. When not provided, current session will be ended.

Example

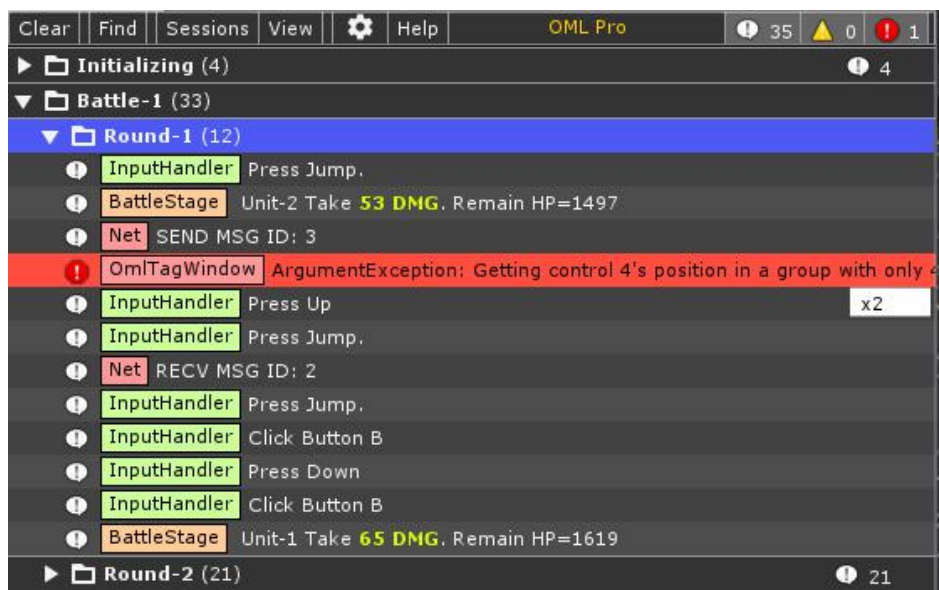
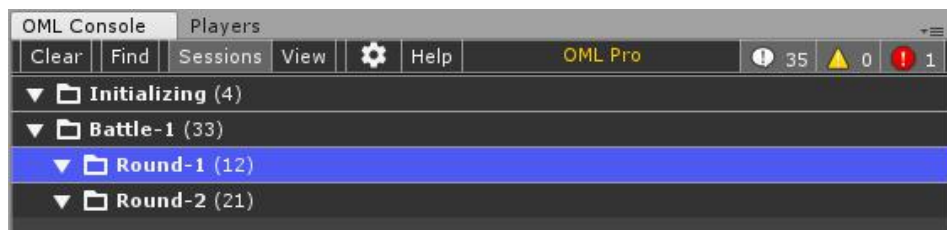
```
Debug.Log("/session PlayerTurn");
```

NOTE: When a session is started during another session, the old session will be ended automatically.

## Session List View



The session list view mode is enabled by clicking on the [Sessions] button on OML top menu. It allow you to quickly see all the sessions and go to them more quickly.



Select a session row and press Enter to view a particular session OR press [Sessions] button again to restore original log view.

## Display Settings

You can access view settings from view setting bar OR from the option panel.  
View setting bar can be toggled by holding down Ctrl Key.



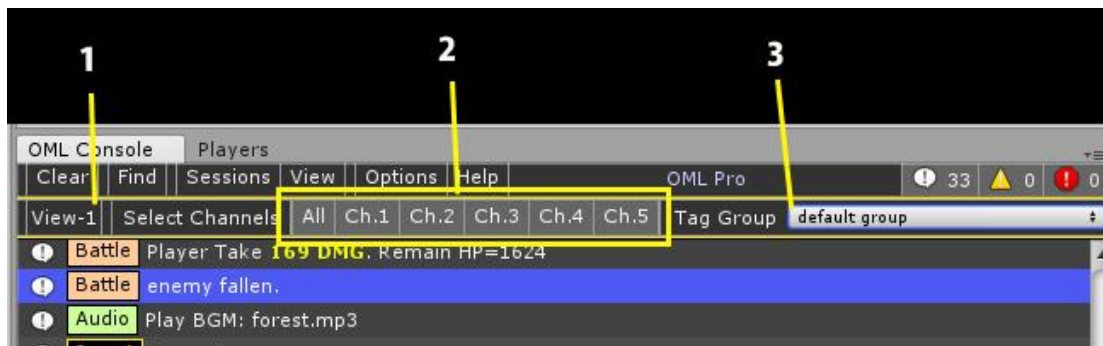
Log Icon	<p>Toggle whether log row show log type icon</p>
Time	<p>Toggle whether log row show timestamp</p>
Tags	<p>Toggle whether log row show tags</p>
Channel	<p>Toggle whether log row show channel number</p>
Sessions	<p>Toggle whether log row show session rows</p>
View Tabs Bar	<p>Toggle whether to display Advance Filtering bar. (will talk more about this later)</p>
Split View	<p>Toggle between single view mode and horizontal/vertical split view mode. Click a activated mode button again to restore single view mode.</p>

## Advance Filtering (View Tabs Bar)

Advance filtering consist of two main filtering mechanisms: Tag Group and Channel  
They both have their own unique characteristic that make them best suit for certain scenarios.

### Interface

1. The adv. filtering bar. It also show the current view number.  
If you have enabled split-view, you will see one showing View-1 and the other showing View-2 on its own adv. filtering bar.
2. Log Channel to be applied to the current log stream.  
You can normal Click to select a channel or Ctrl+Click to toggle a channel (to make combination).
3. Tag group to be applied to the current log stream.

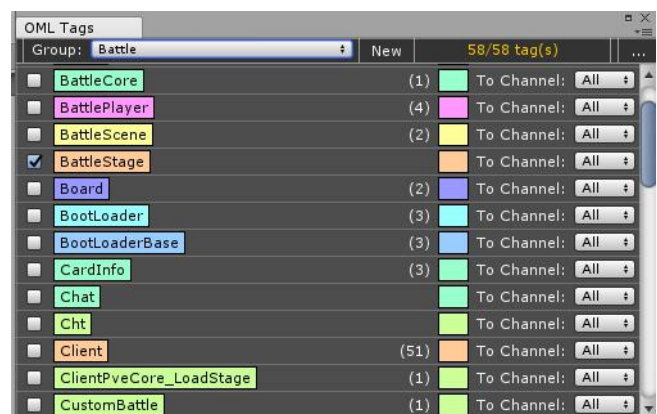
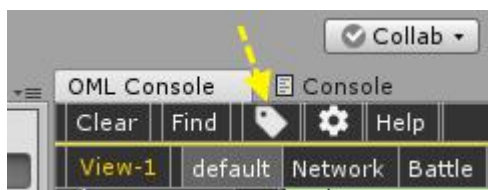


### Tag Group

Tag group is just a group of tag with its state saved in group so that you could apply them to your logs without toggling those tags one by one every time.

E.g. When I was working on the networking part of my game, I would have a tag group named Networking with all the networking related tags checked. Then I could have one log stream showing the networking activity and the other log stream showing other gaming logs in OML *split views*.

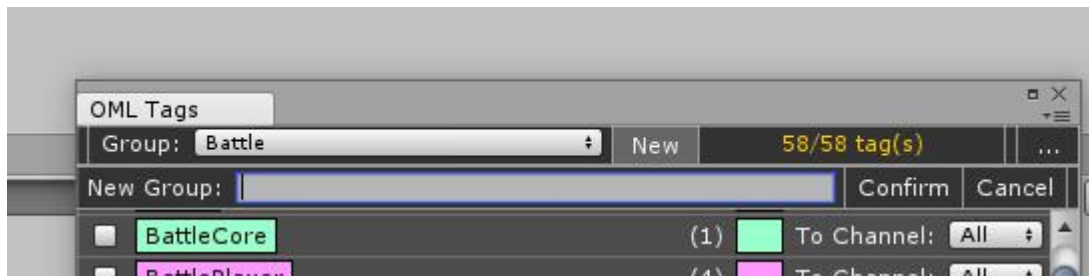
If you want to check out the list of current tags, you can open the OML Tags panel from Unity Menu -> [Tools] -> [Oh-My-Logs!] -> [Tags]  
OR simply press the tag icon button from top menu.





## Managnig Tag Group

You can create new tag group by press the [New] button on the top.

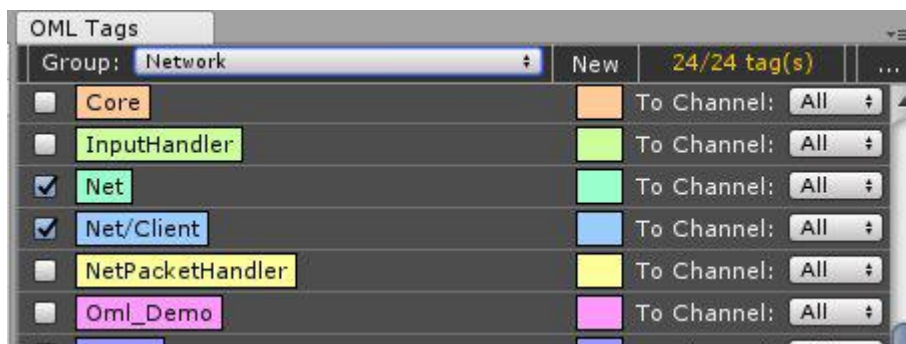


If you need to delete a tag group, hold down Ctrl key to toggle hidden UI. The [delete] button is right next to the [new] button.

## Assign tags to Tag Group

You can assign a tag to a tag group in two different way.

1. Assign from tag panel.
  - A. .Select the group you want to assign to.
  - B. Find the tag and check the check box on the left.  
(you can easily find any existing tag by just typing on the panel.)



2. Assign from console
  - A. Right-click on the tag
  - B. Check the target tag group



## Log Channel

Logs by default will reach all channels. If a log is set to a channel, only the view with that channel enabled will see that log.

Channel filtering is VERY useful when you have log message printing the same thing from different instances of the same kind. In that case, they will generally have the same tags and it is very hard to distinguish them visually even if you print out their ID. With log channel, you can do filtering on them.

Clear	Find	Sessions	View	Options	Help		
View-1	Select Channels	All	Ch.1	Ch.2	Ch.3	Ch.4	Ch.5
❗ Battle	Unit-1 Take 143 DMG. Remain HP=1481						
❗ Battle	Unit-1 shoot						
❗ Battle	Unit-3 Take 64 DMG. Remain HP=1834						
❗ Battle	Unit-3 Take 86 DMG. Remain HP=1187						
❗ Battle	Unit-3 Take 74 DMG. Remain HP=1775						
❗ Battle	Unit-3 Take 120 DMG. Remain HP=1851						
❗ Battle	Unit-3 Take 51 DMG. Remain HP=1244						
❗ Battle	Unit-1 Take 109 DMG. Remain HP=1787						
❗ Battle	Unit-3 Take 119 DMG. Remain HP=1009						
❗ Battle	Unit-3 shoot						
❗ Battle	Unit-1 fallen						
❗ Battle	Unit-1 move						
❗ Battle	Unit-2 Take 96 DMG. Remain HP=1198						
❗ Battle	Unit-3 Take 67 DMG. Remain HP=1469						
❗ Battle	Unit-2 Take 183 DMG. Remain HP=1254						
❗ Battle	Unit-2 stopped						
❗ Battle	Unit-3 Take 97 DMG. Remain HP=1185						
❗ Battle	Unit-1 Take 156 DMG. Remain HP=1833						
❗ Battle	Unit-3 stopped						
❗ Battle	Unit-1 Take 116 DMG. Remain HP=1713						
❗ Battle	Unit-1 shoot						
❗ Battle	Unit-3 Take 105 DMG. Remain HP=1767						

Log Icon	Timestamp	Tags	Sessions	Adv. Filtering	Collapse		
View-1	Select Channels	All	Ch.1	Ch.2	Ch.3	Ch.4	Ch.5
❗ Battle	Unit-1 Take 143 DMG. Remain HP=1481						
❗ Battle	Unit-1 shoot						
❗ Battle	Unit-1 Take 109 DMG. Remain HP=1787						
❗ Battle	Unit-1 fallen						
❗ Battle	Unit-1 move						
❗ Battle	Unit-1 Take 156 DMG. Remain HP=1833						
❗ Battle	Unit-1 Take 116 DMG. Remain HP=1713						
❗ Battle	Unit-1 shoot						
❗ Battle	Unit-1 Take 187 DMG. Remain HP=1422						
❗ Battle	Unit-1 Take 177 DMG. Remain HP=1176						
View-2	Select Channels	All	Ch.1	Ch.2	Ch.3	Ch.4	Ch.5
❗ Battle	Unit-2 Take 198 DMG. Remain HP=1846						
❗ Battle	Unit-2 Take 103 DMG. Remain HP=1046						
❗ Battle	Unit-2 Take 89 DMG. Remain HP=1506						
❗ Battle	Unit-2 fallen						
❗ Battle	Unit-2 Take 151 DMG. Remain HP=1551						
❗ Battle	Unit-2 Take 70 DMG. Remain HP=1636						
❗ Battle	Unit-2 Take 169 DMG. Remain HP=1915						
❗ Battle	Unit-2 Take 167 DMG. Remain HP=1575						
❗ Battle	Unit-2 Take 102 DMG. Remain HP=1091						
❗ Battle	Unit-2 shoot						

On the left without channel filtering, logs about unit taking damage are crossing with each other between Units. On the right, you can use channel to separate them to track a certain instance much more easily. You can also use split view to observe on different things.

To write a log to a channel, prepend your log message with `[#<channel>]` syntax.

Example:

```
Debug.Log("[#1] sample log message."); // log to channel 1
Debug.Log("[Battle#1] sample log message."); // log to channel 1 with a tag.
Debug.Log("[Data*#1] sample log message."); // similar but allow class tag to be added on the right.
```

## Highlights

Highlights help you colorize your log and make key words or information stand out so that you can spot them more easily.

### Setting Interface

Open the setting interface from [Options] and go to tab [Text Highlight].

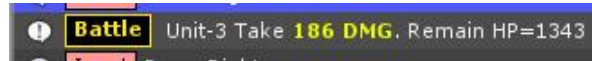


Use	Whether enable this highlight or not.
Search Pattern	Text or regex you want to highlight
Tag	Empty to apply to all tags. Type a tag path to restrict search range and improve performance.
Color	Highlight color, use transparent if you only want to apply font style like bold or italic
Style	Set highlight font style. (normal, bold, italic)
Case S.	Set whether the search should be case sensitive
Delete Button	Delete this highlight

Example: highlighting damage with Regex.



Before



After

NOTE:

Due to unity rich text limitation , you cannot have multiple highlights partially overlapping with each other. Only the 1<sup>st</sup> hit will be applied.

## Some Useful Settings

The screenshot shows the OML settings window. It is divided into several sections. The top section contains options for 'On play...', 'When playing...', and 'When copying logs on Console... (Ctrl+C)'. The 'Collapse' section is highlighted with a blue '1'. The middle section, 'Force certain logs to show even if hidden by tag filter or session folding.', is highlighted with a blue '2'. The bottom section, 'Import / Export settings.', is highlighted with a blue '3'.

**On play...**  
☒ Clear the console.

**When playing...**  
☐ Auto pause on Warning.  
☐ Auto pause on Error.

**When copying logs on Console... (Ctrl+C)**  
☒ include Tags  
☒ include Stack Trace

**1 Collapse**

Info	Off	Neighbor	Global
Warning	Off	Neighbor	Global
Error	Off	Neighbor	Global

**2 Force certain logs to show even if hidden by tag filter or session folding.**

☒ Force Warning Visible  
☒ Force Error Visible  
\*check these if you don't want to miss them in hidden tags or sessions.  
☒ Show a lock icon to those logs.

**3 Import / Export settings.** (For sharing settings or version control)

Personal Settings (Display Preference, etc)

Project Settings (Highlights and Tag-Groups)

### 1. Collapsing settings

In OML, you can enable/disable collapse to each of the log types( info, warning and error) independently. Sometimes you only want to collapse warning and error but let info to print as usual.

You can also set “Neighbor mode” to only collapse logs with neighbors so that new repeating logs won’t ‘sink’ which in turn reduce the chance you not seeing them appear.

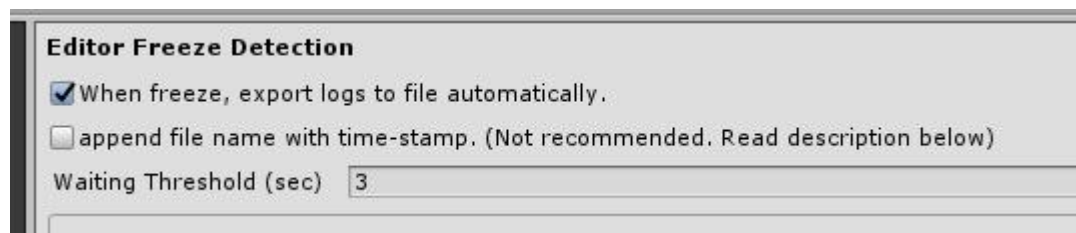
### 2. Force Warning, Error to display, ignoring tag and section filter.

Since the introduction of filtering, I found that there is a chance of missing error/warning if they were hidden by the current tag or section state. These option make them ignore these two filters and always visible.

### 3. Import/Export settings

Use this to share OML settings between computers or user.  
(Buy your legit OML copies to be used by different users!)

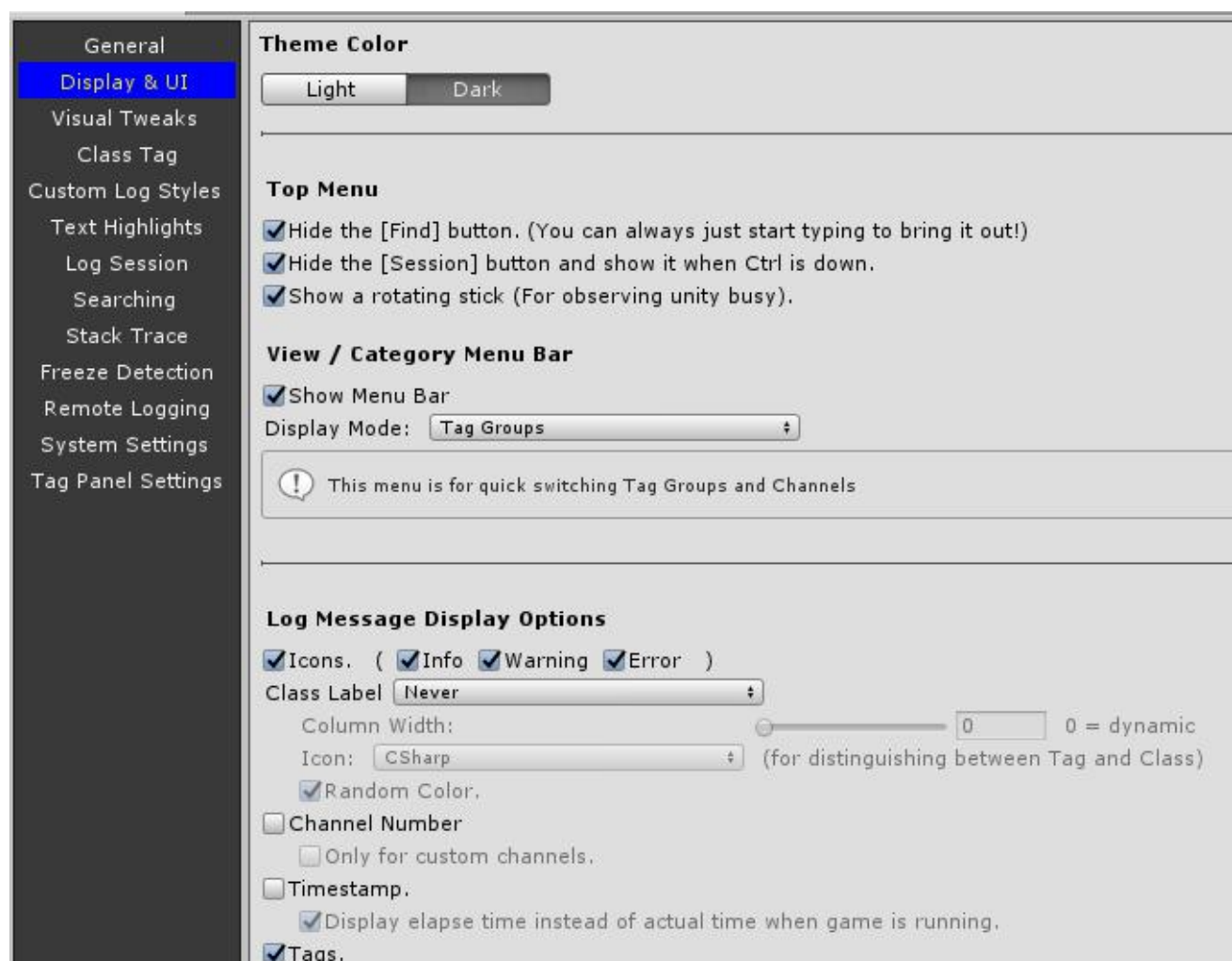
## Unity Freeze Detection and auto export



When enabled, OML will auto export current logs to project folder if freeze is detected.

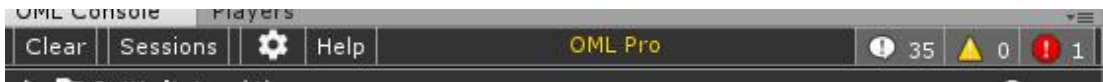
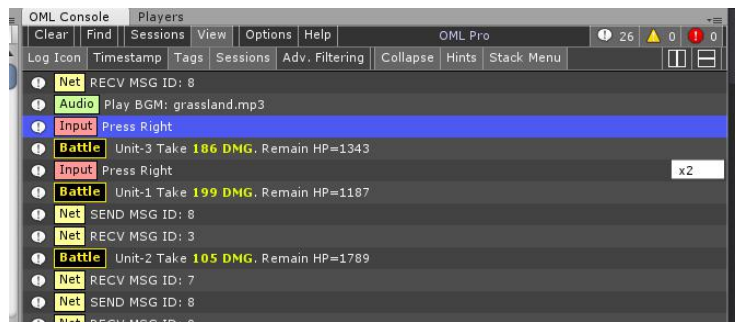
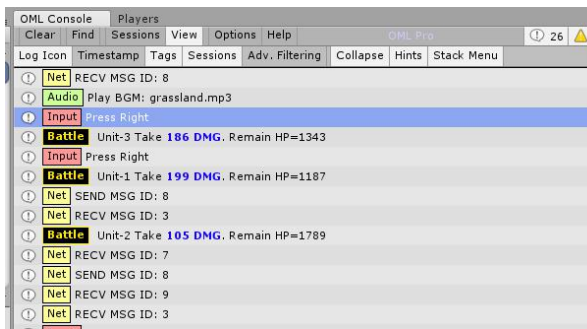
When Unity editor is frozen (e.g. dead loop situation) , everything including the console is not responsive. This could be a way to help you understand what happens before the freeze.

Many of the settings are self explanatory.



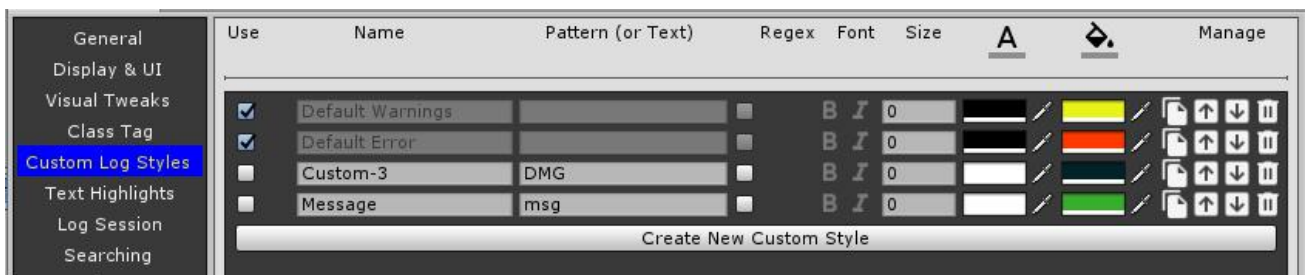


OML currently have two themes: Light and Dark



After hiding Find and View button

## Custom Log Styles



Custom Log Styles allow you to customize how logs is stylized when shown.

To customize the style for certain log, you first need to define a search pattern to identify the log message. It can be regex search or simple text search. After that, any targeted log will be applied with the specified colors and font styles.

The styles at higher place in the list will be test first.

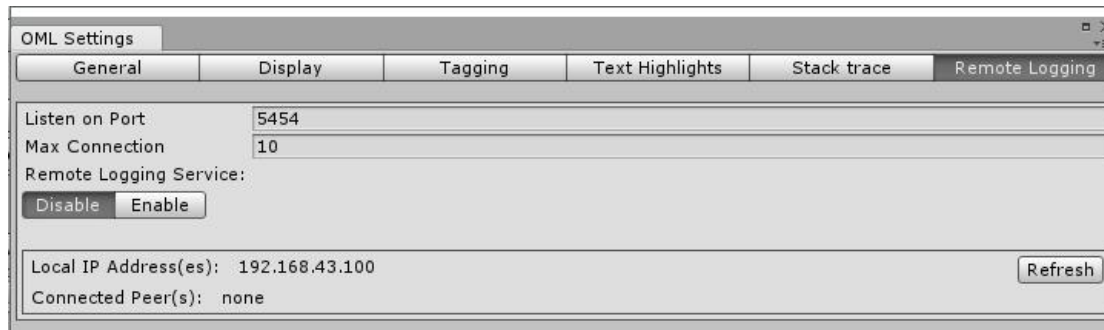




## Remote Logging

OML also allow you to have logs message pushed from a build (exe, android, etc) so that you can see how things are doing even outside the Unity editor.

It is as simple as placing an prefab to your application's first scene.



### **Listen on Port**

The UDP network port number to be used by OML to receive log message from a build.

If you have firewall enabled, make sure your firewall setting allow message to pass through this port.

### **Max Connection**

Set a max number of connection allowed to connect.

E.g. when multiple clients are connecting to a server project.

To start the service, simply press [Enable] button.

Local IP address are at the info box for your convenience.

## Client Side Config

### **Run Condition**

Set whether log upload should only be enabled in Development build or even release build.

### **Include Stacktrace**

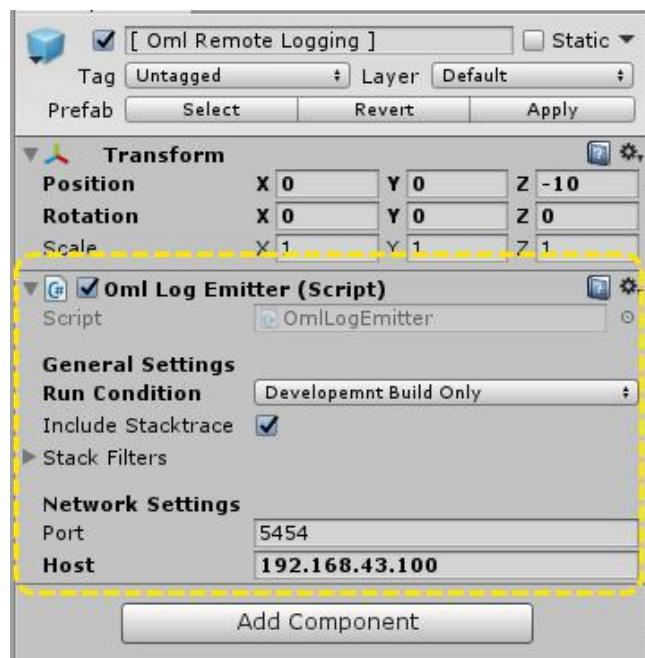
Turn off to have better performance if you don't want to have stacktrace included.

### **Stacktrace filters**

For filtering unimportant stacktrace.

### **Network Settings**

Port and host IP address to OML running computer.



## OML Watch Panel

OML also support convenient variable watching with log command on a dedicated panel. Logs with the set-watch command won't get into the console view and help keep your log stream clean.

The watch panel also allow you to group related fields into a single row for a more compact and easy-to-read view.

Example.

```
Debug.Log("/set Coins=15");
```

```
Debug.Log("/set Player.ATK=80");
```

When Edit mode is turned on, you can select each data entry to change their color.

OML Watch		
Edit	Coloring Rules	Clear
Coins	15	
Score	990	
Time	99	
Enemy.ATK	80	
Enemy.DEF	50	
Enemy.HP	100	
Enemy.MP	10	
Player.ATK	80	
Player.DEF	50	
Player.HP	100	
Player.MP	10	

You can also multi select them by shift click and merge them as a single row for more compact view.

Edit	All	None	Same Parent	
<input type="checkbox"/> Coins				-
<input type="checkbox"/> Score				-
<input type="checkbox"/> Time				-
<input checked="" type="checkbox"/> Enemy.ATK				-
<input type="checkbox"/> Enemy.DEF				-
<input type="checkbox"/> Enemy.HP				-
<input type="checkbox"/> Enemy.MP				-
<input type="checkbox"/> Player.ATK				-
<input type="checkbox"/> Player.DEF				-
Change Color:				
<div> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> <span>Abc</span> </div>				
Selected : 1				
Merge All				

[illegible]

OML Watch											
Edit	Coloring Rules										Clear
Coins	15			Score	990			Time	99		
Enemy	ATK	80		DEF	50		HP	100		MP	10
Player	ATK	80		DEF	50		HP	100		MP	10

## Exporting logs

You can export logs to text file or HTML file

To export current logs,

1. Press hold down [Ctrl] key to bring out the [Export] button

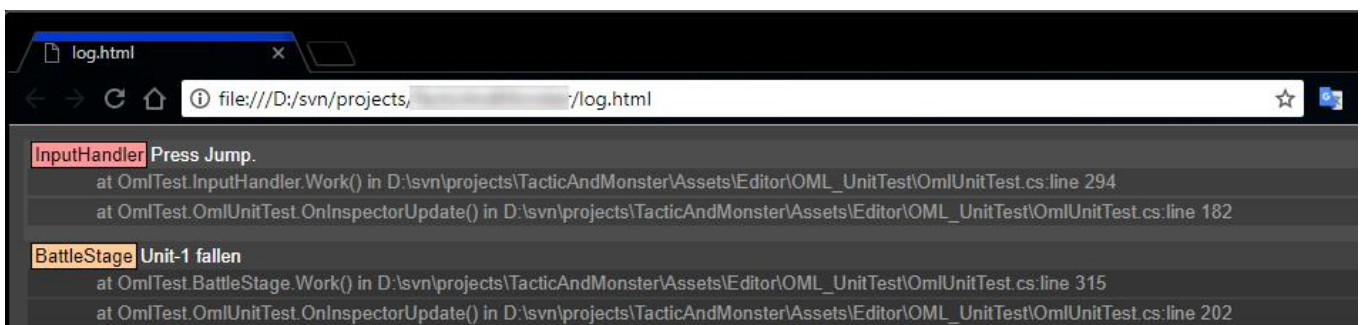


2. Click the [Export] button to bring out export panel



3. Press [Save HTML] or [Save Text] to start export

\* file will be exported to Project Root directory.



Sample HTML export with little coloring for better viewing experience

Thank you for your support.

If you found any bug, feel free to report to me preferably with a direct email to [omlconsole@outlook.com](mailto:omlconsole@outlook.com) or write to the corresponding unity forum thread (not yet created as of this writing) pointed from asset store.

You can also send me suggestions and do feature request to me.

I will implement your idea if you idea is good and practical and will benefit lots of OML users.

Complex new features may take time to implement.

Enjoy~

-Nicromancer