Jahro

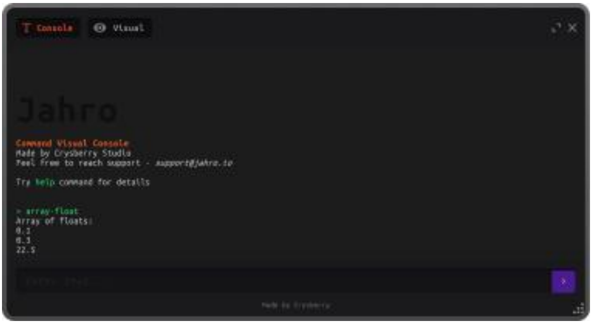
Console User Guide v.2.1.0

~1. What is Jahro Visual Console?

Jahro Console is a brand new plugin for Unity 3D. It strives to provide an amazing development experience by an opportunity to use the console for executing commands in a Unity Project directly or in a build of an app on a target device (after integration and building)

Jahro plugin supports:

- a native application for Unity 3D - UnityEditor;
- mobile devices - iOS, Android;
- desktop OS - Universal Windows Platform, Linux, MacOS;
- API - WebGL.



```
Command Visual Console
Made by Crysberry Studio
Feel free to reach support - support@jahro.io

Try help command for details


> array-float
Array of floats:
0.1
0.3
22.5
```

Native version

Mobile version

Strengths

- Undemanding to resources and light as a feather.
- Non-developers’ oriented UX and UI.
- Flexibility of customization and effortless control.
- Simple access to commands during runtime.
- Easy parameters input for commands.

Tricks

- Possibility to select Assemblies as commands source easily.
- Easy removal from a build without any harm and stress.
- Customizable launch options for both mobile and desktop versions.
- Keyboard shortcuts for launching, closing, switching modes.
- Innumerous command parameter types support: from Int to Vector and Enum.
- Console or Visual view for running commands.
- Easy change of window position or size.
- Predictive commands input for Console mode.
- Access to history: entered commands for Console mode and parameter value for Visual mode.
- Convenient work with Enums in Visual mode.
- Easy work with Arrays of different types.
- Possibility to group commands.
- Access to 10 recent commands.
- Possibility to gather favorite commands in one place.
- Possibility of creating group foldouts
- Plain and understandable design for anyone.
- Possibility to search, look through and send working logs with stack trace.

~1.1 Requirements for UnityEditor

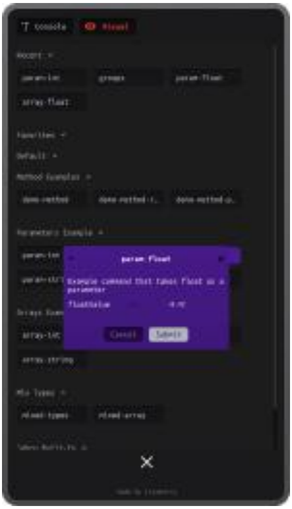
The following Unity 3D software versions make business with Jahro: 2019.3.0f5 and higher.

~1.2 Requirements for Mobile version

Specially designed for the mobile app debugging process, Jahro brings all its capabilities with the application version on a mobile device.

Jahro is compatible with the following software:

- Android v.5 and higher;
- iOS v.10 and higher.



Example command that takes float as a parameter

~2. Plugin Management How-Tos

Note: Mobile version is available in builds by default and can be launched right away - check section 2.3 Launch. There is no way to get rid of it in any case as Deactivation and Removal are not possible for the Mobile version.

~2.1 Installation

The plugin installation process is comfortable and simple as :

1. Open **Unity Editor.**
2. Go to **Window menu** -> **Package manager**.
3. Select **My Assets** from the package scope drop-down menu.
4. Find the Jahro package (or type Jahro in the search field) to install from the list of packages.
5. Once the package information appears in thedetails panel, select the required version from the drop-down menu (or just select the latest one).
6. Click **Download** and **Install** buttons.
7. Once the process is finished, the new package is ready to use.

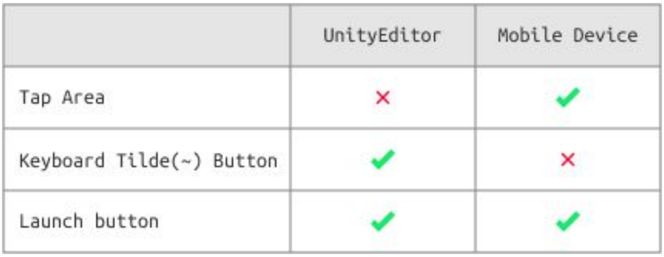
More information about package installation is available at Unity official docs:
https://docs.unity3d.com/Manual/upm-ui-update.html

~2.2 Activation

Created from an idea to provide an easy to use a tool without any headache and drag-and-drop prefab experience, Jahro Console is enabled right after installation.

~2.3 Launch

Providing a unique user experience Jahro plugin launch is available in several options:



	UnityEditor	Mobile Device
Tap Area	✗	✓
Keyboard Tilde(~) Button	✓	✗
Launch button	✓	✓

Note: It’s possible to use API integration for UnityEditor and create a Call Console in your custom Menu option.

~2.3.1 Launch Tap Area

Using a mobile device, Jahro can be easily launched by fast four-taps in the upper ¼ area of the screen.

Note: If your taps are not fast enough and it drives you crazy, this option can be disabled in Jahro settings. Check 5. Settings section for more details.

~2.3.2 Keyboard Tilde Button

The desktop version allows launching Jahro in UnityEditor with pressing ~ on the keyboard

Note: If this hotkey is not exact enough to use, it can be disabled in Jahro settings. Check 5. Settings section for more details.

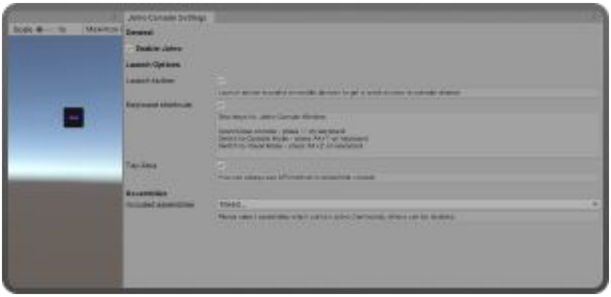
~2.3.3 Launch Tap Area

Note: To be true we are still in doubt about how to call this button properly. If you want to save plugin developers from Holy War and see the next version of this awesome tool, we are begging to send your ideas about the button naming to improve@jahro.io.

For launching the plugin in UnityEditor or in a build it's possible to use the **Lauch** button after adding it in the project's settings.

Steps to add the Jahro Launch Button:

1. Open **Unity Editor**.
2. Go to **Tools menu -> Jahro Settings.**
3. Find **Launch option.**
4. Click **Launch button**.



Jahro Console Settings

General

□ **Disable Jahro**

Launch Options

Launch button ☑

[illegible]

Keyboard shortcuts ☑

[illegible]

Tap Area ☑

[illegible]

Assemblies

Included assemblies [illegible]

[illegible]

This button allows launching in one click. Also,**Launch** button can be moved to any place on the screen using drag-and-drop.

Note: Check 5. Settings section to get more control over Jahro.

~2.3.4 API

Jahro Console namespace provides access to Jahro API. Using Jahro class access code it's possible to call different Static Commands.

Static Properties:
IsOpen - Checks if the console window is open.
Enabled - Checks if console Enabled in ProjectSettings.

Static Commands:
InitView - Calls initialization of the console
ShowConsoleView - Shows main console view.
CloseConsoleView - Closes main console view.
ShowStatusButton - Enables and shows the Launch button.
HideStatusButton - Hides the Launch button.
IsStatusButtonEnabled - Returns status of the Launch button visibility.
Release - Releases and destroys console instances.

Static Events:
OnConsoleShow - Event runs when the console has been shown.
OnConsoleHide - Event runs when the console has been hidden.

~2.4 Deactivation

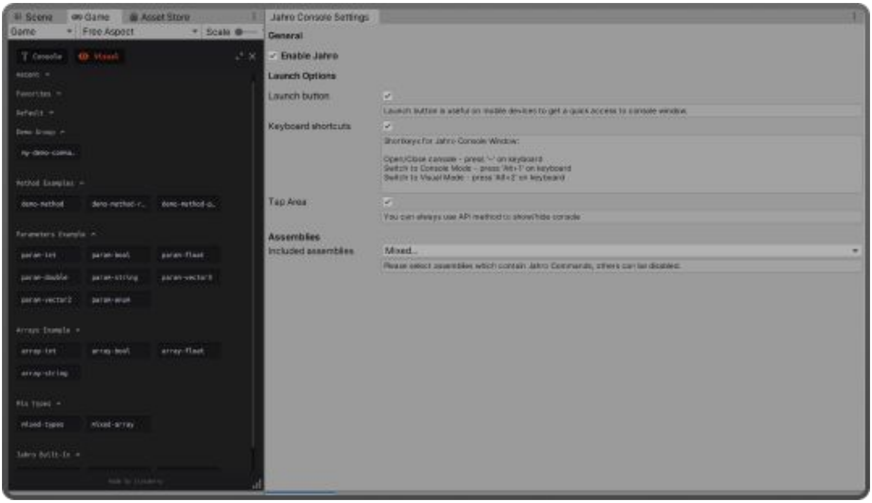
Note: By default, Jahro Console is enabled in Dev and Prod builds. To be careful in delivering builds to non-Jahro experienced users, it can be deactivated from a build.

Taking mom-style care about the project's data, Jahro deactivation will not affect the application code and the performance of a build.

For safe deactivation:

1. Open **Unity Editor.**
2. Go to **Tools menu -> Jahro Settings.**
3. Remove the checkmark next to **Jahro Enabled.**

A disabled built-in package can be **re-enabled** by setting the checkmark next to **Jahro Enabled.** In the case of the plugin's reactivation, Jahro code lines become available in a project again.



Jahro Console Settings

General

☑ Enable Jahro

Launch Options

Launch button ☑

Launch button is useful on mobile devices to get a quick access to console window.

Keyboard shortcuts ☑

Shortkeys for Jahro Console Window:

Open/Close console - press '~' on keyboard
Switch to Console Mode - press 'Alt+1' on keyboard
Switch to Visual Mode - press 'Alt+2' on keyboard

Tap Area ☑

You can always use API method to show/hide console

Assemblies

Included assemblies: Mixed...

Please select assemblies which contain Jahro Commands, others can be disabled.

~2.5 Removal

Note: Removal can bring harm and real troubles to the existing commands and API callbacks. Using Deactivation instead of Removal to save a build’s performance, time, and nerves.

The plugin removal process includes the following steps:

1. Open **Unity Editor.**
2. Go to **Window menu** -> **Package manager.**
3. Select **Jahro console** from the list.
4. Press the **Remove** button in the lower-left corner of the window.

~3. Interface and main elements.

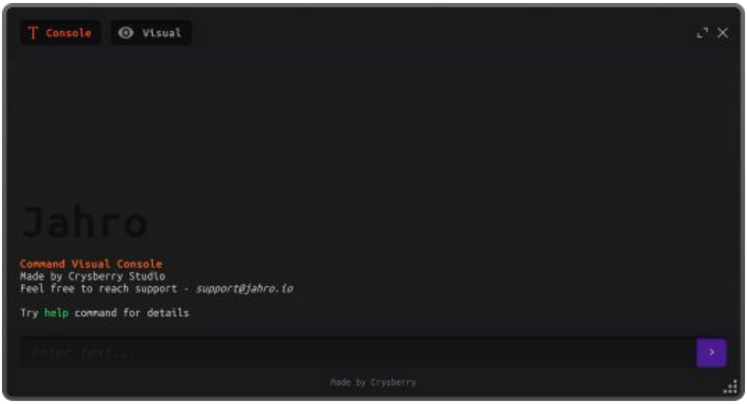
The main elements for **desktop version** in view of the plugin contain:

A - Mode Buttons - allow switching between Visual and Consol modes with Logs indicator.

B - Exit Button - allows closing the window.

C - Maximize Button - helps to return to full screen (not available for Mobile version).

D - Resize Option - allows altering the window dimensions by pulling one edge of the window and setting the required size (not available for Mobile version).



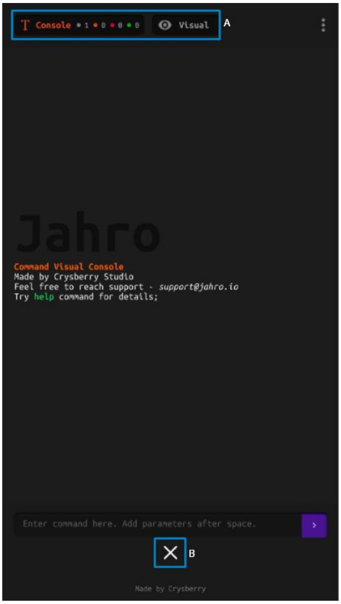
```
Command Visual Console
Made by Crysberry Studio
Feel free to reach support - support@jahro.io

Try help command for details
```

Also, it's possible to change the position of the window by holding down the primary mouse button and then dragging the mouse pointer to the required place on the screen.

The main elements for the **mobile version** are the same, but we made some modifications to improve user experience.

A - Mode Buttons - allow switching between Visual and Consol modes with Logs indicator.

B - Exit Button - allows closing the window.



```
Command Visual Console
Made by Crysberry Studio
Feel free to reach support - support@jahro.io
Try help command for details;
```

~3.1 Modes

Despite being created as a console window, Jahro plugin provides a comfortable possibility to enter parameters both in the old school console form and in the visual one.

Mode switching is available by:

- using **Visual or Console** buttons in the upper right part of the screen,
- or pressing **+** (for Visual view) and **+** (for Console view).

~3.1.1 Console mode

The Console view of the plugin in the **desktop version** consists of:

A - Search bar - helps to locate information in all logs, and highlights in the result output.

B - Logs filtering buttons - allow turning on/off log type.

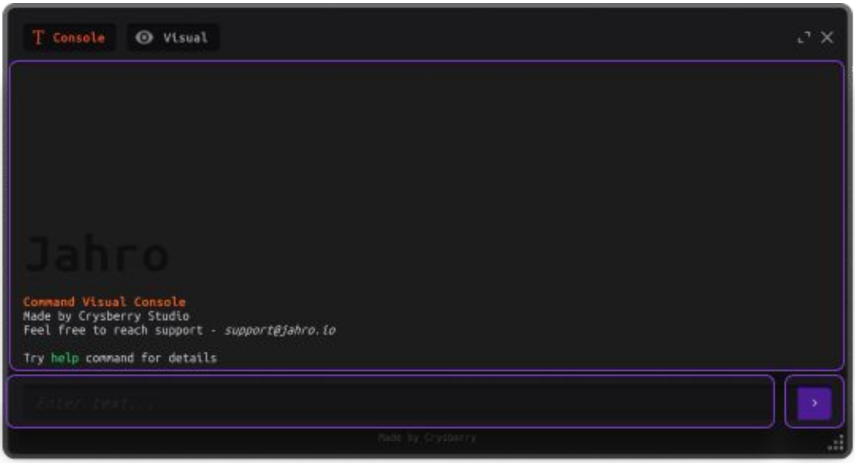
C - Command line - a place to enter a needed command.

D - Run Button - allows executing an entered command.

E - Scroll down button - allows look through the text on the screen.

F - Execution Results Output Area - shows the results of entered commands.

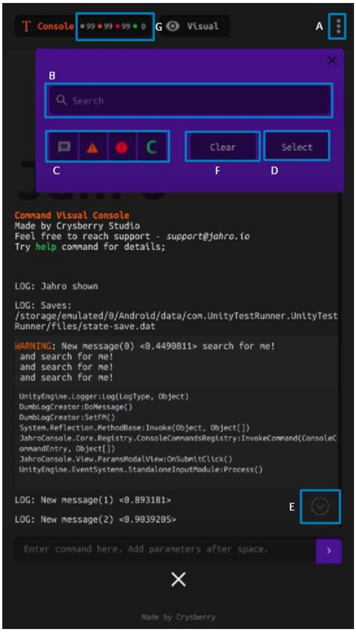
G - Select Button - allows choosing several logs for sharing or making a copy.
H - Copy button - allows copying the selected log.
I - Share Button - allows sharing the selected log.
J - Log details button - opens log details with stack trace.
K - Clear Execution Results Output Area - delete all logs and messages
L - Logs counter - indicates how many logs you have



```
Command Visual Console
Made by Crysberry Studio
Feel free to reach support - support@jahro.io

Try help command for details
```

Interface of the **mobile version** in the Console view has the following elements”

A - **Menu button** - opens options for filtration, search, and allows selecting multiple logs.
B - **Search bar** - helps to locate information in all logs, and highlights in the result output.
C - **Logs filtering buttons** - allow turning on/off log type.
D - **Select button** - allows choosing several logs for sharing or making a copy.
E - **Scroll down button** - allows looking through the text on the screen.
F - **Clear Execution Results Output Area** - delete all logs and messages
G - **Logs counter** - indicates how many logs you have



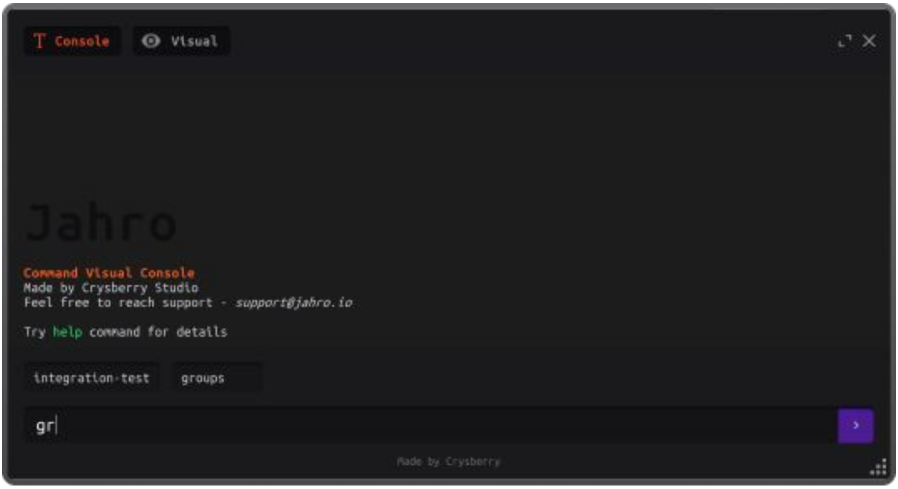
```
Command Visual Console
Made by Crysberry Studio
Feel free to reach support - support@jahro.io
Try help command for details;


LOG: Jahro shown

LOG: Saves:
/storage/emulated/0/Android/data/com.UnityTestRunner.UnityTest
Runner/files/state-save.dat

WARNING: New message(0) <0.4490811> search for me!
 and search for me!
 and search for me!
 and search for me!

 UnityEngine.Logger:Log(LogType, Object)
 DumbLogCreator:DoMessage()
 DumbLogCreator:SetFM()
 System.Reflection.MethodBase:Invoke(Object, Object[])
 JahroConsole.Core.Registry.ConsoleCommandsRegistry:InvokeCommand(ConsoleC
 ommandEntry, Object[])
 JahroConsole.View.ParamsModalView:OnSubmitClick()
 UnityEngine.EventSystems.StandaloneInputModule:Process()

LOG: New message(1) <0.893181>

LOG: New message(2) <0.9039205>
```

~3.1.1.1 Navigation between entered commands

All commands input in Console mode are saved and available for reuse. The last command entered will automatically fill in the command line by pressing and for looking through the list. It's possible to access previously entered commands using the and further.

~3.1.1.2 Autocomplete

Autocomplete is a functionality that helps to select a command from the list while entering the first letters. It allows easily choosing a command from the proposed ones and then clicking the **Run** button to execute it.



```
Command Visual Console
Made by Crysberry Studio
Feel free to reach support - support@jahro.io

Try help command for details
```

```
gr
```

~3.1.2 Logs

Console mode shows errors, warnings and other messages generated by Unity 3D and Jahro. All logs from Unity 3D are duplicated to Jahro.
It is possible to filter logs and console output using filtering system

~3.1.2.1 Navigation between logs

It’s possible to look through, copy, and share a separate log or several ones.
For a separate log hover the mouse over the log (press and hold the log for mobile version) to see 3 action buttons next to it in the right side of the screen:
For several logs use the Select button to open the list of logs. Action buttons can be found at the bottom of the screen (right side of the screen for mobile version).
Also, log search is available via the Search bar.

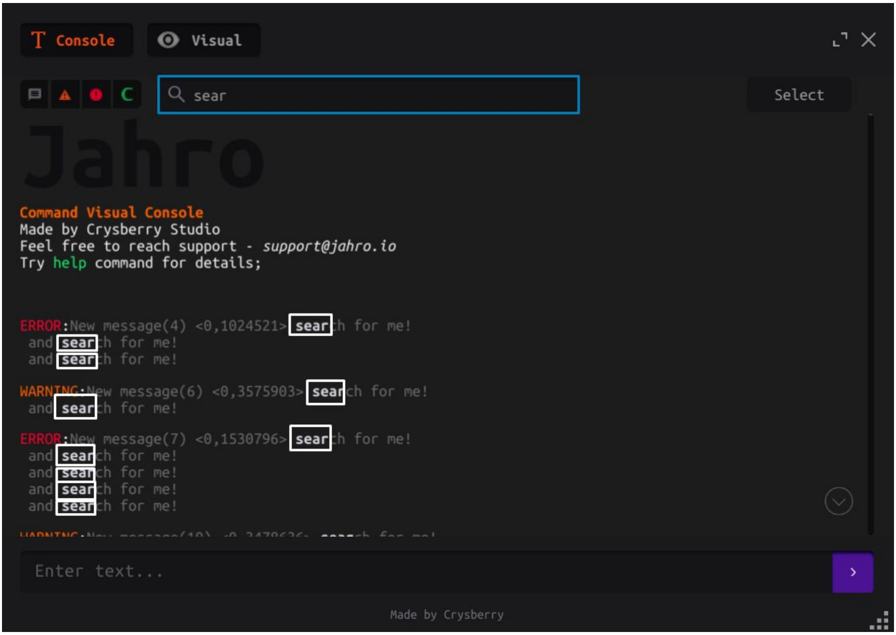
~3.1.2.1.1 Filtering and Searching through Console output

It’s possible to search through Console messages using specific keywords in the Jahro search bar. Everything that appears in Console messages can be used as a searching parameter, including numbers and specific characters. For example, it’s possible to search for the time a message was logged or a keyword contained in the message.

After typing a searching parameter the Console filters messages to display only those that match an entered text.

The search process looks for exact matches of whatever is typed in the search bar. It's not possible to search for two different terms at once or use common search engine operators.

Also, the Console highlights all searching matches in the message text in the visible part of the message. Subsequent matches will be highlighted in the same message while scrolling through.



```
Command Visual Console
Made by Crysberry Studio
Feel free to reach support - support@jahro.io
Try help command for details;


ERROR:New message(4) <0,1024521> search for me!
 and search for me!
 and search for me!

WARNING:New message(6) <0,3575903> search for me!
 and search for me!

ERROR:New message(7) <0,1530796> search for me!
 and search for me!
 and search for me!
 and search for me!
 and search for me!

WARNING:New message(10) <0,3478636> search for me!
```

All the matches for the searching parameter "sear" are highlighted in each message (white).

To filter Console messages by type click the buttons next to the search bar:

▤ **Click to show/hide messages log**

⚠ **Click to show/hide warning log**

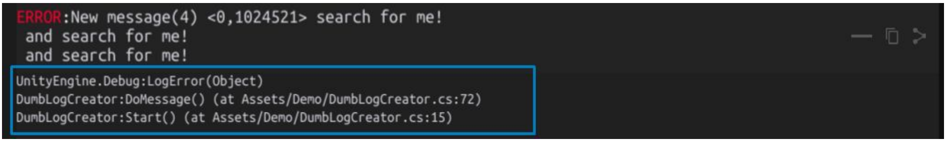
● **Click to show/hide errors log**

C **Click to show/hide Jahro commands log**

~3.1.2.1.2 Logs Stack trace

Jahro includes detailed stack trace information when it prints messages to the Console. This is useful when an error message is not very clear, for example. In that case, the stack trace helps to determine the source of the error.

To see these details click + next to the log to expand it in full:



```
ERROR:New message(4) <0,1024521> search for me!
 and search for me!
 and search for me!
UnityEngine.Debug:LogError(Object)
DumbLogCreator:DoMessage() (at Assets/Demo/DumbLogCreator.cs:72)
DumbLogCreator:Start() (at Assets/Demo/DumbLogCreator.cs:15)
```

~3.1.2.1.3 Actions with logs: Sharing, Copying, Checking details.

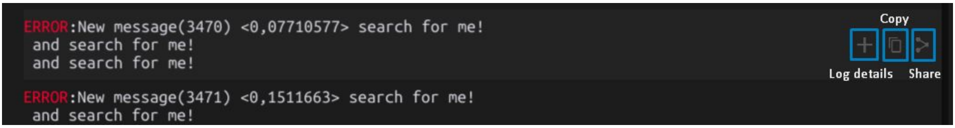
For Desktop version:

To share or copy a separate log:

1. Select a log.
2. Use one of the buttons displayed on the screen for the required action: Log details, Copy, and Share:

To copy or share several logs:

1. Press the Select button on the right upper corner.
2. Choose several or all logs.
3. Press Copy or Share buttons according to the required action.



```
ERROR:New message(3470) <0,07710577> search for me!
 and search for me!
 and search for me!

ERROR:New message(3471) <0,1511663> search for me!
 and search for me!
```

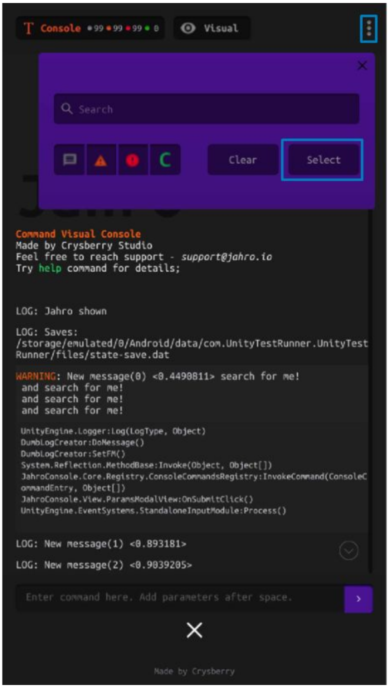
For Mobile version:

To copy or share logs a separate log:

1. Select a log.
2. Press and hold on the log to launch Copy and Share Buttons.

To copy or share several logs:

1. Use the Menu button in the upper right corner of the screen.
2. Select the needed logs.
3. Press Copy or Share buttons according to the required action.

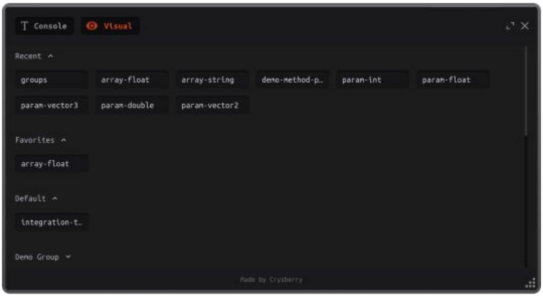


```
Command Visual Console
Made by Crysberry Studio
Feel free to reach support - support@jahro.io
Try help command for details;

LOG: Jahro shown

LOG: Saves:
/storage/emulated/0/Android/data/com.UnityTestRunner.UnityTest
Runner/files/state-save.dat

WARNING: New message(0) <0.4490811> search for me!
 and search for me!
 and search for me!
 and search for me!
 UnityEngine.Logger:Log(LogType, Object)
 DumbLogCreator:DoMessage()
 DumbLogCreator:SetFM()
 System.Reflection.MethodBase:Invoke(Object, Object[])
 JahroConsole.Core.Registry.ConsoleCommandsRegistry:InvokeCommand(ConsoleC
 ommandEntry, Object[])
 JahroConsole.View.ParamsModalView:OnSubmitClick()
 UnityEngine.EventSystems.StandaloneInputModule:Process()

LOG: New message(1) <0.893181>

LOG: New message(2) <0.9039205>
```

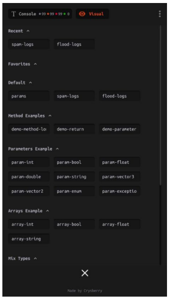
~3.1.3 Visual mode

Created to simplify experience on mobile platforms, the visual mode allows executing any command available in a Console view. It also supports parameter input for each command.
The interface shows groups, which can be easily created and customized by a user.

Note: Check 3.1.2.2 Grouping section for more details.



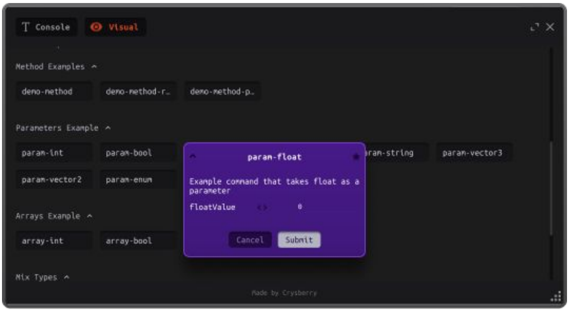
Native version



Mobile version

~3.1.3.1 Parameter Input view

Visual mode allows entering parameters for any command using the parameter input window. It becomes available after clicking on the command.

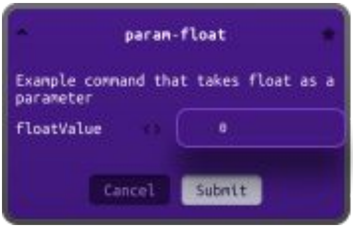


Example command that takes float as a parameter

floatValue 0

It is possible to enter values in two ways:

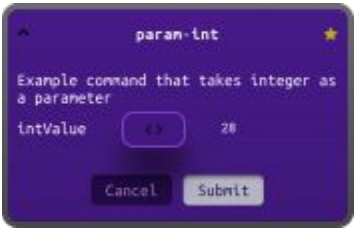
- **Keyboard input** - simply enter the required value using the keyboard;
- **Drag input:**
  - Click **<>** button and use the left arrow to increase or the right arrow to decrease the value.
  - Move a slider to change an array size.



param-float

Example command that takes float as a parameter

floatValue: 0



param-int

Example command that takes integer as a parameter

intValue: 28

The command will be executed automatically after entering parameters and clicking **Submit.**
Switching to **Console mode** will allow checking output of a command executed in **Visual mode.**

Note: The values entered in the parameter input window are saved. So next time this window is launched, the previous values are available for the adjustment.

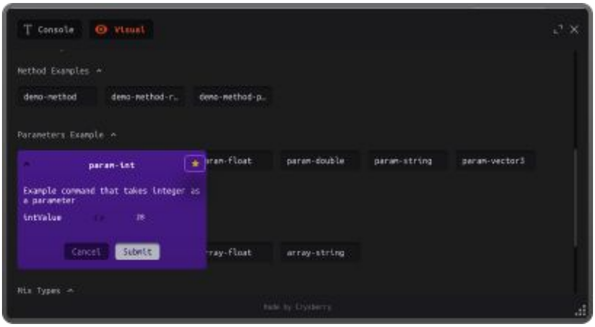
Check 5. Settings section to get more control over parameters.

~3.1.3.2 Grouping

Jahro provides a convenient option to systematize commands – grouping. It’s possible to configure groups and add/remove commands within them. By default, Jahro offers the following groups:

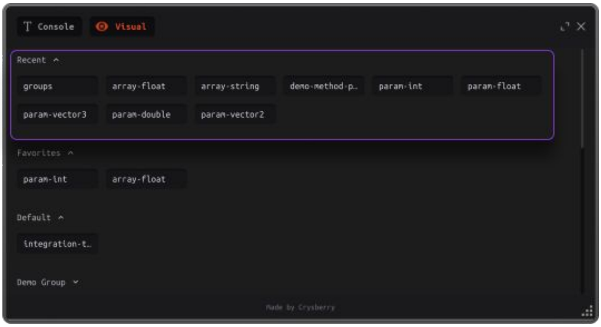
- **Favorites** – a group for quick access to the most used commands:
  - To add a command to Favorites, click on the star icon.
  - To remove a method from Favorites, just deselect it.

Note: A command added to Favorites is still available in any other group it’s added to.



Example command that takes integer as a parameter

- **Recent** – a group, which stores up to 10 commands used recently.



Any user’s preferences can be applied for group creation and configuration.

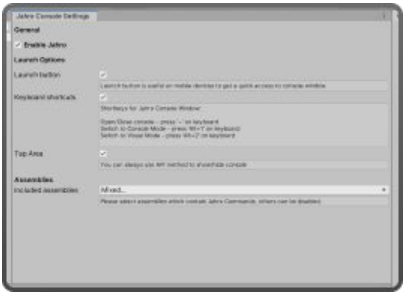
~4. Settings

Note: Settings menu is not available in Mobile version.

Jahro is customization friendly and flexible.

To configure the plugin:

1. Go to the **Tools** menu of the Unity main window.
2. Select **Jahro Setting.**



Jahro Console Settings

General

☑ Enable Jahro

Launch Options

Launch button ☑

Launch button is useful on mobile devices to get a quick access to console window

Keyboard shortcuts ☑

Shortkeys for Jahro Console Window

Open/Close console - press '~' on keyboard
Switch to Console Mode - press 'Alt+1' on keyboard
Switch to Visual Mode - press 'Alt+2' on keyboard

Tap Area ☑

You can always use API method to show/hide console

Assemblies

Included assemblies Mixed...

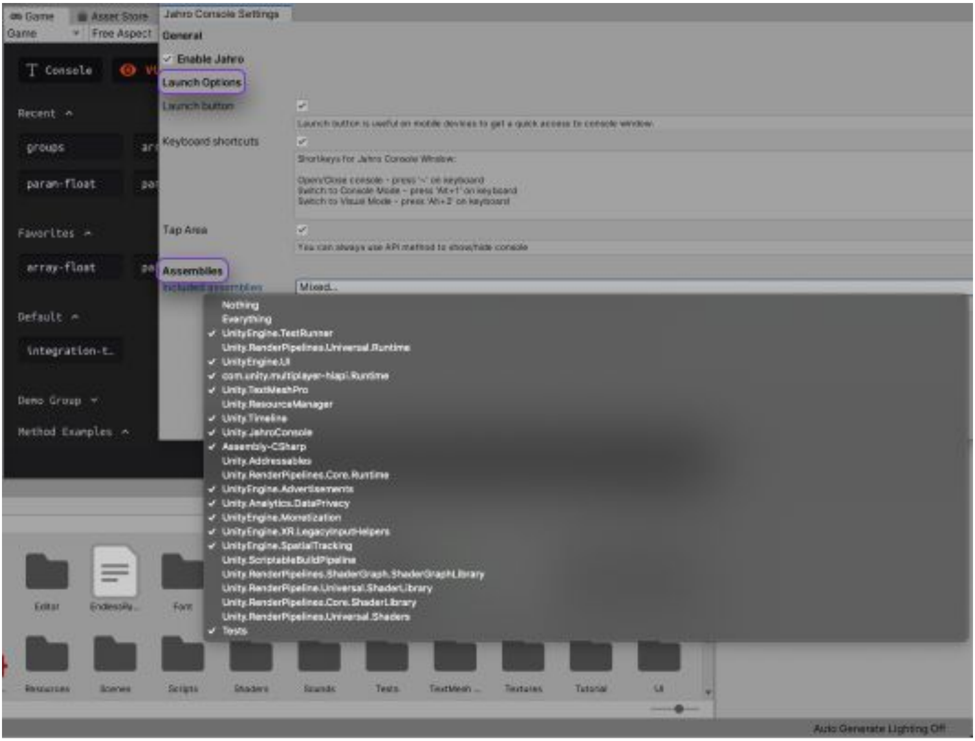
Please select assemblies which contain Jahro Commands, others can be disabled

General settings include:

- **Enable Jahro** - adding or removing the checkmark next to this option Activates/Deactivates Jahro console.
- **Launch Button** - adding or removing the checkmark next to this option enables/disables Launch Button.
- **Keyboard shortcuts** - adding or removing the checkmark next to this option allows using some shortcuts for managing console.
  Note: It's not possible to add custom shortcuts at the moment. Stay tuned for next releases
- **Tap Area** - adding or removing the checkmark next to this option allows using fast four taps launch on Mobile version.

Assemblies section allows selecting assemblies that contain Jahro.

Assemblies section allows selecting assemblies that contain Jahro.



Jahro Console Settings

General

✓ Enable Jahro

Launch Options

Launch button ✓

Launch button is useful on mobile devices to get a quick access to console window.

Keyboard shortcuts ✓

Shortkeys for Jahro Console Window:

Open/Close console - press '~' on keyboard
Switch to Console Mode - press 'Alt+1' on keyboard
Switch to Visual Mode - press 'Alt+2' on keyboard

Tap Area ✓

You can always use API method to show/hide console

Assemblies

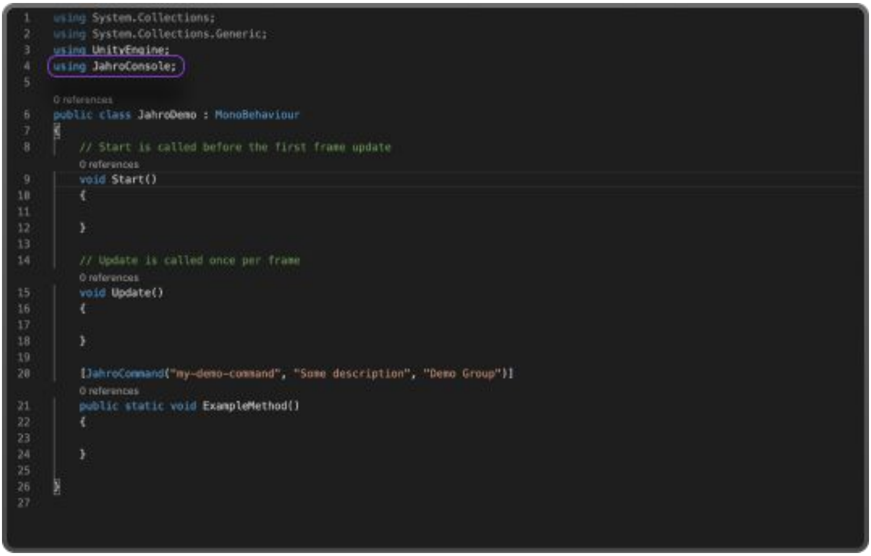
Included assemblies: Mixed...

Nothing
Everything
✓ UnityEngine.TestRunner
Unity.RenderPipelines.Universal.Runtime
✓ UnityEngine.UI
✓ com.unity.multiplayer-hlapi.Runtime
✓ Unity.TextMeshPro
Unity.ResourceManager
✓ Unity.Timeline
✓ Unity.JahroConsole
✓ Assembly-CSharp
Unity.Addressables
Unity.RenderPipelines.Core.Runtime
✓ UnityEngine.Advertisements
✓ Unity.Analytics.DataPrivacy
✓ UnityEngine.Monetization
✓ UnityEngine.XR.LegacyInputHelpers
✓ UnityEngine.SpatialTracking
Unity.ScriptableBuildPipeline
Unity.RenderPipelines.ShaderGraph.ShaderGraphLibrary
Unity.RenderPipeline.Universal.ShaderLibrary
Unity.RenderPipelines.Core.ShaderLibrary
Unity.RenderPipelines.Universal.Shaders
✓ Tests

~5. Commands

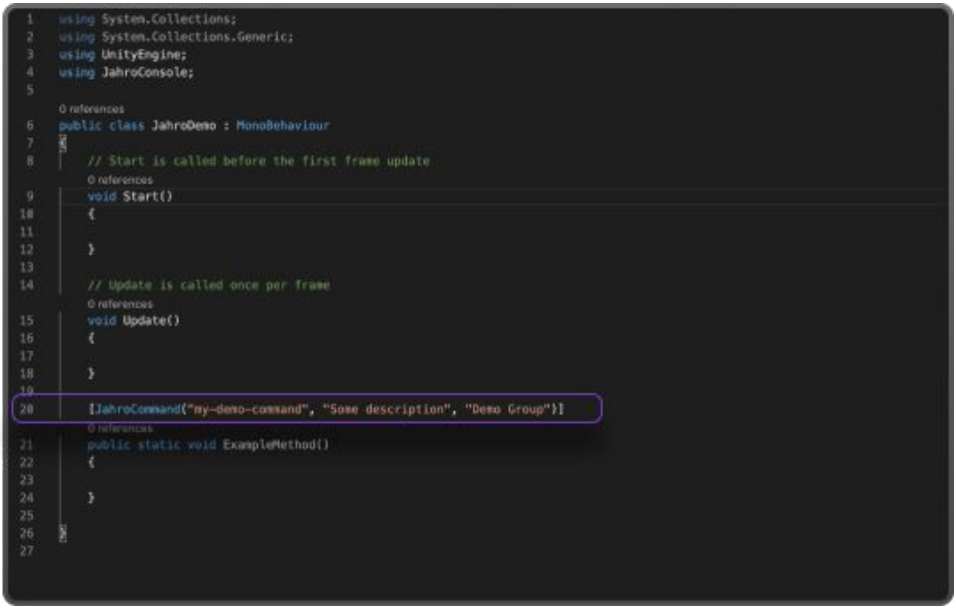
It’s possible to create custom commands for Jahro Console or apply the existing ones.

1. Open C# script you use for commands creation in Project.
2. Add JahroConsole namespace *“**using JahroConsole**”* to connect it to Jahro.



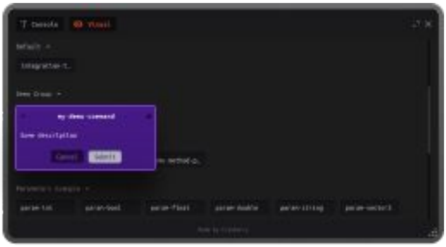
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using JahroConsole;

0 references
public class JahroDemo : MonoBehaviour
{
    // Start is called before the first frame update
    0 references
    void Start()
    {

    }

    // Update is called once per frame
    0 references
    void Update()
    {

    }

    [JahroCommand("my-demo-command", "Some description", "Demo Group")]
    0 references
    public static void ExampleMethod()
    {

    }
}
```

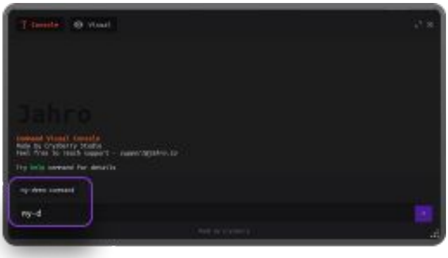
4. Add the required command to the document using the following attributes:
   - **JahroCommand** - Used to mark a command that will be used as a Command in Console and Visual mode.
   - **methodName** - Used to identify the command.
   - **methodDescription** - Adds a description to a command.
   - **groupName** - Assigns a command to a group it should be related to.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using JahroConsole;

0 references
public class JahroDemo : MonoBehaviour
{
    // Start is called before the first frame update
    0 references
    void Start()
    {

    }

    // Update is called once per frame
    0 references
    void Update()
    {

    }

    [JahroCommand("my-demo-command", "Some description", "Demo Group")]
    0 references
    public static void ExampleMethod()
    {

    }
}
```

4. Once a command is added, it will be possible to use it in both Console and Visual modes.





Supported parameters:

- int
- bool
- string
- float
- double
- Vector2
- Vector3
- enum
- and arrays:
  - int[]
  - float[]
  - bool[]
  - string[]
  - float[]
  - double[]
  - Vector3[]
  - Vector2[]

...and any combination of primitive types described above.

Supported returned valuei:

- void
- string

~5.1 “Help” command

It’s possible to use an internal helper added to Jahro - **Help** command:

```
"Help is on the way bro:
Create a public static or private static method anywhere
in your project.
Put an attribute for this method - [JahroCommand("command
name", "description", "group name")]
Use string returning value to output some info or just
use to void return.
What parameters can you use to those methods?
Supported parameters are:
- int
- float
- bool
- string
- Vector3
- array of types described above
- Enums
Use as many parameters as you can deal with!
Don't forget to check Tools -> Jahro Console section to
get more control over the console.”
```