Welcome to Evryway ADB Control!

Evryway ADB Control allows you to control your android devices from inside the Unity Editor.

- Start, pause and stop your android app.
- Inspect the logcat output directly in the editor, with customisable filters.
- Switch between multiple devices, or enable TCP wireless connections with a button press.
- Take screenshots and record video from the device.
- Extend with your own custom build pipeline functions, and easily write your own ADB utilities.

First steps

Install the package from the Asset Store.

If you wish, you can move the ADBControl directory anywhere in your project.

You should have a "Tools/Evryway/ADB Control" menu option - select it, and the "ADB Control" window will appear.

If you are using an older version of Unity, please ensure you have enabled ".NET 4.x Equivalent" for your Scripting Runtime Version in Player Settings.

ADB Control Window

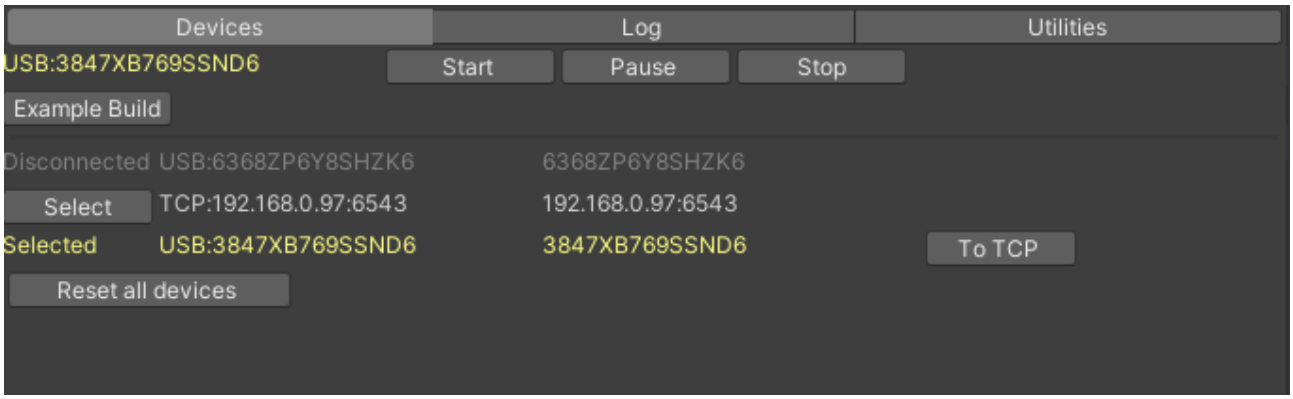
You'll see three panes in the window.

At the top of each pane, you will be able to see the currently selected device, as well as the common control buttons. To select a specific pane, click on the name.

Common Controls

By default, the common controls will contain Start, Pause and Stop buttons, which allow you to control your application on your selected device. The application which is controlled by these on your android device is specified by the Android player package name in your Unity project settings. Simply change the package name if you wish to control a different application. If you wish to control more than one application, or you wish to control the application on multiple devices, you can create your own controls to achieve this.

Devices



Here you can:

- See a list of available devices
- Select your current device from the list
- Change a device from a USB connection to a TCP connection
- Reset all available devices

All devices that have been connected are stored in the list. The connections are cached in a file in your unity project at:
Library/Evryway/AdbControl/adb_control_devices.json.

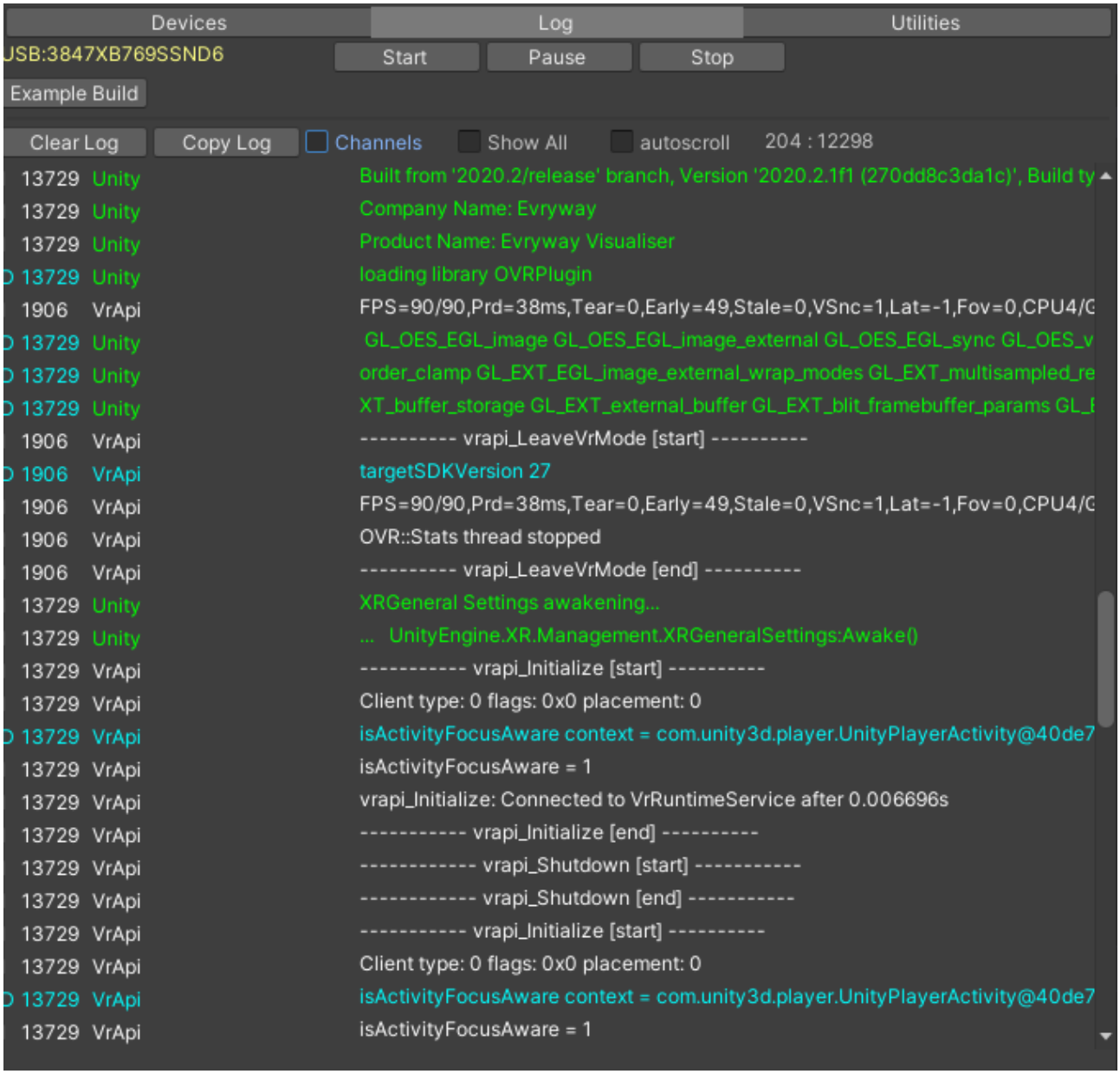
When a device is connected, you are able to select it. The currently selected device will be shown in yellow - any commands that require a specific device will target the selected device (e.g. logcat, run and stop, etc).

Sometimes it's helpful for a device to be connected in TCP (wireless) mode, rather than wired via a USB cable. When a device is connected with USB, you can select "To TCP" and the device will be reconnected as a TCP device. You can then disconnect the USB cable and still access the device via ADB over the TCP connection.

If the device shows “Unauthorized” then you will need to authorize your computer on the device. Check the device screen for the authorization prompt, and grant your computer permission to connect to the device.

Any commands that act on a specific device will use the currently selected device.

Log



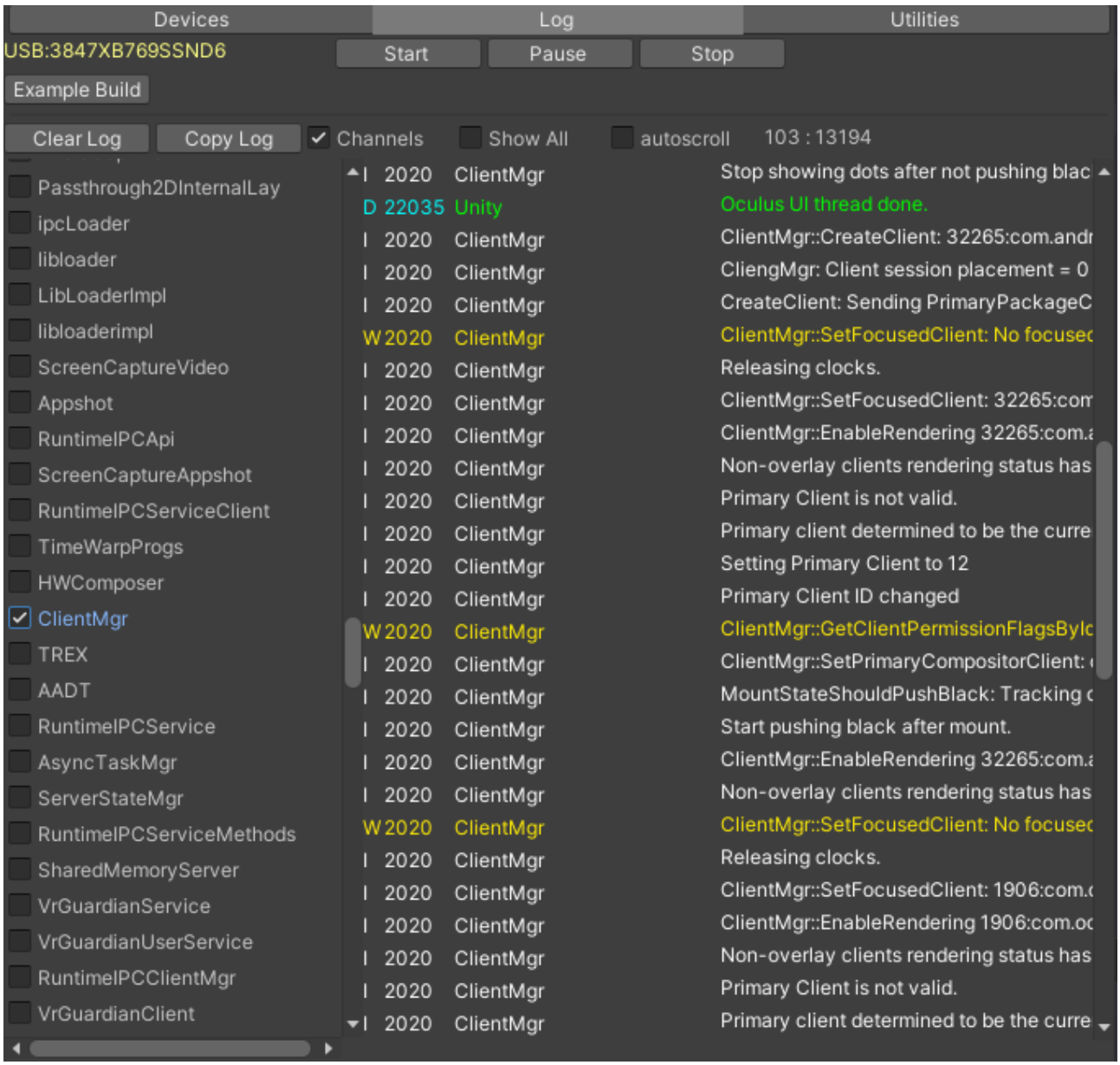
Here you can:

1. View the logcat output for the currently selected device
2. Copy and reset the log output
3. Select active channels for the filter

The Log window gives you a via of the logcat logs for the currently selected device. By default, the "Unity" log channel will always be displayed, as will the "ADBControl" log channel. Selecting "Channels" will give you a list of channels which can be toggled on and off.

selecting "Show All" lets you see all output, unfiltered.

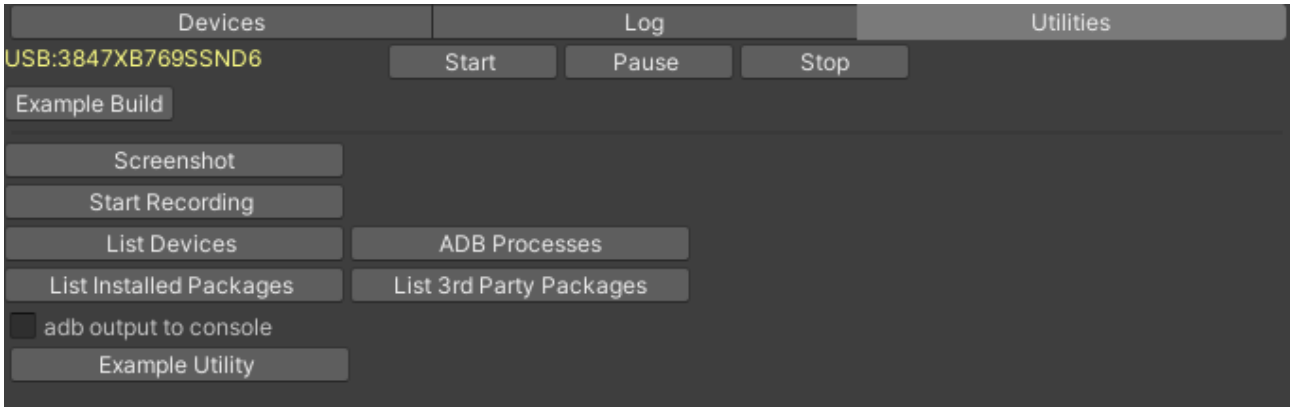
Normally, you want the log to track changes as they come in – select “autoscroll” to enable this, and deselect it if you want the window not to update as changes happen. Dragging the scroll bar to the bottom will automatically enable autoscroll.



Each log line can be selected and copied. To deselect, click in a different editor pane.

If you wish to work with multiple log lines, select “Copy Log” and then paste into your preferred editor.

Utilities



Here you can:

- Toggle the ADB command output in the editor console
- Take a screenshot, or record video
- Assign your own custom utility functions to a button

If an ADB command does not function correctly, you can enable command output to the console to help diagnose the issue.

Screenshots

Pressing "Screenshot" will capture a screenshot from the currently active device.

Video

Pressing “Start Recording” will start an ADB video capture process. Pressing “Stop recording” will stop that process, and copy the file locally.

Built-in utilities

There are a set of useful utilities provided. Here you can list all connected devices, show any running ADB processes, list all installed packages, and list installed 3rd-party packages.

Custom Utility command buttons

To assign a custom utility command, see the "How to configure ADB Control" section for more details.

How to configure ADB Control

take a look at Editor/ADBCConfigExample.cs. You'll see examples of how you can:
1. Set an override path to adb.exe
2. Provide a build function
3. Provide a utility function

Please ensure you place your version of this class in an editor assembly - either under an Editor directory in your project,
or controlled by an editor-only assembly definition file.
Your config class must derive from Evryway.ADBC.ADBCConfig. In the example script, you'll see the using Evryway.ADBC line provides the namespace, and the class inherits ADBCConfig.

Set an override path to adb.exe

In your Setup function, call

```
ADBControl.SetExePathOverride("z:/path/to/adb.exe");
```

ensure that adb.exe exists at the relevant path.

Set the Screenshots and Video output directories

You can change the directory screenshots are saved into by calling ADBControl.SetScreenshotPath(..). the default is "../Screenshots". Paths are relative to Application.dataPath.

You can change the directory videos are saved into by calling ADBControl.SetVideoPath(..). The default is "../Videos". Paths are relative to Application.dataPath.

Provide a common function

In your Setup function, call

```
window.AddCommonCommand("Example funcion", SomeFunction );
```

Here, you will get a button saying "Example function" on your build tab, which calls the function SomeFunction.
If necessary, you can use a lambda to provide parameters, e.g.

```
window.AddCommonCommand("Example build", () =>
BuildMyApp("Standard"));
```

Provide a utility function

Similarly, in your Setup function, call

```
window.AddUtilityCommand("Example Utility", UtilityFunction);
```

This will provide a button titled "Example Utility" on your utility pane, which calls UtilityFunction when pressed.

Filter log output

If you wish to explicitly filter certain log output (for example, your application is spamming the log with a specific line that you wish to hide) you can call

```
LogLine.AddFilter(exact_string);
```

and any matches of the exact string will be removed from the log output.

Running ADB Commands from Editor scripts

Ensure you are using the Evryway.ADBC namespace.

Pass your ADB commandline arguments in a string list to ADBControl.Run.

```
var ok = ADBControl.Run(new List<string>{ "devices" }, out string output, out string error);
```

any non-zero process exitcode will return false.

The default timeout for a command is 5 seconds. You can pass an additional float value to increase this if required.

You can also run asynchronous ADB processes. This is much more complex to do correctly – please get in touch for details!

Feedback and support

We'd love to hear your feedback (good and bad!) for ADB Control. If you have any problems, or great ideas for how to improve
ADB Control, please get in touch:

* email support@evryway.com
* twitter @evryway
* https://evryway.com

Evryway ADB Control Version 1.2.4, 20211106

Changelog:

1.2.4 20211106	Fixed another issue with ADB server process hanging Unity on exit. ADBControl now correctly respects the currently selected ADB path in Preferences/External Tools.
1.2.3 20211103	Fixed 2021.2 issue with forked ADB server process hanging Unity on exit.
1.2.2 20210616	Fixed Clear Log lockup when no device attached.
1.2.1 20210405	Added filters to log lines. Fixed Copy Log button.
1.2.0 20210208	Removed Builds pane, made build buttons common.
1.1.0 20210120	Added video recording. Various bug fixes.
1.0.2 20210114	Added support for unauthorized devices
1.0.1	Bugfixes, unity version support down to 2018.2
1.0.0	First Release