PLATFORM INFO

View online: https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/platform-info

The **Platform info** object returns information about the system, device and browser.

Scripting

When using JavaScript or TypeScript coding, the features of this object can be accessed via the IPlatformInfo script interface, typically accessed via the `runtime.platformInfo` property. Note that this interface is always available in script regardless of whether the Platform Info object has been added to the project.

Platform info conditions

Is Cordova export

True when running after a Cordova-based export. Cordova is used to export Android and iOS apps, and implies that the platform is mobile and the OS is either Android or iOS.

Is Linux wrapper export

True when running after a Linux wrapper export, and implies the platform is desktop and the OS is Linux.

Is macOS wrapper export

True when running after a macOS wrapper export, and implies the platform is desktop and the OS is macOS.

Is web export

True when running after a web-based export. This is also true in preview mode, so that any features based on these export conditions will act as if it's in a web-based export in preview mode.

Is Windows wrapper export

True when running after a Windows wrapper export, and implies the platform is desktop and the OS is Windows.

On network change

Triggered when the network connection changes, e.g. when moving from a Wifi network to a cellular data network. The network-related expressions will update in this trigger to reflect values for the new network.

Is on Android

Is on Chrome OS

Is on iOS

Is on Linux

Is on macOS

Is on Windows

These conditions check which operating system (OS) the project is currently running on. For example in Chrome on Windows, *Is on Windows* will be true; in an Android app, *Is on Android* will be true; and so on. Note these conditions are not exhaustive: there may be platforms where none of these six conditions are true.

macOS was formerly known as OS X.

Is on mobile

Test if the current device reports itself as a "mobile" device.

This condition may not work like you expect. Many devices lie about their device class - for example modern iPads tell websites they are macOS desktop devices in order to get the desktop experience. Further it is probably unwise to rely on this detection anyway, as there is no clear definition of "mobile device". For example tablets can be used with the touchscreen only and act like a mobile device, or have a keyboard and/or mouse attached and be used like a desktop. It would also be wrong to use this condition to check if the device will use touch input (see the detecting input method for a better approach). This condition just reports whatever the device indicates to web content, which may or may not be what you want.

Is wake lock active

True if a *Request wake lock* action has successfully completed and the screen is currently being kept on.

Is wake lock supported

True if the current browser/platform supports using wake locks to keep the screen on.

On wake lock acquired

Triggered after the *Request wake lock* action if the lock was successfully acquired. The screen will be kept on until *On wake lock released* triggers.

On wake lock error

Triggered if an error occurs while attempting to request a wake lock. The screen will not be kept on.

On wake lock released

Triggered when an existing wake lock is released, meaning the screen may once again automatically turn off due to inactivity. This can happen after a *Release wake lock* action, or automatically in some circumstances, such as if the browser tab or app goes in to the background. To keep the screen on after this trigger, another wake lock must be requested.

Platform info actions

Request wake lock

Request a wake lock to keep the screen on permanently, preventing it automatically dimming or switching off due to inactivity. Note this increases power usage so will drain battery more quickly. If successful, *On wake lock acquired* triggers; otherwise *On wake lock error* triggers.

> *A wake lock might only be granted in a user input trigger, such as On touch end.*

> *The wake lock could be released at any time after being acquired. Use the On wake lock released trigger to identify if the wake lock ended.*

Release wake lock

If a wake lock is currently active, release it, so the screen may once again dim or turn off due to inactivity.

Platform info expressions

DeviceMemory

Return the approximate amount of device memory (RAM) in gigabytes. For privacy reasons some platforms will round the result, so it may not match the exact amount installed on the system. Additionally this value is not available on some browsers or platforms, in which case it will return 0.

HardwareConcurrency

Return the number of hardware threads supported by the CPU. This is normally at least the number of CPU cores. Many modern CPUs support multiple hardware threads on a single CPU (e.g. Hyper-threading), and a common case is for each CPU to support two hardware threads, so this is often double the number of CPU cores. For privacy reasons some platforms will round the result, so it may not match the exact number of hardware threads/CPU cores available on the system. If the value is not available, it will return 0. However almost all consumer devices have at least two CPU cores.

CanvasCssWidth

CanvasCssHeight

Return the size of the main display canvas in CSS pixels. This does not correspond exactly to device (physical display) pixels, but is the appropriate size from a web design perspective in `px` units.

CanvasDeviceWidth

CanvasDeviceHeight

Return the size of the main display canvas in device (physical display) pixels. Unlike the CSS size, this size reflects the number of actual pixels used in the display.

DevicePixelRatio

Return the number of device (physical display) pixels per CSS pixel. For example a high-DPI display may have a device pixel ratio of 2, meaning there are two device pixels per CSS pixel. This value also reflects the browser zoom level, which works by adjusting the device pixel ratio. The concept of the device pixel ratio also allows web pages using `px` units to appear the same size on higher density displays where physical pixels are much smaller.

Renderer

Return a string indicating the graphics rendering technology in use. This can be `"webgl1"`, `"webgl2"` or `"webgpu"`. It can also have a `-software` suffix if a "major performance caveat" is detected, which normally indicates slower software rendering, typically due to unreliable GPU drivers. For example, software-rendered WebGL 1 support would return `"webgl1-software"`.

RendererDetail

Return a string describing system-specific detail about the graphics hardware in use. This typically describes the GPU manufacturer and model name, and sometimes some hardware capabilities.

SafeAreaInsetLeft

SafeAreaInsetTop

SafeAreaInsetRight

SafeAreaInsetBottom

The inset around the edges of the screen in CSS pixels of a rectangular area that is always visible (hence safe to use for displaying anything important without it risking being cut off). This only applies for devices with non-rectangular screens, notably mobile devices with a notch, or in some cases devices with rounded edges in the corners of the screen. Devices with a standard rectangular display will return 0 for these values.

ScreenWidth

ScreenHeight

Get the size of the current display screen. Note this often includes areas not available to applications, such as a desktop taskbar, or mobile status bar.

WindowInnerWidth

WindowInnerHeight

Get the inner size of the current window. This is the size of the window content area that is available to the application.

WindowOuterWidth

WindowOuterHeight

Get the outer size of the current window. This includes the window browser, caption, browser address bar etc. which is not generally available to the application.

ConnectionEffectiveType

Return a string rating the effective type of the connection based on the comparable cellular data connection generation, e.g. `"2g"`, `"3g"`, `"4g"`.

ConnectionRTT

Return the estimated round-trip time (latency) of the connection in milliseconds. This is the time it takes for a network message to be sent to the remote host and a reply to be received back.

ConnectionType

The type of network connection technology in use, e.g. `"cellular"`, `"wifi"` or `"ethernet"`. If the connection type cannot be detected or the platform does not support this feature, returns `"unknown"`.

Downlink

The estimated effective download bandwidth in megabits per second. Returns 0 if unable to detect.

DownlinkMax

The maximum downlink speed in megabits per second of the underlying connection technology. This is normally the theoretical maximum the current network technology's specification allows, for example 4G cellular allows a higher maximum speed than 3G under ideal signal conditions; however the actual available bandwidth will depend on other factors like signal quality and other parts of the network. Returns 0 if unable to detect.

FramesPerSecond

How many frames per second (FPS) the project is rendering. The most common display refresh rate is 60 Hz, so typically an efficiently designed project will render at 60 FPS. Note however if nothing is changing on-screen, then nothing is rendered, and so the FPS measurement may fall to 0 or display a lower result; this does not indicate poor performance, only that fewer frames are necessary to render. The *TicksPerSecond* expression indicates how frequently the engine is stepping, which may be different to the frames rendered per second.

TicksPerSecond

How many ticks per second (TPS) the project is running at. Each tick processes the logic of the game. Usually a new frame is also rendered every tick, but if nothing changes then rendering a frame is skipped; further, depending on the framerate mode, stepping the engine and drawing frames may happen at different rates. Therefore the ticks per second may produce a different measurement to the frames per second. Usually the project will continually tick even if nothing is visually changing, and only stop ticking if the project is suspended, such as by being minimized or going in to the background.