

# IMPORTING CONSTRUCT 2 PROJECTS

View online: <https://www.construct.net/en/make-games/manuals/construct-3/tips-and-guides/importing-c2-projects>

Construct 3 is largely backwards-compatible with Construct 2. You can usually import Construct 2 projects in to Construct 3 and it will open them so you can continue working on them in Construct 3. One easy way to do this is to drag-and-drop a Construct 2 .capx file in to the Construct 3 window. However sometimes you will need to make changes to an imported Construct 2 project to continue developing it in Construct 3.

*Note that Construct 3 saves projects in a different format to Construct 2. While C3 can open C2 projects, C2 can't open C3 projects. Make sure you are ready to move entirely to Construct 3 before making significant changes to an imported Construct 2 project.*

This guide covers some points to be aware of when importing Construct 2 projects to Construct 3.

*Construct 2 was retired in July 2021 and is no longer officially supported. For more information see the blog post [Sunsetting Construct 2](#). However you can still import your C2 projects in to C3.*

## Replaced features

Construct 3 originally inherited all of Construct 2's features. Over the years some key features have been replaced and the old feature retired. These include:

- The entire runtime was rewritten and upgraded to an enhanced, modernized and faster engine, referred to as the **C3 runtime**. This is highly compatible with the old runtime but there are some compatibility differences.
- The Function plugin has been replaced with a **built-in functions** feature
- Construct 3 now exports projects with a proper **"folder" file structure**, preserving subfolders and filename case, instead of the legacy "flat" mode which exported everything in the root folder with a lowercased name.

You may need to update an imported C2 project to take in to account these changes. For more information, see the [deprecated features](#) section. The rest of this guide also includes details about other compatibility differences.

## Compatibility differences

This section covers some changes made either with Construct 3 or its new runtime that may affect how imported Construct 2 projects work.

### Setting effect color parameters

In Construct 2, some effects like *Tint* and *Set color* specified a color parameter as three separate parameters for the red, green and blue components. In Construct 3, these parameters have been replaced by a single color parameter. Construct 3 will show a color picker instead of three number fields for red, green and blue.

Construct 3 should correctly import the right color value. However if you use *Set effect parameter* in your events to change the effect's color, this will need updating. Previously you may have used actions like:

- *Set effect "Tint" parameter 0 to redAmount*
- *Set effect "Tint" parameter 1 to greenAmount*
- *Set effect "Tint" parameter 2 to blueAmount*

In Construct 3 there is now only one parameter instead of three. So this action needs to be replaced by:

- *Set effect "Tint" parameter 0 to rgbEx(redAmount, greenAmount, blueAmount)*

### Physics behavior stepping mode

In Construct 2, the Physics behavior defaulted to *fixed* stepping mode. This meant simulations were deterministic, but also meant it assumed a fixed display rate of 60 Hz. This was a reasonable assumption in the early 2010s, but now devices with varying refresh rates are much more common, such as phones with 90 Hz displays, tablets with 120 Hz displays, and gaming monitors with 144Hz+ displays. On these devices the fixed framerate mode of the Physics behavior appears to run in fast-forward mode.

To avoid this, the Physics behavior in Construct 3 instead defaults to *framerate independent* mode. This ensures it works at the same speed on all displays, but means the simulation is no longer deterministic. If you want to restore a deterministic simulation you can use the *Set stepping mode* action to switch back to *Fixed* mode - but then note you will have the aforementioned problem of the gameplay running too fast on high refresh rate displays.

### Dynamically creating objects

In the C2 runtime, creating an object which is not placed on the layout (and so is not loaded in to memory) immediately loads images for the object, which can "jank" the game (cause a short pause while loading is done). In the C3 runtime, in this case the game continues to run and the images are loaded in parallel. This provides better loading performance and avoids janking the game. However the object can exist for a short period of time while the images are still being

loaded. During this time the object is not drawn, as if it were set to invisible. In some cases this can cause a noticeable flicker, especially if the object is intended to cover up something else. The workaround is to place the object in the layout, and destroy it in *On start of layout* if it is not needed. This ensures Construct loads its images when loading the layout, avoiding any delay when creating it.

## System object changes

The following changes were made to the System object, and system expressions, in the C3 runtime:

- The *WindowWidth* and *WindowHeight* expressions have been removed. These were confusingly named, but if you need the same values, use the Platform Info object's *CanvasDeviceWidth* and *CanvasDeviceHeight* expressions. Usually it's better to use the new *ViewportWidth* and *ViewportHeight* system expressions instead, which return a size in layout pixels rather than device (physical display) pixels.
- The *Is on mobile* and *Is on platform* conditions have been moved to the Platform Info object.
- The *Renderer* and *RendererDetail* expressions have been moved to the Platform Info object.
- The *rgb* expression has been removed. Instead use the new *rgbEx* or *rgba* expressions. Note *rgb* used values in the 0-255 range, but *rgbEx* and *rgba* use values in the 0-100 range, and also support a wider range with better precision.

## User Media object changes

In the C3 runtime, the User Media object's speech synthesis, speech recognition, and canvas recording features have been moved to separate plugins (*Speech Synthesis*, *Speech Recognition* and *Video recorder*). Existing projects using these User Media features can be used in C3, but those features will no longer work until you replace them with the new plugins.

Also note that the *Snapshot* action no longer provides the snapshot immediately: you must use the *On snapshot ready* trigger to know when *SnapshotURL* is available.

## Browser object changes

The following changes have been made for the Browser object in the C3 runtime:

- The battery features have been removed due to lack of browser support.
- The network information and various display related expressions have been moved to the Platform Info object.
- *On suspended* and *On resumed* have been moved to the System object. The *Page is visible* condition has been replaced with the *Is suspended* system condition (which makes the inverse check).

## Third-party addons

If your project uses third-party plugins, behaviors or effects, these need to be available in Construct 3 as well before you can import a Construct 2 project using them. Note that Construct 2 addon files (.c2addon) cannot be used in Construct 3. The addon needs to be updated to work with Construct 3, and distributed as a Construct 3 addon file (.c3addon).

Note that third-party addon developers are independent of Scirra and we cannot provide support for their addons. If you need help with third-party addons you'll need to contact the addon developer.