

# PORTING CONSTRUCT 2 PLUGINS/BEHAVIORS

View online: <https://www.construct.net/en/make-games/manuals/addon-sdk/guide/porting-c2-addons>

To port a Construct 2 plugin or behavior to Construct 3, use the following checklist.

- 1 Start by copying the template SDK to a new folder.
- 2 Update the **addon metadata** in `addon.json`.
- 3 Update the icon. An SVG icon is preferable. The .ico files Construct 2 uses are not supported in Construct 3, but you can use a PNG icon. Just delete `icon.svg`, add `icon.png`, and call `this._info.SetIcon("icon.png", "image/png");` in the plugin/behavior constructor.
- 4 Update the plugin/behavior constants and identifiers in `plugin.js/behavior.js`, `type.js` and `instance.js`, as described in **configuring plugins/configuring behaviors**.
- 5 Match your Construct 2 addon's configuration by making calls to **IPluginInfo/IBehaviorInfo** in the addon constructor. For example if your Construct 2 plugin was a single-global plugin, the Construct 3 plugin should call `this._info.SetIsSingleGlobal(true);` in the plugin constructor.
- 6 Add equivalent properties as the Construct 2 addon has. See *Specifying plugin properties* in **configuring plugins**. (The process is identical for behaviors.)
- 7 Create corresponding action, condition and expression definitions. See **defining actions, conditions and expressions**. The key point to ensure Construct 2 projects using your addon can be imported to Construct 3 is:
  - 8 Give every action, condition and expression a new `id` based on a string
  - 9 Also set the `c2id` property to the corresponding numeric ID that the Construct 2 addon used
- 10 Update **the language file** to contain the UI strings for the addon, properties, and ACEs.
- 11 You'll then need to port the runtime script to the C3 runtime, since Construct 3 introduced an entirely rewritten engine.

Once complete, zip all the addon files and rename the .zip to .c3addon. You should now have a addon you can install via the Addon Manager in Construct 3.