PROJECT FILES

View online: https://www.construct.net/en/make-games/manuals/construct-3/project-primitives/files

Any external files can also be imported to your project via the Project Bar. This is useful for including any other resources your project might need, such as videos, additional images, JSON/XML/CSV/text data, HTML, CSS, documents, or other general files you might need in your project. Often project files are requested in events with the **AJAX** object, allowing data files to be read by the game.

Project files (excluding sounds and music) can be categorised in to folders for **Videos**, **Fonts**, **Icons** (see Icons & splash) and **Files** (for anything that does not belong to one of the other folders). Files can be imported by right-clicking one of the folders in the Project Bar and selecting the **Import** option. Note importing files copies them to the project.

Previewing files

Several kinds of files can be previewed simply by **double-clicking** their item in the Project Bar:

- Like sounds and music, previewing a video file will play it in a video player.
- Fonts preview in a dialog showing some sample text using the font.
- SVG images are previewed in a resizable dialog that displays the image.
- Other images than SVG can be opened in the Animations Editor for viewing and editing. This is particularly useful with icon files.
- Array, dictionary and text-based files can be viewed and edited in the Array editor, Dictionary editor and Text editor respectively Paid plans only.

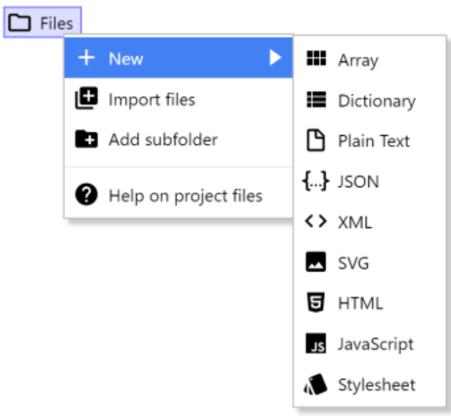
Other text-based files, like HTML and CSS files, can also be edited with a built-in text editor by double-clicking them Paid plans only.

Using Web Fonts

You can use any web font (.woff or .woff2) files imported to the Fonts folder in Text objects. For more information see the section on *Using web fonts* in the Text object.

Adding new files

Paid plans only When right-clicking the general-purpose **Files** folder, there is also a **New** submenu which provides a range of common kinds of file that you might want to add to the project.



Adding a new project file

Adding an **Array** adds a JSON file in a format that can be loaded by the Array object, and opens the Array editor. Adding a **Dictionary** adds a JSON file that can be loaded by the Dictionary object, and opens the Dictionary editor. Each allows the data in the files to be edited in a visual editor. The other files open a Text editor for editing the contents of the file. Once added, **double-click** the file in the Project Bar to re-open its editor.

Adding HTML files

Paid plans only HTML files can be added to the project, and edited with a built-in text editor. This is useful for writing long HTML content to be displayed in the HTML Element object. The content can then be fetched using the AJAX object, and then set as the content of the HTML Element.

HTML project files can also have a *Purpose* set in the Properties Bar. This allows using them to insert content in to the exported HTML file. If the file is used this way, it will no longer be exported as an additional file. The possible purposes are:

- **(none)**: the HTML file will not be automatically used anywhere. It will be exported as an additional file.
- **End of <head>**: the contents of the HTML file will be inserted just before `</head>` in the exported HTML.
- **Start of <body>**: the contents of the HTML file will be inserted just after `<body>` in the exported HTML.
- **End of <body>**: the contents of the HTML file will be inserted just before `</body>` in the exported HTML.

Adding stylesheets

Paid plans only Stylesheets, using the CSS file extension (short for *Cascading Style Sheet*), can be added to the project and edited with a built-in text editor. This is particularly useful for customising the appearance of the HTML Element object.

Custom CSS files are also useful for customising the appearance of other HTML-based objects like Button and Text Input. These objects have *ID* and *Class* properties which can be used to identify them so custom styles can be applied from stylesheets.

> *Construct sets a CSS variable named `--construct-scale` on the root `html` element of the document, with a number representing the canvas scale as a multiplier. You can use this to scale CSS properties to match the displayed canvas size. For example `height: calc(var(--construct-scale) * 2em);` sets a height size of 2em at 100% scale, but also adjusts the height to follow Construct's fullscreen scaling.*

CSS files have a *Purpose* property in the Properties Bar. The default is *Stylesheet* which means Construct will use it as an additional stylesheet for your project. The purpose can also be set to *(none)* in which case its styles will not automatically be applied, and the stylesheet will merely be exported as an additional file on export. This may be useful if the stylesheet is needed for other purposes, such as for the content of an iframe.

MIME types

When publishing to the web, you must ensure the server hosting the project has the correct MIME types set up for all the kinds of project file you use. Otherwise some project files may fail to load, or could be loaded with corrupted data.