FILE CHOOSER

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

The **File chooser** plugin is a form control that allows the user to pick a file on their local system and load it. For example, a text file can be chosen and its contents read, or a picture chosen and the image displayed in-game. Some mobile browsers also allow a picture to be taken with the device camera instead of choosing an existing file, then the taken photo image file is passed as the chosen file.

The appearance of the file chooser button varies depending on the platform. Several browsers display a "Choose file" button.

Scripting

When using JavaScript or TypeScript coding, the features of this object can be accessed via the IFileChooserInstance script interface.

Layering HTML objects

This object displays using a HTML element rather than drawing in to the canvas. This means its layering works differently to other objects. To learn more about how to layer HTML objects, see HTML layers.

Using files

Once a file has been selected, it can be accessed using a URL. While many URLs reference web resources on the Internet, the File Chooser plugin returns a special URL that refers to the local file. This URL typically starts with "blob:" (since the browser's term for an arbitrary set of data is a "blob") followed by some kind of unique ID. By accessing this URL the local file is accessed, without using any Internet connection, so this also works offline.

The file URL can be used like any other URL. For example, it can be requested by the AJAX object to load its contents as text, or passed to the Sprite object's *Load image from URL* action to display it.

File Chooser properties

Accept

The file type filter to accept. This can be a comma-separated list of file extensions, e.g. ".png,.jpg,.jpeg", or a MIME type, such as *image/** for any image file, *video/** for any video file, and so on. If left empty, the file chooser will be able to choose any kind of file.

> *For best cross-platform compatibility, where possible use a MIME type instead of a file extension. For example prefer using application/json instead of the .json file extension.*

Select

Whether to allow selecting a single file only, or multiple files in one go.

Initially visibile

Whether the control is initially visible or hidden.

ID Optional

An optional *id* attribute for the element in the DOM (Document Object Model). This can be useful for CSS styling.

Class Optional

An optional *class* attribute for the element in the DOM (Document Object Model). This can be useful for CSS styling.

File Chooser conditions

See common conditions for features shared between form control objects.

On changed

Triggered when a file or set of files has been chosen from the input control.

File Chooser actions

See common actions for features shared between form control objects.

Clear

Reset the control to its initial state with no selection, clearing any prior picked file.

Click

Acts as if the file chooser was clicked, which brings up the system file picker. This allows the button to be completely customised, as the actual File Chooser object can be moved offscreen, and a custom sprite or other object used to activate the *Click* action of File Chooser instead.

> *Normally this action can only be used in a user input trigger, e.g. in a mouse click event, touch start event, etc.*

Release file

Release a previously returned file URL. This allows the browser to release memory for the file. If the user chooses lots of files or chooses them regularly this action should be used to release the files after they are no longer needed.

File Chooser expressions

FileCount

The number of chosen files. If *Select* is *Single*, this is always 1. If *Multiple* then this returns the number of chosen files.

FileNameAt(index)

Return the name of the chosen file at an index. In *Single* select mode, the index should always be 0, since only one file can be chosen.

FileSizeAt(index)

Return the size of the chosen file in bytes at an index. In *Single* select mode, the index should always be 0, since only one file can be chosen.

FileTypeAt(index)

Return the MIME type of the chosen file at an index. The MIME type is the standardised Internet media type of the file contents. For example, this can be *image/png* for a PNG image, *image/jpeg* for a JPEG image, or *text/plain* for a plaintext file. In *Single* select mode, the index should always be 0, since only one file can be chosen.

FileURLAt(index)

The URL to access the local file's contents at an index. For more information see *Using files* above. In *Single* select mode, the index should always be 0, since only one file can be chosen.