

CSV

View online: <https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/csv>

The **CSV** plugin supports reading and writing data in [comma-separated values](#) (CSV) format, as well as other similar formats that use a different delimiter such as [tab-separated values](#) (TSV).

Delimiter-based formats like CSV and TSV are simple and easy to use, but are limited in their ability to represent more complex data. For more advanced uses consider using [JSON](#) instead.

Scripting

When using JavaScript or TypeScript coding, the features of this object can be accessed via the [ICSVObjectType](#) script interface.

Using CSV files

CSV and TSV files can be added as [project files](#) in the Project Bar and edited directly in Construct. As with other data formats, these can then be loaded by requesting them with the [AJAX object](#).

Using tab delimiters

Due to the use of the tab key as a shortcut in the editor, it can be difficult to type a tab in the Parameters Dialog for the delimiter parameter if you want to use tab-separated values. To make this easier the CSV plugin provides a *TabCharacter* expression which just returns a string with a single tab character in it, so it can be conveniently used as a delimiter parameter.

Using array for storage

The CSV plugin merely parses and generates CSV. It uses an [Array object](#) to store the actual data, as CSV can conveniently be represented as a two-dimensional array, and it allows using the full features of the Array object to read and manipulate the data.

CSV conditions

On parse error

Triggered when using the *Parse* action with invalid data, such as a quoted item that is missing its end quote. If a parse error occurs then no data will be read from the file and the specified Array object will be unmodified.

On parse success

Triggered when using the *Parse* action with valid data that is successfully parsed.

CSV actions

Generate CSV

Generate a string of CSV data using data from a specified two-dimensional Array object. A custom delimiter can be set for other delimiter-separated formats like tab-separated values. After this action the resulting string is available via the `GeneratedCSV` expression.

Parse CSV

Read a string of CSV data in to a specified Array object. The resulting data will be stored as a two-dimensional array. A custom delimiter can be set for other delimiter-separated formats like tab-separated values. If any of the rows are different lengths, the width of the array will be the maximum row length. If the data is invalid for any reason, *On parse error* will be triggered; otherwise if valid *On parse success* will be triggered. The *Data type* parameter determines whether values are read as strings or numbers. Its possible values are:

- **Auto:** automatically determine whether values are strings or numbers, based on whether the string looks like a number. For example "hello" will be stored as a string, but "100" will be stored as a number.
- **String:** treat every value as a string. Note this means a number like "100" will be stored as a string with that sequence of characters rather than an actual number type.
- **Number:** treat every value as a number. Note this means strings like "hello", which are not valid numbers, will result in the special "Not A Number" (NaN) value.

CSV expressions

GeneratedCSV

After the Generate CSV action, this expression returns a string of the generated CSV data.

TabCharacter

This expression merely returns a string with a single tab character in it. This can be useful to use as a delimiter parameter for tab-separated values (TSV), as a tab character can be difficult to type in the editor as it functions as a keyboard shortcut to move focus.