Card Maker JavaScript Translator

Table of Contents

Introduction

This document covers the JavaScript translator functionality. This is not the default translator for CardMaker but is available for users needing advanced functionality. There are a number of advantages and unfortunately a few disadvantages to using JavaScript translation for your project / references. This documentation is written with the assumption that you are already familiar with many concepts in the CardMaker application.

CardMaker can be switched to the JavaScript translator by right-clicking the Layouts node and selecting the Project Settings... menu item.

Critical

If you plan to use JavaScript for your translation language it must be applied to all your references in addition to the project file. You cannot mix translator types.

Reference Files

Column Names

Column names have slightly more strict rules. The column name should follow the same rules as a valid JavaScript variable name. Any spaces in the name are automatically replaced with underscores.

Plain Strings

Plain string values in references work as normal. CardMaker will automatically assume the value in the spreadsheet is a raw string (exceptions are listed below). Plain strings are automatically quoted with single quotes for use by the JavaScript interpreter.

Functions

Any Reference string that begins with `function(` is assumed to be a function and is left as-is.

Single Quoted Strings

Any Reference string that begins with a quoted string is left as-is. This allows for the string to contain references and/or function calls.

Excel / Calc Note

Single quoted strings may be mutated and/or completely trashed by the editor. To help users avoid this problem they can put a ~ (tilde). Example:

```
~'give each player ' + cointcount + ' coins
```

Tilde Escape Character

If the reference value is intended to be a JavaScript variable use the tilde to avoid treating the value as a string. The example below would be translated as though j, k, and l are variables.

```
~j + k + l
```

Escape Codes

The same escape codes (like **\n** for newline) are translated for JavaScript based references with one exception: you **MUST** use two blackslashes: **\\n** instead of **\n** – this is due to JavaScript interpreting the backslash character as an indicator of an escape code. CardMaker needs the backslash to perform the expected translation. Graphics paths are also affected by this! Be sure to indicated two backslashes in the image path (or just use forward slashes).

Working with the Element Definition

Column References / Defines

For column references just enter in the name of the column in **all lower case characters**. Example: `moneyearned`

Remember this is true across all JavaScript when referencing column values!

Raw Strings

Raw strings must be single quoted to appear. An example: `'This is a raw string.'`

The quotes are necessary so JavaScript treats the contents as a string not as variables.

Mixing Column References / Defines and Raw Strings

The quoted text will be treated as a raw string and any unquoted text will be interpreted as a variable (column reference/define). Example: `'Earn $' + money` where money is a column/define.

Building Variable Names

With JavaScript it is possible to construct a string and then look up the value by the name.

```
idx = 5;
columnName = 'specialcolumn_' + idx;
finalValue = this[columnName]
```

Pre-Defined Variables

These values are always defined and usable in JavaScript.

Variable	Value
deckIndex	The index of the card in relation to the deck
cardIndex	The index of the card itself (if the count is > 0)

Element Overrides

The AddOverrideField method is available (with generally the same rules as a reference override).

Usage: AddOverrideField(fieldName, fieldValue);