Card Maker 1.0.0.2

by Tim Stair

Table of Contents

Introduction..3
Warning..3
User Interface...3
  Main Window..3
  Project window..5
    Renaming a Layout..5
    Re-Order Layouts...5
    Context Menu Items (Layouts Node)...5
    Context Menu Items (Layout Node)...5
    Context Menu Items (Reference Node)..7
  Layout Control window...7
    Card Count Group..7
    Card Layout..7
    Context Menu Items..8
    Multi-Select..8
  Element Control window...9
    Element..9
    Font (Text Elements)..9
    Graphic (Graphic Elements)...10
    Shape (Shape Elements)...11
    Back...11
    Border...11
    Outline..11
  Canvas window...12
    Zoom...12
    Canvas Alignment...13
    Dividers...13
  Logger window..13
  Defines window...13
CSV Reference Files...14
  Creation...14
  Special Functions..14
  Color String Format...14
    RGB Color Format...14
    Color Name Format...15
  Font String Format...15
  Element Overrides..15
    Data Based Overrides..15
    Definition (Variable) Based Overrides (Incept Translator only at this time).........16
    Override Fields..16
  Define File...17

Google References / Defines....18
Google Authentication Setup....18
Translator....18
Incept (Default)....19
JavaScript....19
Definition / Scripting Reference (Incept Translator)....19
All Elements....19
Inline Background Shapes....23
Basic....23
Advanced....23
Inline Background Graphic....23
Basic....23
Advanced....23
Graphic Elements....24
Text Elements....24
Formatted Text Elements....24
Shape Elements....27
Additional Parameters by Shape Table....27
Grid Note....28
Order of Translation....28
Settings....28
General....28
PDF Export Settings....29
Exporting Layouts....29
Layout Notes....29
Where is the print option?....29
Layout Templates....30

Introduction

Card Maker is an application for generating layouts of information in a graphical representation. (a wordy way of saying “it can make business cards or game cards”) Card Maker was inspired by the need to generate cards for prototyping board games quickly without a lot of overhead to make minor changes. The goal of the application is to simplify the process by which a card layout is created and decrease the amount of time required to make minor changes.

Warning

All information in this document is subject to change... of course. Card Maker should be used AT YOUR OWN RISK. The author of the application is not responsible for any damages caused by its use. (though I certainly hope nothing horrible will happen)

User Interface

Main Window

The main window has limited functionality and acts as a container for all of the other windows.

File Menu Item	**Function**
New Project	Creates a new default project.
Open Project...	Opens a project file.
Save Project...	Saves the current project to the loaded file (or prompts the user if the file does not yet exist).
Save Project As...	Allows the user to save the project as a different file name.
Export Project to Images...	Displays the Image Export dialog for the project.
Export Project to PDF...	Displays the PDF Export dialog for the project.
Close	Exits Card Maker

Edit Menu Item	**Function**
Undo	Undoes the last element or layout change.
Redo	Performs the last undone element or layout change.

View Menu Item	**Function**
Draw Element Borders	Toggles the red outline drawn around each element.
Always Show Selected Element	Toggles the outline for the selected element.
Draw Selected Element Guides	Toggles the visibility of the guide lines indicating the

	location of the Element as related to others.
Draw Formatted Text Word Borders	Toggles an outline drawn around each word in a formatted text field.

Project Menu Item	**Function**
Clear Image Cache	Clears any cached images from memory forcing them to reload from disc. (allowing changes made by external programs)
Update Known Issues	Displays / Updates the known issues dialog. Each problem area will be transitioned to upon clicking on an issue. Note: Your project ***must*** be saved to use this feature.
Import Layouts...	Allows the import of layouts from other CardMaker project files.
Clear All Google Cache Entries	This clears the entire Google Cache
Project Settings...	Allows you to configure project level settings. You can set the project-wide translator type and the default define reference type. The Default Define Reference Type setting allows you to specify where to attempt to read the _defines file for the project on layouts without any references. CSV is the fallback if nothing is found in the Google spreadsheets.

Layout Menu Item	**Function**
Reload References	Reloads the references for the currently selected Layout. If Google Cache is enabled the references associated with the layout are downloaded to update the cache.

Tools Menu Item	**Function**
Project Manager	Viewer for all projects within a given directory structure.
Color Picker	The color selection dialog (useful if you need the color value)
Remove Layout Templates	Dialog for removing layout templates.
Illegal File Name Character Replacement...	Configures the characters to replace those that are invalid on export of the project. (empty by default)
Update Google Credentials...	Allows you to update the AccessToken used when connecting to a Google Spreadsheet reference.

Settings...	Configure the general settings for Card Maker.

Windows Menu Item	Function
(Window Name)	Opens the given window

Project window

The Project window contains a tree view indicating the card layouts and their associated reference files.

Renaming a Layout

To rename a Layout left-click a layout node and don't move the mouse until the editable text field appears. (like renaming a folder in windows explorer)

Re-Order Layouts

You can drag and drop layouts in the project view to change the order in the project.

Context Menu Items (Layouts Node)

Menu Item	Node Type	Function
Add Card Layout...	Layouts	Adds a new layout based the parameters supplied in the dialog.
Add Card Layout From Template...	Layouts	Adds the selected Layout template from the dialog with the name specified.
Set Name Format...	Layout(s)	Adjusts the naming scheme associated with the project when exporting images. Special codes in the name format: @[column_name] – A value from the reference / define ## – card number (1 based) #SC – Sub card number (1 based) #L – layout name
Project Settings...	Layouts	Same functionality as the Project > Project Settings… item.

Context Menu Items (Layout Node)

Menu Item	Function
Duplicate Layout	Duplicates the selected layout
Duplicate Layout (Custom)...	Duplicates the layout based on the settings specified.

Resize Layout...	Assists with configuring the width/height of the layout based on the desired measurement (and DPI). Uses the configured measurement unit from the application wide pdf export settings dialog.
Define as Template Layout...	Adds the selected layout to the list of layout templates. All aspects of the layout (including elements) are persisted except the name.
Remove Card Layout	Removes the selected layout
Export Card Layout as Images...	Displays the Image Export dialog for the selected layout. There are a few special settings: Stitch Skip Index: A value to indicate any gaps to intentionally leave in the stitched export. File Name Format (see Configure Layout Export for details)
Export Card Layout as PDF...	Displays the PDF Export dialog for the selected layout
Add Reference...	Adds a reference file (CSV) to be associated with the selected layout
Add Google Spreadsheet Reference...	Adds a reference to a Google Spreadsheet. Each time you open the project the AuthToken will be requested.
Configure Layout Settings...	Configures numerous settings for the Layout: **Export** Naming scheme associated with the project or layout: Special codes in the name format: @[column_name] – A value from the reference ## – card number #L – layout name Export rotation if your images need to be rotated 90/-90 degrees. This applies to pdf, and image export (adjusting your card between portrait or landscape). Stitched columns/rows configures the width and height based on the buffer and layout size. Export width and height for export to stitch multiple images together into one. Export with a transparent background instead of a white one.

Context Menu Items (Reference Node)

Menu Item	Function
Set as Default Reference	Sets the selected Reference as the default for the layout associated with it.
Remove Reference	Removes the selected Reference from the list associated with the parent layout.

Layout Control window

Card Count Group

Control	Function
#	Sets the default card count. This is intended for layouts that do not have a reference. Layouts with references only use this number if no items are in the associated reference (either due to a sparse reference or the use of the allowed layout functionality).
Card Numeric	This sets the card index for this layout. Use this to cycle through all of the cards associated with this layout.
Row Numeric	This sets the current row from the reference source, jumping duplicates.

Card Layout

Control	Function
Buffer	Sets the vertical and horizontal buffer between cards when exporting to PDF.
Export DPI	Sets the DPI of the given layout. This only affects the exported DPI. The layout is still rendered 1:1 on the screen. (example: if you want a 3.5” wide card at 300dpi then the width needs to be 3.5 x 300 = 1050)
Width	The width of the layout in pixels.
Height	The height of the layout in pixels
Resize (button)	Assists with configuring the width/height of the layout based on the desired measurement (and DPI). Uses the configured measurement unit from the application wide pdf export settings dialog.
Draw Border	Draws a black border 1 pixel wide on the edges of the layout. This is primarily intended for cutting cards out after printing.
Load All References	This option will attempt to load all the associated references. The project wide defines are loaded based on the default reference only. All reference define files are loaded as well. The column names are driven by the default reference only. Any additional references **must** have the same column order. Any extra columns in further references are ignored. Press F9 to reload references after toggling this option.

Control	Function
Elements	A list of all of the elements in the layout. If an element is highlighted it is the selected item and can be affected in the Canvas and Element Control windows. The enabled value indicates whether to draw the element or not. You can toggle the selected elements visibility by double-clicking or pressing the space bar. The order of the elements indicates how the items will be drawn. The bottom item is draw first and proceeds to draw the others up to the top element.
Add	Adds a new element.
Dupe	Duplicates the selected element. If multiple elements are selected the same number of new element names must be specified to duplicate.
Remove	Removes the selected element.
Rename	Renames the selected element. Note: Duplicate names of elements are not allowed within a layout.
Scale	Scales the selected elements by a decimal value (1.0 is no change)
Resize	Scales the selected elements by the specified amounts (0 is no change). Negative values indicate the shrink the elements.
Up / Down	Moves the element up or down in the list affecting draw order.

The table below contains shortcuts for use in the Elements list box.

Description	Keyboard Shortcut
Move selected element up 1 level	Shift + Up
Move selected element down 1 level	Shift + Down
Select the element 1 level up	Up
Select the element 1 level down	Down

Context Menu Items

Menu Item	Function
Copy	Copies the selected Element(s) to the clipboard.
Paste	Pastes the Elements in the clipboard into the Layout. This is intended for copy Elements from one Layout to another. Note: You cannot paste an Element of the same name.
Paste Settings...	Applies the specified settings from the Element in the clipboard to all selected Elements.

Multi-Select

You can also select 1 or more of the layout elements to move them together in the Canvas window. This will maintain their relative location to one another. Other changes, such as resizing, in the Canvas with multiple items selected will only affect the given selected element you are manipulating. If multiple items are selected all elements will be affected by changes to the Element Control, such as changing the font or border.

Element Control window

Element

Control	Function
Type	The type of element.
X	The x position of the element. 0 is the far left. X increases moving right.
Y	The y position of the element. 0 is the far left. Y increases moving down.
Width	The width of the element.
Height	The height of the element
Opacity	The opacity of the element.
Rotation	Rotation of the element around the center of the element definition position.
Spreadsheet Data	Above the Definition text area there is a view of the current data associated with the given card index. You can right-click in the view to add a reference (to the given column) to the Definition field.
Definition	The text string defining the element contents. (see Definition / Scripting Reference (Incept Translator) for more details) When working with a Graphic element type the "..." button may be used to browse to and select supported graphic files. The "+" button to the right of the Definition field will pop up a list of possible items to insert into the Definition text area. (see Definition / Scripting Reference (Incept Translator) for more details)

Font (Text Elements)

Control	Function
Font	The font to render the text with.
Color	The color to render the text with.
Bold	Draws the font in bold. (may not be supported by all fonts)
Italic	Draws the font italicized. (may not be supported by all fonts)
Strikeout	Draws a line through the rendered text. (may not be supported by all fonts)

Control	**Function**
Underline	Draws a line under the rendered text. (may not be supported by all fonts)
Size	The size of the font to render.
Auto-Scale	Attempts to scale the font of any text that overlaps the element borders down to the desired element size. Warning: This can be CPU intensive. You should also setup your element with an optimal font size. The font size you specify in the element is the target font size maximum The more scaling performed the less accurate the output. (though your string should still appear in its entirety) **Note:** This can only be used with Text elements.
Line Spacing	Adjusts the spacing between text lines. **Note:** This can only be used with FormattedText elements.
Word Spacing	Adjusts the spacing between words (can be negative). **Note:** This can only be used with FormattedText elements. This is used to tweak the size of a space (and space markup).
H Alignment	The horizontal alignment of the text within the element space.
V Alignment	The vertical alignment of the text within the element space.

Notes:

- The text will appear to be red until the font has been set once.
- Any strings that are too large to fit in the desired element will be drawn with ellipsis to help find issues. The ellipsis will ***not*** be present if Auto-Scale is used.
- OTF (Open Type Fonts) are not supported at this time.

Graphic (Graphic Elements)

Control	**Function**
Lock Aspect Ratio	Locks the image aspect ratio when drawing.
Keep Original Size	Locks all drawing of the image to the original size. If oversized the image is cropped (based on alignment settings). The Lock Aspect Ratio option has no effect on this setting (as it will be the original image aspect ratio).
H Alignment	The horizontal alignment of the graphic within the element space. *Note: This only apples when Lock Aspect is checked.*
V Alignment	The vertical alignment of the graphic within the element space. *Note: This only apples when Lock Aspect is checked.*
Set Size To Image	This will set the size of the element to be that of the original image file.

Control	Function
Tile Size	This defines the size of the tile for the graphic (if you want to tile at all). The format is #x# (width x height). Example: **50x45** You can use a variable width or height that will match the aspect ratio by specifying '-' as the width or height (one of the two should be specified) Examples: **-x45** **50x-** You can specify '-' or a blank string if you do not want tiling at all.

Shape (Shape Elements)

Control	Function
Color	The color to render the shape with.

See shape information in the Definition / Scripting Reference (Incept Translator) section. There is also an assistant to help define the shape. Each time you change a property value the definition of your element will automatically change to the settings indicated.

Back

Back is the background color for the element rectangle.

Control	Function
Color	The color of the background.
X	Clears the background color completely (none is drawn)

Border

Border is an outline for the element itself (the rectangle defining the object).

Control	Function
Color	The color of the element border.
Thickness	The thickness of the border (in pixels). If this is 0 no border is drawn.

Outline

Outline is an outline for the drawn element itself. (text/shape outline)

Control	Function
Color	The color of the outline border.
Thickness	The thickness of the outline (in pixels). If this is 0 no outline is drawn.

Canvas window

The Canvas is a WYSIWYG (what you see is what you get) editor for the element shape and location. It directly effects the Element Control window values. It is highly recommended that you further tweak any location/scale values in the Element Control window for neatness sake!

The currently selected Element in the Layout Control window will be drawn with a green border. This item can be moved and manipulated like many other standard graphics tools. I'm not going to detail out how to do this. If you're new to graphics editing you should spend some time in the Canvas manipulating an element to see how it functions.

Note: The green border on selected items will not appear if you have disabled the **Draw Element Borders** option.

You can use the right mouse button to select the element under the cursor. If more than one item is under the cursor a context menu will display a list to select from.

You can press and hold the middle mouse button to pan around the layout. This is useful if zoomed in or any time the scroll bars are visible.

The table below contains shortcuts for use in the Canvas.

Description	Keyboard Shortcut
Move selected element up 1 level	Shift + Up
Move selected element down 1 level	Shift + Down
Select the element 1 level up	Control + Up
Select the element 1 level down	Control + Down
Move selected elements up/down 1 pixel	Up / Down
Move selected elements left/right 1 pixel	Left / Right
Lock to axis	Shift (and move the mouse along the desired axis to lock to)
Zoom In / Out	Ctrl + +/- (+ zooms in, - zooms out)
Toggle Mode – switch between resize-and-move(normal) and move-only	m
Toggle Mode – switch between move(normal) and rotate-only	r

Zoom

Control	Function
Fit	Sets the zoom value so the canvas is filled with the layout (width or height)
Fit Horizontal	Sets the zoom value so the canvas view is filled with the layout (width wise)

Note: Zoom may result in a slightly inaccurate representation on things that are measured programatically(this is a bug). Formatted Text elements in particular may be laid out slightly differently when zoomed in. It is recommended that you always review your work at a zoom of 1.0.

Canvas Alignment

Control	Function
Center Elements Horizontally	Center the elements horizontally in the layout (only the x value is affected)
Custom Align Elements Horizontally	Customizable Horizontal Align
Center Elements Vertically	Center the elements vertically in the layout (only the y value is affected)
Custom Align Elements Vertically	Customizable Vertical Align
Custom Align Elements	Customizable Align

Under the custom options there are a few settings to further tweak how to align the items. Including the Element Width/Height makes the elements to be aligned without overlap. Vertical/Horizontal centering can either be based on the Element or the Layout itself. Pixel spacing indicates the number of pixels between the placement of each element (if the width/height is not included the Elements will likely overlap).

Dividers

You can configure dividers to help layout elements by dividing the card. These lines can be toggled off and on with the Toggle Dividers button. They are never exported to the final output.

Logger window

Context Menu

Control	Function
Copy Line to Clipboard	Copies the selected line to the clipboard.
Copy all text to Clipboard	Copies all of the text in the logger to the clipboard.

Defines window

The defines window is used to display the current applicable defines. You can use Ctrl+C to copy the selected define as a data reference item (“@[define]”). There is a right-click context menu with similar options.

CSV Reference Files

Creation

CSV reference files have 4 strict content rules:

1. The separator character must be a comma. Some editors may switch this on you!
2. The first row must contain the column names. The columns may be named with spaces but should not use the [or] characters (or anything else that might confuse the parser when trying to translate references).
3. The first column may be named anything, but the contents should always be the card count of the given row. (an empty entry assumes 1 instance of the given row)
4. Never use quotes in the strings in Excel or Calc (OpenOffice). Both applications handle this differently and it makes a mess of things. The automatic quotes around a string (if you look at your CSV file in a text editor) are necessary and should be left as is. See the escape codes in the Text Elements section.
5. Newlines within a value are supported but not recommended!

Special Functions

Function	**Definition Details**
Comment out a line.	In the first column set the card count to 0.
Specifying the layout	You can specify the allowed layout for a card by name. Any row without a matching layout name with the current layout will not be included in the list of cards. Name the column **allowed_layout** in your CSV file and specify the desired layout for every card.

Color String Format

RGB Color Format

Colors are formatted into strings as follows:

RRRGGGBBB or RRRGGGBBBAAA

Each block is a 3 digit representation of the color value for the given color. (R – Red; G – Green; B – Blue; A - Alpha)

Example: 012123167

You can also specify a color using hexadecimal values:
RRGGBB or RRGGBBAA

Example: 882859

Be sure to include any leading zeroes. The type of color value you specify (hex vs. decimal) is based solely on string length: 6/8(hex)

Some editors may remove the leading zeroes. If you are specifying the color as hex you can insert **0x** at the beginning of the string (ie. 0X882859). **I highly recommend doing so.**

Color Name Format

You may also specify colors by using the names supported by the .NET framework. See: https://msdn.microsoft.com/en-us/library/ie/aa358802%28v=vs.85%29.aspx

Font String Format

Fonts are formatted into strings as follows:

fontname;fontsize;bold;underline;italicized;strikeout

fontname – string name of a font (“Arial”)
fontsize – size of the font (like the size in a word processor)
bold – 1 or 0 (on or off)
underline – 1 or 0 (on or off)
italicized – 1 or 0 (on or off)
strikeout – 1 or 0 (on or off)

Note: See one of your project files for an example font string (they are readable in notepad or any other text editor). An invalid font string will be ignored and may result in crashes or other undesirable outcomes!

Element Overrides

Data Based Overrides

You can override an element's settings per card in the CSV by providing a column with the following format:

override:[element name]:[element value]

Example: **override:text_field:x**

Any values specified in this column will override the **text_field** element's **x** value. When rendered the element's x location would be that of the override value. (locking the horizontal position in the example)

Any colons involved in the name of an element will cause the override to fail. **DO NOT USE COLONS IN YOUR ELEMENT NAME**.

This will completely override any changes you make in Card Maker. When rendered the override will take effect. Any empty value specified within the column is ignored.

The value of an override can use the same functionality as described in Definition / Scripting Reference (Incept Translator). This allows the override value to be defined based on other properties of the given card.

Definition (Variable) Based Overrides (Incept Translator only at this time)

Overrides can also be created within the Definition field itself using the following macro:

$[field:value]

Example: $[y:40]

The above would lock the y value to 40.

Override Fields

The various field type values must be specified correctly for the program to function correctly. See the table below for formatting the content of your override column. This table also contains the recognized field names. All field names are lower case.

Fields	Override string format
x y width height borderthickness opacity outlinethickness lineheight wordspace (+/-)	**(int)** Integer number values.
autoscalefont enabled lockaspect keeporiginalsize justifiedtext	**(bool)** Boolean values: **true** or **false** (case should not matter)
rotation	**(float)** In the case of rotation use only integer number values.
bordercolor elementcolor outlinecolor backgroundcolor	**(string)** (See the section Color String Format)
tilesize	**(string)** (See the section Graphic (Graphic Elements))
font	**(string)** (See the section Font String Format)

Fields	Override string format
verticalalign	**(int)** Numeric value indicating the vertical alignment: 0 – Near 1 – Center 2 – Far
horizontalalign	**(int)** Numeric value indicating the horizontal alignment: 0 – Near 1 – Center 2 – Far
variable	**(string)** The **Definition** field. This can only be overridden in a data file
type	**(string)** One of the following strings: **Text** **Shape** **Graphic** **FormattedText**

Define File

In addition to a CSV reference file you can have a set of defines associated with each reference. This is for the definition of values so you do not have to repeat them in the main reference file. Each reference can have a defines file. The project can also have a defines file if there are values you need across all references. Sample of a define file shown below:

Define	Value
coin	Awesome Token
roll_hint	You must roll 1D6 to determine the outcome.
game_title	Space Adventure 2048
nested_coin	(@[coin])

Notes:

- Each define name follows the same rules as a column name in a reference file.
- The first row is ignored. The column names are there only for reference. The first column in all rows after the first contains the define name and the second column contains the value.

The define file should be named the same as the main reference file with **_defines** appended to the name and should reside in the same folder as the reference file. Example:

- Reference file: **items.csv**
- Define file: **items_defines.csv**

- Project Wide Defines: [project file name without extension]**_defines.csv**

Google References / Defines

References stored as Google spreadsheets follow the same general pattern as local CSV References. The one exception is that the **defines** functionality operates based on a sheet in the same spreadsheet with **_defines** appended to the name. Sample:

- Main sheet: **items**
- Defines sheet: **items_defines**
- Setting up Project Wide Defines
  Spreadsheet Name: [project file name without the extension] (this is case sensitive!)
  Sheet Name: **defines** (always this name)

  Alternative / Override: If you don't want to name your Spreadsheet as described above you can change the Spreadsheet to look for the **defines** sheet in under the Layouts > Project Settings… menu item in the Project window. Specify the desired sheet in **Google Project define spreadsheet override** (case sensitive). You can enter the name manually or use the "..." button to navigate.

Google Authentication Setup

Configuring Google connectivity is reasonably easy. It used to be as trivial as entering you Google credentials but Google has disabled that style of authentication in favor of OAUTH2. See the instructions below for enabling access to your Google spreadsheets.

1. Launch CardMaker and select **Tools** > **Update Google Credentials...**
2. Either copy the Auth URL and browse to it in your preferred browser or click **Browse To URL**
3. Google should prompt you to allow CardMaker to access your spreadsheets and then navigate to the CardMaker site with the AccessToken field already selected.
4. Copy the **AccessToken** value and return to CardMaker.
5. You can manually paste the **AccessToken** value into the **Access Token** field or press the **Paste Token** button. If you use the **Paste Token** button the dialog will close immediately (slight time savings!).
6. Now when you add a Google Reference you should be able to access your Spreadsheets. You will need to setup the Access Token each time you run CardMaker. No credentials data is persisted to any files and the Access Token has an expiration anyway.

Translator

CardMaker supports two different translators. Each translator operates differently and requires different formatting in the reference files. The translator can be configured on a per project basis. Translator types cannot be mixed within a single project.

Incept (Default)

This is the default translator for CardMaker. This document includes the details on how to work with this translator.

JavaScript

The alternative (and experimental) translator is JavaScript based. This allows for much more complex logic and offers full scripting capabilities. See the Card_Maker_JavaScript.pdf file for more information.

Definition / Scripting Reference (Incept Translator)

The information below is intended for use in the Definition field of the Element Control or in the CSV Reference (or define) file itself.

#empty – This can be used in place of an empty string. Example:
#if(x == then a)#
is the same as
#if(x == #empty then a)#

All Elements

Function	**Definition Details**
Referencing a Data Source item	@[**columnname**] Replace **columnname** with the desired column name or define from the referenced data source.

Function	**Definition Details**
Referencing a Data Source item with parameters	@[**columnname**,parameter1,parameter2,(repeats...)] Replace **columnname** with the desired column name or define from the referenced data source. Replace parameter1 (etc.) with the desired input for the define. When constructing the resulting value any instances of {#} will be replaced with the given parameter. Be aware that spaces after/before the commas will be included in the parameter strings. Example: **Defines** smallImgTag = <img={1};.90;0;3> theCoin = \images\coin.png **Input** @[smallImgTag,@[theCoin]] **Output** <img=\images\coin.png;.90;0;3>
Card Values	There are some special values that are taken from the current context of the layout/card. ![deckindex] – This is the current deck index ![cardindex] – This is the current card index (see Sub Counter) ![cardcount] – The number of cards in this deck ![elementname] – The name of the element
Substring	%[string,index,count] string – some sequence of characters index – the index to begin extracting characters from (0 based!) count – the number of characters to extracted Example: %[sample,4,2] would result in **le** This allows for a string to have a given number of characters extracted.
Counter	##x;y;z# This will display a number based on the following properties: x + (current card index * y) with left padded 0's numbering z Example (on card index 5) with ##1;1;5# would be **00005**

Function	**Definition Details**
Sub Counter	#sc;x;y;z# The sub card index is the index of the card based on the number of items indicated in the count column of the reference file. This will display a number based on the following properties: x + (sub card index * y) with left padded 0's numbering z Example (on sub card index 5) with #sc;1;1;5# would be **00005**
Repeat	#repeat;x;y# This will repeat the string of characters represented by y x times.
If Statement	Samples: #(if x == y then a)# #(if x != y then a)# #(if x == y then a else b)# #(if x != y then a else b)# **==** - x is equal to y (case insensitive) **!=** - x is not equal to y (case insensitive) **>** - x is greater than y **<** - x is less than y **>=** - x is greater than or equal to **<=** - x is less than or equal to x,y,a,b – string values **#nodraw** is supported as a result to trigger the element to not be drawn at all. You can specify an empty string by simply entering nothing. An example of an empty comparison is as follows: #if(x == then a)#

Function	**Definition Details**
If Statement (grouped)	You can use the if statement with groups. The x and y values listed above can be formatted as: [val1;val2;val3] Allowing multiple items to be compared to multiple other items. The x set of strings is sought in the y set. The comparison is successful if all of the items in x exist in y. (the y set is not checked for existing in x) NOTE: This does not apply to numeric comparisons. Only **==** and **!=** are supported. TBD – keep this functionality?
Switch Statement	#(switch;key;keytocheck1;value1<repeat>)# key – The value to check the others for. keytocheck1 – A comparison string value1 – A value string (this will be the overall result if key matches keytoccheck1) ;keytoccheck1;value1 can be repeated multiple times with varying strings. There should always be a key and a value. (an empty string is valid) If the keytocheckis **#default** the value will be the default if no other keys match. (example: #(switch;15;10;Y;#default;Z)# – would default to Z because “15” is not the same string as “10”). **#switchkey** can be used in the switch statement to reference the key value. **#nodraw** is supported as a result to trigger the element to not be drawn at all.
Switch Statement (alternative)	#(switch//key//keytocheck1//value1<repeat>)# This is the same as the above just using ‘//’ as a delimiter. The delimiter is completely customizable. The first 2 characters after the word *switch* will be used as the delimiter.
Random Number	#random;min;max# min – minimum value (inclusive) max – maximum value (inclusive) This operates with positive/negative integers only.

Function	Definition Details
Greater than character	**&gt;** This is critical for anyone needing to use the > character in a value used by an if statement.
Less than character	**&lt;** This is critical for anyone needing to use the < character in a value used by an if statement.

Inline Background Shapes

Inline background shapes are a convenient way to create backgrounds for various elements.

Normally you would need to create an independent element and keep its size up-to-date with the foreground element.

There are two formats:

Basic

#bgshape::[shape definition string]::[color string]#

Example: **#bgshape::#roundedrect;0;-;-;30#::0xff00ff#**

Advanced

#bgshape::[shape definition string]::[color string]::[x offset]::[y offset]::[width adjust]::[height adjust]::[outline thickness]::[outline color]#

Example: **#bgshape::#roundedrect;0;-;-;30#::0xff00ff::-20::-20::40::40::5::0xff0000#**

See the Shape Elements section for how to define the shape. The offset and adjust settings may be negative.

Inline Background Graphic

Inline background graphics are a convenient way to create backgrounds for various elements.

Normally you would need to create an independent element and keep its size up-to-date with the foreground element.

There are two formats:

Basic

#bggraphic::[image path]#

Example: **#bggraphic::images/Faction_empire.bmp#**

Note: The aspect ratio is not enforced. The graphic will stretch to the bounds of the element.

Advanced

#bggraphic::[image path]::[x offset]::[y offset]::[width adjust]::[height adjust]::[lock aspect ratio]::[tile size]::[horizontal align]::[vertical align]#

Example: **#bggraphic::images/Faction_empire.bmp::-5::-5::10::10::true::-::1::1#**

See the Graphic (Graphic Elements) and Override Fields sections for how to configure the various settings. The offset and adjust settings may be negative.

Graphic Elements

Function	**Definition Details**
Referencing an Image	The path to the file should be specified. The path may also be relative to the project file. Supported Image Types: BMP, GIF, EXIG, JPG, PNG, TIFF, and PSD. PSD file support is rudimentary. 8 bit per channel PSD images should be functional. (RGBA) Relative path example: **images\file.bmp** would be valid if the project file were stored in the folder where the folder **images** existed
Draw no image	**#nodraw** is supported like other elements. **none** may also be specified to indicate to draw no image (note that this is different than no draw as the other components of the element will be rendered (example: border)**.** Note: Any invalid file path / name will also not draw.

Text Elements

Function	**Definition Details**
New Line	**\n** – Will be replaced with a new line
Quote Character	**\q** – Will be replaced with a “ (necessary for CSV file annoyances such as OpenOffice and Excel handling of strings with commas in them)
Comma	**\c** – Will be replaced with a comma

Formatted Text Elements

These elements support HTML like tags for mid-line formatting. Some fonts do not display the same between styles. They will appear to be vertically offset by some number of pixels. I recommend finding a font that does not have this issue (as there may be little I can do to resolve the issue on a per font basis). To close a tag specify the tag name with a ‘/’ character before it (Example: this word will be <b>bold</b> and the rest will not).

Using the outline functionality with Formatted Text is not recommended. There are numerous draw issues that may come up. Be very careful to check the results before assuming everything is working correctly if you use outlined Formatted Text. Generally in the documentation below just about anything you specify in the definition field (even words and spaces) translate internally to a markup (a component to determine how to render).

Function	**Requires Closing Tag**	**Definition Details**
Bold	Y	**<b>** – Will change the text within to bold
Italic	Y	**<i>** – Will change the text within to italicized
Strikeout	Y	**<s>** – Will change the text within to be drawn struck out
Underline	Y	**<u>** – Will change the text within to be underlined
New Line	N	** ** – Will be replaced with a new line
Space	N	**<spc=#>** - Number of spaces to add. May also be used without specifying the number to indicate a single space: **<spc>** The width of this markup is numberOfSpaces * (fontWidthSize + WordSpacing).
Push	N	**<push=#;#>** - (x;y) The amount to push the current x/y values. **<push=#>** - (x) The amount to push the current x value. This provides direct control over the position of the next markup (including text/spaces).
Background Color	Y	**<bgc=#########;#>** - The first part is the color; the second is the additional vertical pixels to add to the default height (some fonts are given a whole lot of space at the top, but lacking at the bottom) **<bgc=#########>** - Will create a background color within the tag. (See the section Color String Format)
Background Image	N	**<bgi=[filename]>** OR **<bgi=[filename];[#];[#];[#];[#]>** - Will create a background image behind each given word in the tag. The first option will draw the image at full size in the upper left hand corner of the given word. The second option has the following fields: 1. x offset 2. y offset 3. width 4. height

Function	Requires Closing Tag	Definition Details
Image	N	**<img=[filename]>** OR **<img=[filename];[percent]>** OR **<img=[filename];[xoffset];[yoffset]>** OR **<img=[filename];[percent];[xoffset];[yoffset]>** OR **<img=[filename];[xoffset];[yoffset];[width];[height]>** • percent – percentage of the current font height the image height should be (aspect ratio is maintained) – value from 0.00 to 1.00 would be smaller than the line. 1.00 represents 100%. • xoffset – The x draw offset • yoffset – The y draw offset • width – The width to draw • height – The height to draw This will render an image and consume horizontal space like text does. If an image is too wide it will be shrunk horizontally to fit.
Font Size	Y	**<fs=#>** - Will change the size of the font for the words within the tag.
Font Color	Y	**<fc=########>** - Will change the font color within the tag. (See the section Color String Format)
Font	Y	**<f=fontname;fontsize;bold;underline;italicized;strikeout>** (See the section Font String Format)
Y Draw Offset	Y	**<yo=#>** - Will affect the Y offset of the markups when drawing. This is recommended when a style (bold or italic) does not align correcly with normal text. The number value may include the decimal point for more precise tuning.
X Draw Offset	Y	**<xo=#>** - Will affect the X offset of the markups when drawing. This is recommended when a style (bold or italic) does not align correcly with normal text. The number value may include the decimal point for more precise tuning.
Pixel	N	**<px=#;#>** - (x,y) Adjusts the draw location to the point specified within the element (will self adjust if specified value is outside the bounds). If either value is -1 the existing x/y location is maintained. **<px=#>** - y is defaulted to -1 as defined above.
Align Center	Y	**<ac>** - Overrides the horizontal alignment to be centered (this tag has no effect if using justified).
Align Left	Y	**<al>** - Overrides the horizontal alignment to be align left (this tag has no effect if using justified).

Function	Requires Closing Tag	Definition Details
Align Right	Y	**<ar>** - Overrides the horizontal alignment to be align right (this tag has no effect if using justified).
Line Spacing	Y	**<ls=#>** - Overrides the current line spacing value. Upon closing tag the previous value is restored.
Quote	N	**<q>** - Will be replaced with a double quote (“) character
Comma	N	**<c>** - Will be replaced with a comma (,) character
Escape Characters	N/A	The ‘\’ character can be used to indicate to ignore any special meaning of the following character. Examples: **\>** - will results in the > character (not treated as part of a tag) **\<** - will results in the < character (not treated as part of a tag)

NOTE: The line spacing must be configured. Lines will draw on top of each other without it.

Shape Elements

Basic shape elements are described by this format: (Rectangles/ Ellipse)
#**shapename**;**thickness**;**horizontal_size**;**vertical_size**#
Other complex types have the same basic format with extra parameters:
#**shapename**;**thickness**;**horizontal_size**;**vertical_size;[extra parameters]**#

Example: **#roundedrect;0;-;-;45#** (see below for what the **45** indicates)

Value	Description
shapename	the name of the shape to be drawn rect, roundedrect, grid, ellipse, triangle
thickness	The thickness of the pen to draw with. 0 will fill the shape.
horizontal_size / vertical_size	The horizontal/vertical size in pixels of the rectangle to draw within. This overrides the Element size. The values may be negative to draw above or to the left of the Element.

Additional Parameters by Shape Table

shapename	Additional Parameters
roundedrect (rectangle with rounded edges)	**[EdgeSize]** EdgeSize – The distance to round from the corner.

shapename	Additional Parameters
grid	**[AllowPartialGrid];[GridWidth];[GridHeight]** AllowPartialGrid – Allows the grid to draw in an incomplete form on the right and bottom GridWidth – The width of a single rectangle in the grid. GridHeight – The height of a single rectangle in the grid.
triangle	**[Corner]** Corner – The corner to draw the triangle from. 0 is upper left, 1 is upper right, 2 lower right, and 3 is lower left (default is 0)

Grid Note

- The partial grid option is only a best guess. If your thickness and other variables are not configured precisely you may find the last line draw incorrectly or missing entirely (generally obvious with 1 thickness). This issue may be resolved with a code fix in the future.

Order of Translation

1. Card Values (![x])
2. Data Source References (@[x]) & Substring (%[x,y,z]) (starting with right most first)
3. If/Switch (starting with the right most first)
4. Text translations (Note: These only apply to Text elements)
5. Definition overrides ($[…] Incept translations) are evaluated/extracted

Note: It is possible to create an endless loop if you dare enter @[?] values in the csv file.

Settings

General

Setting	Description
Enable Google Cache	When enabled any Google References will be persisted locally until you either clear the cache (Project > Clear All Google Cache Entries) or reload references (Layout > Reload References). This allows you to work offline after you have loaded a given layout (or multiple). The cache is persisted to disk and is available after relaunching the application.
Export Layout Border	When exporting a border will be drawn around the layout. This can be useful for cutting after printing.

Setting	Description
Default Translator Type	Specifies the default translator to set when you create a new project.
Log Incept Translation	If enabled, all translations performed with the Incept translator are logged (useful for debugging issues).

PDF Export Settings

The print settings can be found under **Tools > Settings... > PDF Export Tab**

Setting	Description
Page Width	The full width of the paper to print to (inches)
Page Height	The full height of the paper to print to (inches)
Page Horizontal Margin	The size of the horizontal paper margin (inches) – This is the horizontal space on the left/right side of the paper that Card Maker will not draw beyond.**
Page Vertical Margin	The size of the vertical paper margin (inches) – This is the vertical space on the top/bottom of the paper that Card Maker will not draw beyond.**
Auto-Center Layouts on Page	If enabled layouts will print to the page horizontally centered. Note: When switching layouts while printing the new layout will automatically be placed below the previous layout.
Print Layouts On New Page	When printing or exporting to PDF start a new page for each layout

** - If the layout switches to landscape the margins apply as if the output were in portrait.

Exporting Layouts

By default Card Maker attempts to export the entire project performing some basic space optimizations to combine as many layouts onto a single page as possible. In some cases you may want to select and print the individual layouts to avoid the combining of multiple layouts.

Layout Notes

Be sure to make your layouts the desired size based on your target DPI. For example: if you want a card to be 4” x 3” and the target print DPI is 300 you will need to configure your layout be 4x300(1200) width and 3x300(900) height. This is especially critical if you are targeting PDF as it compresses to fairly low quality JPEGs when storing images.

Where is the print option?

For now, printing has been removed. Image export and PDF export are both excellent options that should cover just about every use case needed.

Layout Templates

CardMaker allows for the creation of individual layout templates. These files can be found in the folder where CardMaker is started from under the \templates folder. If this folder does not exist no templates will be available to the application. If you download any templates you may have to create the \templates folder.