

 [Berteh / ScribusGenerator](#) Öffentlich

Create beautiful documents with data. Open source pdf (and Scribus) template and mail-merge alternative.

[berteh.github.io/scribusgenerator/](https://github.com/berteh/ScribusGenerator/)

 MIT license

 199 stars  32 forks

 Star

 Notifications

[Code](#) [Probleme](#) [37](#) [Pull-Anfragen](#) [Aktionen](#) [Wiki](#) [Sicherheit](#) [Einblicke](#)

 Meister ▾

[Zur Datei](#)



S1SYPHOS ...

✓ on 24 Feb 

[View code](#)

ScribusGenerator

Vorlage und Mail-Merge-ähnliche Engine, um schnell nützliche Dokumente aus Ihrem Tabellenkalkulationsinhalt im PDF-Format von Scribus OpenSource PAO zu generieren.



[Scribus Generator: Erstellen Sie schöne Dokumente mit Daten.](#) Open Source

hochwertige PDF-Vorlage und Mail-Merge-Alternative. Ihre Vorstellungskraft ist die Grenze für die Erstellung schöner Jahrbücher, personalisierter Einladungen, Spielkartendecks, Vereinslisten, Kunst- oder Arbeitsportfolio und [viele mehr](#).

Was ist Scribus Generator?

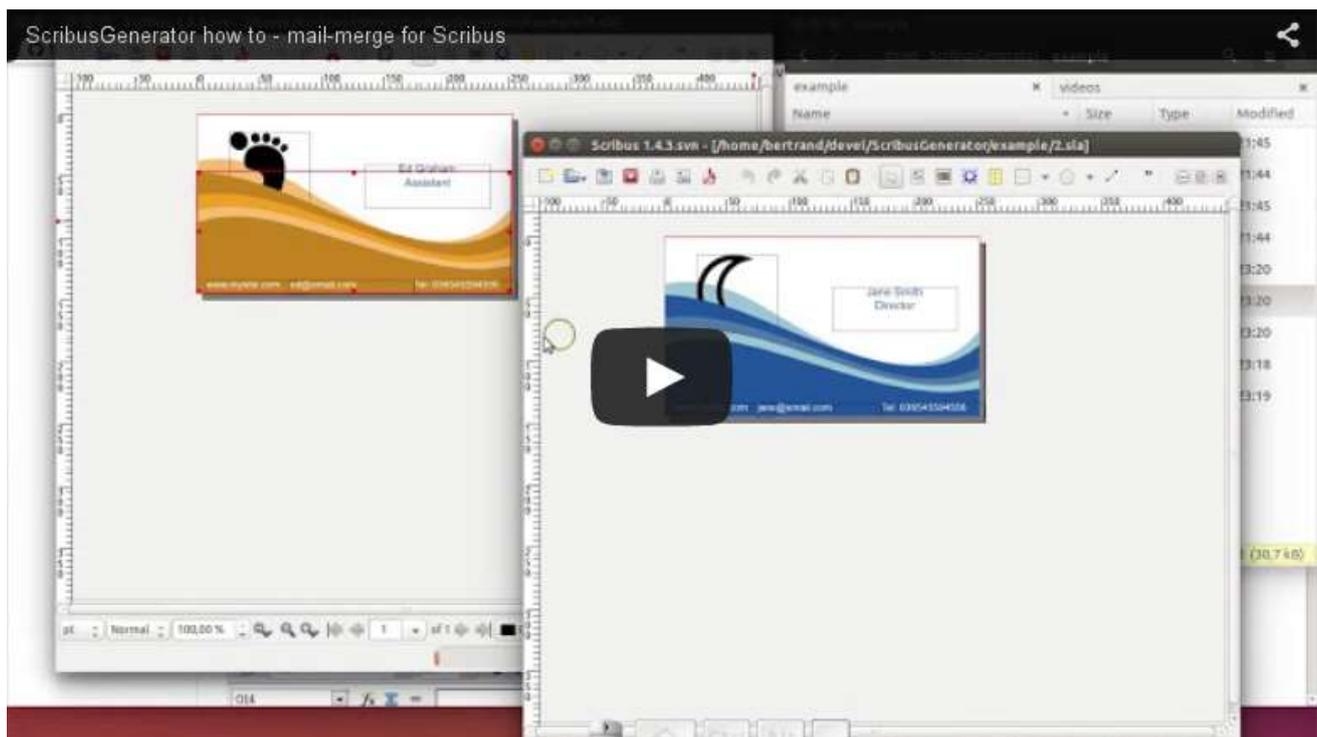
[Scribus Generator](#) verwandelt jedes schöne Dokument in eine Vorlage, um ein Original-Set basierend auf Ihren Daten zu erstellen, zusätzlich zum [Scribus](#) Open-Source-PAO-Studio. Verwenden Sie es, um Kartendecks, Visitenkarten, personalisierte Briefe, Broschüren usw. zu erstellen. Genau wie Mail Merge in Ihrer bevorzugten Office-Anwendung, aber schön und flexibel.

Mit **Scribus Generator** können Sie:

- Texte und Bilder dynamisch ersetzen
- Objektfarben, Position, Textschriftart oder -größe dynamisch ändern
- Generieren Sie separate PDF- (oder Scribus-) Dateien für jeden Dateneintrag oder eine einzelne Datei aus all Ihren Daten
- Arbeiten Sie direkt in Scribus mit einer schönen Benutzeroberfläche oder über die Befehlszeile
- Verwenden Sie eine beliebige Datenquelle (Excel, OpenOffice, MySQL, Notepad, ...), die in CSV exportiert werden kann.
- und vieles mehr...

Im Allgemeinen ersetzt **Scribus Generator** Text durch Daten, um automatisch Dateien zu generieren (z. B. SLA, PDF). Es wurde ursprünglich von [Ekkehard Will](#) geschrieben und von [Berteh](#) weiter ausgebaut.

Ein [kurzes Video stellt diesen](#) Scribus Generator vor. 6 erste Minuten für die grundlegende Übersicht, 12 letzte für einige erweiterte Funktionen.



Wie installiere ich Scribus Generator ?

Für Scribus 1.4.x bis 1.5.5: Laden Sie das Skript herunter und dekomprimieren [Sie](#) es an einer beliebigen Stelle auf dem lokalen Computer in einem Ordner, in den Ihr Benutzer schreiben kann. **Scribus Generator** kann dann gestartet werden, indem Sie das Skript () im Dialog: «Scribus → Script → Execute Script» oder über die [Kommandozeile](#) auswählen. ScribusGenerator.py

Für Scribus 1.5.6+ laden Sie bitte [die \(aktuellere\) ScribusGenerator für Python3-Version herunter](#). Ein paar kleine Funktionen hinzugefügt und syntaktische Updates für die neuere Python3-Engine.

MacOSX-Probleme

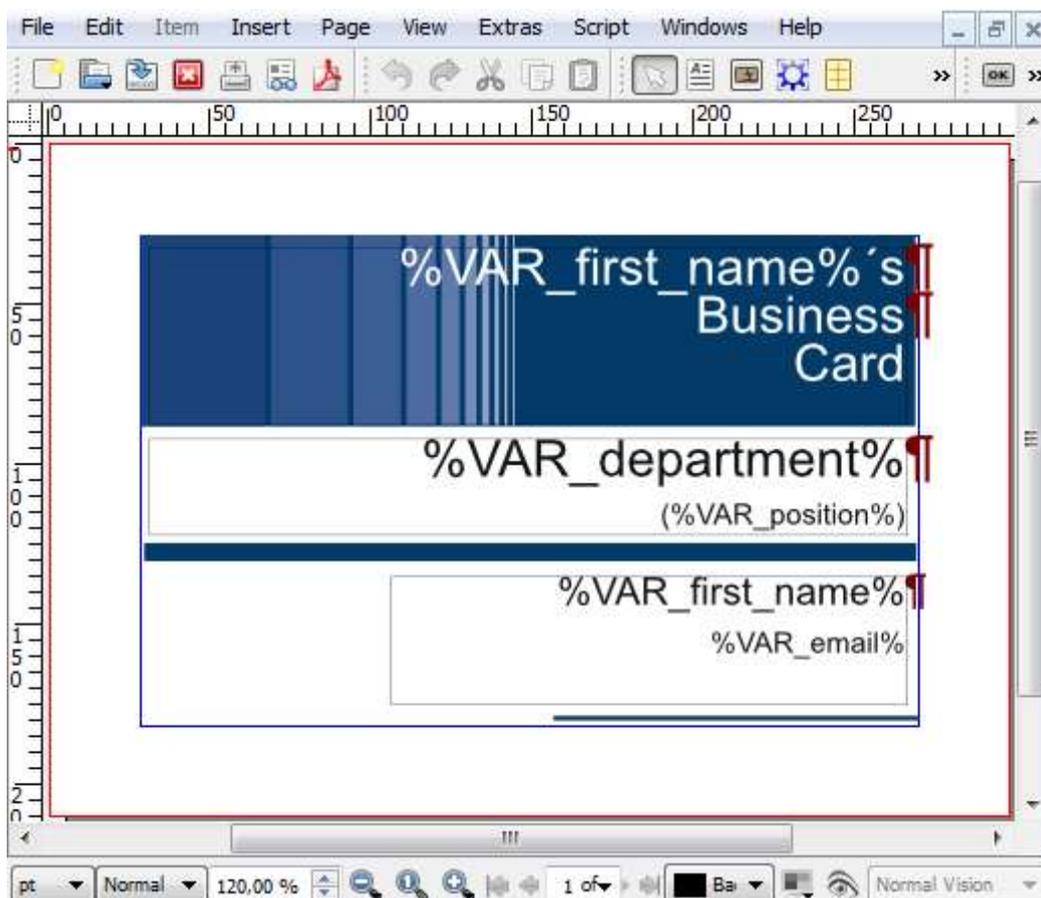
Wir empfehlen, ScribusGenerator in MacOSX über die [Befehlszeile](#) auszuführen, da die grafische Oberfläche von Scribus Generator erfordert, dass Tkinter in Ihrem Python-Setup installiert ist, was unter MacOSX schwierig sein kann.

So verwenden Sie Scribus Generator

Erstellen Sie Ihre Scribus "Template" -Datei

Erstellen und gestalten Sie Ihre Scribus-Datei wie jede andere. An den Stellen, an denen Sie Text durch Daten ersetzen müssen, fügen Sie einfach hinzu, wo sich die Spalte Ihrer Datendatei befindet. Schauen Sie sich unsere [Galerie mit Beispielen](#) an, um zu sehen, was andere getan haben. [Jede vorhandene Scribus-Datei](#) kann als Vorlage verwendet werden. `%VAR_name% name`

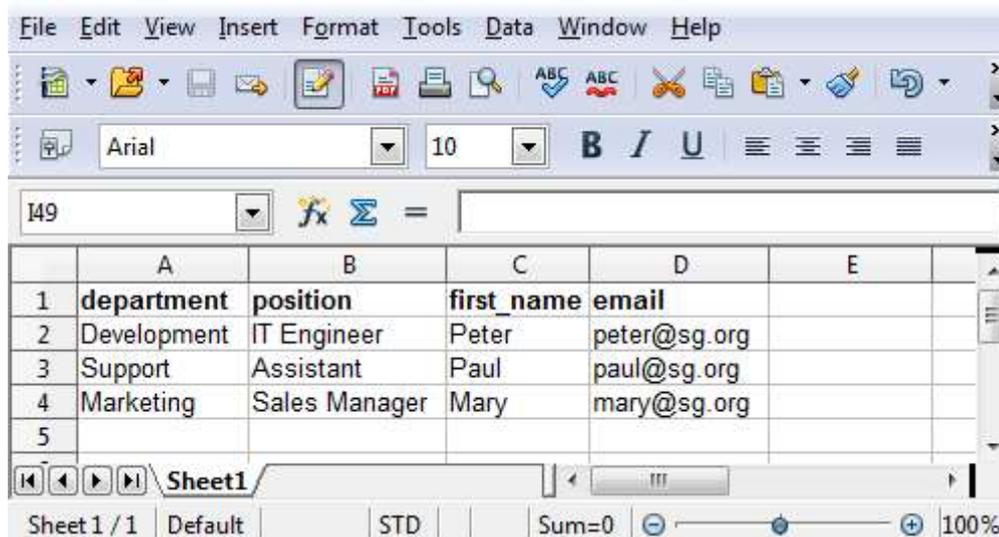
Sie können die Variable an einer beliebigen Position innerhalb eines Textrahmens platzieren. Wenden Sie das gesamte Format und den gesamten Stil auf die Variable an, die Sie auf den endgültigen Text anwenden möchten.



Wenn Sie eine Seite (oder viele) für jede Dateneingabe generieren möchten, die Sie abgeschlossen haben, herzlichen Glückwunsch! Wenn Sie lieber viele Dateneinträge auf einer einzigen Seite anzeigen möchten, fügen Sie einfach den Text **vor jedem Eintrag hinzu, aber den ersten**: ScribusGenerator lädt den neuen Datensatz automatisch, sobald er dieses Token erkennt, siehe [die entsprechende Dokumentationsseite](#). %SG_NEXT-RECORD%

Erstellen Sie Ihre (CSV-) Datendatei

Scribus Generator erwartet eine CSV-Datei (*Comma Separated Values*), die mit einem Standard-Tabellenkalkulationseditor (wie [LibreOffice](#), Excel oder GoogleDoc) sehr einfach zu erstellen ist: Geben Sie die Daten ein und speichern / exportieren Sie sie als CSV. Stellen Sie einfach sicher, dass Ihre CSV-Datei in UTF-8 codiert ist, um einen vollständigen Zeichensatz zu haben (Akzente, Brailleschrift, Mathematik, Kyrillisch, Symbole,...).



	A	B	C	D	E
1	department	position	first_name	email	
2	Development	IT Engineer	Peter	peter@sg.org	
3	Support	Assistant	Paul	paul@sg.org	
4	Marketing	Sales Manager	Mary	mary@sg.org	
5					

Es ist wichtig sicherzustellen, dass die Spalten die gleichen Variablen () haben, auf die Sie in der von Ihnen entworfenen Scribus-Vorlagendatei verweisen. name %VAR_name%

Wir empfehlen, in UTF-8-Codierung zu speichern, um den vollständigen Satz akzentuierter Zeichen, Chinesisch, Kyrillisch, mathematische Symbole, Pfeile, Brailleschrift, Symbole und [viele mehr](#) zu aktivieren. Kopieren Sie einfach diejenigen, für die Ihnen eine Tastaturkombination fehlt.

Das Exportieren gut formatierter CSV-Dateien in UTF-8-Codierung ist mit OpenOffice oder LibreOffice Calc kinderleicht, mit Excel weniger. Wenn Sie Microsofts Excel verwenden, könnten Sie an diesem kostenlosen Add-In interessiert sein, das gute Export- / Importfunktionen bietet: <http://www.csvio.net/>

CSV-Dateien können leicht aus vielen vorhandenen Datenquellen generiert werden (inkl. ETL-Plattformen der Enterprise-Klasse, den meisten Datenbanken wie MySQL, PostgreSQL, SQLite3 und mehr), siehe unsere Wiki-Seite für die Verwendung [anderer Datenquellen](#)

Ausführen des Generatorskripts - Einstellungen

Führen Sie das Skript über das Menü aus: und starten Sie `. Script > execute Script ScribusGenerator.py`

Im Skript-Dialog können Sie die Eingabe- und Ausgabeeinstellungen für **Scribus Generator** konfigurieren.

The screenshot shows the Scribus Generator dialog box with the following settings:

- Input Settings:**
 - Scribus File: `C:/projects/cards/businessCard.sla` (with file explorer and refresh icons)
 - Data File: `C:/projects/cards/businessCard.csv` (with file explorer icon)
 - Data Field Separator: `,` (opt.) use partial data, only from: to:
- Output Settings:**
 - Output Directory: `C:/projects/cards/out` (with file explorer icon)
 - Output File Name: `card-%VAR_name%` Save Settings:
 - Merge in Single File: Output Format: `PDF` (dropdown) Keep Scribus Files:

Buttons at the bottom: **Generate** (with checkmark), **Cancel** (with X), and **Help** (with question mark).

Eingabeeinstellung	Erklärung
Scribus-Datei	Wählen Sie die Scribus-Datei aus, die die zu ersetzenden Variablen enthält. Klicken  , um einen Datei-Explorer zu öffnen, und  , um die Scribus Generator-Einstellungen aus dieser Datei zu laden.
Datei	Wählen Sie die Datendatei aus, die die durch Kommas getrennten Werte enthält. Klicken  , um einen Datei-Explorer zu öffnen.
Datenfeld-Trennzeichen	Zeichen, das das Datenfeld in Ihrer CSV-Datei löscht, standardmäßig Komma (,).
Von - Bis	Um Scribus Generator für eine Teilmenge Ihrer Daten auszuführen, erwähnen Sie die Anfangs- und / oder letzte Zeile dieser Teilmenge, ohne die Kopfzeile zu zählen. Lassen Sie einfach leer, um von Anfang an (oder bis zum letzten) Dateneingabe zu generieren.

Ausgabeeinstellung	Erklärung
--------------------	-----------

Ausgabeeinstellung	Erklärung
Ausgabeverzeichnis	Wählen Sie den Pfad zu einem vorhandenen Verzeichnis aus, in dem das Ergebnis gespeichert werden soll. Klicken  , um einen Datei-Explorer zu öffnen.
Name der Ausgabedatei	Sie können die gleichen Variablen verwenden, die in der Scribus-Datei/Datendatei definiert sind. Sie können die Variablen auch mit anderen Zeichen mischen. Wenn Sie das Feld leer lassen, wird ein Index verwendet. Der Index/die resultierenden Dateien werden nach dem Vorkommen von Zeilen in der Datendatei sortiert.
Einstellungen speichern	Speichern Sie die aktuellen Scribus Generator-Einstellungen in der Quell-SLA-Datei für die zukünftige Verwendung.
Zusammenführen in einer einzigen Datei	Wählen Sie diese Option, um eine einzelne Ausgabedatei (SLA und/oder PDF) zu generieren, die alle Datenzeilen kombiniert.
Ouput-Format	Wählen Sie das Ausgabeformat der generierten Dateien, entweder PDF oder Scribus.
Scribus-Dateien aufbewahren	Wählen Sie diese Option, um die generierten Scribus-Dateien beizubehalten. Andernfalls werden sie nach der PDF-Generierung gelöscht. Diese Option hat keine Auswirkung, wenn Sie das Scribus-Ausgabeformat wählen.

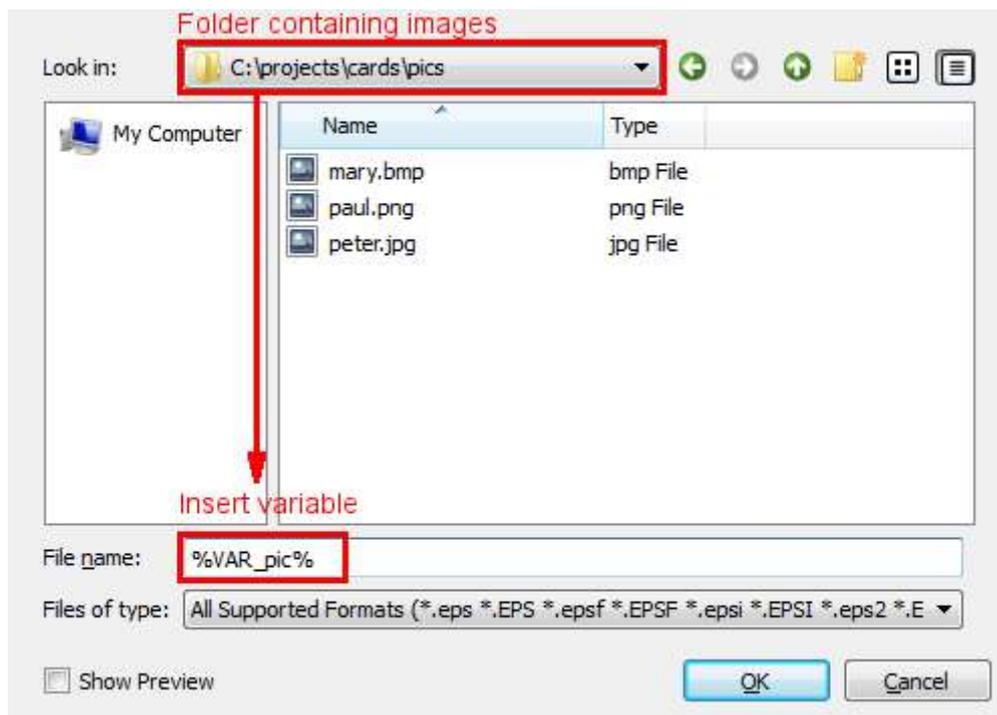
Zusätzliche (technischere) Optionen können eingestellt werden, um die automatische Aufzeichnung von Scribus Generators-Aktionen in Ihrem System durch Bearbeiten der Datei anzupassen. Sie können beispielsweise den Speicherort der Protokolldatei (standardmäßig) in ein Verzeichnis verschieben, das keine Administratorrechte zum Bearbeiten in Windows benötigt (), oder die Dateiprotokollierung durch Ihren Standard-Systemprotokollierer ersetzen (SysLogHandler unter Linux, NTEventLogHandler unter Windows). Alle Einstellungen (und mehr) werden in der [Dokumentation zur Python-Protokollierung](#) beschrieben. `logging.conf scribusGenerator.log C:\tmp\scribusGenerator.log`

Dynamische Bilder

Bildreferenzen können auch dynamisch mit **Scribus Generator** geändert werden. Sehen Sie sich das [Screencast-Video @6:10](#) an.

Alle von **Scribus** unterstützten Bilder können verwendet werden. Um jedoch mit ScribusGenerator zu arbeiten, müssen sie sich alle in einem einzigen Ordner befinden, der alle Bilder enthält. Dieser Ordner kann sich an einer beliebigen Stelle auf Ihrem PC befinden. Duplizieren Sie ein beliebiges Bild in diesem Ordner und benennen Sie es um (und ähnlich für jeden anderen Variablennamen, den Sie für Bilder verwenden müssen,). %VAR_pic% %VAR_pic2%, %VAR_photo%

Fügen Sie einen Bilderrahmen an einer beliebigen Stelle in der Scribus-Datei hinzu («Scribus → Insert → Insert Image Frame») und öffnen Sie den Dialog zum Abrufen eines Bildes (z. B. Rechtsklick auf Bilderrahmen → Bild abrufen... auf ausgewähltem Rahmen). Wählen Sie das gewünschte "Platzhalter"-Bild aus.



Die Bilddatei kann wie jede andere zuvor beschriebene Variable definiert werden. Es muss nur eine Spalte mit einem Spaltennamen vorhanden sein, der dem Variablennamen in der Scribus-Vorlagendatei entspricht. Stellen Sie nur sicher, dass Sie die Dateierweiterung entweder im Dateinamen oder in den CSV-Daten behandeln, aber nicht in beiden.

	A	B	C	D	E
1	department	position	first_name	email	pic
2	Development	IT Engineer	Peter	peter@sg.org	peter.jpg
3	Support	Assistant	Paul	paul@sg.org	paul.png
4	Marketing	Sales Manager	Mary	mary@sg.org	mary.bmp
5					

Das Importieren von Vektorbildern (SVG, PDF oder andere) als Bildrahmen funktioniert in ScribusGenerator, genau wie bei PNG- oder JPG-Bildern. Stellen Sie einfach sicher, dass Ihr (relativer oder absoluter) *SVG-Dateipfad mit dem **Ausgabeverzeichnis des Generators** übereinstimmt*, da dies der Ort ist, von dem aus Scribus beim Transformieren des SLA in Ihr PDF-Format suchen wird.

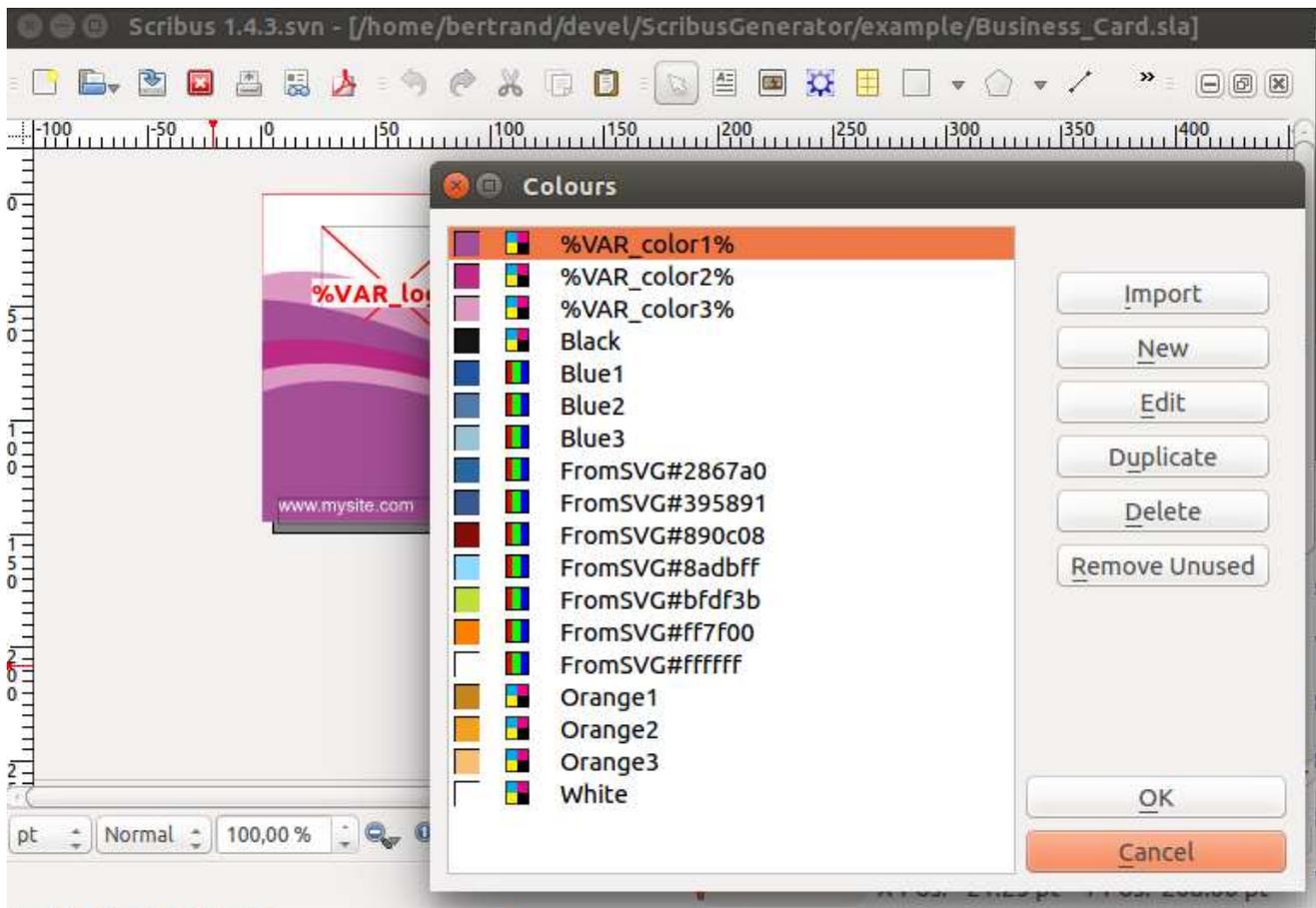
Scribus rendert manchmal das enthaltene Bild in wirklich niedriger Auflösung, daher sollten Sie die Auflösung (dpi) (und / oder Größe) oder Ihr Quellmaterial überprüfen. Um SVG-Objekte in mehreren Auflösungen schnell zu exportieren, könnte das [Objektexportskript von Inkscape](#) interessiert sein.

Es gibt leider keine Möglichkeit, dynamische Vektordateien (pdf oder svg) mit ScribusGenerator einzubinden, da Scribus sie importiert (aus dem Menü Datei > Import > Vector), indem es sie im Handumdrehen in Scribus-Objekte konvertiert. Das ist ideal für die direkte Bearbeitung in Scribus, aber nicht für das Einbinden externer Dateien als Referenz.

Dynamische Farben

Farben können wie Text dynamisch ersetzt werden. Sehen Sie sich das [Screencast-Video @8:31 an](#) oder einfach

1. Bearbeiten Sie die Farben Ihrer Scribus-Datei () und benennen Sie die Farben um, die Sie durch Variablennamen ersetzen möchten (normalerweise durch die jetzt bekannte Form ersetzen). `edit > colours FromSVG#whatever %VAR_name%`
2. Definieren Sie die Farben, die Sie im endgültigen Dokument verwenden möchten, und verwenden Sie ihre Scribus-Namen als Werte in Ihrem Farbdatenfeld.

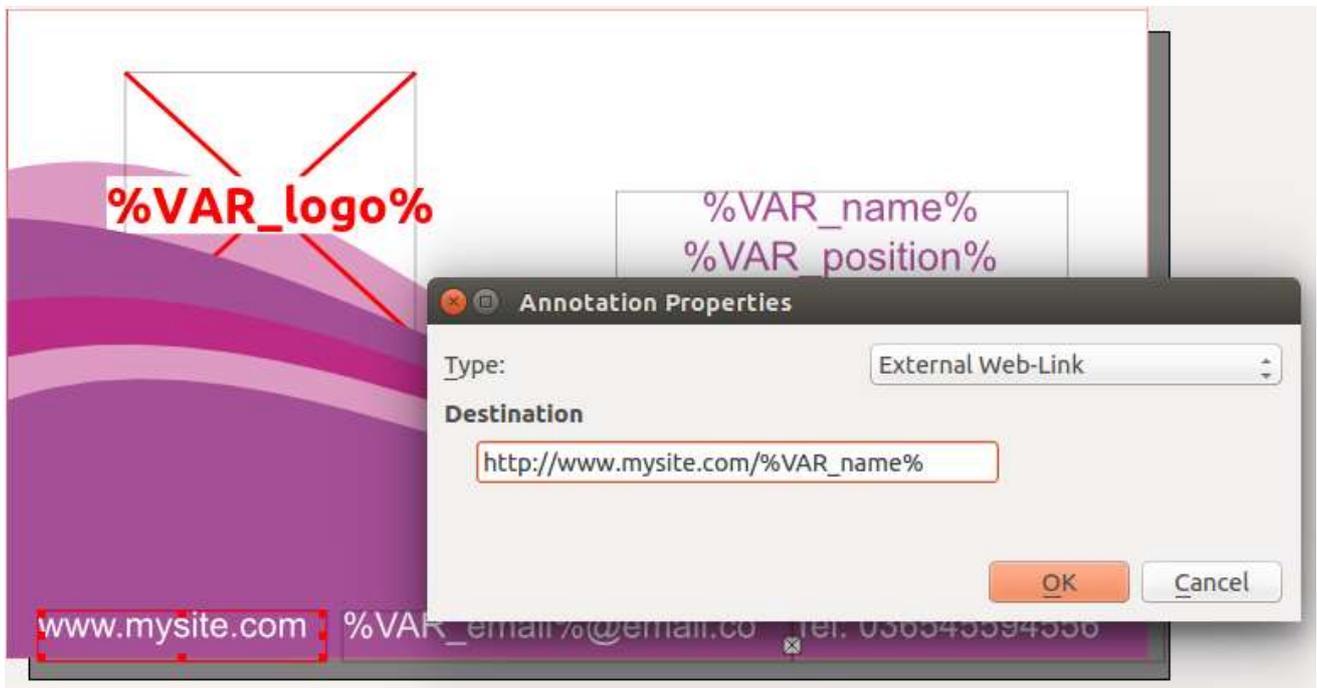


Verwenden Sie dies zusammen mit den vielen bekannten Farbpaletten, die direkt in Scribus enthalten sind, um Ihr Dokumentendesign reichhaltig und ansprechend zu gestalten!

Dynamische Links

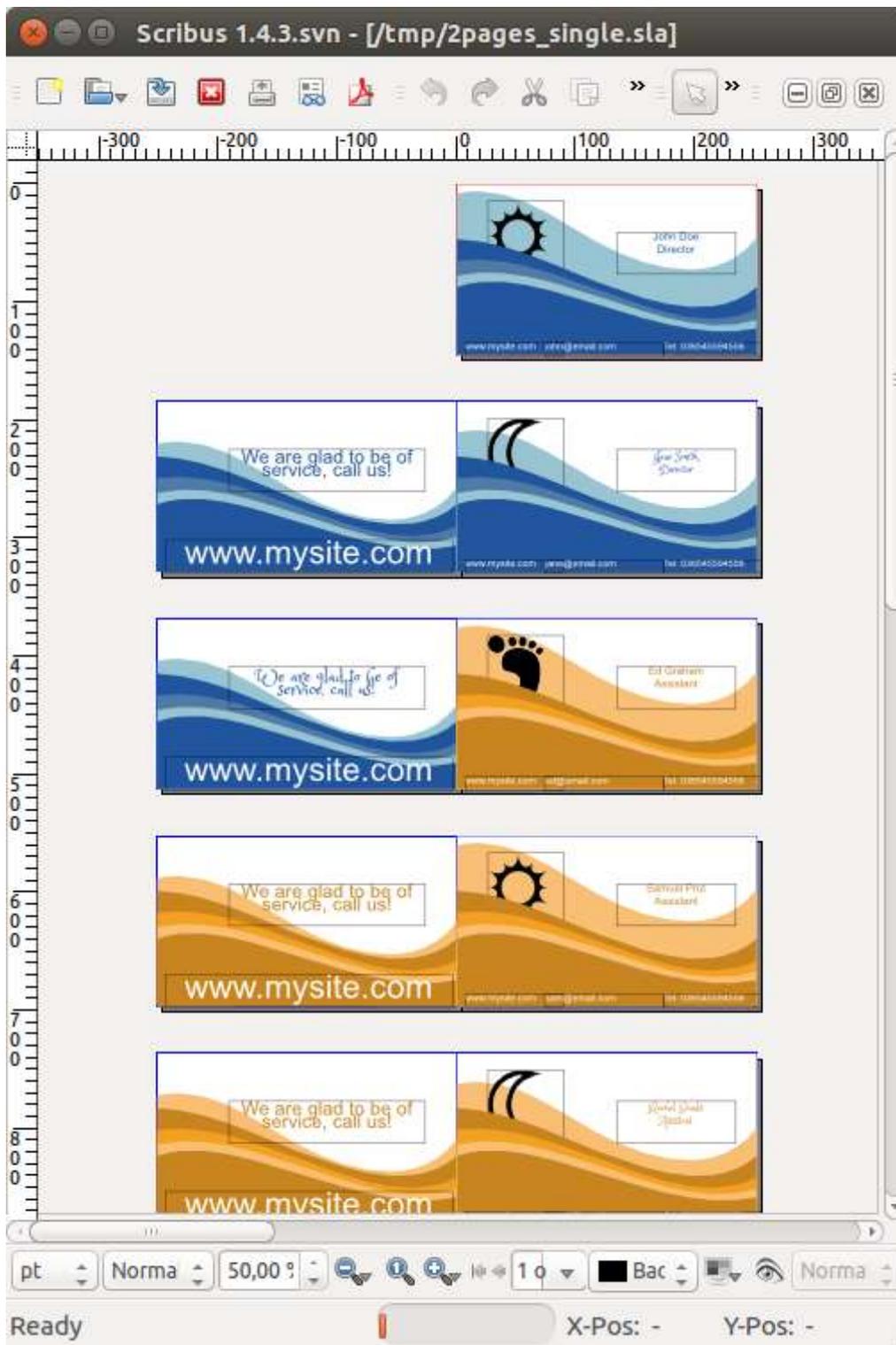
In 2 Schritten kann auf nahezu jedem Scribus-Objekt ein anklickbarer (Web-)Link eingefügt werden:

1. right-click > PDF Options aktivieren Is PDF Annotation
2. right-click > PDF Options > Annotation Properties , wählen Sie Typ und geben Sie die Ziel-URL ein, wobei Sie natürlich auch Variablen verwenden können, die ersetzt werden sollen, wie unten dargestellt. External Web-Link



Seriendruckausgabe - einzelnes Dokument

Anstatt eine einzelne Datei (SLA oder PDF) für jede Datenzeile zu generieren, können Sie eine einzelne Datei generieren, die all dies zusammenführt. Wählen Sie einfach die Option entsprechend aus, um das unten dargestellte Ergebnis in einer einzigen Scribus- (und/oder PDF-) Datei zu erhalten. Wenn Ihr Dokumentlayout mehrere Seiten umfasst (doppelseitiges oder gefaltetes Merkblatt), ist es wichtig, dass Scribus Generator die Dateien zusammenführen kann, für die Ihre Vorlage die gleiche Anzahl von Seiten (oder ein Vielfaches davon) hat. Für ein 3-faches Dokument sollte Ihre Quell-SLA beispielsweise 3, 6 oder 9 Seiten (oder ein beliebiges Vielfaches von 3) haben.

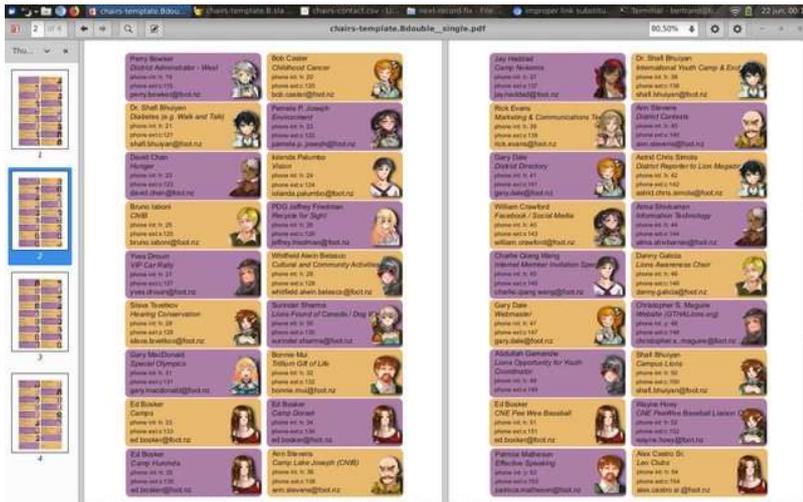


Mehrere Datensätze auf einer einzigen Seite

Mit **Scribus Generator** können Sie beliebige Dateneinträge in einem einzigen Dokument anzeigen. Großartig, um Ihre eigenen Angebote, Teamdiagramme, Spielkarten, Who's Who-Poster und mehr zu generieren.

Legen Sie einfach den Text in Ihrem Dokument *vor jeder Dateneingabe mit Ausnahme des ersten Datensatzes* ab, der unserem [Beispieldokument](#) folgt. Bitte beachten Sie, dass bei diesem Token die Groß-/Kleinschreibung beachtet wird und es in *Großbuchstaben* geschrieben werden muss. Scribus Generator lädt automatisch den nächsten Datensatz, sobald er es erkennt und wenn es das Ende Ihrer Vorlage erreicht. %SG_NEXT-RECORD% %SG_NEXT-RECORD%

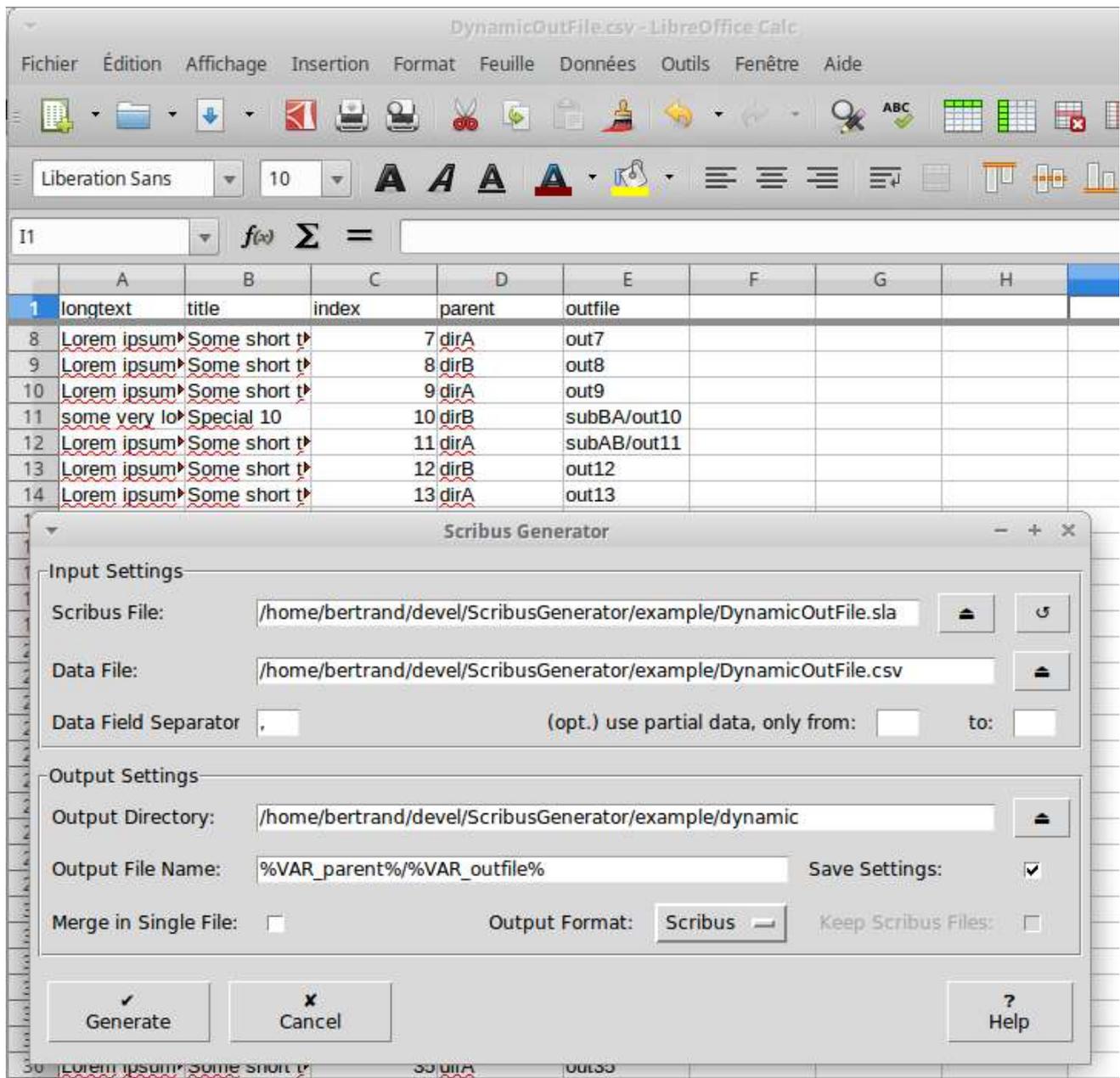
Die Funktion, mehrere Datensätze auf einer einzigen Seite (oder einem Dokument) aufzunehmen, funktioniert großartig, ist aber fehleranfällig, bitte lesen (und verbessern) Sie unsere [spezielle Wiki-Seite für mehrere Datensätze, SG_NEXT-RECORD](#).



Speicherort der dynamischen Ausgabedatei

Mit [Scribus Generator](#) können Sie den Namen (und Speicherort) des generierten Dokuments einfach anpassen. Fügen Sie den gewünschten Ausgabedateinamen in Ihren Daten hinzu und verwenden Sie die entsprechende Variable (oder Kombination mehrerer Variablen) im Feld "Name der Ausgabedatei". Bitte beachten Sie, dass dieser Dateiname immer relativ zum Ausgabeverzeichnis ist und *keine Erweiterung* hat.

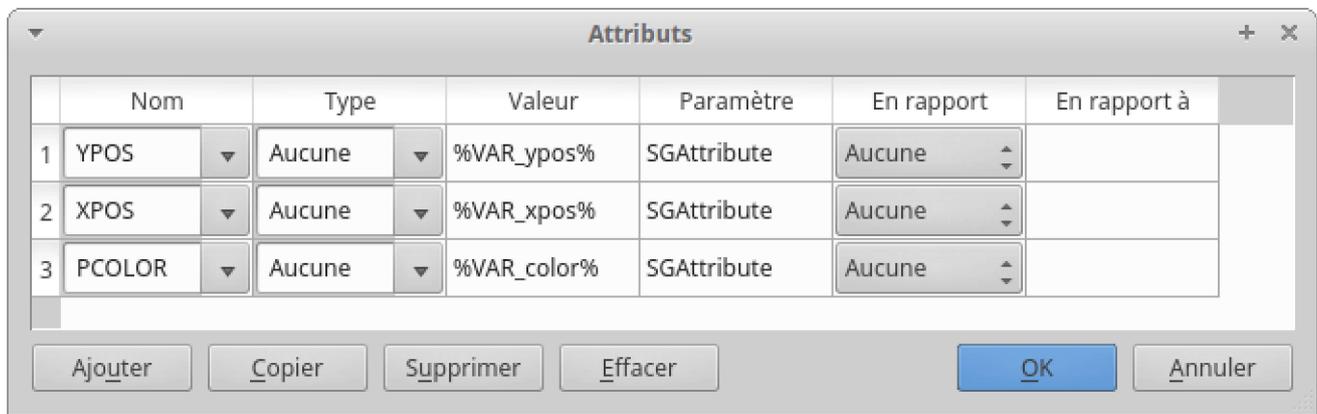
Werfen Sie einen Blick auf die Kombination von Werten von und in unserem [Beispieldatensatz](#) und der zugehörigen [Beispielvorlage](#). parent outfile



Fortgeschrittenere Anwendungen

Scribus Generator ermöglicht es technisch versierteren Benutzern, die generierten Dokumente noch mehr anzupassen und praktisch jedes Attribut eines Objekts zu ändern, z. B. die Füll- oder Konturfarbe einer Zeile, die Farbe eines Textes, eine Linienstärke, eine Objektposition,... Siehe das [Screencast-Video @13:13](#).

Um beispielsweise die *Schriftart* eines Objekts dynamisch zu ändern, fügen Sie ihm ein *Attribut* hinzu («right-click on object → Attributes → Add»). Das **Parameter-Attribut** muss auf `sgAttribute`, seinen Namen auf die **Objekteigenschaft**, die Sie ändern möchten (in diesem Fall), und seinen Wert auf den gewünschten dynamischen **Wert** (in der Regel einige) festgelegt werden. Ihre Datendatei (CSV) sollte dann den Schriftnamen enthalten, der in der Spalte der angegebenen Variablen verwendet werden soll, z. B. "Arial Regular" oder "Courier New Bold". `FONT %VAR_name%`



Um die Eigenschaften eines Unterelements (z. B. einer bestimmten Textzeile in einem Textrahmen) zu ändern, können Sie das **Feld RelationshipTo** verwenden, um zu definieren, welche Unterelemente die neue Eigenschaft erhalten sollen. Verwenden Sie diese Option, um alle direkten untergeordneten Elemente zu ändern, oder einen anderen [vereinfachten XPATH-Ausdruck](#), um nur eine Teilmenge zu ändern. *

Ausgewählte Beispiele für SGAttributes:

Name	Wert	Parameter	BeziehungZu	Erklärung
FONT	%VAR_font%	SGAttribute	//ITEXT[2]	Schriftart der 2D-Textzeile in einem Rahmen, wie "Arial Regular"
FONTSIZE	%VAR_size%	SGAttribute	//ITEXT	Textgröße aller Textzeilen in einem Rahmen, z. B. "14"
LINESP	%VAR_spacing%	SGAttribute	//para[last()]	Fester Zeilenabstand des letzten Absatzes in einem Frame, z. B. "9.5"
YPOS	%VAR_top%	SGAttribute		Rand von oben für vertikale Elementposition, wie "22.04"
ROT	%VAR_degrees%	SGAttribute		Drehung des aktuellen Objekts, in Grad, [0 , 359]

Bitte beachten Sie, dass ScribusGenerator das Attribut nicht erstellt, sondern nur versucht, seinen Wert zu aktualisieren. Möglicherweise müssen Sie das Objekt also leicht drehen (geben Sie einfach einen beliebigen Wert außer 0 in das Drehfeld ein, es wird sowieso ersetzt), um sicherzustellen, dass das Attribut tatsächlich von Scribus beim Speichern der Datei verwendet wird. ROT

Änderungen in der SGAttribute-Syntax

Seit ScribusGenerator v2.7 (veröffentlicht im April 2018) wurde die Syntax der SGAttributes leicht geändert, um Scribus 1.5.3 zu unterstützen. Wenn Sie Vorlagen für eine ältere Version des Skripts geschrieben haben, aktualisieren Sie bitte Ihre Vorlagen gemäß der obigen Dokumentation.

Dies kann mit den klassischen Scribus Editors-Fenstern erfolgen, indem Sie einfach Ihre Werte von nach verschieben und auf

setzen. Parameter RelationshipTo Parameter SGAttribute

Wenn Sie über viele solcher Dateien oder Attribute verfügen, wird ein Skript vorgeschlagen, um diese Aktualisierung automatisch vorzunehmen. Führen Sie es von der Befehlszeile aus wie folgt aus, es erstellt automatisch eine Sicherungskopie Ihrer alten Dateien:

```
python ./ConvertSGAttributesToSG27.py *.sla
```

Ausführen von Scribus Generator über die Befehlszeile

Es ist möglich, Scribus Generator über die Befehlszeile auszuführen, und es ist schnell! Großartig, um Ihren Workflow zu automatisieren oder mit anderen Tools zu integrieren.

Bitte beachten Sie, dass nur die SLA-Generierung über die Befehlszeile funktioniert. Die PDF-Generierung wird im Moment nicht über die Scribus Generator-Befehlszeile unterstützt.

Alle benötigten Informationen finden Sie in der Skripthilfe: `./ScribusGeneratorCLI.py --help`

positional arguments:

infiles SLA file(s) to use as template(s) for the generation, wildcards are supported

optional arguments:

-h, --help show this help message and exit
-c CSVFILE, --csvFile CSVFILE
CSV file containing the data to substitute in each template during generation. Default is scribus source file(s) name with "csv" extension instead of "sla". If csv file is not found, generation from this particular

```

        template is skipped.
-d CSVDELIMITER, --csvDelimiter CSVDELIMITER
        CSV field delimiter character. Default is comma: ","
-n OUTNAME, --outName OUTNAME
        name of the generated files, with no extension.
        Default is a simple incremental index. Using SG variables
        is allowed to define the name of generated documents.
-o OUTDIR, --outDir OUTDIR
        directory where generated files are stored. Default is
        the directory of the scribus source file. outputDir
        will be created if it does not exist.
-m, --merge, --single
        generate a single output (SLA) file that combines all
        data rows, for each source file.
-from FIRSTROW, --firstrow FIRSTROW
        Starting row of data to merge (not counting the header
        row), first row by default.
-to LASTROW, --lastrow LASTROW
        Last row of data to merge (not counting the header
        row), last row by default.
-s, --save
        Save current generator settings in (each) Scribus
        input file(s).
-l, --load
        Load generator settings from (each) Scribus input
        file(s). Overloads all default values, not provided
        command line arguments.

```

requirements

This program requires Python 2.7+

examples:

```
ScribusGeneratorCLI.py my-template.sla
generates Scribus (SLA) files for each line of 'my-template.csv'
by substituting the provided values into 'my-template.sla' to the
current directory.
```

```
ScribusGeneratorCLI.py --outDir "/home/user/tmp" example/Business_Card.sla
generates Scribus files for each line of example/Business_Card.csv
in the "/home/user/tmp" subdirectory.
```

```
ScribusGeneratorCLI.py --outName "card_%VAR_email%" */*.sla
generates Scribus files for each sla file in any subdirectory
that has a csv file with a similar name in the same directory.
Generated files will have a name constructed from the "email" field
data, and are stored in their respective sla file directory.
```

```
ScribusGeneratorCLI.py --single -c translations.csv -n doc_ lang/*.sla
generates a single Scribus file for each sla file in the lang/ subdirectory
using all rows of the translations.csv data file.
Generated files will have a name constructed from the "doc_" prefix
```

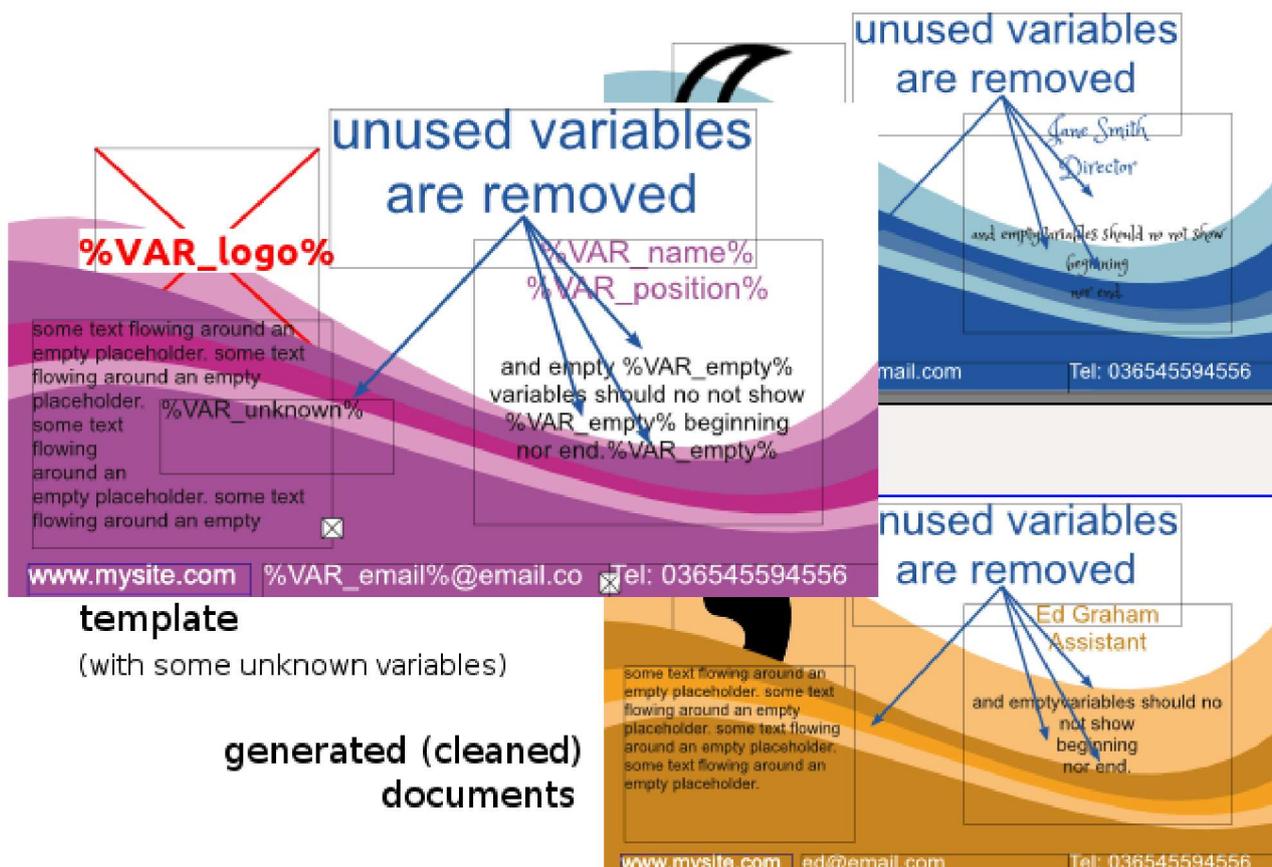
☰ README.md

more information: <https://github.com/berteh/ScribusGenerator/>

Mehr Details

Sauberer Output

Scribus Generator entfernt nicht verwendete Variablen aus den generierten Dokumenten, zusammen mit ihrem Textrahmen, wenn dieser Textrahmen nur Variablen enthielt.



Wenn Sie diese nicht verwendeten Variablen und leeren Texte beibehalten möchten, ändern Sie einfach die Standardeinstellung entsprechend in , Einstellung auf

0. ScribusGeneratorBackend.py CLEAN_UNUSED_EMPTY_VARS

Wenn diesen nicht verwendeten Variablen oder leeren Texten ein einfacher Text *vorangestellt* wurde (z. B. ein einzelner Zeilenvorschub oder ein Zeichen wie eines von), werden diese listenartigen Trennzeichen entfernt, sodass Sie eine saubere Enumaration erhalten können, indem Sie Ihre Variablen auf einfache Weise wie , oder in einem tabellarischen Layout mit (einzelnen) Zeilenvorschüben anhängen: , ; - %VAR_n1%, %VAR_n2%, %VAR_n3%, %VAR_n4%.

The following will show nicely even if some variable is empty or missing:

```
%VAR_n1%
%VAR_n2%
%VAR_n3%
%VAR_n4%
```

Wenn Sie diese Trennzeichen beibehalten möchten, ändern Sie einfach die Standardeinstellung entsprechend in `,` Einstellung auf

`0`. `ScribusGeneratorBackend.py REMOVE_CLEANED_ELEMENT_PREFIX`

Zeilenumbrüche und **Tabellierungen** in Ihren CSV-Daten werden durch das Scribus-Äquivalent (Zeilenumbrüche und Tabellierungen) ersetzt. Um sie zu entfernen und in einfache Leerzeichen zu verwandeln, setzen Sie die Einstellung auf `0`. `KEEP_TAB_LINEBREAK`

Protokollierung und Debuggen

ScribusGenerator zeichnet alle seine Aktionen in einer Protokolldatei auf, die sich (standardmäßig) **in Ihrem Benutzerverzeichnis (Home-Verzeichnis)** befindet. Wenn Sie auf ein unerwartetes Verhalten stoßen, lesen Sie den Inhalt von `,` um mehr zu erfahren. Sie können die Protokollierungseinstellungen in `logging.conf` ändern (weitere Optionen finden Sie unter [Python-Protokollkonfiguration](#)). `.scribusGenerator.log`

Bitte kopieren Sie die relevanten (normalerweise letzten) Zeilen von Ihnen, wenn Sie [ein Problem melden](#) möchten. `.scribusGenerator.log`

Variablennamen

Versuchen Sie, einfache Wortzeichen (`()`) für Variablennamen ohne Leerzeichen oder andere Sonderzeichen (wie `'&'`) zu verwenden. Z.B. verwenden `und` anstelle von `und` . Die Spalten der Datendatei (CSV) wären dann `und` . `A-Za-a0-`

```
9_ %VAR_first_name% %VAR_zip_code% %VAR_first
name% %VAR_&zip#code% first_name zip_code
```

Dies ist nur für Variablennamen in der Scribus-Datei und *Spaltennamen* der Datendatei wichtig. Die Daten FIELDS (die Zeilen des CVS) dürfen natürlich BELIEBIGE Zeichen enthalten.

Datenbankquelle

Um Daten aus einer Datenbank anstelle einer (manuellen) Tabelle zu verwenden, können Sie das zugehörige Abfrageergebnis einfach in eine CSV-Datei exportieren. Im Folgenden finden Sie einige Beispiele für gängige Datenbank-Engines. Erfahren Sie mehr über die Verwendung externer Datenquellen in unserem [Wiki](#) .

Mysql:

```
mysql --delimiter="," -u myuser -p mydb -e "select f1,f2 from mytable" >
/tmp/mydata.txt
```

oder

```
mysql -u myuser -p mydatabase -e
"select field1 , field2 FROM mytable INTO OUTFILE
```

```
'/tmp/myfilename.csv' FIELDS TERMINATED BY ','  
ENCLOSED BY '\"' LINES TERMINATED BY '\n' "
```

Mehr über INTO OUTFILE bei <http://dev.mysql.com/doc/refman/5.1/en/select.html>

Postgresql

Nicht ausgerichteten Ausgang mit dem Schalter umschalten, aktivieren Sie ein Komma als Trennzeichen mit . Senden Sie die Ausgabe an eine Datei mit , und fragen Sie dann Ihre Datenbank ab. \a \f , \o myfile.csv

SQLITE3

Sie können in der Befehlszeile oder in der interaktiven Shell von sqlite verwenden `sqlite3 -csv .mode csv`

Skripten

Verschiedene Skripte sind im Ordner verfügbar, um Ihnen die Verwendung von Scribus Generator zu erleichtern. Schauen Sie sich die [zugehörige README](#) an ! `utils`

Bekannte Probleme

Mac OSX-Fehlerbehebung

Einige Installationen von Python unter Mac OSX liefern kein funktionierendes Tkinter-Paket aus, das für die ScribusGenerator-GUI erforderlich ist. Finden Sie entweder einen Weg, eine kompatible TCL / Tk-Umgebung einzurichten, oder verwenden Sie einfach die [Scribus Generator-Befehlszeilenschnittstelle](#).

Wenn Sie eine weitere GUI zu Scribus Generator beitragen möchten, die unter Mac OSX funktioniert, zögern Sie nicht! Einfach [verzweigen](#), [verzweigen](#), [programmieren](#) und eine [Anfrage ziehen](#), um Ihren Beitrag überprüfen zu lassen!

Unterstützen

Schauen Sie sich das [Wiki](#) für weitere Ideen an, schauen Sie sich die [gelösten](#) und [offenen Probleme](#) an und melden Sie dann freundlicherweise ein [Problem](#) online und kopieren Sie dort die letzten Zeilen Ihrer Protokolldatei (es ist in Ihrem Benutzerverzeichnis (Home), um den Grund für dieses schlechte Verhalten zu finden, zusammen mit einer kurzen Erklärung Ihres Problems. `.scribusGenerator.log`

Lizenz

The MIT License

Copyright

(c) 2011, Ekkehard Will (www.ekkehardwill.de) (c)

2014-2021, Berteh (<https://github.com/berteh/>)

Hiermit wird jeder Person, die eine Kopie dieser Software und der zugehörigen Dokumentationsdateien (die "Software") erhält, die Erlaubnis erteilt, mit der Software uneingeschränkt zu handeln, einschließlich, aber nicht beschränkt auf die Rechte zur Nutzung, Vervielfältigung, Änderung, Zusammenführung, Veröffentlichung, Verbreitung, Unterlizenzierung und/oder zum Verkauf von Kopien der Software, und Personen, denen die Software zur Verfügung gestellt wird, zu gestatten, dies zu tun, unter folgenden Bedingungen:

Der obige Urheberrechtshinweis und dieser Genehmigungshinweis sind in allen Kopien oder wesentlichen Teilen der Software enthalten.

Die Software wird "wie besehen" bereitgestellt, ohne jegliche ausdrückliche oder stillschweigende Gewährleistung, einschließlich, aber nicht beschränkt auf die Gewährleistung der Marktgängigkeit, der Eignung für einen bestimmten Zweck und der Nichtverletzung von Rechten. In keinem Fall haften die Autoren oder Urheberrechtshaber für Ansprüche, Schäden oder andere Haftungen, sei es in einer Vertragshandlung, unerlaubten Handlung oder anderweitig, die sich aus, aus oder in Verbindung mit der Software oder der Nutzung oder dem sonstigen Handel mit der Software ergeben.

Auslösungen 15

 3.0 Neueste
on 16 Feb

+ 14 Veröffentlichungen

Pakete

Keine Pakete veröffentlicht

Verwendet von 3



@daybugging / **pcbis2pdf**



@fundevogel / **scribus-toolbox**

Beitragende 7



Sprachen

- Python 100.0%