Erzeugen von PDF-Dateien ohne ODS: Das Makro Lst2Pdf

Stefan Beimel
Merz Pharmaceuticals GmbH
Eckenheimer Landstr. 100
600318 Frankfurt / Main
stefan.beimel@merz.de

Zusammenfassung

SAS/ODS PDF zeigt viele Schwächen bei der Erzeugung von PDF-Dateien, z. B. die unzureichende Gestaltung der Lesezeichen und die Schwierigkeit, die PDF-Dateien nach ihrem Erzeugen automatisch zusammen zu fassen oder zu formatieren. Letzteres liegt in der Natur von PDF-Dateien. Gerade bei der Auswertung klinischer Studien besteht aber Bedarf, sehr viele Ausgabe-Dateien, d. h. sowohl Text als auch Grafiken, in einem wohlformatierten Dokument möglichst ohne manuelle Eingriffe zu sammeln. Dazu ist SAS/ODS PDF ungeeignet.

Das SAS Makro Lst2Pdf wandelt SAS Text Dateien (traditioneller SAS output) und PostScript-Dateien, die mit SAS/Graph erzeugt wurden, ohne ODS in PDF-Dateien um. Der Benutzer hat dabei über Makroparameter etliche Gestaltungsmöglichkeiten:

- Schriftgröße und Schriftart (hier kommen nur nicht-proportionale Schriften wie Courier oder SAS Monospace in Frage)
- Deckblattgestaltung
- Erstellung eines automatischen Inhaltsverzeichnisses
- Drucken eines Wasserzeichen (z. B. DRAFT)
- automatisches Nummerieren von Tabellen und Listen
- Kopf- und Fußzeile inkl. durchgängiger Seitennummerierung
- Lesezeichen in der PDF-Datei

Im Gegensatz zu traditionellen Methoden (Einsammeln und Formatieren in einem Textverarbeitungsprogramm, Formatierung beim Ausdruck über den Druckertreiber, Ausdruck der einzelnen Ausgabedateien,...) hat Lst2Pdf einige entscheidende Vorteile:

- Die Ausgabe besteht aus nur einer Datei und nicht, wie üblich, aus einer oder mehreren Dateien pro SAS Programm.
- Das Formatieren und Umwandeln dauert nur Sekunden.
- Es ist kein manueller Eingriff notwendig.
- Die Navigation am Bildschirm ist aufgrund der erzeugten Lesezeichen sehr einfach.
- Es wird ein automatisches Inhaltsverzeichnis erzeugt.
- Die Tabellennummerierung erfolgt automatisch, so dass Umsortierungen oder das Hinzufügen von Tabellen kein Problem ist.

Lst2Pdf 'schreibt' ein PostScript-Programm. Es greift dabei nicht auf Druckertreiber zu, sondern benutzt einfache PUT-Statements. Mit Hilfe von GhostScript, einer frei zugänglichen Software, wandelt dann Lst2Pdf diese PostScript-Dateien ins PDF-Format um. Aufgrund der PUT-Statements und dadurch, dass es GhostScript für praktisch jedes Betriebssystem gibt, ist Lst2Pdf nahezu betriebssystemunabhängig.

1 Was kann Lst2Pdf?

Bei der Auswertung einer klinischen Studie liegen viele SAS Listen, Tabellen und Grafiken vor, die in einem Dokument zusammengefasst werden müssen.

Bisher wurde großer manueller Aufwand betrieben. Das Zusammenfügen der Texte und Graphiken erfolgte in MSWord, inkl. Zeichenformatierung (die LST-Dateien enthalten keinerlei Formatierungsinformationen), Kopf- und Fußzeile, Abschnittswechsel bei Wechsel der Orientierung, Einfügen von CGM-Grafiken und Erstellung eines Inhaltsverzeichnisses.

Das Makro Lst2Pdf fasst SAS Listings und Graphiken in einem PDF-Dokument zusammen, wobei

- ein Deckblatt erstellt wird,
- Kopf- und Fußzeilen erzeugt werden,
- ein Wasserzeichen gedruckt werden kann,
- ein Inhaltsverzeichnis erzeugt wird und
- Tabellen automatisch nummeriert werden.

Es ist nur ein Makroaufruf notwendig im Gegensatz zu viel Handarbeit beim Formatieren mit einem Textverarbeitungsprogramm.

1.1 Makroparameter

Tabelle 1 zeigt sämtliche Makroparameter von Lst2Pdf. Sie soll einen Überblick darüber geben, welche Gestaltungsmöglichkeiten Lst2Pdf bietet. Eine ausführliche Beschreibung ist im User Manual [3] zu finden.

Tabelle 1: Makroparameter

parameter <optional>	**description**	**default**
infiles	input file(s) separated by blanks (no blanks in file names are allowed!) a P or L can be placed between file names to switch to portrait or landscape orientation, respectively	--
pdffile	name of PDF file to be created	--
path	path for infiles and pdffile	--
<font>	font for output	SASMono
<fontsize>	font size for included text files	8
<watermark>	text for a watermark (up to 8 letters)	
<orientation>	page orientation, may be overwritten by switches in infiles	Landscape
Table of Contents and Replacement of main table numbers		
<toc>	type of table of contents	Full
<titid>	title ID, included in single quotes, more than one separated by vertical bar	'14.' | '16.2.'
<replace>	levels of table numbers to be replaced	0
<fsttabno>	first table number	1
Layout for the pagination header		
<header>	type of header	Full

parameter <optional>	**description**	**default**
<hdreport>	1st line centered (type of report)	
<hdstudy>	2nd line centered (study number)	
<hdstatus>	3rd line centered (status of report)	
<hddate>	3rd line right (date of report)	current date
<hdpage>	Prefix for page number	
Layout for the cover page		
<cover>	cover page is printed	Yes
<cover1>	1st line on cover page	
<cover2>	2nd line on cover page	
<cover3>	3rd line on cover page	

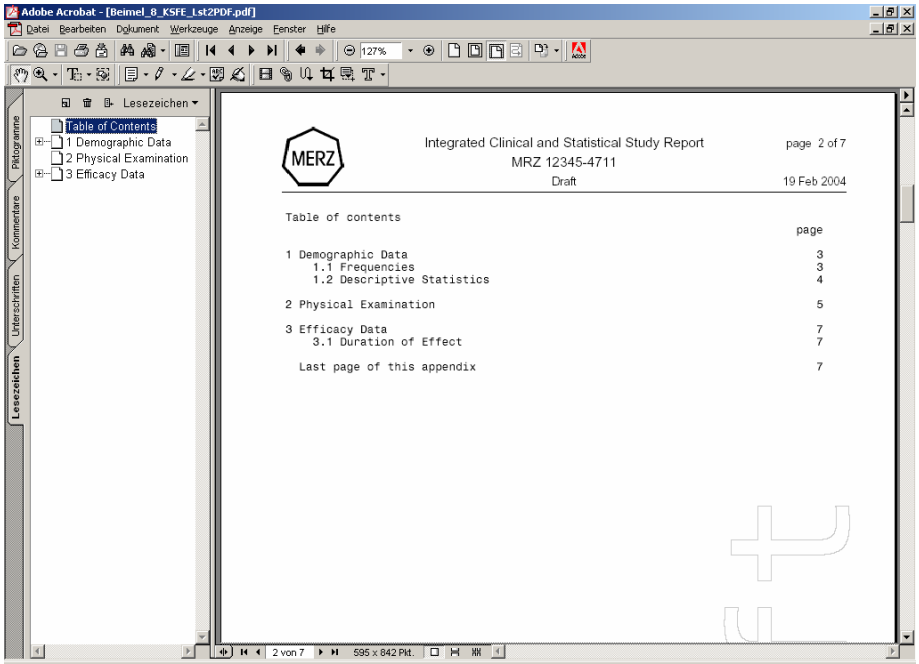

Abbildung 1: Inhaltsverzeichnis, Kopfzeile und Lesezeichen, wie sie von Lst2Pdf erzeugt werden

1.2 Weitere Vorteile

Es ist kein manueller Eingriff beim Erstellen des PDF-Dokuments notwendig. Das ist besonders interessant aus Sicht der Validierung, denn wo man nichts macht, kann man keine Fehler machen.
Das Formatieren und Umwandeln dauert nur Sekunden und ist beliebig oft wiederholbar.
Das Makro kann von CROs benutzt werden, Listen und Tabellen werden Merz dann im PDF-Format zur Verfügung gestellt.

2 Warum kein ODS?

Es gibt einige bekannte Bugs in ODS bei der Formatierung der Ausgabe von Proc Report. Da bei der Auswertung klinischer Studien sehr häufig Proc Report genutzt wird, ist SAS/ODS PDF für Merz nicht geeignet.
Aufgrund der Eigenschaften von PDF-Dateien ist ein Zusammensetzen einzelner kleiner Dateien zu einer großen, umfassenden Datei schwierig.
Es kann kein automatisches Inhaltsverzeichnis erstellt werden. ODS ist zwar in der Lage, Lesezeichen zu erzeugen, diese Lesezeichen sind aber unzureichend.

3 Besondere Herausforderungen

3.1 PostScript-Datei erzeugen

Um PDF-Dateien zu erzeugen, wird von Lst2Pdf zuerst eine PostScript (PS)-Datei erzeugt. Die Programmiersprache PostScript erlaubt volle Kontrolle über alle Formatierungen und die Seitengestaltung.

Die Umwandlung in eine PDF-Datei ist z. B. mit Acrobat Distiller oder GhostScript möglich. Lst2Pdf nutzt GhostScript, ein Programm, das für sehr viele Betriebssysteme zur Verfügung steht.

PostScript ist eine Programmiersprache mit einigen Eigenheiten, z. B. der Umgekehrten Polnischen Notation.

Jede PostScript-Datei ist eine einfache Textdatei, die mit %!PS beginnt. Sämtliche Koordinaten und Größen müssen in Point (pt) angegeben werden. Das ist die gleiche Einheit, wie sie z. B. für die Schriftgröße in MSWord verwendet wird.

Die folgenden Programmzeilen zeigen ein sehr kurzes, aber lauffähiges PostScript-Programm:

```
%!PS

%Font definieren
/Helvetica findfont 100 scalefont setfont

%Gehe zur Koordinate (200,600)
200 600 moveto

%Drucke einen Text
(Hallo) show
```

Eine ausführliche Beschreibung der PostScript-Sprache ist in [1] zu finden.

3.1.1 PDFMark

PostScript ist eine Seitenbeschreibungssprache und dient der Druckersteuerung. PDF-Dateien enthalten noch zusätzliche Informationen wie z. B. Verknüpfungen und Lesezeichen.

Diese Befehle können in der PostScript-Datei als so genannte PDFMarks abgelegt werden. Das sind zusätzliche Zeilen im Programmcode.

Ein einfaches Beispiel ist das Drehen der Bildschirmansicht. Der 'rotate' Befehl der PostScript-Sprache dreht die Schrift auf der Seite mit dem Effekt, dass in der PDF-Datei die Schrift auf der Seite liegt. Das PDFMark 'rotate' dagegen dreht die Seite inklusive der Schrift - eine Operation, die wohl die wenigsten Drucker beherrschen:

```
[{Page1} << /Rotate 90 >> /PUT pdfmark
```

3.2 Erkennen der Titelzeilen mit Regulären Ausdrücken

Lst2Pdf muss die Titelzeilen an ihrem Muster erkennen, da kein spezieller Marker verwendet wird. Dabei ist es notwendig, das Muster so genau wie möglich zu beschreiben, um nicht aus Versehen normalen Text als Titelzeile zu erkennen.

Reguläre Ausdrücke (Regular Expressions oder auch kurz RX) dienen der Textmustererkennung. Es gibt sie auch in anderen Programmiersprachen (z. B. Perl, Unix-Script). Sie sind überall ähnlich, aber nicht gleich.

RX sind deutlich mächtiger als konventionelle Stringfunktionen in SAS (index(), verify(), indexc(), scan(), ...).

3.2.1 Definieren mit RXParse

RX müssen erst mit der Funktion RXParse definiert werden. Diese Funktion gibt eine Nummer zurück, mit der der Reguläre Ausdruck dann in anderen Funktionen oder Call Routinen genutzt werden kann.

```
rx_no=RXParse("` Regulärer Ausdruck  ");
```

Der Ausdruck besteht aus Elementen, die angeben, was gesucht wird, wie oft gesucht wird, in welche Richtung gesucht wird usw. Tabelle 2 enthält die Elemente, die im folgenden Beispiel genutzt werden. Eine komplette Übersicht gibt die SAS-Online Dokumentation [2] unter der Beschreibung der Funktion RXParse.

Tabelle 2: Ausgewählte Elemente Regulärer Ausdrücke in SAS

	Element	**Bedeutung**
Was?	A..Z	die Buchstaben A-Z
	?	ein beliebiges Zeichen
	$d	eine Ziffer
	''	maskiert Sonderzeichen
Wie oft?	m+	mindestens ein Muster m
	m*	null oder mehr Muster m
	( )	Gruppierung

3.2.2 Beispiel

Lst2Pdf findet Titelzeilen, die von links nach rechts wie folgt aufgebaut sind:

- das Wort 'Table'
- beliebig viele Leerzeichen,
- die erste Tabellennummer,
- beliebig viele Untertabellennummern, die durch einen Punkt getrennt sind,
- ein Doppelpunkt und
- der eigentliche Text des Titels

Beispiel:

Table 2.1.3: Demographic Data

In RX-Sprache kann man dieses Muster so definieren:

```
Table ' '*          - das Wort 'Table' und die Leerzeichen
$d+ (. $d+)*        - die Tabellennummer
':' ?+              - der Doppelpunkt und der Text des Titels
```

Der Aufruf der Funktion RXParse sieht dann so aus:

```
rx_no=RXParse("` Table ' '* $d+ (. $d+)*  ':' ?+");
```

3.2.3 Anwendung

SAS bietet die Funktionen RXMatch, Call RXChange und Call RXSubstr an, um den mit RXParse definierten Ausdruck anzuwenden.

RXMatch überprüft, wo das Muster in einem String gefunden wird:

```
IsTitle=RXMatch(rx_no, line);
```

Hat die Variable IsTitle einen Wert > 0, wird die Textzeile als Titelzeile angesehen und die Weiterverarbeitung mit den Call Routinen kann beginnen. Mit ihnen wird gefundener Text extrahiert (für das Inhaltsverzeichnis) oder ersetzt (bei der Ersetzung der Tabellennummern).

Literatur

[1] Adobe Systems Incoporated (1999), PostScript language reference manual, third edition, Addison-Wesley Publishing Company

[2] SAS Online Dokumentation

[3] S. Beimel (2004), User Manual Lst2Pdf, Merz Pharmaceuticals GmbH