Dokumente für die Ewigkeit — oder doch nicht?

Pfaffhausen, 20. August 2013: *Im Alltag haben sich PDF-Dokumente längst durchgesetzt. Kaum eine Anwendung, welche das Erstellen von PDF-Dokumenten nicht unterstützt. Ob unter Windows, Mac, Linux, auf dem Handy, überall können PDF-Dateien dargestellt werden. Meist geht dies ohne Probleme, und weil dem so ist, werden PDF-Dateien auch in der Archivierung eingesetzt. Ein Beispiel aus der Praxis zeigt, wo die Tücken liegen. Zunächst jedoch ein kurzer Rückblick zur Entstehung des PDF-Formates bzw. der PDF-Unterstützung in ArchivistaDMS.*

Gleicher Ursprung, unterschiedliche Wirkung

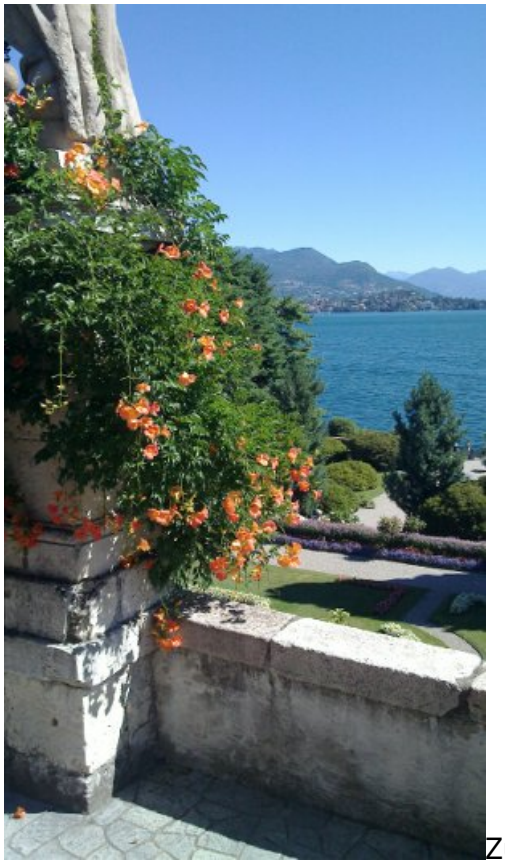
Als die erste Version unserer DMS-Lösung Archivista im Jahre 1998 auf den Markt kam, konnte das PDF-Format bereits den fünfjährigen Geburtstag feiern, wurde die **PDF-Spezifikation doch bereits im Jahre 1993 veröffentlicht.** Die Ursprünge gehen aber weiter zurück, als **Vorläufer zum PDF-Format gilt Postscript;** diese Sprache bzw. die ersten Drucker dazu kamen bereits 1984 auf den Markt. Im Unterschied zu anderen Drucksprachen hatte Postscript seit jeher die Möglichkeit, Elemente frei im Druckspeicher zu positionieren, und diese erst am Schluss zu Papier zu bringen. Damit war Postscript anderen Drucksprachen weit überlegen, die nur Zeile um Zeile zu Papier bringen konnten. Anfänglich waren Postscript-Drucker unsäglich teuer, unter einem fünfstelligen Betrag war ein Drucker (Schwarz/Weiss in 300dpi) nicht erhältlich.

Der hohe Preis war dadurch bedingt, dass dei Drucker-Hersteller hohe Lizenzgebühren zahlen mussten, um einen 100% kompatiblen Postscript-Drucker auf den Markt zu bringen. Für deutlich weniger Geld gab es PS-Printer (PS steht hier für Postscript), welche die Postscript-Befehle verarbeiten konnten, die aber nicht lizenziert wurden.

Noch günstiger war es, eine Postscript-Datei mit einer Software (Ghostscript sprach sich bald herum) in eine Bilddatei umzuwandeln, um sie anschliessend mit einem kostengünstigen Laser-Drucker zu Papier zu bringen. Dieses Verfahren hatte den Nachteil, dass das Erstellen einer Seite “unendlich” lange dauern konnte. Auf einem 386er mit 4 MByte RAM **dauerte es im Jahre 1990 ca. 2 Stunden, um eine einzelne Seite zu erstellen (rastern).** Wehe, wenn es dabei bei 98 Prozent ein Problem gab, wehe wenn noch Tippfehler bereinigt werden sollten… Einmal gerasterte Seiten konnten anschliessend aber problemlos am Bildschirm dargestellt werden, und auch das Drucken mit einem günstigen Drucker war keine Hexerei.

Zu jener Zeit gab es auch erste Versuche, Postscript-Dateien direkt am Bildschirm (in Echtzeit) darzustellen. Dieser Versuch scheiterte, wer wartet schon zwei Stunden auf eine Seite? Daher wurde **1993 eine vereinfachte Variante von Postscript, das PDF-Format, veröffentlicht.** Wie der Name **P**ortable **D**ocument **F**ormat besagt, sollte es dadurch möglich werden, Dokumente unabhängig von einer Software oder einem Betriebssystem darzustellen. Dafür veröffentliche der Hersteller (Adobe) die ersten Versionen von Acrobat, wobei die **erste Version des Viewers z.B. nicht kostenfrei verteilt werden konnte, genauso wie die damals erstellten Dateien unsäglich gross waren,** da PDF-Dateien zu Beginn nur ASCII-Zeichen enthalten konnten.

Das Darstellen eines gescannten Logos z.B. konnte enorm viel Platz beanspruchen, selbst wenn eine platzsparende komprimierte Tiff-Datei vorlag; die Bilddatei wurde mühsam in Textzeichen umgewandelt und benötigte anschliessend gut und gerne um den **Faktor 100 mehr Platz.** Und ja, die Zuverlässigkeit von PDF-Dateien liess oft zu wünschen übrig. Die Darstellung einer PDF-Datei wich zuweilen wesentlich von einer Ursprungsdatei ab. Manchmal enthielten die erstellten Dateien Fehler, mal gab es Probleme mit dem PDF-Anzeigeprogramm, mal fehlten Schriften, sodass diese Dateien nicht mehr geöffnet werden konnten, oder dann derart verhackelt waren, dass die Lesbarkeit nicht mehr gegeben war.

 Zu dieser Zeit entstand die erste Version von Archivista. Und **weil es mit Archivista möglich sein sollte, Dokumente langfristig verfügbar zu halten, kam das PDF-Format als "Originaldatei" nicht in Frage.** Die PDF-Spezifikation änderte zwischen 1993 und 1998 bereits viermal, derzeit (nach 20 Jahren) sind wir bereits im zweistelligen Bereich. Zum Vergleich: Mit Archivista werden 30 Jahre Lesbarkeit der gleichen Datei garantiert. Im Moment steht es 20:11 beim PDF-Format zu 15:1 für Archivista. Will heissen, das **PDF-Format wurde in 20 Jahren 11 mal geändert** (die Unterarten PDF-X und PDF-A bzw. deren Versionen nicht mal eingerechnet), bei der **ArchivistaBox war dies in 15 Jahren bisher noch nie der Fall.**

Zugegeben, die Verbreitung der PDF-Dateien ist immens, nicht weniger problematisch ist aber das langrfristige Speichern der Dokumente. Bei der ArchivistaBox funktioniert dies in dem Sinne, dass wir **jederzeit PDF-Dateien in Bild-Dateien umwandeln und aus den archivierten Daten jederzeit wieder PDF-Dateien erstellen können.** Dabei erleben wir immer wieder Problemfälle, bei denen der Kunde meist davon ausgeht, das Problem liege bei der ArchivistaBox.

Acrobat zeigt PDF ohne Probleme an…

Seit mehreren Jahren archivieren wir Daten aus einer verbreiteten ERP-Lösung, wobei die Dokumente im PDF-Format angeliefert werden. Nach einem Versionswechsel bei der ERP-Software gab es plötzlich Probleme mit den verarbeiteten PDF-Dateien. Etwa die Hälfte der Textzeilen war nicht korrekt ausgerichtet (meist links aussen am Rand).

Sowohl der **Kunde wie auch der Lieferant beteuerten, die PDF-Dokumente könnten in Acrobat problemlos dargestellt werden.** Daraus ergab sich die folgende Mail-Konversation:

> Ich möchte Sie höflich anfragen, ob Sie sich dem von Herrn Hofer> beschriebenen Problem annehmen konnten?> Auf unserer Seite drängt es ein bisschen mit der Zeit, da wir seit gut> einem Monat die "elektronischen Belege" nicht mehr archivieren können.> Besten Dank für Ihr kurzes Feedback

Die Recherchen von unserer Seite führten zu der folgenden Antwort:

Ich hatte bereits letzte Woche versucht, Herrn Hofer zu erreichen. Ichhätte damals jedoch noch keine Resultate vorweisen können, sondern einzigeine Vermutung. Die Vermutung, welche durch erneute Recherchen von heuteabend bestätigt werden konnte, lautet, dass die PDF-Dateien nicht korrekterstellt werden. Doch zunächst der Reihe nach, d.h. eine Zusammenfassung,welche Schritte unternommen wurden, um anhand der Datei GU1000439.pdf zudiesem Schluss zu kommen.

Erste Annahme: Veraltete Bibliotheken auf ArchivistaBox

Installation neuste Debian-Version und entsprechende Tools (Aufwand: 1,25 Std).Resultat: PDF-Datei enthält weiterhin Fehler

Zweite Annahme: Fehler in PDF-Datei

Überprüfen der Datei mit verschiedenen Acrobat-Readern. Getestet und installiert wurden Acrobat 6.0 unter Windows und 9.x unter Linux. Datei wird mit Acrobat 6.0 nicht korrekt dargestellt, mit 9.x dagegen schon. Die Datei wird aber in derVersion 1.4 angeliefert (müsste daher bereits mit Acrobat 5.x korrekt dargestellt werden), siehe dazu:

```
pdfinfo GU1000439.pdf
Title:          GU1000439.pdf
Author:         Oracle
ReportsCreator: Oracle11gR1 AS Reports Services
Producer:       Oracle PDF driver
CreationDate:   Wed Jul 17 14:39:57 2013
ModDate:        Wed Jul 17 14:39:57 2013
Tagged:         no
Pages:          1
```

```
Encrypted:      noPage
size:           595 x 842 pts (A4)
File size:      342918 bytes
Optimized:      noPDF
version:        1.4
```

Folglich müsste die Datei mit Acrobat 6.0 auch korrekt dargestellt werden,was aber nicht der Fall ist. Um es zu verifizieren, habe ich zusätzlichden sehr zuverlässigen PDF-Viewer Foxit installiert. Die Installation vonAcrobat benötigt aufgrund der Grösse leider etwas viel Zeit (Aufwand: 1.5Std).

Dritte Annahme: Irgendwo müssten bei der Verarbeitung Fehlermeldungen auftreten

Bei der aktuellen ArchivistaBox konnte ich keine Fehlermeldungen vorfinden(zu alte Ghostscript-Version). Darauf installierte ich Ghostscript 9.x inder Testumgebung. Aufgrund der verschiedenen Bibliotheken (Abhängigkeiten) istauch dies nicht ganz einfach, daher hatte ich hier 2.0 Std Aufwand. Am Endekonnte ich die folgenden Fehlermeldungen vorfinden (Auszug aus Log-Datei bei Verarbeitung mit Ghostscript 9.x):

Filename: /home/data/archivista/ftp/pdf/GU1000439.pdfpdf **** Unknown operator: '–10.16' looks like a malformed number, replacing with 0. **** Unknown operator: '–10.20' looks like a malformed number, replacing with 0. **** Unknown operator: '–10.20' looks like a malformed number, replacing with 0. **** Unknown operator: '–51.44' looks like a malformed number, replacing with 0. **** Unknown operator: '–21.40' looks like a malformed number, replacing with 0. **** Unknown operator: '–32.08' looks like a malformed number, replacing with 0. **** Unknown operator: '–21.40' looks like a malformed number, replacing with 0. **** Unknown operator: '–32.08' looks like a malformed number, replacing with 0. **** Unknown operator: '–206.88' looks like a malformed number, replacing with 0. **** Unknown operator: '–12.36' looks like a malformed number, replacing with 0. **** Unknown operator: '–12.40' looks like a malformed number, replacing with 0. **** Unknown operator: '–13.04' looks like a malformed number, replacing with 0. **** Unknown operator: '–12.36' looks like a malformed number, replacing with 0. **** Unknown operator: '–292.00' looks like a malformed number, replacing with 0. **** Unknown operator: '–1.40' looks like a malformed number, replacing with 0.

Nun kommen wir der Problematik näher. Die angelieferte Datei wird nicht korrekt dargestellt,die Positionen der

Textelemente stimmen nicht. In PDF-Dateien werden oft relativePositionierungen angegeben; ein Textelement wird beispielsweise 1,4 cm links davonpositioniert (-1.40). Genau darin liegt die Problematik, findet sich doch die Fehlermeldung --1.40 mit zwei Minuszeichen (malformed number). Ghostscript ist hier deutlich wenigerfehlertolerant als Acrobat 9.x. Ich denke, dies ist auch richtig so, denn -1.40 ist nichtgleich --1.40). Wie auch immer, korrekt ist die Angabe nicht, eine Zahl mag eine negativeGrösse haben, nicht aber zweimal Minus Minus (--). Aufwand für diese Erkenntnis (1,25 Std).

Vierte Annahme: Es müsste doch machbar sein, die Datei selber zu korrigerien

Die angelieferte PDF-Datei wird komprimiert angeliefert (Platzersparnis). Das Entpacken einer PDF-Datei und anschliessende erneute Speichern traute ich zunächst nur pdflib zu. Einmal kompiliert musste ich erkennen, dass ich für das Schreiben eines Programms im Minimum einige Stunden (wenn nicht Tage) Zeit benötigen würde. Folglich suchte ich weiterund wurde bei pdftk mit den enstprechenden Optionen fündig:

```
pdftk GU1000439.pdf output temp.pdf flatten uncompress
```

Damit liegt eine PDF-Datei im ANSI-Format vor, d.h. die Minuswerte können in der Dateitemp.pdf korrigiert werden (anschliessendes Speichern unter fixed.pdf). Beim ersten Versucherfolgte dies von Hand, und siehe da, es funktionierte (siehe Beilage fixed.pdf). Aufwandhier: 1,5 Std.

Fünfte Annahme: Mit einem Programm geht es schneller

Anschliessend schrieb ich ein Skript, mit dem nun Massentests gefahren werden könnten(Aufwand: 0.5 Std):

```
============================================================
#!/usr/bin/perl
use strict;
use lib qw(/home/cvs/archivista/jobs);
use AVJobs;
my $file = shift;
```

```
my $ftmp = "tmp.pdf";
my $fok = "fixed.pdf";
my $cmd = "pdftk $file output $ftmp flatten uncompress";
my $res = system($cmd);
if ($res==0) {
  my $content = "";
  readFile2($ftmp,$content);
  $content =~ s/(--)([0-9]+)(.)([0-9]{2,2})(s)(TD)/-
$2$3$4$5$6/g;
  $content =~s/(--)([0-9]+)(.)([0-9]{2,2})(s)(.*?)(s)(TD)/-
$2$3$4$5$6$7$8/g;
  writeFile($fok,$content,1);
}
```

==========

Hinweis: Das obenstehende Programm entfernt einzig die überflüssigen doppeltenMinuszeichen, es korrigiert nicht die relativen Dateipositionen. Dies hat jedoch(auf den ersten Blick) keine Konsequenzen, sowohl in Acrobat wie auch XPDF oder Foxit.Schlussbemerkung: Im Prinzip sollte beim Verursacher das Problem der doppeltenMinuszeichen gelöst werden. Die PDF-Dateien werden eindeutig nicht korrekt angeliefert.Sollte dies nicht möglich sein, müssten wir einen Massentest sowie einen definitivenFix einbauen. Hier würde wohl ein Aufwand von ca. 10-16 Stunden entstehen.

Derzeit liegt der Aufwand (inkl. Verfassen dieser Mail 0,75 Std) bei 8,75 Stunden für diesen Job. Gerne erwarte ich eine Rückantwort wie wir verfahren können.

Soweit unsere Antwort, der im Prinzip nichts hinzugefügt werden muss. Ein Punkt soll hier dennoch angeführt werden. Die **Diskussion, ob ein Problem ein Bug ist, greift hier und oft zu kurz. Probleme sind da, um sie zu lösen.** Kunden der ArchivistaBox erhalten daher auch dann Support, wo an sich klar ist, dass ein Problem nicht durch die ArchivistaBox verursacht wird. Genauso wie wir klar und offen kommunzieren, wo genau Probleme liegen bzw. wie und mit welchem Auftwand diese aus der Welt "geschafft" werden können.

Verarbeitung heute um Faktor 1:200'000 schneller

Und ja, wir stehen — auch nach 15 Jahren dazu — PDF-Dokumente eignen sich nicht für die Langzeit-Archivierung, gerasterte Bild-Dateien sind hier deutlich sinnvoller. Sollte dabei das Vorurteil bestehen, dass der Rasterprozess zu lange dauren sollte, so sei hier noch angeführt, dass eine einzelne **ArchivistaBox bis zu 2,4 Mio Seiten pro Tag (siehe Kurztipps PDF-Tools bei pro-linux.de bzw. dort Kommentare), oder ca. 28 Seiten pro Sekunde** verarbeiten kann. Dies ist im übrigen um den Faktor 1:200'000 schneller, als vor etwas mehr als 20 Jahren, wo eine einzelne Seite ca. 2 Stunden Zeit benötigte.

Toll, dass die Technologie uns heute Möglichkeiten gibt, die vor 20 Jahren nicht einmal denkbar gewesen wären, dumm nur, dass die Problematiken die gleichen bleiben (sprich ein Minus-Zeichen zu viel und die Datei ist hinüber). Als ArchivistaBox-Kunde können Sie **zweifach zurücklehnen. Erstens erhalten ArchivistaBox-Kunden on-the-fly von allen abgelegten Daten Bild-Dateien und zweitens helfen wir heute und jetzt, allfällige "Problemfälle" speditiv aus der Welt zu schaffen.** Damit auch wirklich alle Daten (inkl. der PDF-Dateien) für die Ewigkeit halten.