Adobes Portable Document Format - PDF

Themenauswahl

Hinweis

Mit (Nachtrag) gekennzeichnete Abschnitte sind Ergänzungen, die im Vortrag nicht enthalten waren.

Überblick - Was ist PDF?

PDF ist ein Format, das ein Dokument geräteunabhängig und plattformübergreifend beschreibt. Es hat Ähnlichkeit mit PostScript und wurde wie dieses von Adobe Systems Incorporated entwickelt. Eine PDF-Datei enthält eine oder mehrere Seiten mit Text, Graphik, Bildern, Hypertext-Links und/oder Formularen. Eine kostenlose Software von Adobe, der Acrobat Reader, ermöglicht das Betrachten von PDF-Dokumenten.

PDF im Browser

Findet ein Browser im Web eine am MIME-Typ application/pdf erkenntliche PDF-Resource vor, so kann er auf verschiedene Weise reagieren. Normalerweise ruft er ein externes Programm (helper application) auf oder speichert die Datei nur auf Platte. Mit einem Plugin oder ActiveX-Control dagegen kann der Browser das Dokument direkt in seinem Fenster anzeigen. Noch einen Schritt weiter geht die Einbettung von PDF in HTML-Seiten. Mit den Tags <embed> (Netscape) bzw. <object> (Internet Explorer) kann ein PDF-Dokument in eine Teilfläche der HTML-Seite eingefügt werden. Anders als im Acrobat Reader fehlen hier aber die Menüs und eventuell auch die Werkzeugleiste zum Navigieren im Dokument. Beim Drucken der Web-Seite bleibt die entsprechende Fläche derzeit noch weiß. Anders als bei GIF- oder JPEG-Bildern gibt es auch keine einfache Möglichkeit zum Abspeichern eingebetteter PDF-Dateien. Des weiteren kann das Plugin bzw. ActiveX-Control nur wenige PDF-Dokumente gleichzeitig anzeigen und mit PDF-Verweise und Artikelflüsse nicht umgehen.

Das Plugin und das ActiveX-Control bieten aber auch Vorteile gegenüber dem Acrobat Reader. Eine mit #xml= an den URL angehängte Datei kann benutzt werden, um dynamisch Textstellen im Dokument hervorzuheben. (Siehe Abschnitt Volltextindizierung.) Des weiteren kann das PDF-Dokument inkrementell geladen werden, das heißt der Benutzer kann Teile der Seite sehen noch bevor die ganze Datei geladen ist. Dazu muß die Datei optimiert sein (Option in Exchange bei Speicher unter ...) und das Byterange-Protokoll (ab HTTP 1.1) unterstützt werden. Das progressive Rendering der Seite läuft dann wie folgt ab. Zuerst werden Hypertextelemente und der Artikelfluß geladen. Damit löst ein Mausklick sofort die vorgesehene Funktion aus. Dann

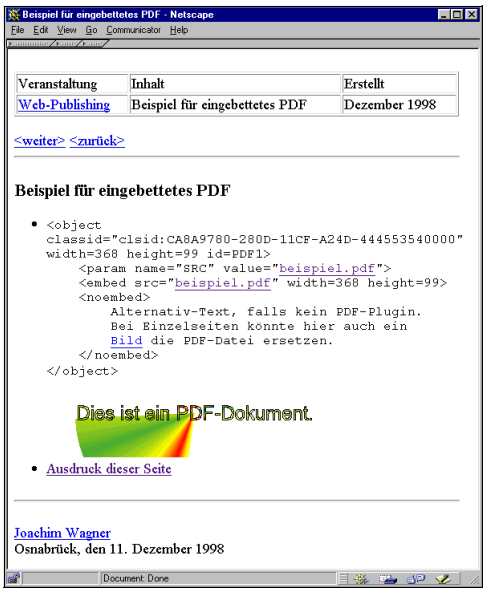
werden die Font-Deskriptoren (Abmaße der Zeichen) eingelesen, die nötig sind, um anschließend den Text an die richtige Stelle setzen zu können. Der Text wird erst danach zusammen mit etwaigen Vektorgraphiken angefordert und auf die Seite plaziert. Mit sogenannten Multiple Master Fonts, die stufenlose Parameter wie Stärke, Stil und Breite haben, werden die gewünschten Schriftarten gemäß den Font-Deskriptoren angenähert. Erst nachdem auch noch Pixelgraphik geladen wurde werden die Fonts abgefragt (Aussehen der Zeichen) und der Text neu gezeichnet. Zuletzt werden Thumbnails, kleine Bilder der Seiten des Dokuments, geladen. Das stückweise Laden der PDF-Datei geschieht über das Byterange-Protokoll. Bei jeder normalen GET -Anfrage kann mit Range: bytes= eine Komma-separierte Liste von Dateibereichen angegeben werden, die dann vom Server als multipart/byteranges bzw. multipart/x-byteranges (alt, vor HTTP 1.1) übertragen werden.

Beispiel für eingebettetes PDF

Mit einem Trick ist es unnötig, per Skript den Browsertyp festzustellen, um dann das richtige Tag, <embed> oder <object>, zu verwenden. <embed> eingeschachtelt von <object> wird ignoriert, wenn <object> unterstützt wird. Im anderen Fall werden die Parameter von <object> überlesen und das <embed>-Tag ausgewertet:

```
<object classid="clsid:CA8A9780-280D-11CF-A24D-444553540000" width=368
height=99 id="PDF1">
    <param name="SRC" value="beispiel.pdf">
    <embed src="beispiel.pdf" width=368 height=99>
    <noembed>
        Alternativ-Text, falls kein PDF-Plugin.
        Bei Einzelseiten könnte hier auch ein
        Bild die PDF-Datei ersetzen.
    </noembed>
</object>
```

Das Attribut classid gibt eine eindeutige Kennung des ActiveX-Moduls an. Mit width und height wird der zu verwendende Platz festgelegt. id gibt einen Namen an, unter dem das Objekt referenziert werden kann, z.B. in einem VBScript-Programm. Mit <param> werden die Parameter für das ActiveX-Control gesetzt. Weitere Daten bis zu </object> werden ignoriert, außer natürlich wenn <object> nicht unterstützt wird. Dann werden die Attribute von <embed> ausgewertet. Nur wenn beide Tags nicht verstanden werden wird der Text zwischen <noembed> und </noembed> angezeigt. Da der Ausdruck eingebetteter PDF-Dokumente noch nicht funktioniert, hier ein Screenshot:



Gestaltungsmöglichkeiten

Ein PDF-Dokument besteht nicht nur aus beliebig angeordneten Text in verschiedenen Schriftarten, sondern kann auch weitere Elemente enthalten, die die Arbeit mit dem Dokument erleichtern. Mit Lesezeichen kann zu einer vorher festgelegten Stelle gesprungen werden. Zwar werden die Lesezeichen häufig als Inhaltsverzeichnis verwendet. Sie sind aber nicht Bestandteil einer Seite des Dokuments, sondern werden außerhalb, im Acrobat Reader links vom Dokument, angezeigt. Die Lesezeichen sind hierarchisch angeordnet und lassen sich im Acrobat Reader entsprechend zusammenfalten und expandieren. Da das Sprungziel eines Lesezeichens auch in einem anderen Dokument liegen darf, läßt sich so ein ganzer Dokumentsatz zusammenfassen. Thumbnails bieten eine zweite Möglichkeit sich im Dokument zurechtzufinden. Die verkleinerten Abbilder der Seiten helfen allerdings nur, wenn das Dokument stark strukturiert ist, da sonst alle Seiten gleich aussehen.

Eher für die Weiterverarbeitung sind Notizen gedacht, da sie nicht im Ausdruck erscheinen. In Acrobat werden sie als gelbe „Klebezettel“ angezeigt, die kleine Mengen Text enthalten können. Jede Notiz hat eine in der PDF-Datei festgelegte Position im Dokument. Notizen eignen sich daher gut für Ergänzungen und Korrekturen. Allgemeine Informationen zum Dokument bringt man besser im Dokumentinfo unter. Neben den vordefinierten Felder wie Titel und Autor können auch eigene Felder definiert werden. Letztere sind bei Suchanfragen in großen Dokumentenbeständen besonders nützlich, wenn eine Zuordnung zu einem Volltextindex hergestellt wurde. (Siehe Volltextindizierung.)

Ein Artikelfluß beschreibt, wo sich am Ende einer „Spalte“ der Text fortsetzt. Im Acrobat Reader erscheint dann jeweils unten rechts ein Pfeil zum Weiterklicken. Bei nicht für die Bildschirmbetrachtung optimierten Layouts erleichtert dies das Lesen sehr. Natürlich gibt es auch richtige Hyperlinks. Man unterscheidet bei PDF zwei Sorten: Web-Links starten automatisch den Web-Browser zur Anzeige des Dokuments. Das Sprungziel muß dabei nicht ein PDF-Dokument sein. Es kann ein beliebiger URL angegeben werden. PDF-Links dagegen verweisen auf eine Stelle im selben oder in einem anderen Dokument. Schließlich kann ein PDF-Dokument Formularelemente wie z.B. Optionsfelder und Schaltflächen enthalten. (Siehe Abschnitt Formulare.)

Viele PostScript-Eigenschaften

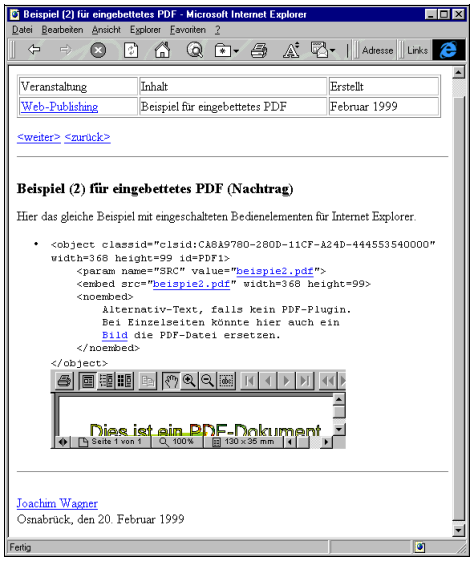
PDF übernimmt viele PostScript-Eigenschaften (Level 1 und 2). Da Text beliebig positioniert, rotiert, skaliert und eingefärbt werden kann, lassen sich beliebig komplexe Formatierungen realisieren. Neun verschiedene Farbmodelle können verwendet werden, darunter RGB, CMYK und Lab. Mit Schwarz- und Weißpunkt im CIE 1931 Farbraum, Gamma-Werten und linearen Abbildungen wird angegeben, wie die Farbwerte zu interpretieren sind. Über das Open Prepress Interface (OPI, ursprünglich von Aldus entwickelt) können externe Bilder eingebunden werden. Neben verlustfreier Kompression, mit der jedes Objekt im PDF verpackt werden kann, steht auch JPEG zur Verfügung.

Bedingtes Anzeigen von Dokumentbestandteilen

Ab PDF 1.2 (Acrobat 3.0) können Dokumentbestandteile bedingt angezeigt werden. Es gibt die Möglichkeit, Anzeige und Drucken oder auch nur das Drucken zu verhindern. Mit Aktionen (s.u.) kann der Zustand gewechselt werden. So kann z.B. ein Hilfetext erscheinen, sobald der Mauszeiger in einen bestimmten Bereich des Dokuments eintritt. Obwohl diese Technik in die Attribute von Notizen eingebaut wurde, was etwas geflickt aussieht, wird sie wohl auch in späteren Versionen unterstützt, da im Reference Manual selbst der Tip gegeben wird, damit eine kontextsensitive Hilfefunktion zu realisieren.

Einfluß auf angezeigte Bedienelemente

Der Browser betrachtet das PDF-Dokument als eine Einheit. Einzelne Seiten können mit seinen Navigationsknöpfen nicht angesprungen werden. In Netscape und Internet Explorer wird daher zusätzlich die Acrobat Werkzeugleiste eingeblendet, wenn PDF alleine angezeigt wird. Bei in HTML eingebetteten PDF-Dokumenten ist dagegen das Verhalten nicht gleich. Netscape verzichtet auf weitere Bedienelemente. Der Internet Explorer entscheidet entsprechend den Vorgaben in der PDF-Datei, was angezeigt wird. Eingestellt werden sie in Acrobat Exchange unter Datei - Dokumentinfo - Öffnen. Dort kann auch der Skalierungsfaktor und die Anzeige von Lesezeichen und Thumbnails festgelegt werden. Hier ein Screenshot mit eingeschalteten Bedienelementen:



Im Netscape Navigator würde die gleiche Seite in etwa so aussehen wie es der Screenshot weiter oben zeigt. An diesem Beispiel sieht man auch, daß genügend Platz für die Bedienelemente vorgesehen werden muß, da sie sonst abgeschnitten werden.

Aktionen

In PDF-Dokumenten kann eine Vielzahl von Ereignissen Aktionen auslösen. Dazu gehören das Bewegen der Maus in eine Fläche, das Auswählen eines Lesezeichens, das Öffnen des Dokuments und vieles mehr. Als Aktionen kommen u.a. Sprünge zu anderen Seiten, Dokumenten oder URLs, Auslösen von Menübefehlen, Versenden von Formulardaten und Starten anderer Anwendungen in Frage. Letztere Aktion ist natürlich ein großes Sicherheitsrisiko, das mit der Version 3.01 von Acrobat dadurch reduziert wurde, daß jetzt eine Warnmeldung dem Start vorausgeht.

Unterschiede PDF - HTML

Zwar nähern sich PDF und HTML immer mehr an. Aber die konzeptionellen Unterschiede wirken sich noch heute aus. Trotz der Möglichkeiten von HTML 4.0 erscheint das Layout im Navigator und Internet Explorer häufig nicht so, wie vom Autor der Seite geplant, da Netscape und Microsoft schwerpunktmäßig an eigene Erweiterungen arbeiten anstatt den Standard vollständig umzusetzen. Ein einheitlicher Weg, um Fonts einzubetten, hat sich noch nicht durchgesetzt. PDF dagegen ist darauf ausgerichtet, Dokumente punktgenau wiederzugeben. Egal ob auf dem Bildschirm oder auf dem Drucker, das Dokument erscheint immer gleich.

Zur Strukturierung des Inhalts leistet PDF so gut wie nichts. Text ist dort einfach eine Ansammlung von Zeilen. Spätestens bei einem Seitenumbruch versagt die Kopierfunktion (Textwerkzeug + Menü Bearbeiten) von Acrobat Reader. In dem dreispaltigen Beispieldokument

quer3.pdf ist es z.B. nicht möglich, einen einzelnen Absatz zu selektieren, da die benachbarten Spalten immer mit markiert werden. HTML ist hier PDF klar überlegen, wenn man einmal davon absieht, daß viele HTML-Autoren die Strukturinformationen durch Layouttricks (Tabellen, die von unsichtbaren GIFs formatiert werden, etc.) zerstören.

Eignung für Bildschirm und Drucker

HTML wird vom Browser aufbereitet und dabei im Idealfall optimal an das aktuelle Ausgabegerät angepaßt. Auf dem Bildschirm nimmt der Text automatisch genau die Fensterbreite ein und hat eine sinnvolle Schriftgröße. Dadurch braucht der Benutzer, wenn überhaupt, nur in eine Richtung zu scrollen. Im Ausdruck ist das Dokument zwar anders formatiert. Der vorhandene Platz wird aber gut genutzt. Und da der Browser eine Schrift wählt, die vom Drucker unterstützt wird, erfolgt der Ausdruck schnell und mit geringer Systembelastung.

PDF dagegen gibt genau vor, wie das Dokument aussieht. Häufig muß der Benutzer sich erst einmal weit genug an den Text heranzoomen, um ihn lesen zu können. Dann muß er horizontal und vertikal scrollen, wenn der Platz im Fenster nicht ausreicht. Beim Ausdruck gibt es auch Probleme, wenn das vorhandene Papierformat nicht mit dem des Dokuments übereinstimmt, z.B. verschenkt man bei 3¾ x 8½ Zoll Faltblättern viel Platz auf A4-Papier. Dafür unterscheidet sich die Seite nicht vom Original. Leider bietet der Acrobat Reader nicht die Möglichkeit, Schneidemarken mitzudrucken.

Dateien und Weiterverarbeitung

Eine HTML-Seite setzt sich häufig aus vielen Resourcen zusammen, die einzeln angefordert und übertragen werden. Speichert man nur die html-Datei, hat man ein unvollständiges Dokument. Bilder fehlen und Java-Code gibt Fehlermeldung aus. Nur mit spezieller Software lassen sich komplexe HTML-Seiten brauchbar von einem Ort zu einen anderen verschieben. Bei PDF-Dokumenten besteht dieses Problem nicht. Alles, was zur Anzeige und Interaktion benötigt wird, ist in der der PDF-Datei enthalten.

Zur Weiterverarbeitung eignen sich PDF-Dateien kaum, da Strukturinformationen fehlen. Größere Änderungen am Text oder Layout sind sehr schwierig bis unmöglich. Es ist dagegen einfach, einzelne Seiten aus dem Dokument herauszunehmen und in ein anderes Dokument einzufügen. Ebenso kann man schnell neue Elemente über eine Seite legen, z.B. mit den Werkzeugen in Acrobat Exchange. Nach dem Import eine HTML-Seite in eine Textverarbeitung sieht die Situation besser aus. Absätze sind wirklich zusammenhängend und können neu umgebrochen werden. Überschriften werden erkannt, Listen korrekt dargestellt, Bilder als externe Verknüpfungen eingebunden usw. Mit Software, die HTML nicht unterstützt, kann man HTML-Seiten sogar manuell verarbeiten, wenn nicht zu viele Tags und Entitäten auftreten. Bei PDF ist so ein Vorgehen wegen der Datenkompression selbst mit einem Editor für Binärdateien sehr schwierig.

Dateigröße

PDF-Dateien, die man im Web antrifft, sind oft sehr groß. Das liegt daran, daß PDF gerade für umfangreiche oder bebilderte Dokumente eingesetzt wird. Diese Ausarbeitung ist sogar als PDF etwa 12% kleiner als HTML, natürlich nur, wenn die Bilder in sehr niedriger Auflösung (hier 24 DPI) übernommen werden. Das liegt daran, daß auch Text komprimiert wird. Ist der Textanteil kleiner, wie z.B. bei den Folien zum Vortrag, dann sind PDF-Dateien meist mehr als doppelt so groß wie die HTML-Fassung.

Fortentwicklung des Formats

HTML wird vom W3C (World Wide Web Consortium, eine Arbeitsgruppe innerhalb der Internet Engineering Task Force, IETF) gepflegt. Das W3C verarbeitet Feedback aus dem Internet und unterhält Kontakte zu verschiedenen Softwareherstellern. Trotzdem erweitern Microsoft und Netscape ihre Browser um hauseigene Besonderheiten und setzen gleichzeitig den HTML-Standard nur unvollständig um. Zu HTML erscheinen viele Veröffentlichungen. Viele Menschen sind mit den Grundlagen von HTML vertraut, da das textbasierte Format zum manuellen Nachbearbeiten einlädt.

Bei PDF ist die Situation komplementär. Adobe entwickelt PDF alleine fort. Sie veröffentlicht eine PDF-Referenz (derzeit 400 Seiten). Die Einarbeitung in das Dateiformat gestaltet sich schwierig. Bücher zum Thema beschränken sich auf die Anwendung von PDF und beschreiben den Aufbau von PDF nur schematisch. PDF-Dateien von Hand zu bearbeiten ist kaum möglich. Man muß programmieren oder vorhandene Software nutzen.

Konvertieren bestehender Dokumente

Nach PDF kann jedes Dateiformat konvertiert werden, wenn der gesamte Inhalt über die Druckfunktion einer passenden Anwendung ausgegeben werden kann. Ist das der Fall, so kann mit PDF-Write oder Distiller (s.u.) gearbeitet werden. Sonst ist man auf Export-Filter angewiesen.

Der umgekehrte Weg ist wesentlich schwieriger. (Siehe auch „Dateien und Weiterverarbeitung“ oben.) Notizen und Formulardaten lassen sich von Acrobat Exchange aus exportieren. Es gibt auch schon einige Kommandozeilentools zum Extrahieren von Text aus PDF-Dateien. Man kann auch PDF zu PostScript konvertieren, z.B. durch Ausdrucken in Acrobat Reader, und anschließend gewöhnliche PostScript-Tools benutzen, um die Resourcen zu trennen. PDF ist also in dieser Hinsicht eher als Sackgasse zu bezeichnen.

Erzeugen von PDF-Dateien

Installiert man das Adobe Acrobat Paket, das im Gegensatz zum Acrobat Reader nicht kostenlos ist, hat man mehrere Tools, die das Erstellen von PDF-Dokumenten ermöglichen. Layout-Programme gehören aber nicht zu Acrobat. Grundsätzlich wird das Aussehen der Seiten durch andere Software bestimmt. Z.B. wurde dieser Text unter StarOffice (Star Division GmbH) geschrieben. (Siehe auch Makros für MS-Word und die Beispiele zu PDF-Formulare.)

PDF-Write

Am einfachsten zu benutzen ist PDF-Write, das wie ein Druckertreiber im System eingebunden ist. Man wählt beim Drucken aus einer beliebigen Anwendung den „Drucker“ PDF-Write. Es erscheint dann ein Dialog zum Speichern der PDF-Datei. Ein Nachteil von PDF-Write ist aber, daß es nicht mit PostScript arbeitet sondern auf die Graphikschnittstelle des Systems aufsetzt (bei Windows GDI). So wird von EPS-Graphiken nur die Bildschirmvorschau (=Pixelgraphik) übernommen, die meistens eine viel geringere Qualität als die eigentliche Graphik hat.

Distiller

Schwierigkeiten mit EPS-Dateien und anderen PostScript-Spezialitäten lassen sich sich vermeiden, indem man aus der Anwendung heraus eine PostScript-Datei erzeugt, und diese dann mit dem Distiller in eine PDF-Datei verwandelt. Der Distiller enthält einen vollständigen PostScript-Interpreter. Dies ist nötig, da trotz der gemeinsamen Eigenschaften vieles, z.B. PostScript Funktionsaufrufe, nicht gradlinig in PDF übersetzt werden kann. Über Distiller-Optionen lassen sich Fonteinbettung, Bildkompression und Farbverwaltung beeinflussen. Die Umwandlung muß nicht manuell durchgeführt werden. Der Distiller-Assistent kann

Verzeichnisse überwachen und PostScript-Dateien automatisch umwandeln. Im Internet gibt es bereits Server, die diesen Dienst anbieten. Man stellt via ftp eine PostScript-Datei in das dafür vorgesehene Verzeichnis und kann kurz darauf die fertige PDF-Datei downloaden.

Mit dem Distiller wird allerdings nur das Problem auf das Erzeugen von PostScript-Dateien verlagert. Unter Windows muß ein Druckertreiber installiert werden, der PostScript erzeugt. Zwar wird jetzt nicht mehr unnötig Vektorgraphik in Pixelgraphik verwandelt. Aber PDF-Eigenschaften, die über PostScript hinausgehen, können so nicht eingesetzt werden. Deshalb hat Adobe PostScript um den Operator pdfmark erweitert. Er wird vom Distiller ausgewertet und erlaubt so z.B. das Anlegen von Hyperlinks oder Lesezeichen. Mittels EPS-Dateien lassen sich diese PDF-mark-Anweisungen in jede Anwendung einschleusen, die EPS einbinden kann. (Siehe Beispiel zu PDF-Formulare.) Einige Anwendungen erzeugen aber auch von sich aus PostScript mit pdfmarks. Z.B. wandelt eine Erweiterung von html2ps Hypertext-Links entsprechend um.

Exchange

Acrobat Exchange ist das Werkzeug von Adobe zum Anpassen vorhandener PDF-Dokumente. Es können die Anordnung der Seiten verändert, mehrere PDF-Dokumente zusammengeschnitten und kleine Tippfehler im Text korrigiert werden. Auch Thumbnails, Lesezeichen, Notizen, Hypertext-Links und Formularelemente können hier angelegt werden. Bei letzteren vier werden viele Einstellmöglichkeiten angeboten, z.B. kann die auszuführende Aktion (s.o.) beliebig gewählt werden. Des weiteren kann Exchange verschiedene Pixelgraphik-Formate (leider nicht JPEG) importieren und Papierdokumente direkt vom Scanner in ein PDF-Dokument einlesen. Zu guter Letzt enthält Exchange das OCR-Modul Acrobat Capture, das Zeichen in der Pixelgraphik erkennt und entsprechende Textfelder erzeugt. So kann man die Dateigröße deutlich reduzieren. Alternativ kann der erkannte Text zusammen mit dem Originalbild gespeichert werden. Dann erscheint das Dokument auch bei Erkennungsfehlern, wie sie bei der OCR leider auftreten, genauso wie das Original, und man hat gleichzeitig den Vorteil, im Text suchen zu können oder sogar das Dokument zu indizieren. (Siehe Volltextindizierung.)

pdfTeX, pdfLaTeX

(Nachtrag)

Eine kleine Gruppe arbeitet daran, TeX so zu erweitern, daß es neben DVI-Dateien auch PDF ausgeben kann. Das Projekt befindet sich klar im Alpha-Stadium. Die aktuelle Version 0.13b (08. Februar 1999) sieht dafür schon sehr vielversprechend aus. Mehr Informationen gibt es unter http://www.tug.org/applications/pdftex/.

PDF-Bibliotheken

Für dynamisches PDF (z.B. mit CGI) ist der Weg über PostScript zu aufwendig, da bei jeder Anfrage mehrere Programme gestartet werden müssen. Ein Lösungsansatz stellen Bibliotheken dar, die beim Schreiben von Programmen helfen, die PDF-Dokumente direkt erzeugen können, z.B. PDFlib von Thomas Merz oder ClibPDF von FastIO Systems. Wer jetzt sofort eigene Programme schreiben will, sollte beachten, das diese Bibliotheken die Kapselung der Dateistruktur und der verschiedenen Elemente übernehmen und nicht für das Layout zuständig sind. Tabellen ausbalancieren, Absätze auf die Seiten verteilen, Kopf- und Fußzeilen verwalten usw. gehört in andere Bibliotheken.

Probleme

Acrobat numeriert die Seiten eines Dokuments immer von eins aufsteigend durch. Dies ist für den Benutzer sehr verwirrend, wenn die Seitenzahlen im Dokument abweichend numeriert sind, z.B. bei gescannten Dokumenten.

Bis zur Version 2.1 von Acrobat wurde LZW als Kompressionsverfahren benutzt. Danach wurde das ZIP (Flate) Verfahren aufgenommen, da es meistens besser komprimiert und darüber hinaus kostenlos verwendet werden kann. Seit einiger Zeit ist eine Lizenz von der Unisys Corporation nötig, um LZW verwenden zu dürfen. Daher ist zu erwarten, daß neue Software, insbesondere freie Software LZW nicht mehr unterstützt. Andererseits schließt man mit der ZIP-Kompression Benutzer alter PDF-Viewer aus.

Durch die vielen Schritte bei der Erstellung von PDF-Dokumenten (Layout in Anwendungssoftware, PostScript, Distiller und Exchange) gibt es viele Fehlerstellen. Bei Problemen muß die ganze Kette abgesucht werden. Zum Vorteil ist natürlich, daß sich so auch Probleme einkreisen und eventuell beheben lassen, z.B. gibt es bereits Distiller aus anderen Quellen, z.B. Ghostscript und ps2pdf unter Linux.

Makros für MS-Word

(Nachtrag)

Während des Vortrags behauptete ich, man könne PDF-mark-Anweisungen in Word manuell einfügen und so Links, Lesezeichen usw. einbinden. Zwar ist dies richtig, aber anspruchsvolle PDF-Dokumente lassen sich mit MS-Word wesentlich einfacher erstellen, indem man Makros verwendet, die vorhandene Formatierung, wie z.B. Absatzformate (Überschrift etc.), auswerten. Benutzt man fertige Makro-Pakete, muß man sich nicht mit PDF-mark-Anweisungen beschäftigen oder extra Acrobat Exchange starten.

Für Word 7.0 gibt es pdfMarker von Joel Geraci. Aufgrund einer Sprachbeschränkungen von Word 7.0 läuft es nur mit der englischen Version. Es wird ein Menüeintrag „Acrobat“ angelegt, über dem man die zu übernehmenden Eigenschaften einstellen kann. U.a. werden Word-Querverweise zu Links umgewandelt und URL-Links vom Internet Assistant übernommen.

Von Adobe selbst gibt es Adobe PDFMaker. Es läuft erst ab Word 97. Sprachprobleme wie bei pdfMarker treten aber nicht auf. Die Liste der unterstützten Funktionen schließt die von pdfMarker ein. Erwähnenswert ist noch, daß Links für Fuß- und Endnoten, Notizen anstelle der Word-Kommentare und Artikelflüsse für Textboxen erzeugt werden. Tip: Auf der Homepage von Adobe ist PDFMaker nicht als eigenes Produkt aufgeführt, sondern man muß im Customer Support bzw. bei Downloadable Files suchen.

PDF-Formulare

PDF-Formulare unterscheiden sich nicht wesentlich von HTML-Formularen. Als Elemente stehen Kontrollkästchen, Optionsfelder, Schaltflächen, Listenfelder, Kombinationsfelder und Textfelder zur Verfügung. Formulare können direkt in Exchange erstellt werden. Die einzelnen Elemente werden per Maus plaziert und lassen sich mir wenigen Tastendrücken gegeneinander ausrichten. Über Kontextmenüs lassen sich die Eigenschaften einstellen. Will man aber Formularelemente in eine aufwendig gestaltete Seite einfügen, stößt man auf Schwierigkeiten, wenn sich das Ausrichtungsgitter von Exchange nicht wie gewünscht einstellen läßt.

Um die Formularseite mit einer anderen Software entwerfen zu können, die PDF nicht unterstützt, bieten sich EPS-Platzhalter mit passenden PDF-mark-Anweisungen an, die der Distiller in entsprechende Elemente umwandelt. (Siehe „Erzeugen von PDF-Dateien“ und „PDF-mark Anweisungen“.)

Schnittstelle zum Server

Zum Versenden der Formulardaten sind zwei Formate vorgesehen: URL-Kodierung und FDF von Adobe. FDF steht für Forms Data Format und kapselt im Prinzip lediglich ein Dictionary, das die Werte der Felder speichert. Zusätzlich kann ein Name für das Formular und eine ID

angegebene werden. Interessant an FDF ist die Möglichkeit, FDF-Daten vom Server an den Browser zurückzuschicken, der die Felder dann aktualisiert. So kann der Benutzer eine Antwort erhalten, ohne daß eine neue Seite geladen werden muß. Um entsprechende CGI-Skripte schreiben zu können, stellt Adobe ein FDF-Toolkit bereit, bislang nur für C.

Die URL-Kodierung verhält sich genauso wie bei HTML-Formularen. Vorhandene CGI-Skripte können daher übernommen werden.

Beispiel

Dieses Beispiel soll zeigen, wie mit Exchange in kurzer Zeit ein Formular erstellt werden kann. Sind viele Zeilen als Beschriftung notwendig, ist es einfacher, mit einer beliebiger Anwendung (hier: CorelDRAW) die Seite zu gestalten. Von dort aus erzeugt man dann die Ausgangs-PDF-Datei, entweder direkt mit PDF-Write oder über den Distiller. In Exchange öffnet man die Datei und fügt mit dem Formular-Werkzeug die Formularelemente ein. Jedes Feld muß einen Namen bekommen. Bei Optionsfelder regelt der Name die Zugehörigkeit. Der Export-Wert sollte dagegen innerhalb von Optionsfeldern eindeutig sein, um die Auswahl später auswerten zu können. Nach diesen Änderungen kann das fertige Formular gespeichert werden.

Probleme

Das Plugin für Netscape unter UNIX unterstützt das Versenden von Formulardaten nicht.

PDF-mark-Anweisungen

Mit PDF-mark-Anweisungen können in PostScript PDF-Elemente eingefügt werden, die über PostScript hinausgehen. Der Acrobat Distiller interpretiert die Anweisung und erzeugt entsprechende Objekte in der PDF-Datei. Aber wie gelangen diese Anweisungen in den PostScript-Code, wenn die Anwendung nicht für PDF vorbereitet ist? Eine Lösung, die mit vielen Programmen funktioniert, ist das Einbinden über EPS (Encapsulated PostScript). Über EPS werden normalerweise Graphiken in ein Dokument eingebunden. Die EPS-Graphik kann auf der Seite frei positioniert werden. Die Koordinatenangaben in der EPS-Datei sind immer relativ zum Ursprung der Graphik zu sehen. Bei der Ausgabe des gesamten Dokuments wird einfach der PostScript-Code aus der EPS-Datei hinzugenommen.

Im Verzeichnis zum PDF-Vortrag finden sich einige EPS-Dateien für Formularelemente zum Experimentieren. Als Muster dienten die Vorschläge in „Mit Acrobat ins World Wide Web“. Bei Kontrollkästchen und Optionsfeldern treten aber noch Ungereimtheiten auf. Die vorhandenen EPS-Dateien reichen aber, um das Layout festzulegen. Später müssen dann in Acrobat Exchange die Namen und Exportwerte der Felder gesetzt werden. Hier kann man auch noch den Typ der Formularelemente wechseln. Als Beispiel hier ein Textfeld:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 360 12
%%EndProlog
/pdfmark where {pop} {userdict /pdfmark /cleartomark load put} ifelse
[ /T (Texteingabefeld)   % title
  /Subtype /Widget
  /FT /Tx
  /Rect [ 0 0 360 12 ]
  /F 4
  /BS << /S /S /W 1 >>   % Border style solid, width = 1
  /MK <<
  /BC [ 1 0 0 ]
  /BG [ 1 1 1 ] >>
/ANN pdfmark
showpage
%%EOF
```

Die Zeilen bis %%EndProlog werden von der Anwendung ausgewertet. Hier steht u.a. wieviel Platz das EPS einnimmt. Die Angaben erfolgen in Punkt (1/72 Zoll). Die Zeile mit /pdfmark sorgt dafür, daß Interpreter, die keine PDF-mark-Anweisungen verstehen, sie überlesen. Dann folgt ab [die eigentliche PDF-mark-Anweisung. Schließlich folgt noch ein showpage, das manche Programme erwarten.

Beispiel

Als Beispiel soll ein Adressformular dienen. Es enthält viele Texteingabefelder. Wie im ersten Formularbeispiel wurde zuerst mit CorelDRAW die Seite gestaltet. Zusätzlichen wurde aber obige EPS-Datei mehrmals eingefügt, passend skaliert und angeordnet. Dieser Entwurf mußte dann nur noch in Exchange leicht überarbeitet werden, um das fertige Formular zu erhalten. Theoretisch könnte man die Eingabefelder auch in Exchange wie im ersten Beispiel anlegen. Da aber die Beschriftungen der Eingabefelder mit einem Symbol verziert sind, können sie nicht erst in Exchange passend zum Ausrichtungsgitter angelegt werden. Dann würde es natürlich schwierig, die Eingabefelder immer im gleichen Abstand zur Beschriftung zu positionieren.

Volltextindizierung

Das Programm Acrobat Catalog erstellt aus mehreren PDF-Dokumenten einen Index, der den PDF-Dokumenten in Exchange über Datei - Dokumentinfo - Index zugeordnet werden kann. Dann steht neben der normalen Suche im Dokument auch Suchfunktion für den gesamten indizierten Dokumentensatz bereit. Leider ist diese Funktion für den Webeinsatz nicht geeignet. Die Indexdatei wird i.d.R. sehr groß und muß lokal vorhanden sein.

Um die Übertragung des Index zu vermeiden, muß der Web-Server die Suche selbst durchführen. Für diese Aufgabe wird seit längerem Indizierungssoftware für Web-Server angeboten, die Indizes für Dokumente in vielen Formaten generieren kann. Da die Suchanfrage über CGI-Aufrufe realisiert ist, braucht der Klient keine spezielle Software sondern nur seinen Browser. Eine Liste von Indizierungssoftware mit PDF-Unterstützung zeigt, daß viele Hersteller PDF berücksichtigen:

- Verity: Search97
- Microsoft Index Server + PDF-IFilter
- Netscape Compass Server
- Infoseek Ultraseek Server
- Cascade: Mediasphere/3
- Excalibur Technologies Corporation
- Fulcrum Surfboard v2i
- Glyphia
- Muscat
- Open Text Corporation: Livelink Search
- Personal Library Software PLWeb

Damit die Fundstellen im PDF-Dokument hervorgehoben werden können, kann dem Browser-Plugin mit #xml= eine Datei angegeben werden, die beschreibt, welche Stellen hervorzuheben sind. Wie folgendes Beispiel zeigt, reicht dem Plugin nicht eine relative Dateiangabe:

- hervorhe.pdf mit hervor.txt als Highlight-Datei:
  - hervorhe.pdf#xml=http://www2.informatik.Uni-Osnabrueck.DE/web_pub_9/hervor.txt
  - hervorhe.pdf#xml=hervor.txt

Die Datei hervor.txt:

```
<XML>
<Body units=words color=#FF0000 mode=active version=2>
```

```
  <Highlight>
    <loc pg=0 pos=5 len=1>
    <loc pg=0 pos=12 len=1>
    <loc pg=0 pos=16 len=3>
    <loc pg=0 pos=22 len=5>
  </Highlight>
</Body>
</XML>
```

Verschlüsselte PDF-Dokumente

In Acrobat Exchange kann man eine Reihe von Zugriffsbeschränkung aktivieren. Das Öffnen, Ändern der Sicherheitsoptionen, Drucken, Ändern des Dokuments, Auswählen von Text und Graphiken und das Hinzufügen oder Ändern von Notizen und Formularfeldern läßt sich verhindern. Die Einhaltung der Beschränkungen muß aber vom PDF-Viewer gewährleistet werden. Nur wenn ein Paßwort zum Öffnen der Datei notwendig ist, wird der Inhalt geschützt.

Adobe setzt zur Verschlüsselung RC4 von RSA Data Security ein. Es wird aber nicht die ganze PDF-Datei verschlüsselt, sondern die Datenbereiche der Objekte. Für jedes Objekt muß ein neues Paßwort aus dem eingegebenen Paßwort generiert werden, da RC4 als „symmetric stream cipher“ benutzt wird. D.h. der Output des RC4-Algorithmus wird mit den Daten lediglich Exklusiv-Oder-Verknüpft. Wäre das Paßwort immer gleich, könnte man Text leicht entschlüsseln.

(Nachtrag)

Im Vortrag behauptete ich, daß wegen obiger Eigenschaft für jedes PDF-Dokument ein neues Paßwort gewählt werden sollte. Begründet wurde dies mit der Formel, mit der das eigentliche RC4-Paßwort bestimmt wird: MD5(key + object ID). Im PDF Reference Manual wird aber auch eine FileID mit einbezogen, also müßte die Formel MD5(key + object ID + file ID) lauten. Die FileID setzt sich u.a. aus Dateiname, -größe und aktueller Uhrzeit zusammen. Das sollte reichen, um zu verhindern, daß zwei Objekte mit dem gleichen Paßwort verschlüsselt werden. MD5 ist eine Funktion, die 128-Bit Zahlen in sich selbst abbildet. Da die Umkehrfunktion schwer zu berechnen ist, verhindert sie, daß jemand das eingegebene Paßwort bestimmen kann, selbst wenn dieser jemand RC4 bricht.

Quellenangaben

- Adobe
  - "Portable Document Format Reference Manual" Adobe Systems Incorporated
  - "Publizieren elektronischer Dokumente" Adobe User Education
  - "Acrobat Exchange 3.0 Online-Handbuch" Adobe User Education
  - "Adobe Acrobat 3.0 Online-Handbuch" Adobe User Education
  - "Acrobat Capture 3.0 Online-Handbuch" Adobe User Education
  - "pdfmark Reference Manual" Adobe Developer Relations
  - "Guidelines for Using Adobe Acrobat 3.0 for Production Printing" Adobe Customer Services
- c't Magazin für Computer und Technik
  - "Gut verpackt - Drucken von JPEG-Bildern mit PostScript Level 2" Heft 6 1994 Seite 236 ff
  - "Elektronen statt Papier - Adobes Internet-Pläne" Heft 1 1997 Seite 49
  - "Auf Papier und Bildschirm - Adobe Acrobat 3.01" Heft 10 1997 Seite 118
- "PDFlib Reference Manual" Thomas Merz
- "Mit Acrobat ins World Wide Web" Thomas Merz
- "PostScript & Acrobat/PDF" Thomas Merz