

# PDF/UA – eine technische Einführung



 **PDFlib**® Whitepaper

## Der Standard PDF/UA für Barrierefreiheit

In der Vergangenheit hatten PDF-Dokumente in Bezug auf ihre allgemeine Zugänglichkeit (Barrierefreiheit) keinen guten Ruf. Während bei PDF das Erscheinungsbild eines Dokuments hervorragend langfristig und eindeutig erhalten werden kann, lässt sich die logische Struktur des Inhalts im Allgemeinen nicht gut bewahren. Das liegt vor allem daran, dass das Format aus der PostScript-Drucktechnologie kommt und nicht aus strukturierten Dokumentformaten wie XML. Dieses Defizit wurde mit der Einführung von »Tagged PDF« in PDF 1.4 (Acrobat 5, veröffentlicht im Jahr 2001) ausgeglichen, allerdings fand es während der ersten zehn Jahre noch keine große Verbreitung. Die PDF/UA-Norm ISO 14289, im Jahr 2012 veröffentlicht und 2014 geringfügig aktualisiert, baut auf den Dokumentstrukturen von Tagged PDF auf und erweitert das Format aus. Aus dem Standard:

»Das Hauptanliegen der ISO-Norm 14289 (auch PDF/UA genannt) ist es zu definieren, wie elektronische Dokumente im PDF-Format in einer Weise dargestellt werden können, die Dateien zugänglich/barrierefrei macht.«

## Web Content Accessibility Guidelines (WCAG) 2.0

Um die Anforderungen an barrierefreies PDF zu verstehen, sollte man einen Blick auf die Anforderungen an Barrierefreiheit für das Web werfen. Das W3C veröffentlichte seine »Richtlinien für barrierefreie Webinhalte (WCAG) 2.0« im Jahr 2008 (auch als ISO/IEC 40500:2012 standardisiert). Das Dokument enthält zwölf allgemeine Leitlinien für die Zugänglichkeit, die sich an den folgenden Grundsätzen orientieren:

- ▶ Inhalte sollten wahrnehmbar sein, für Bilder und andere nicht-textuelle Inhalte sollte zum Beispiel Alternativtext vorhanden sein.
- ▶ Inhalte sollten bedienbar sein, zum Beispiel sollte die komplette Funktionalität über die Tastatur gesteuert werden können.
- ▶ Inhalte sollten verständlich sein, der gesamte Text muss zum Beispiel lesbar sein.
- ▶ Inhalte sollten robust, also auf Kompatibilität mit heutigen und zukünftigen Tools ausgerichtet sein.

Diese Richtlinien beschreiben insgesamt 61 Erfolgskriterien, die erfüllt sein müssen, um Inhalte zugänglich zu machen. Die Erfolgskriterien sind in drei Konformitätsstufen gegliedert, die von A (minimale Konformität) über ein mittleres Niveau AA bis auf die höchste Stufe AAA reichen. Um die minimalen Anforderungen für Barrierefreiheit zu erreichen, müssen nur die Kriterien der Stufe A erfüllt werden.

### Anwendung von WCAG auf PDF

WCAG ist ein Technologie-unabhängiger Standard: Er ist nicht auf bestimmte Technologien wie HTML oder serverseitiges Scripting ausgerichtet, sondern beschreibt Anforderungen an die Barrierefreiheit in einer sehr allgemeinen Weise. Ein empfehlendes (d.h. nicht-normatives) Dokument mit dem Titel »Techniques for WCAG 2.0« stellt allgemeine Verfahren zur Erreichung der WCAG-Konformität und

Verfahren für spezifische Technologien wie HTML, CSS, Flash und PDF vor. Die Empfehlung »PDF Techniques for WCAG 2.0« wendet WCAG-Anforderungen auf PDF an und demonstriert anhand nützlicher Beispiele detailliert, wie bestimmte Anforderungen für Barrierefreiheit eingehalten werden können. Die Beispiele richten sich an Anwender von Acrobat, an Software für die Erstellung von Content (z.B. Microsoft Word, OpenOffice) sowie an PDF-Entwickler.

Als Service für PDFlib-Benutzer erläutert das Dokument »PDFlib Techniques for WCAG 2.0« (verfügbar auf der PDFlib-Website), wie die WCAG-Anforderungen bei der Erstellung von barrierefreien Dokumenten mit PDFlib eingehalten werden können.

## WCAG und PDF/UA

Obwohl die PDF-Verfahren des W3C nützlich sind, decken sie nicht alle technischen PDF-Details zur Erreichung der vollen Zugänglichkeit ab. Der PDF/UA-Standard erklärt und vereinfacht die PDF-Anforderungen zur Einhaltung der WCAG 2.0-Empfehlungen. Während die PDF/UA-Konformität die WCAG-Konformität für native PDF-Inhalte gewährleistet, stellt WCAG in einigen Bereichen weitere Anforderungen. Insbesondere müssen zusätzlich zu PDF/UA-Anforderungen WCAG-Anforderungen für eingebettete Multimedia-Inhalte und Scripting erfüllt sein, um solche Inhalte in PDF barrierefrei zu machen.

In einem von AIIM veröffentlichten Dokument ist die Beziehung zwischen WCAG- und PDF/UA-Regeln ausführlich erläutert. Das Dokument »Achieving WCAG 2.0 with PDF/UA« enthält eine Referenztabelle mit den Zuordnungen zwischen WCAG-Erfolgskriterien und den entsprechenden PDF/UA-Anforderungen.

PDF/UA erweitert nur bei zwei Themen die Anforderungen von WCAG: Zur Verbesserung der Navigation in Dokumenten schreibt PDF/UA anders als WCAG bestimmte Verschachtelungsregeln für Überschriften-Tags vor. Außerdem muss die Artikelfunktion von PDF korrekt angewendet werden; dieses PDF-Feature gilt jedoch als veraltet und wird in neu erzeugten PDF-Dokumenten nur noch selten genutzt.

Beachten Sie, dass ein PDF-Dokument konform zu WCAG 2.0 sein kann, ohne PDF/UA-konform zu sein. Allerdings kann WCAG-Konformität viel einfacher mit PDF/UA als ohne erreicht werden.

## Grundsätze von Tagged PDF

Werfen wir zunächst einen Blick auf Tagged PDF, da PDF/UA darauf basiert. Während PDF ursprünglich darauf ausgerichtet war, das visuelle Erscheinungsbild eines Dokuments zuverlässig zu erhalten, sind Informationen über die Dokumentstruktur nicht unbedingt enthalten. Zum Beispiel können Überschriften in Großbuchstaben oder fett gedruckt sein, aber es fehlt in einfachen PDF-Dokumenten ein expliziter Marker für »Überschrift«. Ähnlich wie bei XML-basierten Markup-Sprachen können die Inhalte im Tagged PDF »markiert« und in einer strukturellen Dokumenthierarchie organisiert werden. Jedem inhaltlich relevanten Element ist eine bestimmte Stelle in dieser Hierarchie zugewiesen. Inhaltlich nicht relevante Elemente (z.B. Seitenzahlen) werden als *Artifacts* ausgezeichnet.

Die logische Struktur in einem Tagged PDF wird von einer Hierarchie der Elemente beschrieben, der sogenannten Strukturhierarchie (auch logische Struktur oder Tag-Baum). Ausgehend von der obersten Ebene (*Root*, oft auch als Element *Document* bezeichnet), besteht die Strukturhierarchie aus einer beliebigen Anzahl von Ebenen. Auf jeder dieser Ebenen kann ein Element null oder mehrere der folgenden Objekttypen enthalten:

- ▶ Weitere Strukturelemente; das Element *Document* kann zum Beispiel mehrere Elemente vom Typ *Article* enthalten und jedes Element *Article* wiederum beliebig viele Elemente vom Typ *P* (Paragraph, Absatz).
- ▶ Seiteninhalt wie markierter Text und Grafiken auf einer Seite, aus importierten Bildern oder PDF-Seiten erzeugte XObjects, Anmerkungen und Formularfelder. Diese Objekte repräsentieren den zu einem Strukturelement gehörenden Inhalt.

Tagged PDF löst den potenziellen Konflikt zwischen der Reihenfolge der Content-Erstellung und der logischen Lesereihenfolge: der Inhalt einer PDF-Seite kann in beliebiger Reihenfolge erstellt werden, aber das Lesen des Textes in dieser ursprünglichen Reihenfolge entspricht nicht unbedingt der logischen Reihenfolge der Seiteninhalte. Im Gegensatz dazu ordnet der Strukturbaum Seiteninhalte nach ihrer logischen Reihenfolge, d.h. in der Reihenfolge, in der ein Mensch erwarten würde, sie zu lesen.

## PDF/UA-1-Anforderungen an das Dateiformat

PDF/UA-1 basiert auf der ISO-Norm 32000-1 (PDF 1.7). PDF/UA fügt keine neuen Features hinzu, aber einige in PDF 1.7 noch optionale Aspekte sind bei PDF/UA-1 verbindlich vorgeschrieben. Die folgenden Voraussetzungen müssen in allen PDF/UA-1-Dokumenten erfüllt sein:

- ▶ Das Dokument muss getagged sein. Während PDF 1.7 Anforderungen in Bezug auf die Verschachtelung und die Beziehung der verschiedenen Strukturelement-Typen stellt, erweitert und verfeinert PDF/UA-1 diese Regeln (siehe unten für Details).
- ▶ Alle im Dokument verwendeten Fonts müssen eingebettet sein (außer den Fonts für unsichtbaren Text wie OCR-Ergebnisse).
- ▶ Bestimmte Optionen für Ebenen sind nicht erlaubt.
- ▶ Externe Inhalte sind nicht erlaubt (referenzierte XObjects wie bei PDF/X-5 vorgeschrieben).
- ▶ Der Dokumenttitel muss in den Metadaten des Dokuments hinterlegt werden.

### Semantische Anforderungen

Bei der Erzeugung der Strukturhierarchie für PDF/UA-1 müssen die folgenden semantischen Aspekte berücksichtigt werden:

- ▶ Beim Tagging müssen die zur Dokumentstruktur passenden Strukturelemente verwendet werden: Eine Überschrift muss als solche ausgezeichnet werden. Eine Tabelle muss als Tabelle ausgezeichnet werden. Eine Liste muss als Liste ausgezeichnet werden.
- ▶ Semantisch nicht relevante Inhalte dürfen nicht in die Dokumenthierarchie aufgenommen, sondern müssen als *Artifacts* gekennzeichnet werden. Typische Beispiele sind hier laufende Kopf-/Fußzeilen, Seitenzahlen und Hintergrundbilder.
- ▶ Strukturelemente müssen in der logischen Lesereihenfolge angeordnet sein.
- ▶ Wenn Informationen aufgrund von Farbe, Format oder Layout nicht anders zugänglich sind, muss der Inhalt entsprechend ausgezeichnet sein.
- ▶ Text in einer Grafik verlangt das Attribut *Alt* und eine Erklärung, falls der Text keine natürliche Sprache ist (z.B. bei Font- oder Skript-Beispielen).
- ▶ Für Bilder muss ein Alternativtext vorliegen; Bildunterschriften müssen mit dem Tag *Caption* versehen sein.
- ▶ Für Links muss das zugehörige Element *Link* hinterlegt sein.
- ▶ Für logisch zusammenhängende Gruppen von Grafikelementen darf nur ein einziges Tag *Figure* vergeben werden.
- ▶ Fußnoten, Endnoten, Labels für Anmerkungen sowie Verweise auf Stellen im Dokument müssen entsprechend als *Note* oder *Reference* ausgezeichnet sein.

Die oben beschriebenen semantischen Voraussetzungen erschweren die automatische Konvertierung nicht getaggtter PDF-Dokumente in konformes PDF/UA oder machen sie sogar unmöglich. Genauso wenig wird die Anwendung von OCR-Verfahren auf gescannte Dokumente ohne menschliche Intervention zu voll konformem PDF/UA führen. Zum Beispiel können alternative Texte für Bilder nicht automatisch hergeleitet werden.

### Anforderungen an spezifische Tags

Alle in PDF 1.7 definierten Standard-Tags können in PDF/UA-1 verwendet werden. Bei der Verwendung anderer Tags muss eine Zuordnung solcher benutzerdefinierter Tags auf die Standard-Tags in der Datenstruktur *Rolemap* des Dokuments hinterlegt werden. Für bestimmte Standard-Elementtypen müssen verschiedene Regeln eingehalten werden:

- ▶ Das Element *Figure* für Bilder und Grafiken, die keine Artefakte sind, verlangt das Attribut *Alt* oder *ActualText*.
- ▶ Für logische Tabellen müssen Elemente vom Typ *Table* erzeugt werden. Für Tabellen, die nur zu Layoutzwecken erstellt wurden, dürfen diese Elemente jedoch nicht benutzt werden. Tabellen-Tags müssen korrekt verschachtelt sein, das Tag *Table* muss zum Beispiel Tabellenzeilen vom Typ *TR* enthalten, die aus Kopfzellen *TH* oder Datenzellen *TD* bestehen. Für Kopfzeilen-Elemente *TH* wird das Attribut *Scope* verlangt.
- ▶ Tags für Überschriften müssen korrekt verschachtelt sein. Bei der Verwendung von Tags für nummerierte Überschriften (*H1*, *H2*, ...) müssen diese korrekt verschachtelt werden (d.h. es darf keine Ebene übersprungen werden). Bei der Verwendung von Tags für unnummerierte Überschriften (sogenannte streng strukturierte Dokumente) muss das Tag *H* verwendet werden, aber nicht mehr als einmal innerhalb eines Knotens im Strukturbaum. Überschriftenelemente dürfen keine Unterelemente haben.

## Anforderungen an bestimmte Inhaltstypen

- ▶ Das Listenelement vom Typ *L* verlangt das Attribut *ListNumbering*, das das in der Liste verwendete Nummerierungssystem angibt, z.B. *Disc* für einen Kreis anstelle von Nummern, *Decimal* oder *UpperRoman*.

Die folgenden Voraussetzungen müssen für verschiedene Typen von PDF-Inhalten erfüllt sein:

- ▶ Die natürliche Sprache eines Textes muss deklariert sein, entweder für das gesamte Dokument über das Dokumentinfo-Element *Lang* oder für einzelne Strukturelemente über das Attribut *Lang*. Unsichtbarer Text muss als *Artifact* ausgezeichnet sein, es sei denn es gibt eine dargestellte Entsprechung (zum Beispiel ein gescanntes Bild).
- ▶ Vektorgrafiken und Rasterbilder müssen als *Figure* oder *Artifact* ausgezeichnet sein.
- ▶ Anmerkungen und Formularfelder müssen im Strukturbaum enthalten sein und verlangen bestimmte Flags, um die Zugänglichkeit zu gewährleisten.

## Unicode-Anforderungen

PDF/UA verlangt korrekte Unicode-Semantik für den gesamten Text im Dokument. Diese Anforderung rührt daher, dass PDF eine Vielzahl von Font- und Encoding-Techniken unterstützt, die nicht alle Unicode-fähig sind. Zum Beispiel unterstützt PDF die in den 1980er Jahren eingeführten PostScript-Type-1-Fonts, während das Unicode-Konsortium seine Arbeit erst im Jahr 1991 aufnahm. Für die PDF/UA-Konformität müssen zusätzliche Unicode-Werte für alle Zeichen von solchen Fonts enthalten sein, die diese Angaben nicht schon intern enthalten. Aber nicht alle Unicode-Werte sind akzeptabel: Werte in der Private Use Area (PUA) sind nicht erlaubt, da sie keine allgemeingültige Interpretation (Semantik) haben.

Symbolfonts wie Fonts mit Logos oder Piktogrammen sind ein wichtiger Bereich dieser PDF/UA-Anforderung. Da standardisierte Unicode-Werte für benutzerdefinierte symbolische Glyphen nicht verfügbar sind, muss geeignete Unicode-Semantik im Attribut *ActualText* übergeben werden. *ActualText* kann einer einzelnen Glyphe oder einer Folge von mehreren Glyphen zugeordnet werden und aus einer beliebigen Unicode-Zeichenfolge bestehen.

Als Beispiel nehmen wir Code 0x1A aus dem gängigen Font Wingdings. Dieses Zeichen zeigt das Bild einer Computer-Tastatur mit dem Glyphnamen *keyboard* und dem PUA-Unicode-Wert U+F037, der in PDF/UA-1 nicht akzeptiert wird. Der Glyphname könnte für die Erzeugung eines passenden *ActualText* verwendet werden, so zum Beispiel »Symbol für Keyboard«. Anzumerken ist, dass das programmatische Erzeugen von *ActualText* eine Notlösung bleibt; von einem menschlichen Betrachter ausgewählte Texte sind maschinengeneriertem *ActualText* immer vorzuziehen.

## Weitere Empfehlungen

Obwohl in PDF/UA-1 nicht explizit vorgeschrieben, werden folgende Elemente empfohlen:

- ▶ Zur besseren Navigation sind Lesezeichen zu empfehlen. Sie sollten die Lesereihenfolge und die Verschachtelung des Inhalts korrekt wiedergeben.
- ▶ Tabellen sollten mit Kopfzeilen versehen sein.
- ▶ Der Inhalt von Dateianhängen sollte selbst auch zugänglich sein.
- ▶ Wenn vorhanden, sollten Seiten-Labels korrekt verwendet werden (z.B. römische Seitenzahlen).

## PDF/UA und PDF/A

Die Archivierungsstandards PDF/A-1a, PDF/A-2a und PDF/A-3a erfordern den Einsatz von Tagged PDF. Obwohl es keine direkte Beziehung zwischen PDF/A-1a/2a/3a und PDF/UA-1 gibt, kann ein Dokument zugleich beiden Standards entsprechen. Wir empfehlen sogar, die PDF/UA-1-Anforderungen einzuhalten, wenn Sie PDF/A mit Kompatibilitätsstufe A erstellen möchten, um die Zugänglichkeit zu verbessern. Weitere Informationen hierzu können Sie unserem Whitepaper über PDF/A auf der PDFlib-Website entnehmen.

Wir empfehlen, statt PDF/A-1a die neueren Standards PDF/A-2a oder PDF/A-3a zu verwenden, denn zwischen PDF/UA-1 und PDF/A-1a besteht ein gewisser Konflikt: PDF/UA-1 erfordert den Eintrag *Tabs* für Seiten mit Anmerkungen. Dieser Schlüssel gibt die Tab-Reihenfolge für die Anmerkungen auf der Seite an und erfordert die Angabe der »Strukturreihenfolge«. Allerdings ist dieser Schlüssel in PDF 1.4 nicht verfügbar und kann daher nicht in kombinierten PDF/A-1a- und PDF/UA-1-Dokumenten verwendet werden.

## Das Matterhorn-Protokoll

Der PDF/UA-1-Standard definiert Regeln für die Einhaltung der Standardkonformität eines Dokuments. Das PDF/UA Competence Center, eine Initiative der PDF Association, hat eine umfassende Liste aller möglichen PDF/UA-1-Standardverletzungen erstellt. Diese Liste, das sogenannte Matterhorn-Protokoll, enthält Richtlinien für Software-Entwickler und Dokumententester zur Erzielung hundertprozentiger Barrierefreiheit.

Das Matterhorn-Protokoll formuliert die Anforderungen an die Standard-Konformität erneut und listet alle möglichen Verstöße als 136 »Fehlerbedingungen« auf, die in 31 Prüfbereiche gruppiert sind. Während die meisten Fehlerbedingungen programmatisch identifiziert werden können, erfordern manche interaktive Unterstützung. Einige Beispiele für Fehlerbedingungen, die programmatisch identifiziert werden können:

- ▶ Einer Abbildung fehlt der Alternativ- oder Ersatztext.
- ▶ Eine Link-Anmerkung ist nicht von einem *Link*-Tag umschlossen.
- ▶ XMP-Metadaten enthalten keinen Titel.

Einige Beispiele für Fehlerbedingungen, die interaktive Bewertung erfordern:

- ▶ Überschriftentext ist nicht mit einem *Heading*-Tag ausgezeichnet.
- ▶ Irrelevante Inhalte sind nicht als *Artifact* markiert.
- ▶ OCR-Text enthält Fehler bei der Bestimmung von Zeichen.
- ▶ Inhalte werden als Liste dargestellt, sind aber nicht als solche markiert.

Das Matterhorn-Protokoll kann als Richtlinie für die Ermittlung von PDF/UA-1-Standardverletzungen verwendet werden. Die Prüfung auf PDF/UA-Konformität kann z.B. mit dem PDF Accessibility Checker (PAC) durchgeführt werden, einem kostenfreien PDF/UA-Validierungstool der Schweizer Stiftung *Zugang für alle*. Das Matterhorn-Protokoll ist in PAC implementiert.

## PDF/UA-konforme Reader und assistive Technologie (AT)

Der PDF/UA-1-Standard spezifiziert nicht nur Anforderungen an das Dateiformat, sondern auch an das Verhalten von »konformen Readern« und »assistiver Technologie« (AT). Ein konformer Reader ist ein Programm, das PDF-Dateien gemäß der Vorgaben von ISO 32000-1 (also PDF 1.7) verarbeitet und zusätzliche Vorschriften einhält. Unter AT versteht man »von Menschen mit einer Beeinträchtigung verwendete Software und/oder Hardware, die eine alternative Steuerung und/oder Bereitstellung ermöglicht, um deren Zugang zu und Einsatz von verfügbarer Funktionalität und Information zu erleichtern«.

AT verarbeitet PDF-Dokumente zumeist nicht direkt, sondern in Kombination mit einem PDF-Reader. Dies kann zum Beispiel über Programmierschnittstellen wie MSAA von Windows realisiert werden. Typische Beispiele für AT sind Screenreader-Programme, die den Bildschirminhalt auf einem Braille-Gerät ausgeben, sowie Software zur Sprachsynthese, die Benutzern den Inhalt von Dokumenten vorliest. Ein konformer Reader muss die für die Zugänglichkeit in einem PDF/UA-Dokument hinterlegte Zusatzinformation korrekt interpretieren und an das AT-System übergeben. Konforme Reader müssen Dokumentnavigation über Seiten-Labels, Strukturhierarchie oder Lesezeichen unterstützen und müssen mit AT-Systemen so zusammenarbeiten, dass die folgenden Elemente zur Verfügung stehen:

- ▶ Strukturelemente und Artefakte;
- ▶ Bedienoberfläche;
- ▶ Text in logischer Lesereihenfolge zusammen mit der natürlichen Sprache des Textes;
- ▶ Informationen über leere Tabellenzellen;
- ▶ Informationen über Einsatz und Namen von Ebenen (optionale Inhalte);
- ▶ Namen und Inhalt von Dateianhängen;
- ▶ Metadaten und Informationen über digitale Signaturen;
- ▶ alternative Beschreibungen von Anmerkungen und Beschreibungen von Formularfeldern.

Ein konformes AT-Gerät muss die von einem konformen Reader bereitgestellte Information verarbeiten und präsentieren können. Ebenso muss es Artefakte, d.h. Inhalte von Dokumenten, die nicht in der Baumstruktur enthalten sind, verarbeiten können. Wie ein konformer Reader muss konforme AT Dokumentnavigation über Seiten-Labels, Strukturhierarchie oder Lesezeichen unterstützen.

Wir gehen davon aus, dass kommerzielle assistive Software den PDF/UA-1-Standard bald unterstützen wird. Das NVDA-Projekt bietet einen kostenlosen Open-Source-Screenreader für Windows an. NVDA hat bereits angekündigt, PDF/UA-Unterstützung zukünftig in seine Software zu integrieren.

## PDF/UA-Unterstützung in Produkten der PDFlib GmbH

Alle Produkte der Produktfamilie PDFlib 9 unterstützen die Erzeugung von konformem PDF/UA gemäß des Standards:

- ▶ Das Basisprodukt PDFlib kann zur Erzeugung von PDF-Dokumenten verwendet werden, die auf Text, Grafik und Rasterbildern basieren, einschließlich der entsprechenden Struktur-Tags. Die Funktionalität Tagged PDF wurde bereits 2004 in PDFlib eingeführt.
- ▶ Die Produkterweiterung PDFlib+PDI erlaubt zusätzlich den Import von Seiten aus PDF/UA-Dokumenten einschließlich der zugehörigen Teile der Strukturhierarchie. PDFlib+PDI 9 ist damit weltweit das erste Produkt, mit dem bestehende PDF/UA-Dokumente zu einer gemeinsamen, wiederum standardkonformen Ausgabe zusammengefügt werden können (auch *Content Aggregation* genannt).
- ▶ PDFlib Personalization Server (PPS) ist das leistungsfähigste Produkt für template-basierte PDF-Erzeugung. Es unterstützt ebenfalls PDF/UA-Ausgabe, womit personalisierte barrierefreie PDF-Dokumente erzeugt werden können.

PDFlib ist kein Low-Level-Toolkit, mit dem irgendwie PDF/UA-1 erstellt werden kann. Vielmehr implementiert PDFlib alle Regeln und Anforderungen des PDF/UA-Standards: Es erstellt, wo möglich, automatisch barrierefreie Konstrukte und verhindert, dass Benutzer gegen die PDF/UA-Anforderungen verstoßen. Die resultierende Ausgabe genügt garantiert allen PDF/UA-Anforderungen, soweit diese programmatisch abprüfbar sind. Kriterien, die menschliche Beurteilung erfordern, liegen selbstverständlich in der Verantwortung der Benutzer.

Die Unterstützung von PDFlib für PDF/A-1a, PDF/A-2a und PDF/A-3a in Kombination mit PDF/UA-1 erleichtert die Erzeugung von PDF-Dateien, die sowohl zugänglich als auch archivierbar sind.

### Erzeugung von PDF/UA mit PDFlib

PDFlib 9 kümmert sich soweit möglich automatisch um die Einhaltung der PDF/UA-1-Anforderungen. Entwickler, die mit PDFlib-Produkten arbeiten, müssen Strukturinformationen an die PDFlib-Programmierschnittstelle übergeben, damit PDFlib die Strukturhierarchie des Dokuments erstellen kann. PDFlib wendet PDF/UA-1-Prüfungen auf alle vom Benutzer gelieferten Struktur-Tags und Attribute an, um standardkonforme Ausgabe zu gewährleisten. PDFlib zeichnet automatisch Tabellen und Artefakte aus, was eine große Zeitersparnis für Entwickler bedeutet und die Erstellung standardkonformer Strukturhierarchie deutlich vereinfacht. Für Bilder und Vektorgrafik kann Alternativtext hinterlegt werden.

Mit den PDFlib-Paketen wird Beispielcode für eine Vielzahl an Programmiersprachen und Entwicklungsumgebungen ausgeliefert. Im PDFlib-Cookbook werden weitere Programmiermethoden vorgestellt.

### Montieren von PDF/UA mit PDFlib+PDI und PPS

Mit PDFlib+PDI und dem PDFlib Personalization Server (PPS) können Seiten aus bestehenden Dokumenten zusammengefügt und mit neuen Inhalten angereichert werden. Mit PDFlib+PDI 9 ist es erstmals möglich, die entsprechenden Teile der Strukturhierarchie mit jeder Seite zu importieren und sie mit der generierten Strukturhierarchie des Dokuments zusammenzuführen. Wenn das importierte Dokument PDF/UA-konform ist, entspricht die resultierende Ausgabe auch der Norm. Mit dieser Funktionalität können barrierefreie Dokumente aus mehreren Einzelteilen zusammenmontiert werden.

### Digitales Signieren von PDF/UA mit PLOP DS

PDFlib PLOP DS ist unser Produkt für das digitale Signieren von PDF-Dokumenten. PLOP DS wendet Signaturen PDF/UA-konform an: ist die Eingabe PDF/UA-konform, so ist garantiert, dass auch die signierte Ausgabe den PDF/UA-Standard einhält.

**PDFlib GmbH**

Franziska-Bilek-Weg 9  
80339 München, Germany  
phone +49 • 89 • 452 33 84-0  
support@pdflib.com  
[www.pdflib.com/de/knowledge-base/pdfua](http://www.pdflib.com/de/knowledge-base/pdfua)

PDFlib GmbH ist auf die Entwicklung von PDF-Technologie spezialisiert. PDFlib-Produkte sind seit 1997 weltweit im Einsatz. Das Unternehmen berücksichtigt wichtige technologische Trends, etwa ISO-Standards für PDF. PDFlib GmbH vertreibt alle Produkte weltweit, wobei Nordamerika, Europa und Japan die wichtigsten Märkte darstellen.



Die PDF/A Association wurde 2006 als PDF/A Competence Center gegründet. Ihr Ziel ist die Einführung und Umsetzung internationaler Standards für die PDF-Technologie.

- ▶ Entwickler nutzen die PDF Association, um Kenntnisse und Erfahrungen mit der PDF-Technologie auszutauschen.
- ▶ Entscheider nutzen die PDF Association, um Funktionen und Möglichkeiten des PDF Standards sowie seiner Sub-Standards im ECM und anderen Anwendungen für digitale Dokumente kennenzulernen.
- ▶ Anwender erhalten Vorteile durch erhöhte Verlässlichkeit, Qualität und Funktionalität sowie Kompatibilität beim Einsatz digitaler Dokumente.