RICOH

RICOH Transforms Suite

PDF2PDFUA-Tool Anwenderanleitung

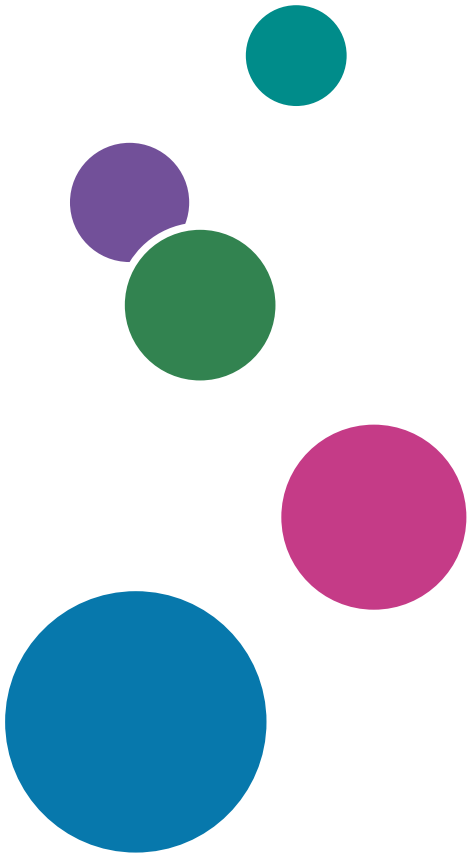
Version 1.5

Verwendung des PDF2PDFUA-Tools 1

Gewährleistung der Konformität mit PDF/UA und WCAG 2

Java-Anwendungsprogrammierschnittstelle 3

Informationen, die nicht in diesem Handbuch enthalten sind, finden Sie im Hilfe-System Ihres Produkts.



INHALT

Einleitung

Wichtig ... 2
Vorsichtshinweise zu dieser Anleitung ... 2
Veröffentlichungen zu diesem Produkt ... 2
Symbole ... 2
Abkürzungen ... 3
Marken ... 3

1 Verwendung des PDF2PDFUA-Tools

Verwendung des PDF2PDFUA-Tools über die AFP2PDF Plus-Umwandlung ... 5
Verwendung des PDF2PDFUA-Tools über die Befehlszeile ... 5
Syntax ... 6
Parameter ... 6
Konfigurieren der Optionsdatei `PDF2PDFUA.cfg` ... 7
Von der PDF2PDFUA-Umwandlung unterstützte Funktionen ... 8

2 Gewährleistung der Konformität mit PDF/UA und WCAG

Anforderungen an die Eingabedatei für die Einhaltung der PDF/UA-Standards ... 9
Überprüfung der PDF-Eingabedatei und der PDF-UA-Ausgabedatei ... 9
Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.0 (Stufe AA) ... 9
Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.0 (Stufe AAA) ... 10
Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.1 (Stufe AA) ... 10
Überprüfung der Konformität mit PDF/UA oder WCAG ... 10

3 Java-Anwendungsprogrammierschnittstelle

Klassen ... 11
Beispiele ... 15

Einleitung

Wichtig

Soweit es die geltenden Gesetze zulassen, haftet der Hersteller in keinem Fall für Schäden, die sich aus Fehlern dieses Produkts, dem Verlust von Dokumenten oder Daten oder der Verwendung oder Nichtverwendung dieses Produkts und der mitgelieferten Bedienungsanleitungen ergeben.

Vergewissern Sie sich, dass Sie immer Kopien oder Sicherungskopien von wichtigen Dokumenten oder Daten haben. Dokumente oder Daten können aufgrund von Bedienungsfehlern oder Fehlfunktionen der Software gelöscht werden. Außerdem sind Sie dafür verantwortlich, Schutzmaßnahmen gegen Computerviren, Würmer und andere schädliche Software zu ergreifen.

Der Hersteller ist in keinem Fall für von Ihnen mit diesem Produkt erstellte Dokumente oder Ergebnisse der von Ihnen ausgeführten Daten haftbar.

Vorsichtshinweise zu dieser Anleitung

- Einige Abbildungen oder Erklärungen in dieser Anleitung können aufgrund von Verbesserungen oder Änderungen am Produkt von Ihrem Produkt abweichen.
- Die Inhalte dieses Dokuments können ohne vorherige Ankündigung geändert werden.
- Kein Teil dieses Dokuments darf ohne vorherige Zustimmung des Anbieters vervielfältigt, dupliziert, in irgendeiner Form reproduziert, verändert oder zitiert werden.

Veröffentlichungen zu diesem Produkt

Diese Veröffentlichung bietet einen Überblick über das Tool sowie Informationen zur Installation und zur Verwendung des PDF2PDFUA-Tools.

Bedienungsanleitungen

Weitere Informationen zum Tool PDF2PDFUA finden Sie in den folgenden Dokumenten:

- *PDF2PDFUA-Installationsanleitung*
- *PDF2PDFUA-Anwenderanleitung*

Symbole

Die nachstehenden Symbole werden in diesem Handbuch verwendet, um Ihnen ein schnelles Auffinden von Inhalten zu ermöglichen.



Dieses Symbol zeigt Punkte an, auf die Sie bei der Verwendung der Anwendung achten sollen. Lesen Sie diese Erläuterungen unbedingt durch.

Hinweis

Dieses Symbol zeigt Zusatzinformationen an, die gegebenenfalls hilfreich, jedoch nicht notwendig sind, um eine Aufgabe abzuschließen.

[Fett]

[Fettschrift] in eckigen Klammern markiert die Namen von Menüs, Menüelementen, Einstellungen, Feldern, Schaltflächen und Tasten.

`Kursiv`

`Kursivschrift` markiert Variablen, die Sie durch eigene Informationen ersetzen müssen.

`Monospace`

`Monospace-Schriftart` markiert Computereingaben und -ausgaben.

...

Auslassungspunkte bedeuten, dass eine Aufzählungsreihe weiter fortgesetzt werden kann.

Abkürzungen

AES

Advanced Encryption Standard

AFP

Advanced Function Presentation

PDF

Portable Document Format

UA

Universelle Barrierefreiheit

WCAG

Richtlinien zur Barrierefreiheit von Webinhalten

Marken

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sowie PDF und das PDF-Logo sind entweder eingetragene Marken oder Marken von Adobe in den Vereinigten Staaten und/oder anderen Ländern.

Linux ist die eingetragene Marke von Linus Torvalds in den USA und anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken oder eingetragene Marken der Microsoft Corporation in den Vereinigten Staaten und/oder anderen Ländern.

Sentinel ist eine eingetragene Marke von Thales DIS CPL USA, Inc.

UNIX ist eine eingetragene Marke von The Open Group in den Vereinigten Staaten und/oder anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicenamen können Marken anderer Hersteller sein.

1. Verwendung des PDF2PDFUA-Tools

- **Verwendung des PDF2PDFUA-Tools über die AFP2PDF Plus-Umwandlung**
- **Verwendung des PDF2PDFUA-Tools über die Befehlszeile**
- **Syntax**
- **Parameter**
- **Konfigurieren der Optionsdatei** `PDF2PDFUA.cfg`
- **Von der PDF2PDFUA-Umwandlung unterstützte Funktionen**

Verwendung des PDF2PDFUA-Tools über die AFP2PDF Plus-Umwandlung

Um AFP-Dateien in PDF/UA-Dokumente zu konvertieren, müssen Sie diesen Optionsparameter in die Konfigurationsdatei von AFP2PDF Plus hinzufügen:

PDFUA_CONTROL_FILE=*Datei*

Gibt den vollqualifizierten Namen der Steuerdatei des PDF2PDFUA-Tools an. Bei dieser Datei handelt es sich um eine XML-Datei, die Informationen zum Inhalt der Eingabedatei enthält. Das PDF2PDFUA-Tool verwendet das Dokument, um den Inhalt gemäß den Anforderungen des PDF/UA-Standards zu kennzeichnen.

PDFUA_CACHE_SIZE=*Zahl*

Legt die Anzahl der am häufigsten verwendeten PDFUA-Steuerdateien fest, die im Cache gespeichert sind. Die Zahl sollte größer als 0 sein.

Hinweis

- Für eine verschlüsselte oder linearisierte PDF/UA-Ausgabedatei nehmen Sie bitte die entsprechenden Einträge in der AFP2PDF Plus-Optionsdatei vor, wenn Sie die AFP2PDF Plus-Umwandlung ausführen:

  Zur Verschlüsselung:

  ```
  Default_Encryption_Permissions
  Default_owner_password
  Default_user_passowrd
  ```

  Zur Erstellung linearisierter Dokumente:

  ```
  Default_linearization
  ```

  Weitere Informationen zur Einstellung der Optionen finden Sie im Kapitel *AFP2PDF Plus-Umwandlungsdatei* in der *AFP2PDF Plus-Anwenderanleitung*.

Einschränkung:

- Bei der Erstellung von PDF/UA-Dateien wird die AES-256-Bit-Verschlüsselung nicht unterstützt.

Verwendung des PDF2PDFUA-Tools über die Befehlszeile

Wenn Sie das PDF2PDFUA-Tool über die Befehlszeile ausführen, kann es sich bei der Eingabedatei um ein PDF-Dokument oder um eine vom PDF2PDFUA-Tool in eine PDF/UA-konforme Datei konvertierte Steuerdatei handeln.

1

Syntax

Je nach Betriebssystem lautet die Syntax für die PDF2PDFUA-Umwandlung wie folgt:

pdf2pdfua.*nnn* [**-o** *Name der Ausgabedatei*] [**-i** *[Dateiname*] [**-b** [*Anzahl_der_Threads*] *Eingabeverzeichnis Ausgabeverzeichnis*] [**-d** *Ausgabe zur Textausgabe*] [**-linearize** *true/false*] [**-verschlüsseln** *Eigentümer-Kennwort Benutzer-Kennwort* Verschlüsselungsflags] [**-decrypt** *Kennwort*] [**-key** *Kennwort*] [**-trace** *regulärer Ausdruck*] [*Eingabedatei*]

wobei *.nnn* für die Dateiendung **.bat** unter Windows und **.sh** unter Linux und AIX steht.

Parameter

-b

Dieser Parameter dient dazu, die PDF2PDFUA-Umwandlung im Stapel-Modus auszuführen. Die Transformation konvertiert alle PDF-Dateien aus einem Verzeichnis mithilfe einer einzigen Steuerdatei, die über den Parameter **-i** angegeben wird, in PDF/UA-Dateien.

Um die Anzahl der Threads festzulegen, fügen Sie die Zahl direkt hinter dem Flag **-b** ein. Die Standard-Thread-Nummer ist 1.

Hinweis

- Im Stapel-Modus sind die Parameter für das *Eingabeverzeichnis* und das *Ausgabeverzeichnis* obligatorisch. Der Parameter *Eingabeverzeichnis* ersetzt den Parameter *Eingabedatei*.

-linearize

Dieser Parameter ist optional.

Der implizite Wert ist nicht festgelegt.

- Ist der Wert „true“, wird die Ausgabe linearisiert.
- Ist der Wert „false“, wird der Ausgang nicht linearisiert, selbst wenn der Eingang linearisiert ist.
- Wird der Wert nicht angegeben, wird der Ausgang nur dann linearisiert, wenn der Eingang linearisiert ist.

-encrypt

Dieser Parameter ist optional.

Standardmäßig verfügt der Benutzer über keine Berechtigung. Die folgenden Flags werden zur Vergabe von Berechtigungen verwendet:

`-allowAnnotations`

`-allowAssemble`

`-allowCopy`

`-allowFill`

`-allowModify`

`-allowPrintDegraded`

`-allowPrintHigh`

`-allowPrint`, `-allowPrintDegraded`, or `-allowPrintHigh`.

-decrypt *Kennwort*

Dieser Parameter entfernt die ursprüngliche Verschlüsselung.

-key *Kennwort*

Dieser Parameter öffnet gesperrte Dokumente, ohne die ursprüngliche Verschlüsselung zu entfernen. Die Berechtigungen bleiben erhalten.

-trace *regulärer Ausdruck*

Dieser Parameter dient zum Debugging von Steuerdateien. Im Modus der Befehlszeile sind standardmäßig alle Trace-Elemente aktiviert.

-validate

Dieser Parameter aktiviert den Validator. Die Funktion kann ohne Steuerdatei verwendet werden, wenn Sie die Eingabedatei vorab validieren. Wird eine Steuerdatei angegeben, wird auch die PDF-Ausgabe überprüft.

Konfigurieren der Optionsdatei PDF2PDFUA.cfg

Zusätzliche Parameter zur Steuerung der Einstellungen für die PDF2PDFUA-Umwandlung werden in einer Optionsdatei festgelegt. Die Datei heißt PDF2PDFUA.cfg. Die Datei PDF2PDFUA.cfg muss sich im Stammverzeichnis des PDF2PDFUA-Tools befinden.

Die Optionsparameter für die Datei PDF2PDFUA.cfg lauten wie folgt:

AlwaysWriteProducer=TRUE | FALSE

Gibt an, dass die ausgegebene PDF-Datei im Feld „PDF Producer“ Angaben zur Version der PDF2PDFUA-Umwandlung und zur Version der Steuerdatei enthalten muss.

Der Standardwert ist **False**.

Wenn der Wert **False** lautet, hat der Parameter dieselbe Funktion, wie sie in der Steuerdatei festgelegt ist.

Wenn der Wert **True** lautet, enthält der Parameter die Metadaten des Erstellers, unabhängig von der Angabe in der Steuerdatei. Die Metadaten sind für die Entwicklung von Nutzen. Dieser Parameter wird nicht mehr benötigt, da neuere Versionen von PDF2PDFUA die Metadaten auf nicht standardkonforme Weise einbinden. Entwickler können die Metadaten extrahieren, doch ein PDF-Reader ignoriert diese einfach.

DefaultLogSize= integer

Der Standardwert beträgt **10**. Dieser Wert gibt die maximale Größe des Standardprotokolls in Megabyte an. Dieser Parameter hat keine Auswirkung, wenn **useDefaultLog** auf „false“ gesetzt ist.

errProgramVersion = true|false

Legt fest, wie die Transformation mit Steuerdateien umgeht, die für neuere Versionen der PDF2PDFUA-Umwandlung konzipiert sind.

Wenn der Wert **true** ist, wird eine Ausnahme ausgelöst, wodurch verhindert wird, dass die Transformation inkompatible Steuerdateien ausführt.

Wenn der Wert **false** ist, wird versucht, inkompatible Steuerdateien auszuführen, wobei eine Warnung im Ausgabehandler (standardmäßig STDOUT) und in den Protokollen ausgegeben wird. Dies könnte zu fehlerhaften PDF-Ausgabedateien führen.

Der Standardwert ist **true**, und es wird dringend empfohlen, diese Option auf **true** belassen.

LegacyLinePosition=TRUE | FALSE

Der Standardwert ist **False**.

Ist der Wert **False**, wird die Position durch das Wort mit der kleinsten Y-Koordinate angegeben.

Wenn der Wert **True** ist, wird die Position durch das Wort mit der größten Y-Koordinate angegeben. Enthält die Zeile Indizes, gibt der Operator holeZeilenPosition die Y-Koordinate des Indexes zurück.

LegacyNextLine= TRUE | FALSE

Der Standardwert ist **True**. Es beeinflusst das Verhalten von **nächsteZeileAbrufen** und **nächsteZeileImBereichAbrufen**, außer für nächsteZeile oder nächsteZeileImBereich. Diese Operatoren wenden stets das neue Verhalten an.

Im Rahmen des bisherigen Verhaltens wird nach der ersten Zeile gesucht, in der y(Zeile) > y(Cursor) gilt. Es funktioniert nicht ordnungsgemäß, wenn die Cursorposition mit **holeZeilenPosition** festgelegt wird.

Im Rahmen des neuen Verhaltens: Es wird nach der ersten Zeile gesucht, bei der y(Zeile) ≥ y (Cursor) gilt. Es behebt zwar bestimmte Probleme, führt jedoch ein weiteres Problem ein:

```
if y(line) = y(cursor)
```

Zum Beispiel:

```
<Cursor> <Y> <Expr> getLinePosition(getNextLine) </Expr> </Y> </Cursor>
<Expr> $line1 = getNextLine </Expr>
<Expr> $line2 = getCurrentLine </Expr>
```

Hier sind Zeile 1 und Zeile 2 identisch.

UseDefaultLog= TRUE | FALSE

Der Standardwert ist **True**.

Wenn der Wert **True** lautet, wird ein Standardprotokoll erstellt, falls keine über die API übergeben wurde. Es ist im Standalone-Modus nützlich.

Wenn der Wert **False** lautet, wird die Protokollierung nur aktiviert, wenn dies über die API konfiguriert wurde.

Von der PDF2PDFUA-Umwandlung unterstützte Funktionen

Mit der PDF2PDFUA-Umwandlung können Sie mehrere Eingabedateien, die eine gemeinsame Schablone haben, im Stapelverfahren konvertieren. Sie können für jede PDF-Datei eine Steuerdatei erstellen, darunter Rechnungen, Kontoauszüge oder Kreditkartenabrechnungen. Sie haben die Möglichkeit, für jede Rechnungsart eine eigene Steuerdatei anzulegen.

PDF2PDFUA Plus bietet erweiterte Funktionen zur Verarbeitung variabler Daten. PDF2PDFUA Plus kann variable Daten an festen Positionen verarbeiten, wie beispielsweise Namen und Adressfelder, aber auch Tabellen mit einer variablen Anzahl von Zeilen oder Tabellen, die sich über mehrere Seiten erstrecken. Mit der PDF2PDFUA-Umwandlung können Sie ausschließlich den eigentlichen Inhalt kennzeichnen und den irrelevanten Inhalt als Artefakt darstellen.

Hinweis

Wenden Sie sich bitte an Ihren Ricoh-Ansprechpartner, um weitere Informationen zum Vorgehen bei der Erstellung einer Steuerdatei zu erhalten.

2. Gewährleistung der Konformität mit PDF/UA und WCAG

- **Anforderungen an die Eingabedatei für die Einhaltung der PDF/UA-Standards**
- **Überprüfung der PDF-Eingabedatei und der PDF-UA-Ausgabedatei**
- **Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.0 (Stufe AA)**
- **Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.0 (Stufe AAA)**
- **Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.1 (Stufe AA)**
- **Überprüfung der Konformität mit PDF/UA oder WCAG**

Anforderungen an die Eingabedatei für die Einhaltung der PDF/UA-Standards

Sollte die Eingabedatei eines dieser Probleme aufweisen, kann die Datei nicht verarbeitet werden oder die Ausgabedatei entspricht nicht den PDF/UA-Standards.

- Die PDF-Datei enthält flackernde Inhalte oder PDF-Elemente, die zu Flackern führen (Matterhorn-Protokoll – Ch03-X).
- Informationen in der PDF-Datei werden durch Kontrast, Farbe oder Formatierung vermittelt (MP Ch04).
- Die PDF-Datei enthält Ton (MP Ch05).
- Die PDF-Datei verwendet Zeichencodes ohne Unicode-Zuordnung (MP Ch10).
- Die PDF-Datei enthält Bilder mit Absätzen (MP C13-06).
- Die PDF-Datei verwendet nicht eingebettete Schriftarten (MP C21).

  Hinweis

  Wenn Sie diese Funktion zusammen mit der AFP2PDF Plus-Umwandlung verwenden, stellen Sie bitte sicher, dass diese so konfiguriert ist, dass die Schriftarten eingebettet werden.
- Die PDF-Datei enthält Formulare.
- Die PDF-Datei enthält Aktionselemente.

Überprüfung der PDF-Eingabedatei und der PDF-UA-Ausgabedatei

Der PDF2PDFUA-Validator generiert Warnmeldungen, die eine Referenz auf bestimmte Indizes aus dem Matterhorn-Protokoll 1.1 der PDF-Spezifikationen enthalten können. Der Pre-Validator überprüft einige der im Matterhorn-Protokoll genannten Bedingungen.

Um weitere Informationen zu den angezeigten Fehlern zu erhalten, suchen Sie im Matterhorn-Protokoll nach der Fehler-ID.

Wenden Sie sich an Ihren PDF-Ersteller, um diesen PDF-Fehler zu untersuchen und zu beheben, damit Sie eine PDF/UA-konforme Ausgabedatei erhalten.

Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.0 (Stufe AA)

Um sicherzustellen, dass die Eingabedatei den Barrierefreiheitsstandards der „Web Content Accessibility Guidelines" (WCAG) 2.0, Stufe AA, entspricht, überprüfen Sie bitte Folgendes:

WCAG 1.4.3: Mindestkontrast

Die Eingabedatei sollte Text und Bilder mit einem Mindestkontrast von 4,5:1 enthalten.

WCAG 1.4.5: Bilder als Text

Die Eingabedatei sollte Informationen in Form von Text liefern und nicht in Form von Bildern, die Text enthalten.

Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.0 (Stufe AAA)

Um sicherzustellen, dass die Eingabedatei den Barrierefreiheitsstandards der „Web Content Accessibility Guidelines" (WCAG) 2.0, Stufe AA, entspricht, überprüfen Sie bitte Folgendes:

WCAG 1.4.6 – Anforderung an den erhöhten Kontrast

Die visuelle Darstellung von Text und Textbildern weist ein Kontrastverhältnis von mindestens 7:1 auf.

WCAG 1.4.8 – Visuelle Darstellung

Die Eingabedatei sollte folgende Anforderungen erfüllen:

1. Die Breite beträgt höchstens 80 Zeichen oder Glyphen (40 bei CJK).
2. Der Text ist nicht zentriert (sondern sowohl am linken als auch am rechten Rand ausgerichtet).
3. Der Zeilenabstand (Leading) beträgt innerhalb von Absätzen mindestens eineinhalb Zeilen, und der Absatzabstand ist mindestens 1,5-mal so groß wie der Zeilenabstand.

WCAG 1.4.9 Bilder mit Text (keine Ausnahme)

Textbilder werden ausschließlich zu rein dekorativen Zwecken verwendet oder in Fällen, in denen eine bestimmte Darstellung des Textes für die Vermittlung der Informationen unerlässlich ist.

Anforderungen an Eingabedateien zur Einhaltung der WCAG 2.1 (Stufe AA)

1.4.11 Kontrast von Nicht-Text-Elementen (Stufe AA)

Die visuelle Darstellung grafischer Objekte weist ein Kontrastverhältnis von mindestens 3:1 gegenüber benachbarten Farben auf.

Überprüfung der Konformität mit PDF/UA oder WCAG

Mit diesen Tools können Sie sicherstellen, dass Ihre PDF-Datei den PDF/UA- oder WCAG-Richtlinien entspricht:

PAC – PDF-Barrierefreiheitsprüfung

Ein kostenlos zum Herunterladen verfügbares Tool, mit dem überprüft werden kann, ob die PDF-Datei den PDF/UA- oder WCAG-Richtlinien entspricht. Sollte die Überprüfung fehlschlagen, können Sie alle Probleme einsehen, die dazu geführt haben, dass die Datei nicht den Anforderungen entspricht. Sie können das Dokument online bei der PDF/UA Foundation herunterladen: https://pac.pdf-accessibility.org/en.

3. Java-Anwendungsprogrammierschnittstelle

- **Klassen**

In diesem Kapitel wird die von RICOH PDF2PDFUA bereitgestellte Java-API beschrieben.

Der von spitzen Klammern umgebene Code enthält eine Referenz auf ein XML-Element in der Steuerdatei. Beispielsweise verweist `<If>` auf bedingte Anweisungen in der Steuerdatei.

Klassen

Diese Klassen sind Teil der Java-Anwendungsprogrammierschnittstelle (API) für RICOH PDF2PDFUA.

enum EncryptionAlgorithm

Paket: **com.ricoh.pdf2pdfua.api**

Option, die die Art der Verschlüsselung der ausgegebenen PDF-Datei festlegt.

Werte:

none

Die Ausgabe ist nicht verschlüsselt.

preserve

Behält die Verschlüsselung der Eingabe-PDF-Datei bei.

Standard128

Eine Stromchiffre nach dem RC4-Verfahren mit einer Schlüsselgröße von 128 Bit.

AES128

Ein fortschrittlicher Standard für die Verschlüsselung, der eine Schlüssellänge von 128 Bit verwendet.

AES256

Ein fortschrittlicher Standard für die Verschlüsselung, der eine Schlüssellänge von 256 Bit verwendet.

interface IControlFile

Paket: **com.ricoh.pdf2pdfua.api**.

Stellt den Syntaxbaum der Steuerdatei dar. Die Steuerdatei wird einmalig analysiert und auf mehrere PDF-Dateien angewendet.

```
IControlFile refresh() throws PreprocessingException;
```

Wird verwendet, um die Steuerdatei auf Änderungen zu überprüfen.

Gibt `this` zurück, wenn die Steuerdatei auf dem neuesten Stand ist.

Gibt ein neues `IControlFile` zurück, wenn die zugrunde liegenden Dateien nicht mehr synchron sind.

interface Pdfua

Paket: **com.ricoh.pdf2pdfua.api**

Die Java-Service-Provider-Schnittstelle (SPI) für RICOH PDF2PDFUA.

```
IControlFile newControlFile(String path);
```

Analysiert die unter „path" angegebene Steuerdatei. Eine einzelne Instanz einer Steuerdatei kann in mehreren Threads wiederverwendet werden.

```
PdfUaOpt newOpt();
```

Erstellt eine Instanz von „PdfUaOpt" mit den Standardeinstellungen.

```
void run(FileInputStream fis, File out, IControlFile cf, PdfUaOpt opt) throws PDFUAExcepti
```

Mit dieser Methode wird die PDF2PDFUA-Konvertierung ausgeführt. Es ist threadsicher und reentrant.


```
byte[] run(InputStream inPdf, IControlFile cf, PdfUaOpt opt) throws PDFUAException;
```

Diese Methode führt die PDF2PDFUA-Konvertierung durch und gibt ein Byte-Array zurück, das die mit Tags versehene PDF-Datei enthält. Es ist threadsicher und reentrant.

Klasse PDFUAException erweitert Exception

Paket: **com.ricoh.pdf2pdfua.api**

Basisklasse für Konvertierungsfehler bei PDF2PDFUA.

class PdfUaLoader

Paket: **com.ricoh.pdf2pdfua.api**

Ein einfacher Hilfsdienst, der die Implementierung des PDF2PDFUA-Dienstes lädt. Der Dienst kann direkt mit einem ServiceLoader geladen werden.

```
public static Pdfua load();
```

Lädt die Implementierung des PDF2PDFUA-Dienstes.

abstract class PdfUaOpt

Paket: **com.ricoh.pdf2pdfua.api**

Dient zur Konfiguration des Verhaltens von PDF2PDFUA. Es handelt sich um einen Konfigurationscontainer, der ausschließlich Getter und Setter enthält.

Dies ist eine abstrakte Klasse, die jedoch von API-Nutzern keinesfalls erweitert werden darf. Um eine Instanz zu erhalten, verwenden Sie `Pdfua.newOpt()`.

Eine einzelne Instanz kann von mehreren Threads verwendet werden, sollte jedoch nicht geändert werden, solange PDF2PDFUA ausgeführt wird.

```
public boolean getAllowAnnotations();
public void setAllowAnnotations(boolean allowAnnotation);
```

Berechtigungsflag für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: false

```
public boolean getAllowAssemble();
public void setAllowAssemble(boolean allowAssemble);
```

Berechtigungsflag für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: false

```
public boolean getAllowCopy();
public void setAllowCopy(boolean allowCopy);
```

Berechtigungsflag für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: false

```
public boolean getAllowFill();
public void setAllowFill(boolean allowFill);
```


Berechtigungsflag für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: false

```
public boolean getAllowModify();
public void setAllowModify(boolean allowModify);
```

Berechtigungsflag für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: false

```
public boolean getAllowPrint();
public void setAllowPrint(boolean allowPrint);
```

Berechtigungsflag für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: false

```
public boolean getAllowPrintDegraded();
public void setAllowPrintDegraded(boolean allowPrintDegraded);
```

Berechtigungsflag für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: false

```
public byte[] getDecryptionKey();
public void setDecryptionKey(byte[] decryptionKey);
```

Diese Option ist erforderlich, wenn die PDF-Datei verschlüsselt ist. Dient dazu, die Eingabe freizugeben, ohne die Verschlüsselung zu entfernen. Der Eigentümer-Schlüssel ist vorzuziehen.

Standardwert: null

```
public EncryptionAlgorithm getEncryption();
public void setEncryption(EncryptionAlgorithm encryption);
```

Legt den Algorithmus für die Verschlüsselung der Ausgabe fest.

Standardwert: EncryptionAlgorithm.preserve

```
public Boolean getLinearize();
public void setLinearize(Boolean linearize);
```

Falls „linearize“ auf „true“ gesetzt ist: Die Ausgabedichtefunktion wird linearisiert.

Wenn „linearize“ auf „false“ gesetzt ist: Die Ausgabedichtefunktion wird nicht linearisiert.

Falls „linearize“ auf „null“ gesetzt ist: Die Ausgabe wird nur dann linearisiert, wenn die Eingabe-PDF linearisiert wurde.

Standardwert: null

```
public Logger getLog();
public void setLog(Logger log);
```

Standardwert: in der Konfigurationsdatei festgelegt.


```
public boolean getOnlyMergeXObjects();
public void setOnlyMergeXObjects(boolean onlyMergeXObjects);
```

Diese Option dient der Entwicklung und dem Testen. Es darf nicht in einer Produktionsumgebung verwendet werden.

Standardwert: false

```
public byte[] getOwnerPassword();
public void setOwnerPassword(byte[] ownerPassword);
```

Das Kennwort für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: null

```
public PdfUaOutputHandler getStderrHandler();
public void setStderrHandler(PdfUaOutputHandler stderrHandler);
```

Dient zur Verarbeitung der Meldungen, die normalerweise an stderr ausgegeben werden.
Standardwert: der in der Konfigurationsdatei angegebene integrierte Handler.

```
public PdfUaOutputHandler getStdoutHandler();
public void setStdoutHandler(PdfUaOutputHandler stdoutHandler);
```

Dient zur Verarbeitung der Meldungen, die normalerweise an stdout ausgegeben werden.

Standardwert: der in der Konfigurationsdatei angegebene integrierte Handler.

```
public PdfUaOutputHandler getTraceHandler();
public void setTraceHandler(PdfUaOutputHandler traceHandler);
```

Wird ausschließlich vom Befehl `<Trace>` verwendet.

Standardwert: der in der Konfigurationsdatei angegebene integrierte Handler.

```
public boolean getUseLatestRevisions();
public void setUseLatestRevisions(boolean useLatestRevisions);
```

Diese Option dient der Entwicklung und dem Testen. Es darf nicht in einer Produktionsumgebung verwendet werden.

Standardwert: `false`

```
public byte[] getUserPassword();
public void setUserPassword(byte[] userPassword);
```

Das Benutzerkennwort für die PDF-Verschlüsselung. Hat keine Auswirkung, wenn die Ausgabedatei nicht mit PDF2PDFUA verschlüsselt wurde.

Standardwert: null

interface PdfUaOutputHandler

Paket: **com.ricoh.pdf2pdfua.api**

Handler für die E/A-Umleitung.

```
void onMessage(Node src, int pageNo, String msg);
```

`src`: Der XML-Knoten der aufrufenden Steuerdatei-Regel. Null, wenn die Nachricht nicht einem XML-Knoten zugeordnet werden kann.

`pageNo`: Die Seitenzahl beginnt bei 1. Dieser Wert ist Null, wenn die Meldung nicht einer Seite zugeordnet werden kann.

`msg`: kann mehrere Textzeilen enthalten.

class PreprocessingException extends Exception

Paket: **com.ricoh.pdf2pdfua.api**

Basisklasse für Fehler, die bei der Vorverarbeitung der Steuerdatei auftreten.

class UaStackTrace extends ArrayList<String>

Paket: **com.ricoh.pdf2pdfua.api**

Ein Stack-Trace, der auf ein XML-Element in der Steuerdatei verweist.

```
public static UaStackTrace unwind(final Node top);
```

Beispiele

Dieses Beispiel zeigt, wie man eine Steuerdatei auswertet und wie man sie auf mehrere Dokumente anwendet. Alle Dokumente in einem Stapel sollen ähnlich sein und mit der Steuerdatei kompatibel sein. Aus Leistungsgründen wird die Steuerdatei nur einmal analysiert.

Es wird davon ausgegangen, dass die Software auf Servern mit mehreren CPU-Kernen läuft. Dieses Beispiel ist multithreaded, da es sich um den Standardanwendungsfall handelt.

```
package com.ricoh.pdf2pdfua.api.example;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import com.ricoh.pdf2pdfua.api.IControlFile;
import com.ricoh.pdf2pdfua.api.PDFUAException;
import com.ricoh.pdf2pdfua.api.PdfUaLoader;
import com.ricoh.pdf2pdfua.api.PdfUaOpt;
import com.ricoh.pdf2pdfua.api.Pdfua;
import com.ricoh.pdf2pdfua.api.preprocessor.PreprocessingException;
```

```
public class Pdf2PdfUaApiExample {
    static class ThreadSafeIterator {
        Iterator<File> i;

        ThreadSafeIterator(List<File> files) {
            i = files.iterator();
        }

        synchronized File next() {
            if (i.hasNext())
                return i.next();
            return null;
        }
    }

    static void exit(String msg) {
        System.err.println(msg);
        System.exit(1);
    }

    static File changeExtension(File f) {
        return new File(f.getAbsolutePath().replace(".pdf", ".ua.pdf"));
    }

    static Thread[] createThreadArray() {
        return new Thread[Runtime.getRuntime().availableProcessors()];
    }

    static void convertBatch(String controlFilePath, List<File> pdfs)
            throws PreprocessingException, InterruptedException {
        final Pdfua pdfuaInstance = PdfUaLoader.load();
        final IControlFile cf = pdfuaInstance.newControlFile(controlFilePath);
        final PdfUaOpt opt = pdfuaInstance.newOpt();
        final ThreadSafeIterator it = new ThreadSafeIterator(pdfs);
        Thread[] threads = createThreadArray();
        for (int i = 0; i < threads.length; i++) {
            threads[i] = new Thread() {
                public void run() {
                    File pdfIn;
                    while ((pdfIn = it.next()) != null) {
                        try {
                            FileInputStream fis = new FileInputStream(pdfIn);
                            File out = changeExtension(pdfIn);
                            pdfuaInstance.run(fis, out, cf, opt);
                        } catch (FileNotFoundException | PDFUAException e) {
                            e.printStackTrace();
                        }
                    }
                }
            };
        }
        for (Thread t : threads)
            t.start();
        for (Thread t : threads)
            t.join();
        System.out.println("Batch complete!");
    }

    public static void main(String[] args)
            throws PreprocessingException, InterruptedException {
        if (args.length < 1)
            exit("No control file!");
```

```
        ArrayList<File> files = new ArrayList<>();
        for (int i = 1; i < args.length; i++)
            files.add(new File(args[i]));
        if (files.isEmpty())
            exit("No PDF file to convert!");
        convertBatch(args[0], files);
    }
}
```

Dieses Beispiel veranschaulicht, wie PDF2PDFUA ohne die Verwendung von Dateien funktionieren kann. Dies ist nützlich, wenn PDF2PDFUA in andere Systeme wie beispielsweise Webanwendungen integriert ist.

```
package com.ricoh.pdf2pdfua.api.example;

import java.io.FileInputStream;
import java.io.FileOutputStream;

import com.ricoh.pdf2pdfua.api.PdfUaLoader;
import com.ricoh.pdf2pdfua.api.Pdfua;

public class Pdf2PdfUaApiExampleBytesOutput {
    public static void main(String[] args) throws Exception {
        Pdfua ua = PdfUaLoader.load();
        byte[] outputBytes = ua.run(
            new FileInputStream("./untagged.pdf"),
            ua.newControlFile("./control-file.xml"),
            ua.newOpt()
        );
        var os = new FileOutputStream("./out.pdf");
        os.write(outputBytes);
        os.close();
    }
}
```

RICOH Transforms Suite PDF2PDFUA-Tool Anwenderanleitung