RICOH

RICOH Transforms Suite

PDF2PDFUA-tool Gebruikershandleiding

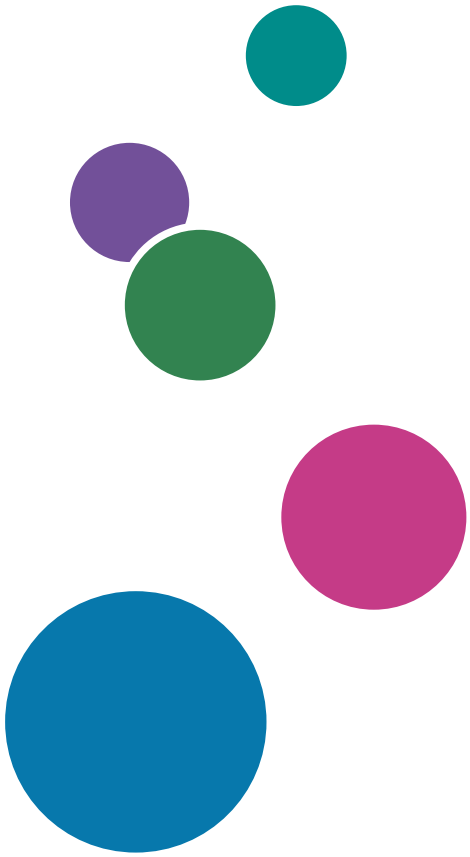
Version 1.5

Gebruik van de PDF2PDFUA-tool 1

Zorgen voor naleving van PDF/UA en WCAG 2

Java-programmeerinterface 3

Bekijk het Help-systeem van uw product voor informatie die niet in deze handleiding staat.



INHOUDSOPGAVE

Introductie

Belangrijk ... 2
Waarschuwingen met betrekking tot deze handleiding ... 2
Publicaties over dit product ... 2
Symbolen ... 2
Afkortingen ... 3
Handelsmerken ... 3

1 Gebruik van de PDF2PDFUA-tool

Gebruik van de PDF2PDFUA-tool via de AFP2PDF Plus-transformatie ... 5
De PDF2PDFUA-tool gebruiken via de opdrachtregel ... 5
Syntaxis ... 6
Parameters ... 6
Het configuratiebestand `PDF2PDFUA.cfg` instellen ... 7
Functies die worden ondersteund door de PDF2PDFUA-transformatie ... 8

2 Zorgen voor naleving van PDF/UA en WCAG

Vereisten voor invoerbestanden met het oog op naleving van de PDF/UA-standaard ... 11
Het controleren van het PDF-invoerbestand en het PDF-UA-uitvoerbestand ... 11
Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.0 (niveau AA) ... 11
Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.0 (niveau AAA) ... 12
Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.1 (niveau AA) ... 12
Controle op naleving van PDF/UA of WCAG ... 12

3 Java-programmeerinterface

Klassen ... 13
Voorbeelden ... 17

Introductie

Belangrijk

Voor zover toegestaan in de betreffende wetten, is de fabrikant in geen enkel geval aansprakelijk voor enige schade die voortvloeit uit storingen van dit product, verlies van documenten of gegevens, of het gebruik of het niet gebruiken van dit product en de gebruikershandleidingen die zijn meegeleverd.

Zorg ervoor dat u altijd een kopie bewaart of back-ups maakt van belangrijke documenten of gegevens. Documenten of gegevens kunnen verloren gaan als gevolg van fouten van uw kant of storingen in de software. Daarnaast bent u zelf verantwoordelijk voor het nemen van beschermende voorzorgsmaatregelen tegen computervirussen, wormen en andere schadelijke software.

De fabrikant is onder geen omstandigheden verantwoordelijk voor documenten die door u zijn gemaakt met dit product of voor resultaten die voortkomen uit door u uitgevoerde gegevens.

Waarschuwingen met betrekking tot deze handleiding

- Sommige afbeeldingen of uitleg in deze handleiding kunnen afwijken van uw product als gevolg van verbeteringen of wijzigingen aan het product.
- De inhoud van dit document kan worden gewijzigd zonder voorafgaande kennisgeving.
- Geen enkel deel van dit document mag zonder de voorafgaande toestemming van de leverancier in enige vorm worden gedupliceerd, gekopieerd, gereproduceerd, aangepast of geciteerd.

Publicaties over dit product

Deze publicatie biedt een overzicht van de tool en informatie over de installatie en het gebruik van de PDF2PDFUA-tool.

Instructiehandleidingen

Raadpleeg de volgende documenten voor meer informatie over de PDF2PDFUA-tool:

- *PDF2PDFUA Installatiehandleiding*
- *PDF2PDFUA Gebruikershandleiding*

Symbolen

In deze handleiding worden ter verduidelijking van de inhoud de volgende symbolen gebruikt.

Belangrijk

Met dit symbool wordt aangegeven dat u bij gebruik van de toepassing ergens op moet letten. Zorg ervoor dat u deze uitleg leest.

Opmerking

Dit symbool duidt op aanvullende informatie die u mogelijk nuttig vindt, maar die niet essentieel is voor de uitvoering van een handeling.

[Vet]

[Vetgedrukte tekst] binnen vierkante haken geeft de namen van menu's, menu-items, instellingen, velden, knoppen en toetsen aan.

`Cursief`

`Cursieve tekst` wordt gebruikt voor variabelen die u moet vervangen door uw eigen gegevens.

`Monospace`

`Monospace` geeft computerinvoer en -uitvoer aan.

...

Een ellipsteken geeft aan dat een reeks kan doorgaan.

Afkortingen

AES

Advanced Encryption Standard

AFP

Advanced Function Presentation

PDF

Portable Document Format

UA

Universele toegankelijkheid

WCAG

Richtlijnen voor Toegankelijkheid van Webcontent

Handelsmerken

Adobe, het Adobe-logo, PostScript en het PostScript-logo, PDF en het PDF-logo zijn geregistreerde handelsmerken of handelsmerken van Adobe in de Verenigde Staten en/of andere landen.

Linux is een geregistreerd handelsmerk van Linus Torvalds in de Verenigde Staten en/of andere landen.

Microsoft, Windows, Windows NT en het Windows-logo zijn handelsmerken van Microsoft Corporation in de Verenigde Staten en/of andere landen.

Sentinel is een geregistreerd handelsmerk van Thales DIS CPL USA, Inc.

UNIX is een geregistreerd handelsmerk van The Open Group in de Verenigde Staten en andere landen.

Andere bedrijfs-, product- of dienstnamen kunnen handelsmerken of dienstmerken van anderen zijn.

1. Gebruik van de PDF2PDFUA-tool

- **Gebruik van de PDF2PDFUA-tool via de AFP2PDF Plus-transformatie**
- **De PDF2PDFUA-tool gebruiken via de opdrachtregel**
- **Syntaxis**
- **Parameters**
- **Het configuratiebestand** `PDF2PDFUA.cfg` **instellen**
- **Functies die worden ondersteund door de PDF2PDFUA-transformatie**

Gebruik van de PDF2PDFUA-tool via de AFP2PDF Plus-transformatie

Om AFP-bestanden om te zetten naar PDF/UA-documenten, dient u deze optieparameter toe te voegen aan het configuratiebestand van AFP2PDF Plus:

PDFUA_CONTROL_FILE=*`bestand`*

Hiermee wordt de volledig gekwalificeerde naam van het controlebestand van de PDF2PDFUA-tool opgegeven. Dit bestand is een XML-bestand dat informatie bevat over de inhoud van het invoerbestand. De PDF2PDFUA-tool gebruikt het document om de inhoud te markeren zoals de PDF/UA-standaard dat voorschrijft.

PDFUA_CACHE_SIZE=*`nummer`*

Hiermee wordt het aantal meest gebruikte PDFUA-controlebestanden gespecificeerd dat in de cache wordt opgeslagen. Het getal moet groter zijn dan 0.

Opmerking

- Voor een bestand met versleuteling of een gelineariseerd PDF/UA-uitvoerbestand dient u de instellingen in het AFP2PDF Plus-optiebestand aan te passen wanneer u de AFP2PDF Plus-transformatie uitvoert:

  Voor versleuteling:

  `Default_Encryption_Permissions`

  `Default_owner_password`

  `Default_user_passowrd`

  Voor het opstellen van lineaire documenten:

  `Default_linearization`

  Voor meer informatie over het instellen van de opties verwijzen wij u naar het hoofdstuk *`AFP2PDF Plus-transformatieoptiebestand`* in de *`AFP2PDF Plus Gebruikershandleiding`*.

Beperking:

- Bij het aanmaken van PDF/UA-bestanden wordt AES-versleuteling met 256-bits niet ondersteund.

De PDF2PDFUA-tool gebruiken via de opdrachtregel

Wanneer u de PDF2PDFUA-tool vanaf de opdrachtregel uitvoert, kan het invoerbestand bestaan uit een PDF-document en een controlebestand dat door de PDF2PDFUA-tool is geconverteerd naar een PDF/UA-conform bestand.

1

Syntaxis

Afhankelijk van het besturingssysteem is de syntaxis voor de PDF2PDFUA-transformatie als volgt:

pdf2pdfua.*nnn* [**-o** *naam van het uitvoerbestand*] [**-i** *naam van het controlebestand*] [**-b** [*aantal_threads*] *invoermap uitvoermap*] [**-d** *uitvoer voor het dumpen van tekst*] [**-linearize** *true/false*] [**-encrypt** *wachtwoord van de eigenaar, gebruikerswachtwoord* versleutelingvlaggen] [**-decrypt** *wachtwoord*] [**-key** *wachtwoord*] [**-trace** *reguliere expressie*] [*invoerbestand*]

waarbij *.nnn* staat voor de bestandsextensie **.bat** voor Windows en **.sh** voor Linux en AIX.

Parameters

-b

Deze parameter wordt gebruikt om de PDF2PDFUA-transformatie in batchmodus uit te voeren. De transformatie zet alle PDF-bestanden uit een map om naar PDF/UA-bestanden aan de hand van één enkel controlebestand dat via de parameter **-i** wordt opgegeven.

Om het aantal threads op te geven, voegt u het getal direct na de optie **-b** toe. Het standaard aantal threads is 1.

Opmerking

- In de batchmodus zijn de parameters voor de *invoermap* en de *uitvoermap* verplicht. De parameter *invoermap* vervangt de parameter *invoerbestand*.

-linearize

Deze parameter is optioneel.

De impliciete waarde is niet gespecificeerd.

- Indien de waarde 'true' is, wordt de uitvoer gelineariseerd.
- Indien de waarde 'false' is, wordt de uitgang niet gelineariseerd, zelfs niet als de ingang wel gelineariseerd is.
- Indien de waarde niet is opgegeven, wordt de uitgang alleen gelineariseerd als de ingang is gelineariseerd.

-encrypt

Deze parameter is optioneel.

Standaard beschikt de gebruiker niet over de benodigde rechten. De volgende vlaggen worden gebruikt om machtigingen te verlenen:

`-allowAnnotations`

`-allowAssemble`

`-allowCopy`

`-allowFill`

`-allowModify`

`-allowPrintDegraded`

`-allowPrintHigh`

`-allowPrint`, `-allowPrintDegraded` of `-allowPrintHigh`.

-decrypt *wachtwoord*

Deze parameter verwijdert de oorspronkelijke versleuteling.

-key *wachtwoord*

Met deze parameter worden vergrendelde documenten geopend zonder dat de oorspronkelijke versleuteling wordt verwijderd. De machtigingen blijven behouden.

-trace *reguliere expressie*

Deze parameter wordt gebruikt voor foutopsporing van controlebestanden. Alle traceerelementen zijn standaard ingeschakeld in de opdrachtregelmodus.

-validate

Met deze parameter wordt de validator ingeschakeld. Het kan zonder controlebestand worden gebruikt wanneer u het invoerbestand vooraf valideert. Indien er een controlebestand is opgegeven, wordt ook de uitvoer-PDF gecontroleerd.

Het configuratiebestand PDF2PDFUA.cfg instellen

Aanvullende parameters voor het instellen van de PDF2PDFUA-transformatie worden opgegeven in een optiebestand. De naam is PDF2PDFUA.cfg. Het bestand PDF2PDFUA.cfg moet zich in de hoofdmap van de PDF2PDFUA-tool bevinden.

De optieparameters van PDF2PDFUA.cfg zijn:

AlwaysWriteProducer=TRUE | FALSE

Geeft aan dat het resulterende PDF-bestand in het veld PDF-producent informatie moet bevatten over de versie van de PDF2PDFUA-transformatie en de versie van het controlebestand.

De standaardwaarde is **False**.

Wanneer de waarde **False** is, heeft de parameter dezelfde functie als is opgegeven in het controlebestand.

Wanneer de waarde **True** is, bevat de parameter de metadata van de producent, ongeacht de specificatie in het controlebestand. De metagegevens zijn nuttig voor de ontwikkeling. Deze parameter is niet langer nodig, aangezien nieuwere versies van PDF2PDFUA de metagegevens op afwijkende manieren opnemen. Ontwikkelaars kunnen de metagegevens extraheren, maar een PDF-reader zal deze gewoon negeren.

DefaultLogSize= integer

De standaardwaarde is **10**. Dit geeft de maximale grootte van het standaardlogboek aan, in megabytes. Deze parameter heeft geen effect als **useDefaultLog** op false is ingesteld.

errProgramVersion = true|false

Hiermee wordt bepaald hoe de transformatie omgaat met controlebestanden die zijn ontworpen voor nieuwere versies van PDF2PDFUA.

Wanneer de waarde **true** is, wordt er een uitzondering gegenereerd, waardoor wordt voorkomen dat de transformatie incompatibele controlebestanden uitvoert.

Wanneer de waarde **false** is, wordt er geprobeerd incompatibele controlebestanden uit te voeren, waarbij een waarschuwing wordt gegenereerd in de uitvoerhandler (standaard STDOUT) en in de logbestanden. Dit kan leiden tot onjuiste PDF-uitvoer.

De standaardwaarde is **true** en het wordt ten zeerste aanbevolen om deze optie op **true** te laten staan.

LegacyLinePosition=TRUE | FALSE

De standaardwaarde is **False**.

Wanneer de waarde **False** is, wordt de positie bepaald door het woord met de kleinste Y-coördinaat.

Wanneer de waarde **True** is, wordt de positie bepaald door het woord met de grootste Y-coördinaat. Indien de regel subscripten bevat, geeft de operator haalRegelPositieOp de Y-coördinaat van het subscript terug.

LegacyNextLine= TRUE | FALSE

De standaardwaarde is **True**. Dit heeft invloed op de werking van **volgendeRegelOphalen** en **deVolgendeRegelInDeRegioOphalen**, met uitzondering van volgendeRegel of volgendeRegelInRegio. Deze operators maken altijd gebruik van het nieuwe gedrag.

Binnen het bestaande gedrag wordt gezocht naar de eerste regel waarbij y(regel) > y(cursor) geldt. Het werkt niet correct als de cursor wordt ingesteld met **haalRegelPositieOp**.

In het nieuwe gedrag: er wordt gezocht naar de eerste regel waarbij y(regel) ≥ y(cursor) geldt. Het lost bepaalde problemen op, maar veroorzaakt tegelijkertijd een ander probleem:

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

Bijvoorbeeld:

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

In dit geval zijn regel 1 en regel 2 identiek.

UseDefaultLog= TRUE | FALSE

De standaardwaarde is **True**.

Wanneer de waarde **True** is, wordt er een standaardlogboek aangemaakt indien er geen logbestand via de API is doorgegeven. Het is handig in de stand-alone-modus.

Wanneer de waarde **False** is, wordt de logboekregistratie alleen ingeschakeld indien dit via de API is geconfigureerd.

Functies die worden ondersteund door de PDF2PDFUA-transformatie

Met de PDF2PDFUA-transformatie kunt u meerdere invoerbestanden die een gemeenschappelijke generatiesjabloon hebben, in één keer converteren. U kunt voor elk PDF-bestand een controlebestand aanmaken, bijvoorbeeld voor facturen, rekeningoverzichten of creditcardafschriften. U hebt de mogelijkheid om voor elk factuurtype één controlebestand aan te maken.

PDF2PDFUA Plus biedt verbeterde ondersteuning voor het verwerken van variabele gegevens. PDF2PDFUA Plus kan variabele gegevens op vaste posities verwerken, zoals namen en adresvelden, maar ook tabellen met een variabel aantal rijen of tabellen die zich over meerdere pagina's uitstrekken. Met de PDF2PDFUA-transformatie kunt u uitsluitend de daadwerkelijke inhoud markeren en het irrelevante artefact verbergen.

Opmerking

Neem contact op met uw Ricoh-vertegenwoordiger voor meer informatie over de procedure voor het aanmaken van een controlebestand.

2. Zorgen voor naleving van PDF/UA en WCAG

- **Vereisten voor invoerbestanden met het oog op naleving van de PDF/UA-standaard**
- **Het controleren van het PDF-invoerbestand en het PDF-UA-uitvoerbestand**
- **Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.0 (niveau AA)**
- **Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.0 (niveau AAA)**
- **Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.1 (niveau AA)**
- **Controle op naleving van PDF/UA of WCAG**

Vereisten voor invoerbestanden met het oog op naleving van de PDF/UA-standaard

Indien het invoerbestand een van deze problemen bevat, kan het bestand niet worden verwerkt of voldoet het resulterende bestand niet aan de PDF/UA-norm.

- Het PDF-bestand bevat flikkerende inhoud of PDF-elementen die tot flikkering leiden (Matterhorn Protocol – Ch03-X).
- De informatie in het PDF-bestand wordt overgebracht door middel van contrast, kleur of opmaak (MP Ch04).
- Het PDF-bestand bevat geluid (MP Ch05).
- Het PDF-bestand maakt gebruik van tekencodes zonder Unicode-toewijzing (MP Ch10).
- Het PDF-bestand bevat afbeeldingen met alinea's met tekst (MP C13-06).
- Het PDF-bestand maakt gebruik van niet-ingesloten lettertypen (MP C21).

  Opmerking

  Indien u dit in combinatie met de AFP2PDF Plus-transformatie gebruikt, dient u ervoor te zorgen dat deze is geconfigureerd om de lettertypen in te sluiten.
- Het PDF-bestand bevat formulieren.
- Het PDF-bestand bevat actie-elementen.

Het controleren van het PDF-invoerbestand en het PDF-UA-uitvoerbestand

De PDF2PDFUA-validator genereert waarschuwingen die kunnen verwijzen naar specifieke indexen uit het Matterhorn Protocol 1.1 van de PDF-verenigingen. De pre-validator controleert een aantal van de voorwaarden die in het Matterhorn-protocol worden genoemd.

Voor meer informatie over de weergegeven fouten kunt u zoeken op de fout-ID in het Matterhorn-protocoldocument.

Neem contact op met uw PDF-producent om deze PDF-fout te onderzoeken en op te lossen, zodat u een uitbestand krijgt dat compatibel is met PDF/UA.

Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.0 (niveau AA)

Om er zeker van te zijn dat het invoerbestand voldoet aan de toegankelijkheidsnormen van de Richtlijnen voor Toegankelijkheid van Webcontent (WCAG) 2.0, niveau AA, dient u het volgende te controleren:

WCAG 1.4.3: Minimale contrastverhouding

Het invoerbestand dient tekst en afbeeldingen te bevatten met een contrast van ten minste 4,5:1.

WCAG 1.4.5: Afbeeldingen als tekst

Het invoerbestand dient informatie in de vorm van tekst te bevatten, en niet in de vorm van afbeeldingen waarin tekst is opgenomen.

Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.0 (niveau AAA)

Om er zeker van te zijn dat het invoerbestand voldoet aan de toegankelijkheidsnormen van de Richtlijnen voor Toegankelijkheid van Webcontent (WCAG) 2.0, niveau AA, dient u het volgende te controleren:

WCAG 1.4.6 – Vereiste inzake verbeterd contrast

De visuele weergave van tekst en afbeeldingen van tekst heeft een contrastverhouding van ten minste 7:1.

WCAG 1.4.8 – Visuele weergave

Het invoerbestand dient aan de volgende vereisten te voldoen:

1. De breedte bedraagt maximaal 80 tekens of glyphs (40 in het geval van CJK).
2. De tekst is niet uitgelijnd (zowel aan de linker- als aan de rechterkant uitgelijnd).
3. De regelafstand (interlinie) bedraagt binnen alinea's ten minste anderhalve spatie, en de afstand tussen alinea's is ten minste 1,5 keer zo groot als de regelafstand.

WCAG 1.4.9 Afbeeldingen van tekst (geen uitzondering)

Afbeeldingen van tekst worden uitsluitend gebruikt ter versiering of wanneer een bepaalde weergave van de tekst essentieel is voor de informatie die wordt overgebracht.

Vereisten voor invoerbestanden met het oog op naleving van WCAG 2.1 (niveau AA)

1.4.11 Contrast van niet-tekstuele elementen (niveau AA)

De visuele weergave van grafische objecten heeft een contrastverhouding van ten minste 3:1 ten opzichte van aangrenzende kleuren.

Controle op naleving van PDF/UA of WCAG

U kunt deze hulpmiddelen gebruiken om te controleren of uw PDF-bestand voldoet aan de PDF/UA- of WCAG-normen:

PAC – PDF-toegankelijkheidscontrole

Een gratis te downloaden hulpprogramma waarmee u kunt controleren of het PDF-bestand voldoet aan de PDF/UA- of WCAG-normen. Indien de controle mislukt, kunt u de problemen bekijken die ertoe hebben geleid dat het bestand niet aan de vereisten voldoet. U kunt het online downloaden via de PDF/UA Foundation: https://pac.pdf-accessibility.org/en.

3. Java-programmeerinterface

- **Klassen**

In dit hoofdstuk wordt de Java-API beschreven die door RICOH PDF2PDFUA wordt aangeboden.

De code tussen puntige haakjes bevat een referentie naar een XML-element in het controlebestand. Zo geeft `<If>` bijvoorbeeld een referentie aan voorwaardelijke constructies in het controlebestand.

Klassen

Deze klassen maken deel uit van de Java-API (Application Programming Interface) voor RICOH PDF2PDFUA.

enum EncryptionAlgorithm

Pakket: **com.ricoh.pdf2pdfua.api**

Optie waarmee het algoritme voor de versleuteling van de uitvoer-PDF wordt gespecificeerd.

Waarden:

none

De uitvoer heeft geen versleuteling.

preserve

Behoudt de versleuteling van de invoer-PDF.

Standard128

Een RC4-stroomcijfer met een sleutellengte van 128 bit.

AES128

Geavanceerde versleutelingstandaard die gebruikmaakt van een sleutellengte van 128 bits.

AES256

Geavanceerde versleutelingstandaard die gebruikmaakt van een sleutellengte van 256 bits.

interface IControlFile

Pakket: **com.ricoh.pdf2pdfua.api**.

Geeft de syntaxisboom van het controlebestand weer. Het controlebestand wordt één keer geparseerd en vervolgens op meerdere PDF-bestanden toegepast.

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

Wordt gebruikt om het controlebestand op wijzigingen te controleren.

Geeft `this` terug als het controlebestand up-to-date is.

Geeft een nieuw `IControlFile` terug als de onderliggende bestanden niet langer synchroon lopen.

interface Pdfua

Pakket: **com.ricoh.pdf2pdfua.api**

De Java Service Provider Interface (SPI) voor RICOH PDF2PDFUA.

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

Parseert het controlebestand dat is opgegeven door het pad. Eén exemplaar van een controlebestand kan in meerdere threads worden hergebruikt.

```
PdfUaOpt newOpt();
```

Maakt een instantie van PdfUaOpt aan met de standaardinstellingen.

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

Met deze methode wordt de PDF2PDFUA-conversie uitgevoerd. Het is thread-safe en herintreedbaar.

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


Deze methode voert de PDF2PDFUA-conversie uit en retourneert een byte-array die de getagde PDF bevat. Het is thread-safe en herintreedbaar.

class PDFUAException extends Exception

Pakket: **com.ricoh.pdf2pdfua.api**

Basisklasse voor conversiefouten in PDF2PDFUA.

class PdfUaLoader

Pakket: **com.ricoh.pdf2pdfua.api**

Een eenvoudige helper die de implementatie van de PDF2PDFUA-service laadt. De service kan rechtstreeks met een ServiceLoader worden geladen.

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

Laadt de implementatie van de PDF2PDFUA-service.

abstract class PdfUaOpt

Pakket: **com.ricoh.pdf2pdfua.api**

Wordt gebruikt om het gedrag van PDF2PDFUA te configureren. Het is een configuratiecontainer die uitsluitend getters en setters bevat.

Dit is een abstracte klasse, maar deze mag nooit door API-gebruikers worden uitgebreid. Om een instantie te verkrijgen, gebruikt u `Pdfua.newOpt()`.

Eén exemplaar kan door meerdere threads worden gebruikt, maar mag niet worden gewijzigd terwijl PDF2PDFUA actief is.

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

Toestemmingsvlag voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: false

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

Toestemmingsvlag voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: false

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

Toestemmingsvlag voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: false

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

Toestemmingsvlag voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: false

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

Toestemmingsvlag voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: false

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

Toestemmingsvlag voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: false

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

Toestemmingsvlag voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: false

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

Deze optie is vereist indien het invoer-PDF-bestand onder versleuteling staat. Wordt gebruikt om de invoer te ontgrendelen zonder de versleuteling te verwijderen. De eigenaarssleutel verdient de voorkeur.

Standaardwaarde: null

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

Hiermee wordt het algoritme voor de versleuteling van de uitvoer gespecificeerd.

Standaardwaarde: EncryptionAlgorithm.preserve

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

Indien 'linearize' op 'true' is ingesteld: wordt de uitvoer-PDF gelineariseerd.

Indien 'linearize' op 'false' is ingesteld: wordt de uitvoer-PDF niet-gelineariseerd.

Indien 'linearize' de waarde 'null' heeft: de uitvoer wordt alleen gelineariseerd als de invoer-PDF gelineariseerd was.

Standaardwaarde: null

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

Standaardwaarde: vastgelegd in het configuratiebestand.

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


Deze optie wordt gebruikt voor ontwikkeling en testen. Het mag niet in een productieomgeving worden gebruikt.

Standaardwaarde: false

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

Het wachtwoord van de eigenaar voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: null

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

Wordt gebruikt om de berichten te verwerken die normaal gesproken naar stderr worden geschreven.
Standaardwaarde: de ingebouwde handler zoals gespecificeerd in het configuratiebestand.

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

Wordt gebruikt om de berichten te verwerken die normaal gesproken naar stdout worden geschreven.

Standaardwaarde: de ingebouwde handler zoals gespecificeerd in het configuratiebestand.

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

Wordt uitsluitend gebruikt door het `<Trace>`-commando.

Standaardwaarde: de ingebouwde handler zoals gespecificeerd in het configuratiebestand.

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

Deze optie wordt gebruikt voor ontwikkeling en testen. Het mag niet in een productieomgeving worden gebruikt.

Standaardwaarde: `false`

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

Het gebruikerswachtwoord voor PDF-versleuteling. Heeft geen effect als de uitvoer niet door PDF2PDFUA is versleuteld.

Standaardwaarde: null

interface PdfUaOutputHandler

Pakket: **com.ricoh.pdf2pdfua.api**

Handler voor IO-omleiding.

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

`src`: het XML-knooppunt van de regel in het aanroepende controlebestand. Null indien het bericht niet aan een XML-knooppunt kan worden gekoppeld.

`pageNo`: het paginanummer begint bij 1. Deze waarde is nul als het bericht niet aan een pagina kan worden gekoppeld.

`msg`: kan uit meerdere regels tekst bestaan.

class PreprocessingException extends Exception

Pakket: **com.ricoh.pdf2pdfua.api**

Basisklasse voor fouten die optreden tijdens het voorbewerken van het controlebestand.

class UaStackTrace extends ArrayList<String>

Pakket: **com.ricoh.pdf2pdfua.api**

Een stacktrace die verwijst naar een XML-element in het controlebestand.

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

Voorbeelden

In dit voorbeeld wordt uitgelegd hoe u een controlebestand kunt parseren en hoe u dit op meerdere documenten kunt toepassen. Alle documenten in een batch dienen vergelijkbaar te zijn en in overeenstemming te zijn met het controlebestand. Om prestatieredenen wordt het controlebestand slechts één keer geparseerd.

De software zal naar verwachting draaien op servers met meerdere CPU-kernen. Dit voorbeeld maakt gebruik van multithreading, aangezien dit het standaardtoepassingsscenario is.

```
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);
    }
}
```

Dit voorbeeld laat zien hoe PDF2PDFUA kan werken zonder gebruik te maken van bestanden. Dit is handig wanneer PDF2PDFUA in andere systemen, zoals webapplicaties, is geïntegreerd.

```
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 Gebruikershandleiding