RICOH

RICOH Transforms Suite

Strumento PDF2PDFUA Guida per l'utente

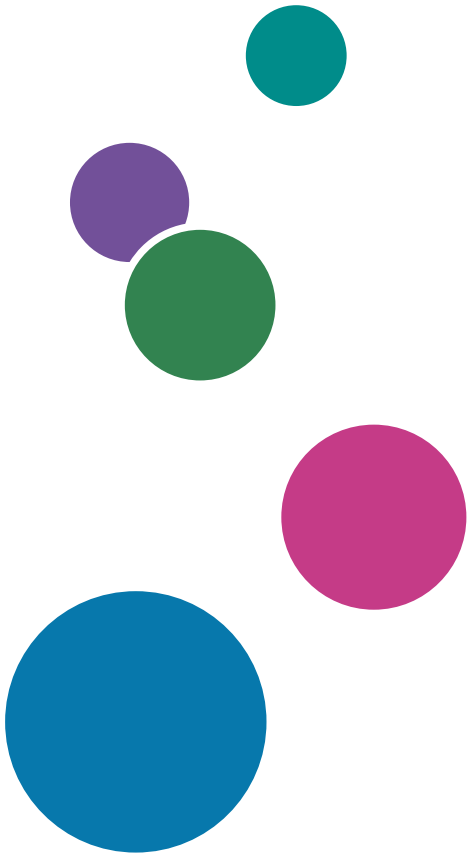
Version 1.5

Utilizzo dello strumento PDF2PDFUA 1

Garantire la conformità alle specifiche PDF/UA e alle WCAG 2

Interfaccia di programmazione delle applicazioni Java 3

Per le informazioni non presenti in questo manuale, fare riferimento alla Guida del prodotto in uso.



SOMMARIO

Introduzione

Importante 2
Precauzioni relative a questa guida 2
Pubblicazioni relative a questo prodotto 2
Simboli 2
Abbreviazioni 3
Marchi 3

1 Utilizzo dello strumento PDF2PDFUA

Utilizzo dello strumento PDF2PDFUA tramite la trasformazione AFP2PDF Plus 5
Utilizzo dello strumento PDF2PDFUA dalla riga di comando 5
Sintassi 5
Parametri 6
Configurazione del file delle opzioni `PDF2PDFUA.cfg` 7
Funzionalità supportate dalla trasformazione PDF2PDFUA 8

2 Garantire la conformità alle specifiche PDF/UA e alle WCAG

Requisiti relativi ai file di input per la conformità allo standard PDF/UA 9
Verifica del file PDF di input e del file PDF UA di output 9
Requisiti relativi ai file di input per la conformità alle WCAG 2.0 (Livello AA) 9
Requisiti relativi ai file di input per la conformità alle WCAG 2.0 (Livello AAA) 10
Requisiti relativi ai file di input per la conformità alle WCAG 2.1 (Livello AA) 10
Verifica della conformità alle specifiche PDF/UA o alle WCAG 10

3 Interfaccia di programmazione delle applicazioni Java

Corsi 11
Esempi 15

Introduzione

Importante

Nella misura massima consentita dalle leggi applicabili, in nessun caso il produttore sarà responsabile per qualsiasi danno derivante da guasti di questo prodotto, perdite di documenti o dati, o l'uso o il non uso di questo prodotto e dei manuali operativi forniti con esso.

Assicurarsi di copiare o avere sempre copie di sicurezza di documenti o dati importanti. I documenti o i dati potrebbero essere cancellati a causa di errori operativi o malfunzionamenti del software. Inoltre, l'utente è responsabile dell'adozione di misure di protezione contro virus, worm e altri software dannosi.

In nessun caso il produttore sarà ritenuto responsabile per i documenti creati dall'utente utilizzando il prodotto e per qualsiasi risultato che l'utente ottiene dai dati.

Precauzioni relative a questa guida

- Alcune illustrazioni o spiegazioni in questa guida potrebbero differire dal prodotto in uso a causa di miglioramenti o cambiamenti nel prodotto.
- Il contenuto di questo documento è soggetto a modifiche senza preavviso.
- Nessuna parte di questo documento può essere duplicata, replicata, riprodotta in qualsiasi forma, modificata o citata senza previo consenso del fornitore.

Pubblicazioni relative a questo prodotto

La presente pubblicazione offre una panoramica dello strumento e fornisce informazioni relative all'installazione e all'utilizzo dello strumento PDF2PDFUA.

Manuali di istruzioni

Per ulteriori informazioni sullo strumento PDF2PDFUA, si prega di consultare i seguenti documenti:

- *Guida all'installazione di PDF2PDFUA*
- *Guida per l'utente di PDF2PDFUA*

Simboli

Di seguito sono riportati i simboli utilizzati in questo manuale per identificare rapidamente i contenuti.

Importante

Questo simbolo indica i punti a cui prestare attenzione quando si utilizza l'applicazione. Assicurarsi di leggere queste spiegazioni.

Nota

Questo simbolo indica informazioni supplementari che potrebbero essere utili, ma non essenziali, per il completamento dell'attività.

[Grassetto]

Il **[Grassetto]** all'interno di parentesi quadre indica i nomi di menu, voci di menu, impostazioni, etichette di campi, pulsanti e tasti.

`Corsivo`

Il *`Corsivo`* indica variabili che devono essere sostituite con informazioni specificate dall'utente.

`Monospazio`

`I caratteri monospazio` indicano input e output del computer.

...

I puntini di sospensione indicano la continuazione di una serie.

Abbreviazioni

AES

Advanced Encryption Standard

AFP

Advanced Function Presentation

PDF

Portable Document Format (Formato documento portabile)

UA

Accessibilità universale

WCAG

Linee guida per l'accessibilità dei contenuti web

Marchi

Adobe, il logo Adobe, PostScript e il logo PostScript, PDF e il logo PDF sono marchi registrati o marchi di Adobe negli Stati Uniti e/o in altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o in altri Paesi.

Microsoft, Windows, Windows NT e il logo Windows sono marchi di Microsoft Corporation negli Stati Uniti, in altri paesi o in entrambi.

Sentinel® è un marchio registrato di Thales DIS CPL USA, Inc.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e in altri paesi.

Altri nomi di aziende, prodotti o servizi potrebbero essere marchi o marchi di servizio di altri.

1. Utilizzo dello strumento PDF2PDFUA

- **Utilizzo dello strumento PDF2PDFUA tramite la trasformazione AFP2PDF Plus**
- **Utilizzo dello strumento PDF2PDFUA dalla riga di comando**
- **Sintassi**
- **Parametri**
- **Configurazione del file delle opzioni** `PDF2PDFUA.cfg`
- **Funzionalità supportate dalla trasformazione PDF2PDFUA**

Utilizzo dello strumento PDF2PDFUA tramite la trasformazione AFP2PDF Plus

Per convertire i file AFP in documenti PDF/UA, è necessario aggiungere questo parametro di opzione nel file di configurazione di AFP2PDF Plus:

PDFUA_CONTROL_FILE=*`file`*

Specifica il nome completo del file di controllo dello strumento PDF2PDFUA. Questo file è un file XML che contiene informazioni sul contenuto del file di input. Lo strumento PDF2PDFUA utilizza il documento per contrassegnarne il contenuto secondo quanto previsto dallo standard PDF/UA.

PDFUA_CACHE_SIZE=*`numero`*

Specifica il numero dei file di controllo PDFUA più utilizzati memorizzati nella cache. Il numero deve essere maggiore di 0.

Nota

- Per ottenere un file di output PDF/UA crittografato o linearizzato, configurare le voci del file di opzioni di AFP2PDF Plus durante la trasformazione AFP2PDF Plus:

  Per la crittografia:

  `Default_Encryption_Permissions`

  `Default_owner_password`

  `Default_user_passowrd`

  Per la creazione di documenti linearizzati:

  `Default_linearization`

  Per ulteriori informazioni su come impostare le opzioni, vedere il capitolo *`File delle opzioni di trasformazione AFP2PDF Plus`* della *`Guida per l'utente di AFP2PDF Plus`*.

Limitazione:

- Durante la creazione di file PDF/UA, la crittografia AES a 256 bit non è supportata.

Utilizzo dello strumento PDF2PDFUA dalla riga di comando

Quando si esegue lo strumento PDF2PDFUA dalla riga di comando, il file di input può essere un documento PDF oppure un file di controllo convertito dallo strumento PDF2PDFUA in un file conforme allo standard PDF/UA.

Sintassi

A seconda del sistema operativo, la sintassi per la trasformazione PDF2PDFUA è la seguente:

PDF2PDFUA.*nnn* [**-o** *nome del file di output*] [**-i** *nome_file_di_controllo*] [**-b** [*numero_di_thread*] *directory di input directory di output*] [**-d** *output per l'esportazione del testo*] [**-linearize** *true/false*] [**-crittografare** *password proprietario password utente* flag di crittografia] [**-decrypt** *password*] [**-key** *password*] [**-trace** *espressione regolare*] [*file_di_input*]

dove *.nnn* rappresenta l'estensione **.bat** per Windows e **.sh** per Linux e AIX.

Parametri

-b

Questo parametro viene utilizzato per eseguire la trasformazione PDF2PDFUA in modalità batch. La trasformazione converte tutti i file PDF presenti in una directory in file PDF/UA, utilizzando un unico file di controllo specificato tramite il parametro **-i**.

Per specificare il numero di thread, inserisca il numero subito dopo l'opzione **-b**. Il numero predefinito dei threads è 1.

Nota

- In modalità batch, i parametri relativi alla *directory di input* e alla *directory di output* sono obbligatori. Il parametro *directory di input* sostituisce il parametro *file di input*.

-linearizzare

Questo parametro è facoltativo.

Il valore implicito non è specificato.

- Se il valore è vero, l'uscita viene linearizzata.
- Se il valore è falso, l'uscita non viene linearizzata, anche se l'entrata è linearizzata.
- Se il valore non viene specificato, l'uscita viene linearizzata solo se lo è anche l'entrata.

-crittografare

Questo parametro è facoltativo.

Per impostazione predefinita, l'utente non dispone di alcuna autorizzazione. Per concedere le autorizzazioni si utilizzano i seguenti flag:

`-consentiAnnotazioni`

`-allowAssemble`

`-allowCopy`

`-allowFill`

`-allowModify`

`-allowPrintDegraded`

`-allowPrintHigh`

`-allowPrint`, `-allowPrintDegraded` o `-allowPrintHigh`.

-decrypt *la password*

Questo parametro effettua la rimozione della crittografia originale.

-key *password*

Questo parametro consente di aprire i documenti bloccati senza la rimozione della crittografia originale. I permessi vengono mantenuti.

-trace *un'espressione regolare*

Questo parametro viene utilizzato per il debug dei file di controllo. In modalità riga di comando, tutti gli elementi di tracciamento sono attivi per impostazione predefinita.

-convalidare

Questo parametro attiva il validatore. È possibile utilizzarlo senza un file di controllo qualora si effettui una pre-convalida del PDF di input. Se viene specificato un file di controllo, viene verificato anche il PDF di output.

Configurazione del file delle opzioni PDF2PDFUA.cfg

I parametri aggiuntivi per il controllo delle impostazioni della trasformazione PDF2PDFUA sono specificati in un file di opzioni. Il nome è PDF2PDFUA.cfg. Il file PDF2PDFUA.cfg deve trovarsi nella directory principale dello strumento PDF2PDFUA.

I parametri dell'opzione PDF2PDFUA.cfg sono i seguenti:

AlwaysWriteProducer=TRUE | FALSE

Indica che il file PDF di output deve contenere informazioni relative alla versione della trasformazione PDF2PDFUA e alla versione del file di controllo all'interno del campo PDF Producer.

Il valore predefinito è **False**.

Quando il valore è **False**, il parametro ha la stessa funzione di quella specificata nel file di controllo.

Quando il valore è **True**, il parametro include i metadati del produttore indipendentemente dalle specifiche del file di controllo. I metadati sono utili per lo sviluppo. Questo parametro non è più necessario poiché le versioni più recenti di PDF2PDFUA includono i metadati in modi non standard. Gli sviluppatori possono estrarre i metadati, ma un lettore di PDF li ignorerà semplicemente.

DefaultLogSize= intero

Il valore predefinito è **10**. Indica la dimensione massima del file di log predefinito, espressa in megabyte. Questo parametro non ha alcun effetto se **useDefaultLog** è impostato su false.

errProgramVersion = true|false

Specifica in che modo la trasformazione gestisce i file di controllo progettati per le versioni più recenti di PDF2PDFUA.

Quando il valore è **true**, viene generata un'eccezione che impedisce al trasformatore di eseguire file di controllo incompatibili.

Quando il valore è **false**, tenta di eseguire file di controllo incompatibili, generando un avviso nel gestore di stampa (STDOUT per impostazione predefinita) e nei registri. Ciò potrebbe comportare la generazione di file PDF di output errati.

Il valore predefinito è **true** e si raccomanda vivamente di mantenere questa opzione impostata su **true**.

LegacyLinePosition=TRUE | FALSE

Il valore predefinito è **False**.

Quando il valore è **False**, la posizione è data dalla parola con la coordinata Y più piccola.

Quando il valore è **True**, la posizione è data dalla parola con la coordinata Y più elevata. Se la riga contiene indici, l'operatore ottieniPosizioneDellaRiga restituisce il valore Y dell'indice.

LegacyNextLine=TRUE | FALSE

Il valore predefinito è **True**. Influisce sul comportamento delle funzioni **ottenereLaRigaSuccessiva** e **ottenereLaRigaSuccessivaNell'Area**, ad eccezione dei parametri rigaSuccessiva o rigaSuccessivaNellArea. Questi operatori utilizzano sempre il nuovo comportamento.

Nell'ambito del comportamento preesistente, viene ricercata la prima riga in cui y(line) > y(cursor). Non funziona correttamente se la posizione del cursore viene impostata con la funzione **ottieniPosizioneDellaRiga**.

Nell'ambito del nuovo comportamento: cerca la prima riga in cui y(line) ≥ y(cursor). Risolve alcuni problemi, ma ne introduce un altro:

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

Ad esempio:

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

In questo caso, line1 e line2 sono identiche.

UseDefaultLog=TRUE | FALSE

Il valore predefinito è **True**.

Quando il valore è **True**, viene creato un file di log predefinito qualora non ne sia stato specificato alcuno dall'API. È utile in modalità autonoma.

Quando il valore è **False**, la registrazione viene abilitata solo se configurata tramite l'API.

Funzionalità supportate dalla trasformazione PDF2PDFUA

Con la trasformazione PDF2PDFUA Plus è possibile convertire in batch diversi file di input che condividono un modello di generazione comune. È possibile creare un file di controllo per qualsiasi file PDF, comprese fatture, estratti conto o estratti conto delle carte di credito. Avete la possibilità di creare un file di controllo per ogni tipo di fattura.

PDF2PDFUA Plus offre un supporto avanzato per la gestione dei dati variabili. PDF2PDFUA Plus è in grado di gestire dati variabili in posizioni fisse, quali nomi e campi di indirizzo, ma anche tabelle con un numero variabile di righe o tabelle che si estendono su più pagine. La trasformazione PDF2PDFUA consente di evidenziare solo il contenuto effettivo e di nascondere gli artefatti irrilevanti.

Nota

Si prega di contattare il proprio rappresentante Ricoh per ricevere ulteriori informazioni sulla procedura di creazione di un file di controllo.

2. Garantire la conformità alle specifiche PDF/UA e alle WCAG

- **Requisiti relativi ai file di input per la conformità allo standard PDF/UA**
- **Verifica del file PDF di input e del file PDF UA di output**
- **Requisiti relativi ai file di input per la conformità alle WCAG 2.0 (Livello AA)**
- **Requisiti relativi ai file di input per la conformità alle WCAG 2.0 (Livello AAA)**
- **Requisiti relativi ai file di input per la conformità alle WCAG 2.1 (Livello AA)**
- **Verifica della conformità alle specifiche PDF/UA o alle WCAG**

Requisiti relativi ai file di input per la conformità allo standard PDF/UA

Se il file di input presenta uno di questi problemi, il file non potrà essere elaborato oppure il file risultante non sarà conforme allo standard PDF/UA.

- Il file PDF contiene contenuti che presentano sfarfallio o elementi PDF che causano lo sfarfallio (Protocollo Matterhorn – Cap. 03-X).
- Le informazioni contenute nel file PDF vengono trasmesse tramite il contrasto, il colore o l'impaginazione (MP, cap. 4).
- Il file PDF contiene audio (MP Ch05).
- Il file PDF utilizza codici di caratteri senza mappatura Unicode (MP, cap. 10).
- Il file PDF contiene immagini con paragrafi di testo (MP C13-06).
- Il file PDF utilizza caratteri non incorporati (MP C21).

  Nota

  Se lo si utilizza insieme alla trasformazione AFP2PDF Plus, si assicuri che sia configurata per incorporare i caratteri.
- Il file PDF contiene dei moduli.
- Il file PDF contiene elementi di azione.

Verifica del file PDF di input e del file PDF UA di output

Il validatore PDF2PDFUA genera avvisi che possono fare riferimento a indici specifici del Matterhorn Protocollo 1.1 per le associazioni PDF. Il pre-validatore verifica alcune delle condizioni menzionate nel Protocollo Matterhorn.

Per ottenere ulteriori informazioni sugli errori visualizzati, cerchi l'ID dell'errore nel Documento di protocollo del Matterhorn.

Si prega di contattare il proprio fornitore di servizi di creazione di file PDF per analizzare e risolvere questo errore, al fine di ottenere un file in formato PDF/UA conforme agli standard.

Requisiti relativi ai file di input per la conformità alle WCAG 2.0 (Livello AA)

Per assicurarsi che il file di input sia conforme agli standard di accessibilità delle Linee guida per l'accessibilità dei contenuti web (WCAG) 2.0 Livello AA, si prega di verificare quanto segue:

WCAG 1.4.3: Contrasto minimo

Il file di input deve contenere testo e immagini con un contrasto minimo di 4,5:1.

WCAG 1.4.5: Immagini come testo

Il file di input dovrebbe fornire informazioni sotto forma di testo, anziché tramite immagini contenenti testo.

Requisiti relativi ai file di input per la conformità alle WCAG 2.0 (Livello AAA)

Per assicurarsi che il file di input sia conforme agli standard di accessibilità delle Linee guida per l'accessibilità dei contenuti web (WCAG) 2.0 Livello AA, si prega di verificare quanto segue:

WCAG 1.4.6 – Requisito relativo al contrasto potenziato

La presentazione visiva del testo e delle immagini contenenti testo presenta un rapporto di contrasto pari ad almeno 7:1.

WCAG 1.4.8 – Presentazione visiva

Il file di input deve soddisfare i seguenti requisiti:

1. La larghezza non supera gli 80 caratteri o glifi (40 se si tratta di caratteri CJK).
2. Il testo non è giustificato (allineato sia al margine sinistro che a quello destro).
3. L'interlinea (spaziatura tra le righe) è pari ad almeno una riga e mezza all'interno dei paragrafi, mentre la spaziatura tra i paragrafi è pari ad almeno 1,5 volte l'interlinea.

WCAG 1.4.9 Immagini di testo (Nessuna Eccezione)

Le immagini contenenti testo vengono utilizzate esclusivamente a scopo puramente decorativo oppure nei casi in cui una particolare presentazione del testo sia essenziale per le informazioni che si intendono trasmettere.

Requisiti relativi ai file di input per la conformità alle WCAG 2.1 (Livello AA)

1.4.11 Contrasto degli elementi non testuali (Livello AA)

La rappresentazione visiva degli oggetti grafici presenta un rapporto di contrasto di almeno 3:1 rispetto ai colori adiacenti.

Verifica della conformità alle specifiche PDF/UA o alle WCAG

È possibile utilizzare questi strumenti per verificare che il proprio file PDF sia conforme agli standard PDF/UA o WCAG:

PAC – Strumento di verifica dell'accessibilità dei file PDF

Uno strumento scaricabile gratuitamente che verifica se un file PDF è conforme agli standard PDF/UA o alle WCAG. Se la verifica non va a buon fine, potrà visualizzare eventuali problemi che hanno determinato la non conformità del file. È possibile scaricarlo online dal sito della PDF/UA Foundation: https://pac.pdf-accessibility.org/en.

3. Interfaccia di programmazione delle applicazioni Java

• **Corsi**

Questo capitolo descrive l'API Java fornita da RICOH PDF2PDFUA.

Il codice racchiuso tra parentesi angolari rappresenta un riferimento a un elemento XML nel file di controllo. Ad esempio, `<Se>` ha un riferimento alle istruzioni condizionali presenti nel file di controllo.

Corsi

Queste classi fanno parte dell'interfaccia di programmazione delle applicazioni (API) Java per RICOH PDF2PDFUA.

enum AlgoritmoDiCrittografia

Pacchetto: **com.ricoh.PDF2PDFUA.api**

Opzione che specifica l'algoritmo di crittografia del file PDF di output.

Valori:

nessuno

L'output non è crittografato.

conservare

Mantiene la crittografia del file PDF di origine.

Standard128

Un cifrario a flusso con chiave da 128 bit basato sull'algoritmo RC4.

AES128

Standard di crittografia avanzato che utilizza una chiave di 128 bit.

AES256

Standard di crittografia avanzato che utilizza una chiave di 256 bit.

interfaccia IControlFile

Pacchetto: **com.ricoh.PDF2PDFUA.api**.

Rappresenta l'albero sintattico del file di controllo. Il file di controllo viene analizzato una sola volta e applicato a più file PDF.

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

Viene utilizzato per verificare la presenza di modifiche nel file di controllo.

Restituisce `questo` se il file di controllo è aggiornato.

Restituisce un nuovo oggetto `IControlFile` se i file sottostanti non sono più sincronizzati.

interfaccia PDFUA

Pacchetto: **com.ricoh.PDF2PDFUA.api**

L'interfaccia SPI (Service Provider Interface) Java per RICOH PDF2PDFUA.

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

Analizza il file di controllo specificato dal percorso. Una singola istanza di file di controllo può essere riutilizzata in più thread.

```
PdfUaOpt newOpt();
```

Crea un'istanza di PdfUaOpt con le impostazioni predefinite.

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

Questo metodo esegue la conversione PDF2PDFUA. È thread-safe e rientrante.


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

Questo metodo esegue la conversione PDF2PDFUA e restituisce un array di byte contenente il PDF con tag. È thread-safe e rientrante.

classe PDFUAException estende Eccezione

Pacchetto: **com.ricoh.PDF2PDFUA.api**

Classe base per gli errori di conversione PDF2PDFUA.

classe PdfUaLoader

Pacchetto: **com.ricoh.PDF2PDFUA.api**

Piccolo programma di supporto che carica l'implementazione del servizio PDF2PDFUA. Il servizio può essere caricato direttamente tramite un ServiceLoader.

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

Carica l'implementazione del servizio PDF2PDFUA.

classe astratta PdfUaOpt

Pacchetto: **com.ricoh.PDF2PDFUA.api**

Serve a configurare il comportamento di PDF2PDFUA. Si tratta di un contenitore di configurazione che dispone esclusivamente di metodi getter e setter.

Si tratta di una classe astratta, ma non deve mai essere estesa dagli utenti dell'API. Per ottenere un'istanza, utilizzi `PDFUA.newOpt()`.

Una singola istanza può essere utilizzata da più thread, ma non deve essere modificata mentre PDF2PDFUA è in esecuzione.

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

Flag di autorizzazione per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: false

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

Flag di autorizzazione per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: false

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

Flag di autorizzazione per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: false

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


Flag di autorizzazione per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: false

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

Flag di autorizzazione per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: false

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

Flag di autorizzazione per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: false

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

Flag di autorizzazione per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: false

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

Questa opzione è necessaria se il file PDF di input è crittografato. Si utilizza per sbloccare l'input senza la rimozione della crittografia. È preferibile utilizzare la chiave del proprietario.

Valore predefinito: null

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

Specifica l'algoritmo di crittografia dell'output.

Valore predefinito: EncryptionAlgorithm.preserve

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

Se linearize è vero: la PDF di uscita viene linearizzata.

Se linearize è falso: la PDF di output non è linearizzata.

Se linearize è nullo: l'uscita viene linearizzata solo se la PDF di ingresso è stata linearizzata.

Valore predefinito: null

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

Valore predefinito: specificato nel file di configurazione.

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


Questa opzione viene utilizzata per lo sviluppo e il collaudo. Non deve essere utilizzato in un ambiente di produzione.

Valore predefinito: false

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

La password del proprietario per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: null

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

Viene utilizzato per gestire i messaggi che normalmente vengono scritti su stderr. Valore predefinito: gestore integrato, come specificato nel file di configurazione.

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

Viene utilizzato per gestire i messaggi che vengono normalmente scritti su stdout.

Valore predefinito: gestore integrato, come specificato nel file di configurazione.

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

Utilizzato esclusivamente dal comando `<Trace>`.

Valore predefinito: gestore integrato, come specificato nel file di configurazione.

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

Questa opzione viene utilizzata per lo sviluppo e il collaudo. Non deve essere utilizzato in un ambiente di produzione.

Valore predefinito: `false`

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

La password utente per la crittografia dei file PDF. Non ha alcun effetto se l'output non è crittografato da PDF2PDFUA.

Valore predefinito: null

interfaccia PdfUaOutputHandler

Pacchetto: **com.ricoh.PDF2PDFUA.api**

Gestore per il reindirizzamento dell'I/O.

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

`src`: il nodo XML della regola del file di controllo che ha generato la chiamata. Null se non è possibile ricondurre il messaggio a un nodo XML.

`pageNo`: il numero di pagina parte da 1. Questo valore è pari a zero se non è possibile ricondurre il messaggio a una pagina.

`msg`: può contenere più righe di testo.

classe PreprocessingException estende Eccezione

Pacchetto: **com.ricoh.PDF2PDFUA.api**

Classe base per gli errori che si verificano durante la pre-elaborazione del file di controllo.

classe UaStackTrace estende ArrayList<STRINGA>

Pacchetto: **com.ricoh.PDF2PDFUA.api**

Una traccia dello stack che rimanda a un elemento XML nel file di controllo.

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

Esempi

Questo esempio illustra come analizzare un file di controllo e come applicarlo a più documenti. Tutti i documenti contenuti in un batch devono essere simili e compatibili con il file di controllo. Per motivi di prestazioni, il file di controllo viene analizzato una sola volta.

Si prevede che il software funzioni su server dotati di più core di CPU. Questo esempio è multithread poiché rappresenta il caso d'uso standard.

```
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);
    }
}
```

Questo esempio illustra come PDF2PDFUA possa funzionare senza ricorrere all'utilizzo di file. Ciò risulta utile quando PDF2PDFUA è integrato in altri sistemi, quali le applicazioni web.

```
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 Strumento PDF2PDFUA Guida per l'utente