RICOH

RICOH Transforms Suite

Herramienta PDF2PDFUA
Guía de usuario

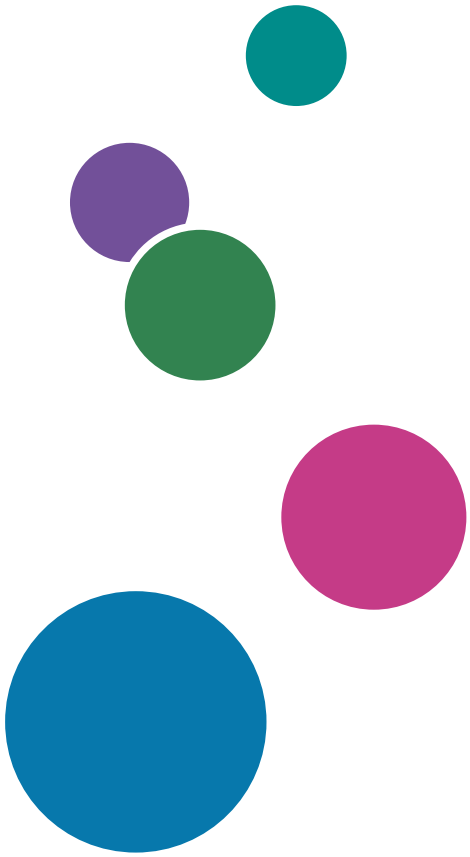
Version 1.5

Cómo utilizar la herramienta PDF2PDFUA 1

Garantizar el cumplimiento de las normas PDF/UA y WCAG 2

Interfaz de programación de aplicaciones de Java 3

Para cualquier información que no se encuentre en este manual, consulte el sistema de ayuda en su producto.



CONTENIDO

Introducción

Importante ... 2
Precauciones a tener en cuenta con respecto a esta guía ... 2
Publicaciones relacionadas con este producto ... 2
Símbolos ... 2
Abreviaciones ... 3
Marcas registradas ... 3

1 Cómo utilizar la herramienta PDF2PDFUA

Uso de la herramienta PDF2PDFUA a través de la transformación AFP2PDF Plus ... 5
Uso de la herramienta PDF2PDFUA desde la línea de comandos ... 5
Sintaxis ... 5
Parámetros ... 6
Configuración del archivo de opciones `PDF2PDFUA.cfg` ... 7
Funcionalidades compatibles con la transformación PDF2PDFUA ... 8

2 Garantizar el cumplimiento de las normas PDF/UA y WCAG

Requisitos del archivo de entrada para el cumplimiento de la norma PDF/UA ... 9
Validación del archivo PDF de entrada y del archivo PDF UA de salida ... 9
Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.0 (Nivel AA) ... 10
Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.0 (Nivel AAA) ... 10
Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.1 (Nivel AA) ... 10
Validación del cumplimiento de las normas PDF/UA o WCAG ... 10

3 Interfaz de programación de aplicaciones de Java

Clases ... 13
Ejemplos ... 17

Introducción

Importante

En la medida en que lo permita la legislación aplicable, el fabricante no será responsable de ningún daño causado por fallos de este software, pérdida de documentos o datos, o el uso o no uso de este producto y los manuales de instrucciones que se proporcionan con el mismo.

Asegúrese de realizar siempre una copia o de contar con copias de seguridad de los documentos o datos importantes. Los documentos o los datos pueden borrarse debido a errores de funcionamiento o fallos del software. Asimismo, usted es el responsable de tomar las medidas preventivas necesarias para evitar virus informáticos, gusanos y cualquier otro tipo de software dañino.

El fabricante no será responsable en ningún caso de los documentos que usted cree con este producto ni de los resultados de los datos que ejecute.

Precauciones a tener en cuenta con respecto a esta guía

- Algunas ilustraciones o explicaciones de esta guía pueden diferir de su producto debido a mejoras o cambios en este último.
- El contenido de este documento está sujeto a cambios sin previo aviso.
- No podrá copiarse, reproducirse de ningún modo, modificarse ni hacerse referencia a este documento, de ninguna forma, sin el consentimiento previo del proveedor.

Publicaciones relacionadas con este producto

Esta publicación ofrece una descripción general de la herramienta, así como información sobre la instalación y el uso de la herramienta PDF2PDFUA.

Manuales de instrucciones

Para obtener más información sobre la herramienta PDF2PDFUA, consulte los siguientes documentos:

- *Guía de instalación de PDF2PDFUA*
- *Guía de usuario de PDF2PDFUA*

Símbolos

Los siguientes símbolos se utilizan en este manual para ayudarle a identificar rápidamente el contenido.

Importante

Este símbolo indica puntos a los que hay que prestar atención al utilizar la aplicación. Asegúrese de leer estas explicaciones.

Nota

Este símbolo indica información complementaria que podría resultar útil, pero que no es esencial para completar una tarea.

[Negrita]

Los corchetes con contenido en **[Tipo en negrita]** indican los nombres de teclas, menús, elementos de menú, ajustes, etiquetas de campos y botones.

`Cursiva`

La *`cursiva`* indica variables que debe sustituir con su propia información.

`Monoespaciado`

`Tipo monoespaciado` indica la entrada y salida del equipo.

...

Una elipsis indica que una serie puede continuar.

Abreviaciones

AES

Advanced Encryption Standard

AFP

Advanced Function Presentation

PDF

Portable Document Format (Formato de documento portátil)

UA

Accesibilidad universal

WCAG

Directrices de accesibilidad al contenido web

Marcas registradas

Adobe, el logotipo de Adobe, PostScript y el logotipo de PostScript, PDF y el logotipo de PDF son marcas registradas o marcas comerciales de Adobe en Estados Unidos y/o en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos y en otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas comerciales de Microsoft Corporation en los Estados Unidos, otros países o ambos.

Sentinel es una marca registrada de Thales DIS CPL USA, Inc.

UNIX es una marca comercial registrada de The Open Group en los Estados Unidos y otros países.

Otros nombres de compañía, producto o servicio pueden ser marcas comerciales o marcas de servicio de otros.

1. Cómo utilizar la herramienta PDF2PDFUA

- **Uso de la herramienta PDF2PDFUA a través de la transformación AFP2PDF Plus**
- **Uso de la herramienta PDF2PDFUA desde la línea de comandos**
- **Sintaxis**
- **Parámetros**
- **Configuración del archivo de opciones** `PDF2PDFUA.cfg`
- **Funcionalidades compatibles con la transformación PDF2PDFUA**

Uso de la herramienta PDF2PDFUA a través de la transformación AFP2PDF Plus

Para convertir archivos AFP en documentos PDF/UA, debe añadir este parámetro de opción al archivo de configuración de AFP2PDF Plus:

PDFUA_CONTROL_FILE=*archivo*

Especifica el nombre completo del archivo de control de la herramienta PDF2PDFUA. Este archivo es un archivo XML que contiene información sobre el contenido del archivo de entrada. La herramienta PDF2PDFUA utiliza el documento para marcar el contenido tal y como exige la norma PDF/UA.

PDFUA_CACHE_SIZE=*número*

Especifica el número de archivos de control PDFUA más utilizados que se almacenan en la caché. El número debe ser mayor que 0.

Nota

- Para obtener un archivo de salida PDF/UA cifrado o linealizado, configure las entradas del archivo de opciones de AFP2PDF Plus al ejecutar la transformación AFP2PDF Plus:

  Para el cifrado:

  ```
  Default_Encryption_Permissions
  Default_owner_password
  Default_user_passowrd
  ```

  Para la creación de documentos linealizados:

  ```
  Default_linearization
  ```

  Para obtener más información sobre cómo configurar las opciones, consulte el capítulo *Archivo de opciones de transformación AFP2PDF Plus* de la *Guía de usuario de AFP2PDF Plus*.

Limitación:

- Al crear archivos PDF/UA, no se admite el cifrado AES de 256 bits.

Uso de la herramienta PDF2PDFUA desde la línea de comandos

Cuando se ejecuta la herramienta PDF2PDFUA desde la línea de comandos, el archivo de entrada puede ser un documento PDF o un archivo de control que la herramienta PDF2PDFUA haya convertido en un archivo compatible con PDF/UA.

Sintaxis

En función del sistema operativo, la sintaxis de la transformación PDF2PDFUA es la siguiente:

PDF2PDFUA.*nnn* [**-o** *nombre del archivo de salida*] [**-i** *nombre del archivo de control*] [**-b** [*número_de_hilos*] *directorio de entrada directorio de salida*] [**-d** *salida para volcar texto*] [**-linearize** *verdadero/falso*] [**-cifrar** *contraseña del propietario*, *contraseña del usuario* indicadores de cifrado] [**-decrypt** *contraseña*] [**-key** *contraseña*] [**-rastreo** *expresión regular*] [*archivo de entrada*]

donde *.nnn* representa la extensión **.bat** para Windows y **.sh** para Linux y AIX.

Parámetros

-b

Este parámetro se utiliza para ejecutar la transformación PDF2PDFUA en modo por lotes. La transformación convierte todos los archivos PDF de un directorio en archivos PDF/UA mediante un único archivo de control que se especifica a través del parámetro **-i**.

Para especificar el número de subprocesos, añada dicho número justo después del indicador **-b**. El número de subproceso por defecto es 1.

Nota

- En el modo por lotes, los parámetros de *directorio de entrada* y *directorio de salida* son obligatorios. El parámetro *directorio de entrada* sustituye al parámetro *archivo de entrada*.

-linearize

Este parámetro es opcional.

El valor implícito no se ha especificado.

- Si el valor es verdadero, la salida se linealiza.
- Si el valor es falso, la salida no se linealiza, aunque la entrada sí lo esté.
- Si no se especifica el valor, la salida solo se linealiza si la entrada está linealizada.

-encrypt

Este parámetro es opcional.

Por defecto, el usuario no tiene ningún permiso. Para conceder permisos se utilizan los siguientes indicadores:

`-allowAnnotations`

`-allowAssemble`

`-allowCopy`

`-allowFill`

`-allowModify`

`-allowPrintDegraded`

`-allowPrintHigh`

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

-decrypt *password*

Este parámetro elimina el cifrado original.

-key *password*

Este parámetro permite abrir documentos bloqueados sin eliminar el cifrado original. Se conservan los permisos.

-trace *expresión regular*

Este parámetro se utiliza para depurar archivos de control. Todos los elementos de seguimiento están activados de forma predeterminada en el modo de línea de comandos.

-validate

Este parámetro habilita el validador. Se puede utilizar sin un archivo de control si se realiza una validación previa del PDF de entrada. Si se especifica un archivo de control, también comprueba el PDF resultante.

Configuración del archivo de opciones PDF2PDFUA.cfg

Los parámetros adicionales para controlar la configuración de la transformación PDF2PDFUA se especifican en un archivo de opciones. El nombre es PDF2PDFUA.cfg. El archivo PDF2PDFUA.cfg debe encontrarse en el directorio raíz de la herramienta PDF2PDFUA.

Los parámetros de opción de PDF2PDFUA.cfgson los siguientes:

AlwaysWriteProducer=TRUE | FALSE

Indica que el archivo PDF de salida debe contener información sobre la versión de la transformación PDF2PDFUA y la versión del archivo de control en el campo PDF Producer.

El valor por defecto es **False**.

Cuando el valor es **False**, el parámetro tiene la misma función que la especificada en el archivo de control.

Cuando el valor es **True**, el parámetro incluye los metadatos del productor, independientemente de lo que especifique el archivo de control. Los metadatos resultan útiles para el desarrollo. Este parámetro ya no es necesario, ya que las versiones más recientes de PDF2PDFUA incluyen los metadatos de formas no estándar. Los desarrolladores pueden extraer los metadatos, pero un lector de PDF simplemente los ignorará.

DefaultLogSize= entero

El valor por defecto es **10**. Representa el tamaño máximo del registro predeterminado, en megabytes. Este parámetro no tiene ningún efecto si **useDefaultLog** es falso.

errProgramVersion = true|false

Especifica cómo gestiona la transformación los archivos de control diseñados para versiones más recientes de PDF2PDFUA.

Cuando el valor es **true**, se lanza una excepción, lo que impide que la transformación ejecute archivos de control incompatibles.

Cuando el valor es **false**, intenta ejecutar archivos de control incompatibles, lo que genera una advertencia en el controlador de salida (por defecto, STDOUT) y en los registros. Esto podría dar lugar a archivos PDF de salida incorrectos.

El valor por defecto es **true** y se recomienda encarecidamente mantener esta opción en **true**.

LegacyLinePosition=TRUE | FALSE

El valor por defecto es **False**.

Cuando el valor es **False**, la posición viene determinada por la palabra con la coordenada Y más pequeña.

Cuando el valor es **True**, la posición viene determinada por la palabra con la coordenada Y más alta. Si la línea contiene subíndices, el operador obtenerPosiciónDeLínea devuelve la coordenada Y del subíndice.

LegacyNextLine= TRUE | FALSE

El valor por defecto es **True**. Afecta al comportamiento de las funciones **irALaLíneaSiguiente** y **obtenerLaLíneaSiguienteEnLaRegión**, salvo en el caso de líneaSiguiente o líneaSiguienteEnLaRegión. Estos operadores siempre utilizan el nuevo comportamiento.

En el comportamiento anterior, busca la primera línea en la que y(línea) > y(cursor). No funciona correctamente si se establece la posición del cursor mediante la función **obtenerPosiciónDeLínea**.

En el nuevo comportamiento: busca la primera línea en la que y(línea) sea ≥ y(cursor). Soluciona algunos problemas, pero plantea otro:

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

Por ejemplo:

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

En este caso, línea1 y línea2 son lo mismo.

UseDefaultLog= TRUE | FALSE

El valor por defecto es **True**.

Cuando el valor es **True**, se crea un registro predeterminado si no se ha pasado ninguno desde la API. Resulta útil en modo autónomo.

Cuando el valor es **False**, el registro de eventos solo se activa si se ha configurado desde la API.

Funcionalidades compatibles con la transformación PDF2PDFUA

Con la transformación PDF2PDFUA Plus, podrá convertir por lotes varios archivos de entrada que compartan una plantilla de generación común. Puede crear un archivo de control para cualquier archivo PDF, incluidas facturas, resúmenes de cuentas o extractos de tarjetas de crédito. Tiene la posibilidad de crear un archivo de control para cada tipo de factura.

PDF2PDFUA Plus ofrece una compatibilidad mejorada para el tratamiento de datos variables. PDF2PDFUA Plus puede gestionar datos variables en posiciones fijas, como nombres o campos de dirección, así como tablas con un número variable de filas o tablas que se extienden a lo largo de varias páginas. La transformación PDF2PDFUA le permite marcar únicamente el contenido real y ocultar los artefactos irrelevantes.

Nota

Póngase en contacto con su representante de Ricoh para obtener más información sobre el procedimiento para crear un archivo de control.

2. Garantizar el cumplimiento de las normas PDF/UA y WCAG

- **Requisitos del archivo de entrada para el cumplimiento de la norma PDF/UA**
- **Validación del archivo PDF de entrada y del archivo PDF UA de salida**
- **Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.0 (Nivel AA)**
- **Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.0 (Nivel AAA)**
- **Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.1 (Nivel AA)**
- **Validación del cumplimiento de las normas PDF/UA o WCAG**

Requisitos del archivo de entrada para el cumplimiento de la norma PDF/UA

Si el archivo de entrada presenta alguno de estos problemas, no se podrá procesar o el archivo resultante no cumplirá con el estándar PDF/UA.

- El archivo PDF contiene contenido que parpadea o elementos del PDF que provocan parpadeos (Protocolo Matterhorn – Cap. 3-X).
- La información contenida en el archivo PDF se transmite mediante el contraste, el color o el diseño de formato (MP, cap. 4).
- El archivo PDF contiene sonido (MP Ch05).
- El archivo PDF utiliza códigos de caracteres sin asignación Unicode (MP, cap. 10).
- El archivo PDF contiene imágenes con párrafos de texto (MP C13-06).
- El archivo PDF utiliza fuentes no incrustadas (MP C21).

  Nota

  Si se utiliza junto con la transformación AFP2PDF Plus, asegúrese de que esté configurada para incrustar las fuentes.
- El archivo PDF contiene formularios.
- El archivo PDF contiene elementos de acción.

Validación del archivo PDF de entrada y del archivo PDF UA de salida

El validador PDF2PDFUA genera advertencias que pueden tener referencias a índices específicos del Protocolo Matterhorn 1.1 de las asociaciones de PDF. El prevalidador comprueba algunas de las condiciones mencionadas en el Protocolo Matterhorn.

Para obtener más información sobre los errores mostrados, busque el ID del error en el documento del protocolo Marterhorn.

Póngase en contacto con su proveedor de servicios de creación de archivos PDF para que analice y resuelva este error del PDF, con el fin de obtener un archivo de salida compatible con el estándar PDF/UA.

Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.0 (Nivel AA)

Para asegurarse de que el archivo de entrada cumple con las normas de accesibilidad de las Pautas de Accesibilidad al Contenido Web (WCAG) 2.0, Nivel AA, compruebe lo siguiente:

WCAG 1.4.3: Contraste mínimo

El archivo de entrada debe contener texto e imágenes con un contraste mínimo de 4,5:1.

WCAG 1.4.5: Imágenes como texto

El archivo de entrada debe proporcionar la información mediante texto, en lugar de imágenes que contengan texto.

Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.0 (Nivel AAA)

Para asegurarse de que el archivo de entrada cumple con las normas de accesibilidad de las Pautas de Accesibilidad al Contenido Web (WCAG) 2.0, Nivel AA, compruebe lo siguiente:

WCAG 1.4.6 – Requisito de contraste mejorado

La presentación visual del texto y las imágenes que contienen texto deben tener una relación de contraste de al menos 7:1.

WCAG 1.4.8 – Presentación visual

El archivo de entrada debe cumplir los siguientes requisitos:

1. La anchura no supera los 80 caracteres o glifos (40 si se trata de caracteres CJK).
2. El texto no está justificado (alineado tanto al margen izquierdo como al derecho).
3. El interlineado (espaciado entre líneas) es de, como mínimo, un espacio y medio dentro de los párrafos, y el espaciado entre párrafos es, como mínimo, 1,5 veces mayor que el interlineado.

WCAG 1.4.9 Imágenes de texto (sin excepción)

Las imágenes de texto solo se utilizan con fines puramente decorativos o cuando una presentación concreta del texto resulta esencial para la información que se desea transmitir.

Requisitos de los archivos de entrada para el cumplimiento de las WCAG 2.1 (Nivel AA)

1.4.11 Contraste de elementos no textuales (Nivel AA)

La presentación visual de los objetos gráficos presenta una relación de contraste de al menos 3:1 con respecto a los colores adyacentes.

Validación del cumplimiento de las normas PDF/UA o WCAG

Puede utilizar estas herramientas para asegurarse de que su archivo PDF cumple con las normas PDF/UA o WCAG:

PAC – Comprobador de accesibilidad de archivos PDF

Una herramienta de descarga gratuita que comprueba si un archivo PDF cumple con las normas PDF/UA o WCAG. Si la verificación falla, podrá consultar los problemas que han provocado que el archivo no cumpla con los requisitos. Puede descargarlo en línea desde la Fundación PDF/UA: https://pac.pdf-accessibility.org/en.

3. Interfaz de programación de aplicaciones de Java

- **Clases**

En este capítulo se describe la API de Java que ofrece RICOH PDF2PDFUA.

El código entre corchetes angulares proporciona una referencia a un elemento XML del archivo de control. Por ejemplo, `<If>` hace referencia a las condiciones del archivo de control.

Clases

Estas clases forman parte de la interfaz de programación de aplicaciones (API) de Java para RICOH PDF2PDFUA.

enum EncryptionAlgorithm

Paquete: **com.ricoh.PDF2PDFUA.api**

Opción que especifica el algoritmo de encriptación del archivo PDF de salida.

Valores:

ninguno

La salida no está cifrada.

conservar

Conserva el cifrado del archivo PDF de origen.

Estándar 128

Un cifrado de flujo RC4 con una clave de 128 bits.

AES128

Estándar de cifrado avanzado que utiliza una clave de 128 bits.

AES256

Estándar de cifrado avanzado que utiliza una clave de 256 bits.

interfaz IControlFile

Paquete: **com.ricoh.PDF2PDFUA.api**.

Representa el árbol sintáctico del archivo de control. El archivo de control se analiza una sola vez y se aplica a varios archivos PDF.

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

Se utiliza para comprobar si se han producido cambios en el archivo de control.

Devuelve `este valor` si el archivo de control está actualizado.

Devuelve un nuevo objeto `IControlFile` si los archivos subyacentes ya no están sincronizados.

interfaz PDFUA

Paquete: **com.ricoh.PDF2PDFUA.api**

La interfaz de proveedor de servicios (SPI) de Java para RICOH PDF2PDFUA.

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

Analiza el archivo de control especificado en path. Una única instancia del archivo de control puede reutilizarse en varios subprocesos.

```
PdfUaOpt newOpt();
```

Crea una instancia de PdfUaOpt con la configuración predeterminada.

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

Este método ejecuta la conversión PDF2PDFUA. Es seguro para subprocesos y reentrante.

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


Este método ejecuta la conversión PDF2PDFUA y devuelve una matriz de bytes que contiene el PDF etiquetado. Es seguro para subprocesos y reentrante.

clase PDFUAException extiende Exception

Paquete: **com.ricoh.PDF2PDFUA.api**

Clase base para los errores de conversión de PDF2PDFUA.

clase PdfUaLoader

Paquete: **com.ricoh.PDF2PDFUA.api**

Ayuda sencilla que carga la implementación del servicio PDF2PDFUA. El servicio se puede cargar directamente mediante un ServiceLoader.

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

Carga la implementación del servicio PDF2PDFUA.

clase abstracta PdfUaOpt

Paquete: **com.ricoh.PDF2PDFUA.api**

Se utiliza para configurar el comportamiento de PDF2PDFUA. Se trata de un contenedor de configuración que solo contiene métodos getter y setter.

Se trata de una clase abstracta, pero los usuarios de la API nunca deben extenderla. Para obtener una instancia, utilice `PDFUA.newOpt()`.

Una única instancia puede ser utilizada por varios subprocesos, pero no debe modificarse mientras PDF2PDFUA esté en ejecución.

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

Indicador de permiso para el cifrado de archivos PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: false

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

Indicador de permiso para el cifrado de archivos PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: false

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

Indicador de permiso para el cifrado de archivos PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: false

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

Indicador de permiso para el cifrado de archivos PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: false

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

Indicador de permiso para el cifrado de archivos PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: false

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

Indicador de permiso para el cifrado de archivos PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: false

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

Indicador de permiso para el cifrado de archivos PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: false

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

Esta opción es obligatoria si el archivo PDF de entrada está cifrado. Se utiliza para desbloquear los datos de entrada sin eliminar el cifrado. Es preferible la clave del propietario.

Valor por defecto: null

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

Especifica el algoritmo de encriptación de la salida.

Valor predeterminado: EncryptionAlgorithm.preserve

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

Si linearize es verdadero: el PDF de salida está linealizado.

Si linearize es falso: el PDF de salida no está linealizado.

Si linearize es nulo: la salida solo se linealiza si la PDF de entrada ya se ha linealizado.

Valor por defecto: null

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

Valor por defecto: se especifica en el archivo de configuración.

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

Esta opción se utiliza para el desarrollo y las pruebas. No debe utilizarse en un entorno de producción.

Valor por defecto: false

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

La contraseña de propietario para el cifrado de archivos PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: null

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

Se utiliza para gestionar los mensajes que normalmente se escriben en stderr. Valor por defecto: el controlador integrado, tal y como se especifica en el archivo de configuración.

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

Se utiliza para gestionar los mensajes que normalmente se escriben en stdout.

Valor por defecto: el controlador integrado, tal y como se especifica en el archivo de configuración.

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

Solo lo utiliza el comando `<Rastreo>`.

Valor por defecto: el controlador integrado, tal y como se especifica en el archivo de configuración.

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

Esta opción se utiliza para el desarrollo y las pruebas. No debe utilizarse en un entorno de producción.

Valor por defecto: `false`

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

La contraseña de usuario para el cifrado del PDF. No tiene ningún efecto si el archivo de salida no ha sido cifrado mediante PDF2PDFUA.

Valor por defecto: null

interfaz PdfUaOutputHandler

Paquete: **com.ricoh.PDF2PDFUA.api**

Manejador para la redirección de E/S.

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

`src`: el nodo XML de la regla del archivo de control que realiza la llamada. Nulo si no es posible rastrear el mensaje hasta un nodo XML.

`pageNo`: el número de página comienza por 1. Este valor es cero si no es posible asociar el mensaje a una página.

`msg`: puede contener varias líneas de texto.

clase PreprocessingException extiende Exception

Paquete: **com.ricoh.PDF2PDFUA.api**

Clase base para los errores que se producen durante el preprocesamiento del archivo de control.

clase UaStackTrace extiende ArrayList<CADENA>

Paquete: **com.ricoh.PDF2PDFUA.api**

Un seguimiento de la pila que apunta a un elemento XML del archivo de control.

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

Ejemplos

En este ejemplo se muestra cómo analizar un archivo de control y cómo aplicarlo a varios documentos. Se supone que todos los documentos de un lote deben ser similares y compatibles con el archivo de control. Por motivos de rendimiento, el archivo de control se analiza una sola vez.

Se prevé que el software funcione en servidores con varios núcleos de CPU. Este ejemplo es multihilo, ya que se trata del caso de uso estándar.

```
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);
    }
```

```
}
```

Este ejemplo ilustra cómo puede funcionar PDF2PDFUA sin utilizar archivos. Esto resulta útil cuando PDF2PDFUA se integra en otros sistemas, como las aplicaciones 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 Herramienta PDF2PDFUA Guía de usuario