

# PDF Tool API V7.0 コマンドライン説明書

アンテナハウス株式会社



## 目 次

第1章 コマンドラインについて .....	1
第2章 動作環境 .....	2
2.1 Windows .....	2
2.2 Linux .....	2
2.3 Amazon Linux2 X86 .....	2
第3章 インストール方法 .....	3
3.1 Windows 版 .....	3
3.1.1 インストール方法 .....	3
3.1.2 アンインストール方法 .....	3
3.1.3 Microsoft Visual C++ ランタイムライブラリについて .....	3
3.1.4 インストールフォルダの構成 .....	4
3.1.5 インストーラによりシステムに設定される内容 .....	4
3.1.6 インストールされるライセンスファイルについて .....	4
3.2 Linux / Amazon Linux2 X86 版 .....	5
3.2.1 インストール方法 .....	5
3.2.2 アンインストール方法 .....	5
3.2.3 インストールフォルダの構成 .....	6
3.2.4 インストールされるライセンスファイルについて .....	6
第4章 コマンドラインのファイル名と依存関係モジュール名 .....	7
4.1 Windows 版 .....	7
4.2 Linux / Amazon Linux2 X86 版 .....	7
第5章 ライセンスファイルについて .....	8
5.1 ライセンスファイルの入手 .....	8
5.2 ライセンスファイルの配置 .....	8
5.2.1 配置方法 1 .....	8
5.2.2 配置方法 2 (Windows 版のみ) .....	8
5.3 ライセンスファイルの入れ替えについて .....	8
5.4 保守期間とコマンドラインの動作について .....	8
第6章 フォントの扱い .....	9
6.1 フォント情報の取得 .....	9
6.2 フォント構築ファイルの設定 .....	9
6.3 環境変数の作成 .....	9
6.4 サポートしているフォントの種類 .....	9

6.5 フォントの埋め込み .....	9
6.6 フォントの統合 .....	10
第7章 処理対象 PDF ファイルについて .....	11
第8章 制限事項 .....	12
第9章 評価版動作仕様 .....	13
第10章 コマンドラインの使い方 .....	14
10.1 Windows 版 .....	14
10.2 Linux / Amazon Linux2 X86 版 .....	14
第11章 コマンドラインの仕様 .....	15
11.1 入出力の指定方法 .....	15
11.1.1 ファイル指定 .....	15
11.1.2 フォルダ指定 .....	15
11.2 コマンドの種類 .....	17
11.3 各コマンドの説明 .....	18
11.3.1 -getFileInfo : PDF の情報取得 .....	18
11.3.2 -changeDocInfo : 文書プロパティ設定 .....	19
11.3.3 -changeViewInfo : 開き方設定 .....	20
11.3.4 -mergeFiles : PDF ファイル結合 .....	22
11.3.5 -getOutlineInfo : しおり情報取得 .....	23
11.3.6 -getAnnotInfo : 注釈情報取得 .....	24
11.3.7 -setPdfWatermark : PDF ファイル透かし設定 .....	25
11.3.8 -setImageWatermark : 画像ファイル透かし設定 .....	27
11.3.9 -setTextWatermark : テキスト透かし設定 .....	29
11.3.10 -setColorWatermark : 色透かし設定 .....	31
11.3.11 -encrypt : PDF セキュリティ設定 .....	33
11.3.12 -decrypt : PDF セキュリティ削除 .....	35
11.3.13 -optimize : PDF の最適化 .....	36
11.3.14 -addEmbeddedFile : ファイル添付 .....	37
11.3.15 -pageRotate : ページ回転 .....	38
11.3.16 -pageDelete : ページ削除 .....	39
11.3.17 -writeText : 文字列追加 .....	40
11.3.18 -exportAnnotsToFDF : 注釈の FDF への出力 .....	42
11.3.19 -importAnnotsFromPDF : 注釈の取り込み .....	43
11.3.20 -insertEmptyPage : 空白ページ挿入 .....	44
11.3.21 -addPageNumber : ページ番号挿入 .....	45
11.3.22 -pageExtract : ページ抽出 .....	47
11.3.23 -embedFonts : フォント埋め込み .....	48
11.3.24 -extractImage : 画像抽出 .....	49

11.3.25	-extractText : テキスト抽出	50
11.3.26	-imageToPdf : 画像ファイルのPDF化	51
11.3.27	-extractEmbeddedFile : 添付ファイルの抽出	52
11.3.28	-importAnnotsFromFDF : FDFからの注釈の取り込み	53
11.3.29	-fixedUpPDFA : PDF/A-1b、-2b変換	54
11.3.30	-validatePDFA : PDF/A-1b、-2b準拠確認	55
11.3.31	-dividePdf : ファイル分割	56
11.3.32	-linearize : Web表示用に最適化	57
11.3.33	-lic : ライセンス情報の取得	57
11.3.34	-h : コマンド名一覧表示	57
11.3.35	-h -all : 全コマンドオプション説明の表示	58
11.3.36	-h [コマンド名] : 個別コマンドオプション説明の表示	58
11.3.37	取得情報の出力内容	59
11.3.37.1	-getFileInfo : PDFの情報取得	59
11.3.37.2	-getOutlineInfo : しおり情報取得	61
11.3.37.3	-getAnnotInfo : 注釈情報取得	62
11.3.37.4	Action情報・Destination情報	64
11.4	コマンドの複数指定	65
11.5	マクロ文字列	67
11.6	エラー処理について	68
11.6.1	コマンドラインの戻り値	68
11.6.2	エラーの内容について	68
11.6.3	PdfTkのエラーコード一覧	69
<b>第12章</b>	<b>V4.0 / V5.0 / V6.0 との相違点</b>	<b>72</b>
12.1	対応コマンド	72
12.2	PDFの情報取得 (-getFileInfo) の変更点	73
12.3	PDFセキュリティ設定 (-encrypt) の変更点	73
12.4	フォルダ指定	73
12.5	マクロ文字列	73
<b>第13章</b>	<b>商標／著作権情報</b>	<b>74</b>
13.1	商標	74
13.2	第三者ライブラリー著作権情報	74
<b>履歴</b>		<b>77</b>



## 第1章 コマンドラインについて

Antenna House PDF Tool API V7.0 コマンドラインは、コマンドとオプションを指定して実行するしくみです。

処理内容は限定されますが、プログラム言語やPDFの仕様についての知識がほとんどない場合でも、開発作業に多くの時間を割くことなく、PDFファイルの情報取得や加工・編集処理の実行が可能です。

PHP、Python、Ruby など、実行ファイルの呼び出しと制御ができるプログラム言語からもかんたんに利用できます。

コマンドライン仕様は、PDF Tool API V7.0 のすべての機能には対応していません。

## 第2章 動作環境

### 2.1 Windows

OSの種類	Microsoft Windows 11 / 10 日本語版 Microsoft Windows Server 2022 日本語版 Windows 10 S モードには対応していません。
必要なランタイムライブラリ	Microsoft Visual C++ 2019 ランタイムライブラリ

※ランタイムライブラリの入手先：最新のサポートされている Microsoft Visual C++再頒布可能パッケージ

「Visual Studio 2015、2017、2019、および 2022」項にある「アーキテクチャ」が「X86」または「X64」をダウンロードしインストールしてください。

32bit 版のコマンドラインを使用するときは「X86」、64bit 版のコマンドラインを使用するときは「X64」のライブラリが必要です。

### 2.2 Linux

OSの種類	X86 64bit
必要なランタイムライブラリ	glibc-2.28 libstdc++.so.6.0.25 もしくはこれらと互換のあるライブラリ ※GCC 8.3.1 でビルドされています。

### 2.3 Amazon Linux2 X86

OSの種類	Amazon Linux2 X86 64bit
必要なランタイムライブラリ	glibc 2.26.amzn2 ※GCC 7.3.1-15.amzn2 でビルドされています。

## 第3章 インストール方法

### 3.1 Windows 版

#### 3.1.1 インストール方法

[コマンドライン版の場合]

- 1) setup-cmd.exe をダブルクリックするなどして起動します。
- 2) ダイアログにしたがってインストールを実行します。デフォルトのインストール先フォルダパスは次の場所です。  
{システムドライブ}\Program Files\Antenna House\PDFToolAPI\_V7\_cmd
- 3) コマンドラインは、インストール後 30 日間有効な評価版としてインストールされます。

[ライブラリ版の場合]

- 1) setup-lib.exe をダブルクリックするなどして起動します。
- 2) ダイアログにしたがってインストールを実行します。デフォルトのインストール先フォルダパスは次の場所です。  
{システムドライブ}\Program Files\Antenna House\PDFToolAPI\_V7\_lib
- 3) コマンドラインは、インストール後 30 日間有効な評価版としてインストールされます。

#### 3.1.2 アンインストール方法

[コマンドライン版の場合]

- 1) 「コントロールパネル-プログラム-プログラムと機能」を開きます。
- 2) プログラム一覧にある「Antenna House PDF Tool API V7.0 Windows コマンドライン」を選択します。
- 3) 「アンインストール」をクリックしてアンインストールを実行します。

[ライブラリ版の場合]

- 1) 「コントロールパネル-プログラム-プログラムと機能」を開きます。
- 2) プログラム一覧にある「Antenna House PDF Tool API V7.0 Windows ライブラリ」を選択します。
- 3) 「アンインストール」をクリックしてアンインストールを実行します。

#### 3.1.3 Microsoft Visual C++ ランタイムライブラリについて

PDF Tool API V7.0 の動作には、Microsoft Visual C++ 2019 ランタイムライブラリが必要です。

setup-cmd.exe / setup-lib.exe は、動作環境にランタイムライブラリが存在しないと判定したとき、Microsoft Visual C++ 2019 再頒布パッケージのインストーラを起動します。

再頒布パッケージのインストーラが起動した場合は、表示されるダイアログにしたがって、インストールを行ってください。

### 3.1.4 インストールフォルダの構成

[コマンドライン版]

インストールフォルダ

|-- bin32 : コマンドラインモジュール (32bit 版)  
|-- bin64 : コマンドラインモジュール (64bit 版)  
|-- fontconfig : フォント構築ファイル  
|-- License : ライセンスファイル  
|-- AHPDFToolAPI\_EULA.txt : 使用許諾契約書

[ライブラリ版]

インストールフォルダ

|-- bin32 : ライブラリモジュール (32bit 版) : コマンドラインを含みます  
|-- bin64 : ライブラリモジュール (64bit 版) : コマンドラインを含みます  
|-- lib32 : lib ファイル (32bit 版) : コマンドラインでは使用しません  
|-- lib64 : lib ファイル (64bit 版) : コマンドラインでは使用しません  
|-- Include : ヘッダファイル : コマンドラインでは使用しません  
|-- fontconfig : フォント構築ファイル  
|-- License : ライセンスファイル  
|-- AHPDFToolAPI\_EULA.txt : 使用許諾契約書

### 3.1.5 インストーラによりシステムに設定される内容

以下の環境変数の作成と値の設定が行われます。

環境変数名 : PTL70\_LIC\_PATH

設定値 : {インストールフォルダ}\License

### 3.1.6 インストールされるライセンスファイルについて

- インストーラによりインストールされるライセンスファイル「ptalic.dat」は、インストール後 30 日間有効の評価ライセンスです。
- PDF Tool API の処理により出力される PDF ファイルの各ページには、透かし文字列が挿入されます。
- 期限を過ぎると、PDF Tool API は利用できません。

## 3.2 Linux / Amazon Linux2 X86 版

### 3.2.1 インストール方法

[コマンドライン版]

- 1) スーパーユーザーでログインします。
- 2) rpm コマンドを実行します。  
`rpm -i setup-cmd.rpm [--prefix インストールパス]`
- 3) インストールパスを指定しなかった場合、「/usr/AHPDFToolV7-cmd」にインストールされます。
- 4) コマンドラインは、インストール後 30 日間有効な評価版としてインストールされます。

※ Amazon Linux2 X86 にインストールする場合は、「{製品データ}\Setup-AmazonLinux2(X86)」の「setup-cmd\_amzn2.rpm.rpm」をご利用ください。

[ライブラリ版]

- 1) スーパーユーザーでログインします。
- 2) rpm コマンドを実行します。  
`rpm -i setup-lib.rpm [--prefix インストールパス]`
- 3) インストールパスを指定しなかった場合、「/usr/AHPDFToolV7-lib」にインストールされます。
- 4) コマンドラインは、インストール後 30 日間有効な評価版としてインストールされます。

※ Amazon Linux2 X86 にインストールする場合は、「{製品データ}\Setup-AmazonLinux2(X86)」の「setup-lib\_amzn2.rpm.rpm」をご利用ください。

### 3.2.2 アンインストール方法

[コマンドライン版]

- 1) スーパーユーザーでログインします。
- 2) rpm コマンドを実行します。  
`rpm -e AHPDFToolV7Cmd`

[ライブラリ版]

- 1) スーパーユーザーでログインします。
- 2) rpm コマンドを実行します。  
`rpm -e AHPDFToolV7Lib`

### 3.2.3 インストールフォルダの構成

[コマンドライン版]

インストールフォルダ

|-- ToolCmd : コマンドライン実行ファイル  
|-- lib : PDF Tool API モジュール  
|-- fontconfig : フォント構築ファイル  
|-- License : ライセンスファイル  
|-- run.sh : コマンドライン実行スクリプト  
|-- AHPDFToolAPI7\_EULA.txt : 使用許諾契約書

[ライブラリ版]

インストールフォルダ

|-- ToolCmd : コマンドライン実行ファイル  
|-- lib : PDF Tool API モジュール  
|-- bin : Java API 用ファイル : コマンドラインでは使用しません  
|-- fontconfig : フォント構築ファイル  
|-- License : ライセンスファイル  
|-- run.sh : コマンドライン実行スクリプト  
|-- AHPDFToolAPI7\_EULA.txt : 使用許諾契約書

### 3.2.4 インストールされるライセンスファイルについて

- インストーラによりインストールされるライセンスファイル「ptalic.dat」は、評価ライセンスです。インストール後、PDF Tool API を初めて実行した日から 30 日間有効です。
- PDF Tool API の処理により出力される PDF ファイルの各ページには、透かし文字列が挿入されます。
- 期限を過ぎると、PDF Tool API は利用できません。

## 第4章 コマンドラインのファイル名と依存関係モジュール名

### 4.1 Windows 版

コマンドラインファイル名	依存関係にあるファイル
AHPDFToolCmd70.exe	PdfTk70.dll PdfTkEx70.dll PtkAHCommon15.dll PtkAHDMC15.dll PtkAHFontService17.dll PtkAHGraphicService11.dll PtkAHPDFEditLib10.dll PtkAHPDFLib40.dll PtkPDFLinearizer60.dll PtkAHCertificate.dll PtkAHPDFFixUp11.dll icudt52.dll icuin52.dll icuio52.dll icule52.dll iculx52.dll icutu52.dll icuuc52.dll

### 4.2 Linux / Amazon Linux2 X86 版

コマンドラインファイル名	依存関係にあるファイル
AHPDFToolCmd70	libPdfTk.so.7.0 libPdfTkEx.so.7.0 libPtkAHCertificate.so.1.0 libPtkAHCommon.so.1.5 libPtkAHDMC.so.1.5 libPtkAHFontService.so.1.7 libPtkAHGraphicService.so.1.1 libPtkAHPDFEditLib.so.1.0 libPtkAHPDFFixUpLib.so.1.1 libPtkAHPDFLib.so.4.0 libPtkPDFLinearizer.so.6.0 libicudata.so.52.1 libicui18n.so.52.1 libicuio.so.52.1 libicule.so.52.1 libiculx.so.52.1 libicutu.so.52.1 libicuuc.so.52.1

## 第5章 ライセンスファイルについて

### 5.1 ライセンスファイルの入手

PDF Tool API を新規にご購入で弊社より CD メディアをお送りしている場合、CD メディア内の「License」フォルダにライセンスファイルが格納されています。

電子納品の場合、ライセンスファイルは別途、弊社よりお送りします。

保守契約を更新された場合、弊社より新しいライセンスファイルを送付いたします。

### 5.2 ライセンスファイルの配置

ライセンスファイルの配置方法には2種類あります。どちらかの方法で配置してください。

#### 5.2.1 配置方法 1

ライセンスファイル (ptalic.dat) を任意の場所に配置します。配置したフォルダパスを下記の環境変数に設定します。

環境変数名：PTL70\_LIC\_PATH

設定値：ライセンスファイルを配置したフォルダパス

#### 5.2.2 配置方法 2 (Windows 版のみ)

Windows 版の場合は、ライセンスファイルを PDF Tool API V7.0 のモジュールファイル「PdfTk70.dll」と同一の場所に配置して使用することも可能です。このとき、環境変数「PTL70\_LIC\_PATH」は必要ありません。

### 5.3 ライセンスファイルの入れ替えについて

評価利用から正規利用にする場合や保守契約を更新した場合、ライセンスファイルを入れ替えます。

ライセンスファイルの入れ替えは、PDF Tool API を利用するプログラムが実行されていないときに行ってください。

ライセンスファイルを入れ替えた後、コンピュータの再起動は必要ありません。

### 5.4 保守期間とコマンドラインの動作について

ライセンスファイルには、保守契約期間情報があります。

保守契約が有効である間にリリースされた改訂版については、アップデートしてご利用いただけます。

保守期間終了後も、それまでと同様にご利用いただけます。

ただし、保守契約終了後にリリースされた改訂版にアップデートしてご利用になると、コマンドラインは評価版として動作します。このため、出力される PDF の各ページに評価用の透かし文字列が挿入されます。

最新の改訂版を正規利用するには、保守契約の更新をご検討ください。

## 第6章 フォントの扱い

テキスト透かしの挿入や、PDF ページのコンテンツに文字列を描画する処理を行うにはフォント情報が必要です。

### 6.1 フォント情報の取得

- Windows 版においては、システムのフォントフォルダ（通常、「{システムドライブ}:\WINDOWS\Fonts」）に存在するフォントを参照します。
- フォントフォルダとは別の場所にあるフォントを使用する場合は、「フォント構築ファイル」を設定します。
- Linux / Amazon Linux2 X86 版においては、「フォント構築ファイル」の設定は必須です。

### 6.2 フォント構築ファイルの設定

フォント情報を特定のフォルダから取得する場合、フォント構築ファイルを設定します。  
フォント構築ファイルは、以下のファイルです。

インストールフォルダ\fontconfig\font-config.xml

フォントファイルが存在するフォルダパスを、「font-folder path」タグに記述します。

font-config.xml 記述例 (Windows 版)

```
<font-config>
  <font-folder path=" C:\TestFont" ></font-folder>
</font-config>
```

### 6.3 環境変数の作成

フォント構築ファイルの設定を反映させるためには、環境変数を作成し値を設定します。

環境変数名： PTL70\_FONT\_CONFIGFILE

設定する値： font-config.xml のフルパス

### 6.4 サポートしているフォントの種類

- TrueType フォント (Unicode cmap を持つもの)
- OpenType フォント (Unicode cmap を持つもの)
- Type1 フォント

### 6.5 フォントの埋め込み

PDF Tool API での処理時にフォントの埋め込みを指示した場合にフォント情報が PDF ファイルに埋め込まれます。  
ただし、次のフォントについては、埋め込み指示があってもフォント情報は埋め込まれません。

- TrueType、OpenType：フォント埋め込みが禁止されたフォント
- Type1：PFB ファイルが存在しないフォント

## 6.6 フォントの統合

- フォント統合は、PDF に埋め込まれているフォントに対して行われます。
- 次のフォントが統合対象です。
  - TrueType、Type1(CFF 形式のみ)、CIDFontType2、CIDFontType0
- 同じフォントタイプであっても Font 辞書の値 (Ascent, CapHeight など) が異なる場合は統合の対象となりません。
- TrueType フォント、CIDFontType2 フォントに対しては、cmap、head、hhea、hmtx、maxp、cvt、fpgm、prep の各テーブルが同一である場合のみ統合の対象となります。

## 第7章 処理対象 PDF ファイルについて

PDF Tool API V7.0 の処理対象 PDF ファイルは、「ISO 32000-2」に準拠した PDF ファイルです。

## 第8章 制限事項

- 入力ファイルとして読み込み (load) 可能な PDF ファイルのファイルサイズは 2GB までです。2GB を超える PDF ファイルの場合は load 時にエラーになります。
- PDF Tool API で保存される PDF のファイルサイズの上限は 2GB です。2GB を超える場合は保存時にエラーになります。※1
- PDF Tool API で取扱い可能な PDF ファイルは、「ISO 32000-2」の仕様に準拠したファイルです。
- 任意の PDF バージョン番号を設定することはできません。
- ダウンサンプリングを行ったとき、カラースペースが元のデータとは異なる場合があります。
- PDF/A-1b、PDF/A-2b、PDF/X の規格を持つファイルに対して編集を行った場合、これらの規格情報は破棄されます。
- ビットマップフォント、Type3 フォントには対応していません。
- 縦書きではイタリックの指定があってもフォントがグリフを持っていない場合はイタリックにはなりません。
- 複数のスレッドでの同時利用はできません。
- Windows においては、コマンドプロンプトに文字数の制限があります。コマンドライン利用時ご注意ください。  
参照：コマンド プロンプト (Cmd. exe) コマンドライン文字列の制限

※1 出力ファイルサイズの上限判定は、PDF Tool API で PDF の加工や編集を行い保存する結果が 2GB を超えるかどうかです。たとえば PDF ファイルを結合する場合、PDF Tool API では物理的に PDF ファイルを結合する処理ではないため、結合前の PDF ファイルサイズの合計が 2GB 未満であっても保存できない場合があります。

## 第9章 評価版動作仕様

- Windows 版ではインストール後 30 日間、Linux / Amazon Linux2 X86 版ではインストール後 PDF Tool API を初めて実行した日から 30 日間、使用できます。期間を過ぎると使用できません。
- コマンドラインから出力される PDF ファイルの各ページに、透かし文字列「Antenna House PDF Tool」が挿入されます。

## 第 10 章 コマンドラインの使い方

### 10.1 Windows 版

コマンドラインアプリケーションは、次のように引数を指定して実行します。

```
AHPDFToolCmd70.exe [コマンド] [パラメータ] -d (入力ファイル/フォルダパス) -o (出力ファイル/フォルダパス)
```

パラメータを持つコマンドの場合、「コマンド」指定後に必ず「パラメータ」を指定してください。

### 10.2 Linux / Amazon Linux2 X86 版

コマンドラインアプリケーションは、次のように引数を指定して実行します。

```
AHPDFToolCmd70 [コマンド] [パラメータ] -d (入力ファイル/フォルダパス) -o (出力ファイル/フォルダパス)
```

Linux / Amazon Linux2 X86 版には、コマンドラインアプリケーションを実行するシェルスクリプト「run.sh」が付いています。シェルスクリプトを使用する場合、次のように引数を指定して実行します。

```
sh run.sh [コマンド] [パラメータ] -d (入力ファイル/フォルダパス) -o (出力ファイル/フォルダパス)
```

パラメータを持つコマンドの場合、「コマンド」指定後に必ず「パラメータ」を指定してください。

## 第 11 章 コマンドラインの仕様

### 11.1 入出力の指定方法

#### 11.1.1 ファイル指定

- 入出力をファイルパスで指定します。PDF の情報取得など出力を伴わないコマンドでは、出力ファイルパスの指定は不要です。

[指定方法]

```
-d(入力ファイルパス) [パスワード] -o (出力ファイルパス) [-linearize]
```

- ファイルパスにスペースが含まれる場合は、ダブルクォーテーション (") で囲んでください。
- 入力ファイルに対しパスワード文字列を指定すると、セキュリティ付き PDF に対しても処理が可能です。文書を開くパスワードと権限パスワードの 2 種類が付与されている場合は、権限パスワードを指定します。
- 入力ファイルのセキュリティ情報は、「PDF ファイル結合」、「PDF セキュリティ設定」、「PDF セキュリティ削除」、「ページ抽出」を除くコマンドにおいて、出力 PDF ファイルに引き継がれます。
- 出力ファイルパスの後ろに「-linearize」を指定すると、「Web 表示用に最適化」された PDF として保存されます。
- 出力先には、存在するフォルダ／ディレクトリのパスを指定してください。パスが見つからない場合、エラーとなります。

[指定例]

<Windows>

```
-d "c:\test\test 1.pdf" ownerpass -o c:\sav\out.pdf -linearize
```

<Linux / Amazon Linux2 X86>

```
-d /home/antenna/test/test.pdf ownerpass -o /home/antenna/sav/out.pdf -linearize
```

#### 11.1.2 フォルダ指定

- 入出力をフォルダ／ディレクトリパスで指定します。PDF の情報取得など出力を伴わないコマンドでは、出力フォルダ／ディレクトリパスの指定は不要です。

[指定方法]

```
-d(入力フォルダ／ディレクトリパス) [パスワード] -o (出力フォルダ／ディレクトリパス) [-linearize]
```

- フォルダ／ディレクトリパスにスペースが含まれる場合は、ダブルクォーテーション (") で囲んでください。
- 指定された入力フォルダ／ディレクトリにある PDF ファイル、あるいは画像ファイルが処理対象となります。サブフォルダ／ディレクトリ内のファイルは処理対象ではありません。
- 入出力については、各コマンドの説明をご参照ください。
- 一部のコマンドはフォルダ／ディレクトリ指定に対応していません。
- 出力先には、存在するフォルダ／ディレクトリのパスを指定してください。パスが見つからない場合、エラーとなります。

[指定例]

<Windows>

-d "c:\in 2" ownerpass -o c:\out -linearize

<Linux / Amazon Linux2 X86>

-d /home/antenna/in ownerpass -o /home/antenna/out -linearize

## 11.2 コマンドの種類

	コマンド	機能	フォルダ指定対応の有無	他のコマンドとの組み合わせ (p. 65)
1	-getFileInfo	PDF の情報取得	○	×
2	-changeDocInfo	文書プロパティ設定	○	○
3	-changeViewInfo	開き方設定	○	○
4	-mergeFiles	PDF ファイル結合	×	○
5	-getOutlineInfo	しおり情報取得	○	×
6	-getAnnotInfo	注釈情報取得	○	×
7	-setPdfWatermark	PDF ファイル透かし設定	○	○
8	-setImageWatermark	画像ファイル透かし設定	○	○
9	-setTextWatermark	テキスト透かし設定	○	○
10	-setColorWatermark	色透かし設定	○	○
11	-encrypt	PDF セキュリティ設定	○	○
12	-decrypt	PDF セキュリティ削除	○	○
13	-optimize	PDF の最適化	○	○
14	-addEmbeddedFile	ファイル添付	○	○
15	-pageRotate	ページ回転	×	○
16	-pageDelete	ページ削除	×	○
17	-writeText	文字列追加	○	○
18	-exportAnnotsToFDF	注釈の FDF への出力	○	○
19	-importAnnotsFromPDF	注釈の取り込み	×	○
20	-insertEmptyPage	空白ページ挿入	×	○
21	-addPageNumber	ページ番号挿入	○	○
22	-pageExtract	ページ抽出	×	○
23	-embedFonts	フォント埋め込み	○	○
24	-extractImage ※V7.0 新機能	画像抽出	○	○
25	-extractText ※V7.0 新機能	テキスト抽出	○	○
26	-imageToPdf ※V7.0 新機能	画像ファイルの PDF 化	○	×
27	-extractEmbeddedFile ※V7.0 新機能	添付ファイルの抽出	○	○
28	-importAnnotsFromFDF ※V7.0 新機能	FDF からの注釈の取り込み	×	○
29	-fixedUpPDFA ※V7.0 改訂 7a 版新機能	PDF/A-1b、PDF/A-2b 変換	○	×
30	-validatePDFA ※V7.0 改訂 7a 版新機能	PDF/A-1b、PDF/A-2b 準拠確認	○	×
31	-dividePdf ※V7.0 改訂 7a 版新機能	ファイル分割	×	×
32	-linearize	Web 表示用に最適化	○	○
33	-lic	ライセンス情報の取得	×	×
34	-h	コマンド名一覧表示	-	-
35	-h -all	全コマンドオプション説明	-	-
36	-h {コマンド名}	個別コマンドオプション説明	-	-

## 11.3 各コマンドの説明

### 11.3.1 -getFileInfo : PDF の情報取得

#### ■コマンド例

[処理例]

Test.pdf の PDF ファイル情報を標準出力に出力する。

[Windows]

```
AHPDFToolCmd70.exe -getFileInfo -d C:\test\test.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -getFileInfo -d /home/antenna/test/test.pdf
```

#### ■処理内容

PDF ファイルから、

- 文書プロパティ情報
- 開き方情報
- セキュリティ設定の有無
- セキュリティ設定がある場合はその詳細情報
- 署名の有無
- PDF/A かどうか
- PDF/X かどうか

を取得し、標準出力に出力します。出力ファイル指定 (-o) は必要ありません。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルの情報を取得し標準出力に出力します。出力フォルダ指定 (-o) は必要ありません。

#### ■パラメータ

パラメータはありません。

#### ■出力内容

『PDF の情報取得』の出力内容 (p. 59)」を参照してください。

### 11.3.2 -changeDocInfo : 文書プロパティ設定

#### ■コマンド例

[処理例]

test.pdf の文書情報「タイトル」に「PDF Tool API テスト」、「作成者」に「アンテナハウス株式会社」を設定し out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -changeDocInfo -title "PDF Tool API テスト" -author アンテナハウス株式会社 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -changeDocInfo -title "PDF Tool API テスト" -author アンテナハウス株式会社 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

文書プロパティ情報を設定します。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに文書プロパティ情報を設定します。入力ファイル名と同一名で出力フォルダに出力します。

#### ■パラメータ

パラメータ	内容
-title Val	「タイトル」の設定。省略可。 指定がない場合、元の内容を引き継ぐ。
-subject Val	「サブタイトル」の設定。省略可。 指定がない場合、元の内容を引き継ぐ。
-author Val	「作成者」の設定。省略可。 指定がない場合、元の内容を引き継ぐ。
-keywords Val	「キーワード」の設定。省略可。 指定がない場合、元の内容を引き継ぐ。
-creator Val	「アプリケーション」の設定。省略可。 指定がない場合、元の内容を引き継ぐ。
-producer Val	「PDF 変換」の設定。省略可。 指定がない場合、元の内容を引き継ぐ。

### 11.3.3 -changeViewInfo : 開き方設定

#### ■コマンド例

[処理例]

test.pdf に対し PDF ファイルを開いたとき、  
連続ページ (=スクロールが有効) / ページサムネールパネル表示 / ツールバーは非表示  
の状態が表示されるよう設定し out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -changeViewInfo -pageLayout 2 -pageMode 3 -hideToolbar true -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -changeViewInfo -pageLayout 2 -pageMode 3 -hideToolbar true -d /home/antenna/test/test.pdf -o /home/  
antenna/sav/out.pdf
```

#### ■処理内容

PDF の開き方を設定します。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに開き方を設定します。入力ファイル名と同一名で出力フォルダに出力します。

#### ■パラメータ

パラメータ	内容
-direction Val	読み上げオプション (縦じ方) の設定。省略可。 1: 左から右 (左縦じ) 2: 右から左 (右縦じ) 指定がない場合、変更なし。 1,2 以外が指定された場合、エラーとなる。
-pageLayout Val	ページレイアウトの設定。省略可。 1: 単一ページ 2: 連続ページ 3: 連続見開きページ 4: 連続見開きページ (表紙) 5: 見開きページ 6: 見開きページ (表紙) 指定がない場合、変更なし。 1~6 以外が指定された場合、エラーとなる。
-pageMode Val	表示状態の設定。省略可。 1: ページのみ 2: しおりパネルとページ 3: ページパネルとページ 5: レイヤーパネルとページ 6: 添付ファイルパネルとページ 指定がない場合、変更なし。 1~3,5,6, 以外が指定された場合、エラーとなる。 -fullScreen true 指定時 -pageMode は無視される。
-fitWindow Val	ページにウィンドウサイズを合わせる。省略可。 true: 合わせる false: 合わせない 指定がない場合、変更なし。
-centerWindow Val	ウィンドウを画面中央に配置。省略可。 true: 配置する false: 配置しない 指定がない場合、変更なし。
-fullScreen Val	フルスクリーンモードで開く。省略可。 true: フルスクリーンで開く false: フルスクリーンでは開かない 指定がない場合、変更なし。 -fullScreen true 指定時 -pageMode が無視される。

-displayTitle Val	ウィンドウのタイトルバーに文書プロパティの「タイトル」を表示する。省略可。 true: 表示する false: 表示しない 指定がない場合、変更なし。
-hideToolbar Val	ツールバーを非表示。省略可。 true: 非表示 false: 表示 指定がない場合、変更なし。
-hideMenubar Val ※	メニューバーを非表示。省略可。 true: 非表示 false: 表示 指定がない場合、変更なし。
-hideWindowUI Val ※	ウィンドウコントロールを非表示。省略可。 true: 非表示 false: 表示 指定がない場合、変更なし。
-viewFit Val	倍率設定。省略可。 1: 100% 2: 全体表示 3: 幅に合わせる 4: 高さに合わせる 5: 描画領域の幅に合わせる 指定がない場合、変更なし。 1~5以外が指定された場合、エラーとなる。
- defaultPageNo Val	最初に関くページ番号設定。省略可。 1ページ目を「0」とする。 指定がない場合、変更なし。

※ 「-hideMenubar true -hideWindowUI true」を設定した PDF ファイルを Acrobat/Adobe Reader で表示した場合、メニューバーは表示されます。

### 11.3.4 -mergeFiles : PDF ファイル結合

#### ■コマンド例

[処理例]

test1.pdf、test2.pdf を結合し Web 表示用に最適化して out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -mergeFiles -d C:\test\test1.pdf -d C:\test\test2.pdf userpass "0,9-14" -o C:\sav\out.pdf -linearize
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -mergeFiles -d /home/antenna/test/test1.pdf -d /home/antenna/test/test2.pdf userpass "0,9-14" -o /home/antenna/sav/out.pdf -linearize
```

#### ■処理内容

複数の PDF ファイルをひとつのファイルに結合します。

ページ番号やページ範囲（ページ）を指定し、任意のページを結合することができます。

注釈やしおりのタイトルは保持されます。ただし、結合するページを指定した場合、そのファイルのしおりは保持されません。

1 番目に指定された入力ファイルにページ指定がない場合、1 番目の入力ファイルの文書情報が出力 PDF ファイルに引き継がれます。

1 番目に指定された入力ファイルにページ指定がある場合、出力 PDF ファイルの文書情報の値は空になります。

入力ファイルのセキュリティ情報は、出力 PDF ファイルには引き継がれません。

#### ※[改訂 8] 新機能

- ファイルに記載された PDF ファイルを結合するパラメータを追加しました。
- 結合時に注釈をコピーするか否かのパラメータを追加しました。
- 結合時にしおりをコピーするか否かのパラメータを追加しました。

#### ■フォルダ指定：対応無し

#### ■パラメータ

パラメータ	内容
-dmerge Val ※改訂 8 にて追加	[Val] に指定されたテキストファイルに記述されている PDF ファイルを先頭行から順に結合する。 [Val] にはフルパス名を指定する。 「-d」と同時に指定された場合、エラーになる。 テキストファイルの記述方法：1 行に 1 ファイルパスを記述する。文字コードは「UTF-8」を使用する。
-copyAnnots Val ※改訂 8 にて追加	ファイル結合時に注釈をコピーするか否かを指定する。 true：コピーする false：コピーしない 指定がない場合はコピーする。
-copyOutlines Val ※改訂 8 にて追加	ファイル結合時にしおりをコピーするか否かを指定する。 true：コピーする false：コピーしない 指定がない場合はコピーする。ただし、結合するページを指定した場合はコピーしない。

### 11.3.5 -getOutlineInfo : しおり情報取得

#### ■コマンド例

[処理例]

test.pdf のしおり情報を標準出力に出力する。

[Windows]

```
AHPDFToolCmd70.exe -getOutlineInfo -d C:\test\test.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -getOutlineInfo -d /home/antenna/test/test.pdf
```

#### ■処理内容

しおり情報を標準出力に出力します。出力ファイル指定 (-o) は必要ありません。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルのしおり情報を取得し標準出力に出力します。出力フォルダ指定 (-o) は必要ありません。

#### ■パラメータ

パラメータはありません。

#### ■出力内容

『しおり情報取得』の出力内容 (p. 61)」を参照してください。

### 11.3.6 -getAnnotInfo : 注釈情報取得

#### ■コマンド例

[処理例]

test.pdf の注釈情報を標準出力に出力する。

[Windows]

```
AHPDFToolCmd70.exe -getAnnotInfo -d C:\test\test.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -getAnnotInfo -d /home/antenna/test/test.pdf
```

#### ■処理内容

注釈（テキスト ※、リンク、スタンプ、ファイル添付）情報を標準出力に出力します。出力ファイル指定（-o）は必要ありません。

※テキスト注釈：PDF ビューアによっては「ノート」という名称が使用されています。

#### ■フォルダ指定：対応有り

入力フォルダの PDF ファイルの注釈情報を取得し標準出力に出力します。出力フォルダ指定 (-o) は必要ありません。

#### ■パラメータ

パラメータはありません。

#### ■出力内容

「『注釈情報取得』の出力内容（p. 62）」を参照してください。

## 11.3.7 -setPdfWatermark : PDF ファイル透かし設定

## ■コマンド例

[処理例]

test.pdf の全ページに mark.pdf の 1 ページ目を 50% の大きさ、透明度 0.7 で透かしとして挿入し out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -setPdfWatermark -pdfPath C:\test\mark.pdf -scale 0.5 -opacity 0.7 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -setPdfWatermark -pdfPath /home/antenna/test/mark.pdf -scale 0.5 -opacity 0.7 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

## ■処理内容

PDF ファイルを透かしとして挿入します。

## ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、PDF ファイルを透かしとして挿入します。入力ファイル名と同一名で出力フォルダに出力します。

## ■パラメータ

パラメータ	内容
-pdfPath Val [パスワード]	透かしにする PDF ファイルパスの指定。指定必須
-scale Val	挿入する透かしの倍率 (範囲 : 0.0~50.0)。省略可。 0.0 を指定すると -rect や -margin で指定された描画矩形にフィットする。 指定がない場合、0.0
-rect left bottom right top	透かしの配置する矩形の指定 (mm 単位)。指定がない場合はページサイズの矩形。
-margin top left bottom right	透かしの配置するときの余白の設定 (mm 単位)。省略可。
-align Val	透かしの配置場所の設定。指定がない場合は「5:中央」に配置。 1: 左上 2: 中央上 3: 右上 4: 左中 5: 中央 6: 右中 7: 左下 8: 中央下 9: 右下 1~9 以外が指定された場合、エラーとなる。
-zorder Val	透かしの Z オーダーの設定。省略可。 1: 前面 2: 背面 指定がない場合、「1 : 前面」に配置。 1,2 以外が指定された場合、エラーとなる。
-pageRange Val	透かしを挿入するページ範囲の設定。省略可。 0: 全ページ 1: 先頭ページのみ 2: 最終ページのみ 3: 奇数ページ 4: 偶数ページ 5: 指定したページ 指定がない場合、「0 : 全ページ」に挿入する。 1~5 以外が指定された場合、エラーとなる。 1 と -notInFirst、2 と -notInLast を指定するとエラーとなる。
-pageRangeCustom Val	透かしを挿入するページ番号あるいはページ範囲を指定。「-pageRange 5」の場合に設定必須。1 ページを「0」とする。 例) -pageRangeCustom "0,2-4" 1 ページと 3~5 ページに挿入される。
-notInFirst	指定がある場合、先頭ページには透かしを挿入しない。 指定がない場合、先頭ページに透かしを挿入できる。 -pageRange 1 と同時に指定するとエラーとなる。
-notInLast	指定がある場合、最終ページには透かしを挿入しない。 指定がない場合、最終ページにも透かしを挿入できる。

	-pageRange 2 と同時に指定するとエラーとなる。
-notDisplay	指定がある場合、透かしを表示しない。 指定がない場合、透かしを表示する。
-notPrint	指定がある場合、透かしを印刷対象としない。 指定がない場合、透かしを印刷対象とする。
-opacity Val	透かしの透明度の設定。省略可。 設定値の範囲：0.0～1.0。0.0が透明、1.0が不透明。指定がない場合、1.0（不透明）。
-tiling	指定がある場合、透かしをタイリング挿入する。-rect 指定があるときは rect 内でタイリングされる。 指定がない場合、透かしをタイリングで挿入しない。

## 11.3.8 -setImageWatermark : 画像ファイル透かし設定

## ■コマンド例

## [処理例]

mark.bmp を透明度 0.8 の透かしとして test.pdf の 1,4,11,12,13 ページめにタイリング状態で挿入し out.pdf に保存する。

## [Windows]

```
AHPDFToolCmd70.exe -setImageWatermark -imagePath C:\test\mark.bmp -pageRange 5 -pageRangeCustom "0,3,10-12"
-opacity 0.8 -tiling -d C:\test\test.pdf -o C:\sav\out.pdf
```

## [Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -setImageWatermark -imagePath /home/antenna/test/mark.bmp -pageRange 5 -pageRangeCustom
"0,3,10-12" -opacity 0.8 -tiling -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

## ■処理内容

画像ファイルを透かしとして挿入します。

## ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、画像ファイルを透かしとして挿入します。入力ファイル名と同一名で出力フォルダに出力します。

## ■パラメータ

パラメータ	内容
-imagePath	透かしにする画像ファイルパスの指定。指定必須 対応形式 : bitmap、jpeg、png、tiff、gif
-scale Val	挿入する透かしの倍率 (範囲 : 0.0~50.0)。省略可。 0.0 を指定すると -rect や -margin で指定された描画矩形にフィットする。 指定がない場合、0.0
-rect left bottom right top	透かしの配置する矩形の指定 (mm 単位)。指定がない場合はページサイズの矩形。
-margin top left bottom right	透かしの配置するときの余白の設定 (mm 単位)。省略可。
-align Val	透かしの配置場所の設定。指定がない場合は「5:中央」に配置。 1: 左上 2: 中央上 3: 右上 4: 左中 5: 中央 6: 右中 7: 左下 8: 中央下 9: 右下 1~9 以外が指定された場合、エラーとなる。
-zorder Val	透かしの Z オーダーの設定。省略可。 1: 前面 2: 背面 指定がない場合、「1:前面」に配置。 1,2 以外が指定された場合、エラーとなる。
-pageRange Val	透かしを挿入するページ範囲の設定。省略可。 0: 全ページ 1: 先頭ページのみ 2: 最終ページのみ 3: 奇数ページ 4: 偶数ページ 5: 指定したページ 指定がない場合、「0 : 全ページ」に挿入する。 1~5 以外が指定された場合、エラーとなる。 1 と -notInFirst、2 と -notInLast を指定するとエラーとなる。
-pageRangeCustom Val	透かしを挿入するページ番号あるいはページ範囲を指定。「-pageRange 5」の場合に設定必須。1 ページを「0」とする。 例) -pageRangeCustom "0,2-4" 1 ページと 3~5 ページに挿入される。
-notInFirst	指定がある場合、先頭ページには透かしを挿入しない。 指定がない場合、先頭ページに透かしを挿入できる。 -pageRange 1 と同時に指定するとエラーとなる。

-notInLast	指定がある場合、最終ページには透かしを挿入しない。 指定がない場合、最終ページにも透かしを挿入できる。 -pageRange 2 と同時に指定するとエラーとなる。
-notDisplay	指定がある場合、透かしを表示しない。 指定がない場合、透かしを表示する。
-notPrint	指定がある場合、透かしを印刷対象としない。 指定がない場合、透かしを印刷対象とする
-opacity Val	透かしの透明度の設定。省略可。 設定値の範囲：0.0～1.0。0.0が透明、1.0が不透明。指定がない場合、1.0（不透明）。
-tiling	指定がある場合、透かしをタイリング挿入する。-rect 指定があるときは rect 内でタイリングされる。 指定がない場合、透かしをタイリングで挿入しない。

### 11.3.9 -setTextWatermark : テキスト透かし設定

#### ■コマンド例

##### [処理例]

メイリオフォント (Windows) / IPA ゴシック (Linux / Amazon Linux2 X86) でフォントサイズ「52」、文字色が {RGB}={100} である「PDF Tool API V7.0 テスト」という文字列を透明度 0.9、挿入角度 20 度の透かしとして test.pdf の全ページの中央に挿入し out.pdf に保存する。

##### [Windows]

```
AHPDFToolCmd70.exe -setTextWatermark -text "PDF Tool API V7.0 テスト" -colorText 1 0 0 -angle 20 -align 5 -opacity 0.9 -font -name メイリオ -size 52 -d C:\test\test.pdf -o C:\sav\out.pdf
```

##### [Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -setTextWatermark -text "PDF Tool API V7.0 テスト" -colorText 1 0 0 -angle 20 -align 5 -opacity 0.9 -font -name ipa-gothic -size 52 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

文字列を透かしとして挿入します。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、文字列を透かしとして挿入します。入力ファイル名と同一名で出力フォルダに出力します。

#### ■パラメータ

パラメータ	内容
-text Val	透かしとして挿入する文字列の設定。指定必須
-colorText r g b	文字の塗りつぶし色の設定。省略可。 設定値の範囲 : r,g,b それぞれ 0.0~1.0 指定がない場合、黒色(0.0 0.0 0.0)
-colorOutline r g b	文字の輪郭の色の設定。省略可。 設定値の範囲 : r,g,b それぞれ 0.0~1.0 指定がない場合、黒色(0.0 0.0 0.0)
-diagonal	指定がある場合、透かしをページの対角線上に挿入する 指定がない場合、透かしは横書き、水平となる。 -tiling が設定されている場合は無効
-angle Val	透かしの挿入角度の設定。反時計周りが正の値。省略可。 指定がない場合、0.0 (水平)。 -diagonal が設定されている場合は無効。
-rect left bottom right top	透かしを配置する矩形の指定 (mm 単位)。指定がない場合はページサイズの矩形。
-margin top left bottom right	透かしを配置するときの余白の設定 (mm 単位)。省略可。
-align Val	透かしの配置場所の設定。指定がない場合は「5:中央」に配置。 1: 左上 2: 中央上 3: 右上 4: 左中 5: 中央 6: 右中 7: 左下 8: 中央下 9: 右下 1~9 以外が指定された場合、エラーとなる。
-zorder Val	透かしの Z オーダーの設定。省略可。 1: 前面 2: 背面 指定がない場合、「1:前面」に配置。 1,2 以外が指定された場合、エラーとなる。
-pageRange Val	透かしを挿入するページ範囲の設定。省略可。 0: 全ページ

	<p>1: 先頭ページのみ 2: 最終ページのみ  3: 奇数ページ 4: 偶数ページ  5: 指定したページ  指定がない場合、「0: 全ページ」に挿入する。  1~5以外が指定された場合、エラーとなる。  1と-notInFirst、2と-notInLastを指定するとエラーとなる。</p>	
-pageRangeCustom Val	<p>透かしを挿入するページ番号あるいはページ範囲を指定。「-pageRange 5」の場合に設定必須。1ページを「0」とする。  例) -pageRangeCustom "0,2-4"  1ページと3~5ページに挿入される。</p>	
-notInFirst	<p>指定がある場合、先頭ページには透かしを挿入しない。  指定がない場合、先頭ページに透かしを挿入できる。  -pageRange 1と同時に指定するとエラーとなる。</p>	
-notInLast	<p>指定がある場合、最終ページには透かしを挿入しない。  指定がない場合、最終ページにも透かしを挿入できる。  -pageRange 2と同時に指定するとエラーとなる。</p>	
-notDisplay	<p>指定がある場合、透かしを表示しない。  指定がない場合、透かしを表示する。</p>	
-notPrint	<p>指定がある場合、透かしを印刷対象としない。  指定がない場合、透かしを印刷対象とする。</p>	
-opacity Val	<p>透かしの透明度の設定。省略可。  設定値の範囲: 0.0~1.0。0.0が透明、1.0が不透明。指定がない場合、1.0(不透明)。</p>	
-tiling	<p>指定がある場合、透かしをタイリング挿入する。-rect指定があるときはrect内でタイリングされる。  指定がない場合、透かしをタイリングで挿入しない。</p>	
-font	<p>透かし文字列のフォント設定。指定必須。</p>	
	-name Val	フォント名
	-size Val	フォントサイズ。省略可。 単位: point 指定がない場合、24.0 point
	-weight Val	太さ。省略可。 有効値: 200,300,400,500,600,700,800,900 指定がない場合、400 有効値以外が指定された場合、エラーとなる。 -boldの指定があるときは-weightの設定は無効。
	-bold	指定がある場合、文字列を太字にする。 指定がない場合、文字列を太字にしない。
	-italic	指定がある場合、文字列を斜体にする。 指定がない場合、文字列を正体にする。
	-embed Val	フォント埋め込み。 true: 埋め込む false: 埋め込まない 指定がない場合、「true: 埋め込む」

## 11.3.10 -setColorWatermark : 色透かし設定

## ■コマンド例

[処理例]

{R G B}={0 1 0} の色を透明度 0.7 の透かしとして先頭ページ以外のページに印刷非対象の状態では test.pdf に挿入して out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -setColorWatermark -color 0 1 0 -notInFirst -notPrint -opacity 0.7 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -setColorWatermark -color 0 1 0 -notInFirst -notPrint -opacity 0.7 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

## ■処理内容

色の透かしを挿入します。

## ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、色の透かしを挿入します。入力ファイル名と同一名で出力フォルダに出力します。

^

## ■パラメータ

パラメータ	内容
-color r g b	挿入する色の設定。指定必須 設定値の範囲 : r,g,b それぞれ 0.0~1.0
-rect left bottom right top	透かしの配置する矩形の指定 (mm 単位)。指定がない場合はページサイズの矩形。
-margin top left bottom right	透かしの配置するときの余白の設定 (mm 単位)。省略可。
-zorder Val	透かしの Z オーダーの設定。省略可。 1: 前面 2: 背面 指定がない場合、「1:前面」に配置。 1,2 以外が指定された場合、エラーとなる。
-pageRange Val	透かしを挿入するページ範囲の設定。省略可。 0: 全ページ 1: 先頭ページのみ 2: 最終ページのみ 3: 奇数ページ 4: 偶数ページ 5: 指定したページ 指定がない場合、「0 : 全ページ」に挿入する。 1~5 以外が指定された場合、エラーとなる。 1 と -notInFirst、2 と -notInLast を指定するとエラーとなる。
-pageRangeCustom Val	透かしを挿入するページ番号あるいはページ範囲を指定。「-pageRange 5」の場合に設定必須。1 ページを「0」とする。 例) -pageRangeCustom "0,2-4" 1 ページと 3~5 ページに挿入される。
-notInFirst	指定がある場合、先頭ページには透かしを挿入しない。 指定がない場合、先頭ページに透かしを挿入できる。 -pageRange 1 と同時に指定するとエラーとなる。
-notInLast	指定がある場合、最終ページには透かしを挿入しない。 指定がない場合、最終ページにも透かしを挿入できる。 -pageRange 2 と同時に指定するとエラーとなる。
-notDisplay	指定がある場合、透かしを表示しない。 指定がない場合、透かしを表示する。
-notPrint	指定がある場合、透かしを印刷対象としない。 指定がない場合、透かしを印刷対象とする。
-opacity Val	透かしの透明度の設定。省略可。 設定値の範囲 : 0.0~1.0。0.0 が透明、1.0 が不透明。指定がない場合、1.0 (不透明)。



## 11.3.11 -encrypt : PDF セキュリティ設定

## ■コマンド例

[処理例]

test.pdf に対し、256bit AES 暗号化で権限パスワード文字列を「pass123」、印刷権限は「低解像度」、内容のコピーを不許可とするセキュリティを設定し out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -encrypt -ownerPass pass123 -filterType 4 -perms2 -print 1 -copy false -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -encrypt -ownerPass pass123 -filterType 4 -perms2 -print 1 -copy false -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

## ■処理内容

セキュリティを設定します。

入力ファイルのセキュリティ情報は、出力 PDF ファイルには引き継がれません。

## ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、セキュリティを設定します。入力ファイル名と同一名で出力フォルダに出力します。

## ■パラメータ

パラメータ	内容	
-userPass Val	文書を開くパスワードの設定。 -userPass と -ownerPass のどちらか指定必須。	
-ownerPass Val	権限パスワードの設定。 -userPass と -ownerPass のどちらか指定必須。 -filterType を指定する場合-ownerPass の指定必須。	
-filterType Val	暗号化レベルの設定。省略可。 2: 128bit RC4    3: 128bit AES 4: 256bit AES 指定がない場合、「4: 256bit AES」 2~4 以外が指定された場合エラーとなる。 -filterType を指定し-ownerPass の指定がない場合エラーとなる。	
-perms2	「-filterType = 2/3/4」の場合のセキュリティ設定。省略可。	
	-print Val	印刷。省略可。 0: 許可しない 1: 低解像度 2: 高解像度 指定がない場合、「0: 許可しない」 0~2 以外が指定された場合エラーとなる。
	-modify Val	文書の編集。省略可。 0: 許可しない 1: ページの挿入、削除、回転 2: フォームフィールドの入力と既存の署名フィールドに署名 3: 注釈の作成、フォームフィールドの入力と既存の署名フィールドに署名 4: ページの抽出を除くすべての操作 指定がない場合、「0: 許可しない」 0~4 外が指定された場合エラーとなる。
	-copy Val	内容のコピー。省略可。 true: 許可 false: 不許可 指定がない場合、「false: 不許可」
	-accessibility Val	スクリーンリーダーデバイスのテキストアクセスを有効にする。省略可。

	<p>true: 有効    false: 無効 指定がない場合、「true : 有効」 「-copy」（内容のコピー）が「true」の場合、自動的に「-accessibility」も「true」となります。 falseを設定する場合は「-copy false」を同時に指定してください。</p>
--	--

### 11.3.12 -decrypt : PDF セキュリティ削除

#### ■コマンド例

[処理例]

test.pdf のセキュリティを削除し out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -decrypt -d C:\test\test.pdf pass123 -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -decrypt -d /home/antenna/test/test.pdf pass123 -o /home/antenna/sav/out.pdf
```

#### ■処理内容

セキュリティを削除します。

入力ファイルのセキュリティ情報は、出力 PDF ファイルには引き継がれません。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルのセキュリティを削除します。入力ファイル名と同一名で出力フォルダに出力します。

#### ■パラメータ

パラメータはありません。

### 11.3.13 -optimize : PDF の最適化

#### ■コマンド例

[処理例]

test.pdf 内の画像をダウンサンプリングし、すべての注釈を削除して out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -optimize -optimizeImage true -removeAnnots true -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -optimize -optimizeImage true -removeAnnots true -d /home/antenna/test/test.pdf -o /home/antenna/sav/  
out.pdf
```

#### ■ 処理内容

画像のダウンサンプリングや、開き方・しおり・注釈・アーティクル・サムネイルの削除、フォントの統合を行うことにより PDF を最適化します。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対して最適化処理を行います。入力ファイル名と同一名で出力フォルダに出力します。

#### ■パラメータ

パラメータ	内容
-optimizeImage Val	画像のダウンサンプリング処理。省略可。 true : する false : しない 省略した場合ダウンサンプリングは行いません。 ダウンサンプリングは次の設定で行われます。 ・バイキュービック法 ・カラー/グレースケール : 解像度が 225dpi 以上の場合 150dpi まで縮小する。 ・白黒 : 解像度が 450dpi 以上の場合 300dpi まで縮小する。
-removeOpenAction Val	開き方設定の削除。省略可。] true : する false : しない
-removeOutlines Val	しおりの削除。省略可。 true : する false : しない 省略した場合削除は行いません。
-removeAnnots Val	注釈の削除。省略可。 true : する false : しない 省略した場合削除は行いません。
-removeArticles Val	アーティクルの削除。省略可。 true : する false : しない 省略した場合削除は行いません。
-removeThumbnails Val	サムネイルの削除。省略可。 true : する false : しない 省略した場合削除は行いません。
-mergeFonts Val	フォントの統合。省略可。 true : 統合する false : 統合しない 省略した場合、統合は行いません。

### 11.3.14 -addEmbeddedFile : ファイル添付

#### ■コマンド例

[処理例]

test.pdf に対し sample.docx を添付して out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -addEmbeddedFile C:\test\sample.docx -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -addEmbeddedFile C:\test\sample.docx -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

ファイルを添付します。

添付ファイルは catalog 辞書にエン트리されます。

複数回指定することにより複数のファイル添付ができます。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、ファイル添付を行います。入力ファイル名と同一名で出力フォルダに出力します。

#### ■パラメータ

パラメータ	内容
[Path]	添付するファイルのフルパス名。指定必須。

### 11.3.15 -pageRotate : ページ回転

#### ■コマンド例

[処理例]

test.pdf の 3 ページ目を 90 度、12 ページ目を 180 度回転し out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -pageRotate 2 90 -pageRotate 11 180 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -pageRotate 2 90 -pageRotate 11 180 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

ページを回転します。

複数回指定することにより複数のページの回転ができます。

#### ■フォルダ指定 : 対応無し

#### ■パラメータ

パラメータ	内容
[Page]	回転するページ番号。1 ページを「0」とする。指定必須。
[Rotation]	回転角度。有効値 : 0,90,180,270 有効値以外が指定された場合、エラーとなる。

### 11.3.16 -pageDelete : ページ削除

#### ■コマンド例

[処理例]

test.pdf の 6 ページ目から 4 ページ分 (6,7,8,9 ページ) を削除して out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -pageDelete 5 4 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -pageDelete 5 4 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

指定した位置のページから指定ページ数を削除します。

1 回のコマンド指定内で複数回の指定はできません。

#### ■フォルダ指定 : 対応無し

#### ■パラメータ

パラメータ	内容
[Index]	削除を開始するページ位置。1 ページを「0」とする。指定必須。
[numPages]	削除するページ数。省略可。 指定がない場合、1 ページ削除する。

### 11.3.17 -writeText : 文字列追加

#### ■コマンド例

[処理例]

メイリオフォント (Windows) / IPA ゴシック (Linux / Amazon Linux2 X86) でフォントサイズ 10 ポイント、文字色が {R G B}={1 0 0}、透明度 0.8 である「アンテナハウス株式会社」という文字列を test.pdf の全ページの右上に挿入して out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -writeText -text "アンテナハウス株式会社" -align 3 -font -name メイリオ -size 10 -colorText 1 0 0 -opacity 0.8 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -writeText -text "アンテナハウス株式会社" -align 3 -font -name ipa-gothic -size 10 -colorText 1 0 0 -opacity 0.8 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

指定ページに文字列を挿入します。

文字列のフォントや色、透明度の指定ができます。挿入位置は、あらかじめ定められた9か所から指定します。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、文字列追加を行います。入力ファイル名と同一名で出力フォルダに出力します。

#### ■パラメータ

パラメータ	内容
-text Val	挿入する文字列。指定必須。
-pageNo Val	文字列を挿入するページ番号。1 ページを「0」とする。省略可。 指定がない場合、全ページに挿入する。
-vertical	縦書き指定。省略可。 指定がある場合、縦書きする。 指定がない場合、横書きする。
-align Val	挿入位置。省略可。 1: 左上 2: 中央上 3: 右上 4: 左中 5: 中央 6: 右中 7: 左下 8: 中央下 9: 右下 指定がない場合「5: 中央」に配置。 1~9以外が指定された場合エラーとなる。
-angle Val	文字列の角度。省略可。 横書きの場合のみ有効。 反時計回りが正の値。
-font	フォント設定。指定必須。
	-name Val      フォント名。指定必須。
	-size Val      フォントサイズ。省略可。 単位 : point 指定がない場合、12 point
	-bold          指定がある場合、文字列を太字にする。 指定がない場合、文字列を太字にしない。
	-italic        指定がある場合、文字列を斜体にする。 指定がない場合、文字列を正体にする。
	-notEmbed     指定がある場合、フォントを埋め込まない。 指定がない場合、フォントを埋め込む。

-colorText r g b	文字の塗りつぶし色。省略可。 設定値の範囲：r, g, b それぞれ 0.0～1.0 指定がない場合、黒色 (0.0 0.0 0.0)
-colorOutline r g b	文字の輪郭の色。省略可。 設定値の範囲：r, g, b それぞれ 0.0～1.0 指定がない場合、黒色 (0.0 0.0 0.0)
-opacity Val	文字列の透明度。省略可。 設定値の範囲：0.0～1.0。0.0 が透明、1.0 が不透明。 指定がない場合、1.0 (不透明)。

### 11.3.18 -exportAnnotsToFDF：注釈の FDF への出力

#### ■コマンド例

[処理例]

test.pdf のページにあるマークアップ注釈を抽出して out.fdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -exportAnnotsToFDF C:\sav\out.fdf -d C:\test\test.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -exportAnnotsToFDF /home/antenna/sav/out.fdf -d /home/antenna/test/test.pdf
```

#### ■処理内容

入力ファイルに存在するマークアップ注釈を FDF ファイルに出力します。

出力ファイル指定 (-o) は必要ありません。

#### ■フォルダ指定：対応有り

入力フォルダの PDF ファイルのマークアップ注釈を FDF ファイルに出力します。入力ファイル名の拡張子を「.fdf」に変更したファイル名で出力フォルダに出力します。

#### ■パラメータ

パラメータ	内容
[outFilePath]	出力する FDF ファイルのフルパス。 指定されたファイル名の拡張子が「fdf」でない場合、コマンドラインが出力時に「fdf」を追加する。

### 11.3.19 -importAnnotsFromPDF : 注釈の取り込み

#### ■コマンド例

[処理例]

annot.pdf に存在するマークアップ注釈を test.pdf に取り込んで out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -importAnnotsFromPDF C:\test\annot.pdf -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -importAnnotsFromPDF /home/antenna/test/annot.pdf -d /home/antenna/test/test.pdf -o /home/antenna/sav/  
out.pdf
```

#### ■処理内容

指定の PDF ファイルに存在するマークアップ注釈を入力ファイルにインポートします。

#### ■フォルダ指定 : 対応無し

#### ■パラメータ

パラメータ	内容
[Path]	インポート元の PDF ファイルのフルパス

### 11.3.20 -insertEmptyPage : 空白ページ挿入

#### ■コマンド例

[処理例]

test.pdf に対し、3 ページ目から 5 ページ分の A4 横サイズの空白ページを挿入して out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -insertEmptyPage -index 2 -number 5 -size 1 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -insertEmptyPage -index 2 -number 5 -size 1 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

指定のページ位置から指定ページ数分の空白ページを挿入します。

1 回のコマンド指定内で複数回の指定はできません。

■フォルダ指定 : 対応無し

#### ■パラメータ

パラメータ	内容
-index Val	空白ページを挿入するページ位置。1 ページを「0」とする。指定必須・
-number Val	挿入するページ数。省略可。 指定がない場合、1 ページ挿入する。
-size Val	挿入するページサイズ。省略可。 0 : A4 縦 1 : A4 横 2 : A5 縦 3 : A5 横 4 : A3 縦 5 : A3 横 指定がない場合、「0:A4 縦」 0~5 以外が指定された場合、エラーとなる。

## 11.3.21 -addPageNumber : ページ番号挿入

## ■コマンド例

[処理例]

test.pdf に対し、1 ページ目をページ番号「11」としてページの右下に {R,G,B} = {0,0,1} の文字色で挿入し out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -addPageNumber -startNumber 11 -align 9 -colorText 0 0 1 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -addPageNumber -startNumber 11 -align 9 -colorText 0 0 1 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

## ■処理内容

ページ番号を挿入します。

ページ番号文字列のフォントや色、透明度の指定ができます。挿入位置は、あらかじめ定められた9か所から指定します。

## ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、ページ番号挿入を行います。入力ファイル名と同一名で出力フォルダに出力します。

## ■パラメータ

パラメータ	内容
-index Val	ページ番号を挿入するページ位置。省略可。 1 ページを「0」とする。 指定がない場合「0」。
-startNumber Val	挿入するページ番号の開始番号。省略可。 指定がない場合「1」。
-align Val	ページ番号挿入位置。省略可。 1: 左上 2: 中央上 3: 右上 4: 左中 5: 中央 6: 右中 7: 左下 8: 中央下 9: 右下 指定がない場合、「8:中央下」に配置。 1~9以外が指定された場合、エラーとなる。
-font	フォント設定。省略可
	-name Val      フォント名。省略可。 指定がない場合、「Helvetica」
	-size Val      フォントサイズ。省略可。 単位 : point 指定がない場合、12 point
	-bold          指定がある場合、文字列を太字にする。 指定がない場合、文字列を太字にしない。
	-italic        指定がある場合、文字列を斜体にする。 指定がない場合、文字列を正体にする。
	-notEmbed     指定がある場合、フォントを埋め込まない。 指定がない場合、フォントを埋め込む。
-colorText r g b	文字の塗りつぶし色。省略可。 設定値の範囲 : r, g, b それぞれ 0.0~1.0 指定がない場合、黒色 (0.0 0.0 0.0)
-colorOutline r g b	文字の輪郭の色。省略可。 設定値の範囲 : r, g, b それぞれ 0.0~1.0

	指定がない場合、黒色 (0.0 0.0 0.0)
-opacity Val	文字列の透明度の設定。省略可。 設定値の範囲：0.0 ~ 1.0。0.0が透明、1.0が不透明。 指定がない場合、1.0 (不透明)。

## 11.3.22 -pageExtract : ページ抽出

## ■コマンド例

## [処理例]

test.pdf から1ページ目と、4,5,6,7ページ目を抽出しそれぞれ別々のファイルに保存する。保存ファイル名は、out\_0.pdf、out\_3-6.pdf となる。

## [Windows]

```
AHPDFToolCmd70.exe -pageExtract "0,3-6" -divideFile -d C:\test\test.pdf -o C:\sav\out.pdf
```

## [Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -pageExtract "0,3-6" -divideFile -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

## ■処理内容

指定されたページやページ範囲を抽出します。

分割指定がある場合、抽出したページやページ範囲ごとに PDF ファイルに保存します

分割指定がない場合、抽出したページやページ範囲をひとつの PDF ファイルに保存します。

しおりや開き方は、抽出後の PDF ファイルにはコピーされません。

また、入力ファイルのセキュリティ情報は、抽出後の PDF ファイルには引き継がれません。

## ■フォルダ指定 : 対応無し

## ■パラメータ

パラメータ	内容
[PageRange]	抽出するページ番号またはページ範囲。指定必須 1 ページを「0」とする。 複数指定する場合はカンマで区切る。(例) "0,2-4"
-divideFile	省略可 指定がある場合、 [PageRange] のカンマごとに分割して PDF ファイルを出力する。出力ファイル名は指定の出力ファイル名の末尾にページ番号がつく。 指定がない場合、抽出したページをひとつのファイルに出力する。
-copyInfo	省略可 指定がある場合、入力ファイルの文書情報を出力ファイルにコピーする。 指定がない場合、文書情報はコピーしない。

### 11.3.23 -embedFonts : フォント埋め込み

#### ■コマンド例

[処理例]

test.pdf に対しフォント埋め込みを行い、out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -embedFonts -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -embedFonts -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

PDF にフォントが埋め込まれていない場合、埋め込み状態に変更します。

フォントを埋め込むことができるのは、動作環境に該当のフォントが存在し、かつ、埋め込みが許可されているフォントの場合です。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルに対し、フォント埋め込み処理を行います。入力ファイル名と同一名で出力フォルダに出力します。

#### ■パラメータ

パラメータはありません。

## 11.3.24 -extractImage : 画像抽出

## ■コマンド例

[処理例]

test.pdf に存在する画像データを JPEG 形式のファイルとして指定フォルダに出力する。

[Windows]

```
AHPDFToolCmd70.exe -extractImage C:\out -format 2 -d C:\test\test.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -extractImage /home/antenna/sav -format 2 -d /home/antenna/test/test.pdf
```

## ■処理内容

PDF のページ上の画像データを抽出します。

ページ全体を画像化する処理ではありません。

## ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルから画像データを抽出します。画像データは出力フォルダの入力ファイル名のサブフォルダに保存します。

## ■パラメータ

パラメータ	内容
出力フォルダパス	[必須] 画像ファイルの出力フォルダパスを指定する。 出力ファイル名は「入力ファイル名_ページ番号_連番」 ページ番号の開始は「1」 連番の開始は「0001」でページごとにリセットする。
-pageNo	画像を抽出するページ番号。省略可。 1 ページ目は「0」指定。 複数指定する場合はカンマで区切る。 指定がない場合、全ページの画像を抽出する。
-format	画像の保存形式。省略可。 0 = AUTO    1 = Bitmap 2 = JPEG    3 = PNG 指定がない場合、「0 = AUTO」

### 11.3.25 -extractText : テキスト抽出

#### ■コマンド例

[処理例]

test.pdf の 1,3,4,5 ページ目上に存在するテキストを座標順に並べ替えて out.txt に出力する。

[Windows]

```
AHPDFToolCmd70.exe -extractText C:\sav\out.txt -pageNo 0,2-4 -sort -d C:\test\test.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -extractText /home/antenna/sav/out.txt -pageNo 0,2-4 -sort -d /home/antenna/test/test.pdf
```

#### ■処理内容

PDF ファイルからテキストを抽出します。

#### ■フォルダ指定 : 対応有り

入力フォルダの PDF ファイルからテキストを抽出します。入力ファイル名の拡張子を「.txt」に変更したファイル名で出力フォルダに出力します。

#### ■パラメータ

パラメータ	内容
出力テキストファイルパス	[必須] 出力先のテキストファイルパスを指定する。 抽出対象が複数ページの場合、先頭行に「pageX」と出力されます。
-pageNo	テキストを抽出するページ番号。省略可。 1 ページ目は「0」指定。 複数指定する場合はカンマで区切る。 指定がない場合、全ページのテキストを抽出する。
-sort	テキストを座標順に並べ替える。

## 11.3.26 -imageToPdf : 画像ファイルの PDF 化

## ■コマンド例

[処理例]

test.jpg を解像度 300dpi で 90 度に回転して out.pdf に出力する。

[Windows]

```
AHPDFToolCmd70.exe -imageToPdf -dpi 300 -rotateAngle 90 -d C:\test\test.jpg -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -imageToPdf -dpi 300 -rotateAngle 90 -d /home/antenna/test/test.jpg -o /home/antenna/sav/out.pdf
```

## ■処理内容

画像ファイルから PDF ファイルを作成します。

出力される PDF ファイルのバージョンは「1.4」です。

## ■フォルダ指定 : 対応有り

入力フォルダに画像ファイルを PDF に変換します。入力ファイル名.拡張子の後ろに「.pdf」を付けたファイル名で出力フォルダに出力します。

## ■パラメータ

パラメータ	内容
-d	[必須] PDF に変換する画像ファイルパスを指定する。 対応形式 : bitmap / jpeg / png / tiff / gif
-o	[必須] 出力ファイルパスを指定する。
-dpi	解像度を整数値で指定する。省略可。 指定がない場合、画像が持つ解像度となる。
-rotateAngle	回転角度を指定する。省略可。 設定値 : 0/90/180/270 (度) で反時計周り。 指定がない場合、「0」(度)

### 11.3.27 -extractEmbeddedFile : 添付ファイルの抽出

PDF ファイルに添付されているファイルを抽出します。

#### ■コマンド例

[処理例]

test.pdf の添付ファイルを指定フォルダに出力する。

[Windows]

```
AHPDFToolCmd70.exe -extractEmbeddedFile C:\out -d C:\test\test.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -extractEmbeddedFile /home/antenna/sav -d /home/antenna/test/test.pdf
```

#### ■処理内容

PDF ファイルからすべての添付ファイルを抽出します。

#### ■フォルダ指定 : 対応有り

入力フォルダに存在する PDF ファイルから添付ファイルを抽出します。

抽出した添付ファイルは、出力フォルダに入力ファイル名のサブフォルダを作成し保存します。

#### ■パラメータ

パラメータ	内容
出力フォルダパス	[必須] 添付ファイルの出力先フォルダパスを指定する。

## 11.3.28 -importAnnotsFromFDF : FDF からの注釈の取り込み

## ■コマンド例

[処理例]

Import.fdf から test.pdf に注釈を取り込み out.pdf に保存する。

[Windows]

```
AHPDFToolCmd70.exe -importAnnotsFromFDF C:\test\import.fdf -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -importAnnotsFromFDF /home/antenna/test/import.fdf -d /home/antenna/test/test.pdf -o /home/antenna/sav/  
out.pdf
```

## ■処理内容

FDF ファイルから注釈をインポートします。

## ■フォルダ指定 : 対応無し

## ■パラメータ

パラメータ	内容
FDF ファイルパス	[必須] FDF ファイルパス

### 11.3.29 -fixedUpPDFA : PDF/A-1b、-2b 変換

#### ■コマンド例

##### [処理例]

プロファイルとして sRGB2014.icc と JapanColor2001Coated.icc を使用して test.pdf を PDF/A-2b に変換する。処理ログを-o で指定した場所に「出力ファイル名.log」のファイル名で出力する。

##### [Windows]

```
AHPDFToolCmd70.exe -fixedUpPDFA -type 2b -RGBProfile sRGB2014.icc -CMYKProfile JapanColor2001Coated.icc -log -d c:\in\test.pdf -o c:\out\out.pdf
```

##### [Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -fixedUpPDFA -type 2b -RGBProfile sRGB2014.icc -CMYKProfile JapanColor2001Coated.icc -log -d /home/antenna/in/test.pdf -o /home/antenna/sav/out.pdf
```

#### ■処理内容

PDF ファイルを PDF/A-1b または PDF/A-2b に変換します。

#### ■フォルダ指定 : 対応有り

#### ■パラメータ

パラメータ	内容
-type [1b/2b]	[必須] PDF/A 規格の種類を指定 1b : PDF/A-1b に変換する 2b : PDF/A-2b に変換する
-RGBProfile [プロファイル名]	RGB カラープロファイル名を指定 フルパスで指定する。ファイル名のみ指定の場合、コマンドラインと同階層に存在するものとする。
-CMYKProfile [プロファイル名]	CMYK カラープロファイル名を指定 フルパスで指定する。ファイル名のみ指定の場合、コマンドラインと同階層に存在するものとする。
-log	処理ログをテキストファイルに出力する。 出力先は「-o」で指定した場所。ファイル名は「-o」で指定したファイル名の拡張子を「.log」に変更した名前となる。

## 11.3.30 -validatePDFA : PDF/A-1b、-2b 準拠確認

## ■コマンド例

## [処理例]

out.pdf が PDF/A-2b に準拠しているかどうかを確認する。処理ログを-d で指定した場所に「入力ファイル名.log」のファイル名で出力する。

## [Windows]

```
AHPDFToolCmd70.exe -validatePDFA -type 2b -log -d c:\out.pdf
```

## [Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -validatePDFA -type 2b -log -d /home/antenna/out.pdf
```

## ■処理内容

PDF ファイルが PDF/A-1b あるいは PDF/A-2b に準拠しているかどうかを確認します。

## ■フォルダ指定 : 対応有り

## ■パラメータ

パラメータ	内容
-type [1b/2b]	[必須] PDF/A 規格の種類を指定 1b : PDF/A-1b の準拠確認を行う 2b : PDF/A-2b の準拠確認を行う
-log	処理ログをテキストファイルに出力する。 出力先は「-d」で指定した場所。ファイル名は、「-d」で指定されたファイル名の拡張子を「.log」に変更した名前となる。

### 11.3.31 -dividePdf : ファイル分割

#### ■コマンド例

[処理例]

test.pdf を 5 ページずつに分割し、入力ファイル名に連番を付与したファイル名で出力する。

[Windows]

```
AHPDFToolCmd70.exe -dividePdf -each 5 -suffix 2 -d c:\in\test.pdf -o c:\out
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -dividePdf -each 5 -suffix 2 -d /home/antenna/in/test.dpf -o /home/antenna/out
```

#### ■処理内容

入力ファイルを指定の条件でファイル分割します。しおりやページ内リンクが存在する場合は、分割後のファイルの該当箇所にジャンプするよう設定しなおします。

「-o」にファイルのフルパスが指定された場合、そのファイル名にサフィックスを付け出力ファイル名とします。

「-o」にフォルダパスが指定された場合、入力ファイル名にサフィックスを付け出力ファイル名とします。

#### ■フォルダ指定 : 対応無し

#### ■パラメータ

パラメータ	内容
-each [ページ数]	指定したページ数ごとに分割
-into [分割数]	指定したファイル数になるよう分割。 指定の数に分割できない場合、suffix は付けずに元のファイル名のまま出力する。
-outline	1階層目のしおりごとに分割。 しおりが存在しない場合、suffix は付けずに元のファイル名のまま出力する。
-suffix [1/2/3/4/5]	出力ファイル名のサフィックス。デフォルトは「1」 「-outline」が指定されていないときに「-suffix [3/4/5]」の指定があるとエラー。 1: 分割したページ範囲を付ける。ページの開始番号は「1」。総ページ数の桁数で表し0詰める。 2: 連番を付ける。連番開始番号は「1」。分割される数の桁数で表し0詰める。 3: しおりのタイトルをファイル名とする。「-outline」が指定されている場合に有効。 4: ページ範囲+しおりタイトルをファイル名とする。ページの開始番号は「1」。「-outline」が指定されている場合に有効。 5: 連番+しおりタイトルをファイル名とする。連番開始番号は「1」。「-outline」が指定されている場合に有効。

※ 「-each」「-info」「-outline」を同時に指定した場合、コマンド文字列の最初に表れたパラメータの条件で分割します。

### 11.3.32 -linearize : Web 表示用に最適化

#### ■コマンド例

[Windows]

```
AHPDFToolCmd70.exe -d C:\in\in.pdf -o C:\sav\out.pdf -linearize
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf -linearize
```

#### ■処理内容

Web 表示用に最適化して PDF 保存を行います。

「-o」で指定されるし出力ファイルパスの後ろに付けてください。

#### ■フォルダ指定 : 対応有り

出力形式が PDF の場合、Web 表示用に最適化して保存します。

#### ■パラメータ

パラメータはありません。

### 11.3.33 -lic : ライセンス情報の取得

#### ■コマンド例

[Windows]

```
AHPDFToolCmd70.exe -lic
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -lic
```

#### ■処理内容

PDF Tool API のライセンス情報を標準出力に出力します。

#### ■フォルダ指定 : 対応無し

#### ■パラメータ

パラメータはありません。

### 11.3.34 -h : コマンド名一覧表示

#### ■コマンド例

[Windows]

```
AHPDFToolCmd70.exe -h
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -h
```

#### ■処理内容

すべてのコマンド名を標準出力に出力します。

「-h」を省略し、コマンドライン実行名（AHPDFToolCmd70）のみを指定して実行した場合も同様の出力結果となります。

### 11.3.35 -h -all : 全コマンドオプション説明の表示

#### ■コマンド例

[Windows]

```
AHPDFToolCmd70.exe -h -all
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -h -all
```

#### ■処理内容

すべてのコマンド名とそのオプションの説明を標準出力に出力します。

「-h」を省略し、「-all」のみを指定して実行した場合も同様の出力結果となります。

### 11.3.36 -h [コマンド名] : 個別コマンドオプション説明の表示

#### ■コマンド例

[処理例]

-extractText のオプションの説明を標準出力に出力する。

[Windows]

```
AHPDFToolCmd70.exe -h -extractText
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -h -extractText
```

#### ■処理内容

指定したコマンドのオプション説明を標準出力に出力します。

「-h」を省略し、コマンド名のみを指定して実行した場合も同様の出力結果となります。

## 11.3.37 取得情報の出力内容

コマンドラインで取得される各情報の出力内容は、次の通りです。

## 11.3.37.1 -getFileInfo : PDF の情報取得

項目名		内容
PDF Version		PDF バージョン番号
	major	メジャーバージョン番号
	minor	マイナーバージョン番号
Document Information		文書情報と総ページ数
	Title	タイトル
	Subject	サブタイトル
	Author	作成者
	Keywords	キーワード
	Creator	アプリケーション
	Producer	PDF 変換
	CreateDate	作成日
	ModifyDate	更新日
	Max Page	総ページ数
View Information		開き方設定内容
	Direction	綴じ方 Left to Right: 左 Right to Left: 右
	PageLayout	ページレイアウト Single Page: 単一ページ One Column: 連続ページ Two Column Left: 連続見開きページ Two Column Right: 連続見開きページ (表紙) Two Page Left: 見開きページ Two Page Right: 見開きページ (表紙)
	PageMode	表示 Use None: ページのみ Use Outlines: しおりパネルとページ Use Thumbs: ページパネルとページ Use OC: レイヤーパネルとページ Use Attachments: 添付ファイルパネルとページ
	FitWindow	ページにウィンドウサイズを合わせる (true / false)
	CenterWindow	ウィンドウを画面中央に配置 (true / false)
	Full Screen	フルスクリーンモードで開く (true / false)
	Display Doc Title	ウィンドウのタイトルバーに文書情報のタイトルを表示する (true / false)
	Hide Menubar	メニューバーを非表示 (true / false)
	Hide Toolbar	ツールバーを非表示 (true / false)
	Hide Window UI	ウィンドウコントロールを非表示 (true / false)
	Action	ファイルを開いたときの動作 「Action 情報 (p. 64)」参照
Security Information		セキュリティ設定内容 Not Encrypted. : セキュリティ設定なし
	Security Handler	セキュリティの扱い方 Standard Security : 標準セキュリティ
	Key Length	暗号化キーの長さ
	Permission Type1	40bit RC4 暗号化

		Print : 印刷の許可／不許可 (allowed / not allowed)
		Modify : 文書変更の許可／不許可 (allowed / not allowed)
		Copy : 内容のコピーの許可／不許可 (allowed / not allowed)
	Permission Type2	128bit RC4,128bit AES,256bit AES の暗号化
		Print : 印刷 not allowed.: 許可しない Low resolution.: 低解像度 High resolution.: 高解像度
		Modify: 変更 not allowed. : 許可しない Inserting, deleting, and rotating pages. : ページの挿入、削除、回転 Filling in form fields, and signing existing signature fields. : フォームフィールドの入力と既存の署名フィールドに署名 Commenting, filling in form fields, and signing existing signature fields. : 注釈の作成、フォームフィールドの入力と既存の署名フィールドに署名 Any except extracting pages. : ページの抽出を除くすべての操作
		Copy : 内容のコピー Enable copying of text, images, and other content. : 内容のコピーが有効 Disable copying of text, images, and other content. : 内容のコピーが無効
		Accesibility : アクセシビリティ Enable text access for screen render devices for the visually impaired. : スクリーンリーダーデバイスのテキストアクセスが有効 Disable text access for screen render devices for the visually impaired. : スクリーンリーダーデバイスのテキストアクセスが無効
Signature		署名が付いている (true / false)
PDF/A		PDF/A 文書である (true / false)
PDF/X		PDF/X 文書である (true / false)

## 11.3.37.2 -getOutlineInfo : しおり情報取得

項目名	内容
Title	しおり名
Indent	しおりの階層。1 階層めが 0
Style	しおり文字列の装飾の種類 normal : 標準 bold : 太字 italic : 斜体 bold & italic : 太い&斜体
Color	しおり文字列の色 (RGB)
Open	true : しおりは開いている (下の階層のしおりが見える状態) false : しおりは閉じている (下の階層のしおりは見えない状態)
Action	しおりがクリックされたときの動作 「Action 情報 (p. 64)」参照

### 11.3.37.3 -getAnnotInfo : 注釈情報取得

項目名		内容
Annot Type		注釈の種類 Link ; リンク Text : テキスト Stamp : スタンプ File Attachment : ファイル添付 Popup : ポップアップ
Link の場合	Highlighting mode	リンクがクリックされたときの表示モード None : なし Invert : リンクの長方形全体が反転する Outline : リンクの枠だけが反転する Push : リンク枠が押し出されるように表示される
	Action	リンクがクリックされたときの動作 「Action 情報 (p. 64)」参照
	※注釈基本情報	「【注釈基本情報】」参照
Text, Stamp の場合	Icon Type	アイコンの種類
	Icon Name	アイコンの名前
	※注釈基本情報	「【注釈基本情報】」参照
	※マークアップ注釈情報	「【マークアップ注釈情報】」参照
File Attachment の場合	Icon Type	アイコンの種類
	Icon Name	アイコンの名前
	File Name	添付ファイル名
	※注釈基本情報	「【注釈基本情報】」参照
	※マークアップ注釈情報	「【マークアップ注釈情報】」参照

## 【注釈基本情報】

項目名	内容
Rect	注釈の位置情報
TextContents	表示されるテキスト（存在する場合）
ModDate	注釈の更新日
Annot Flags	下記の表示状態を表すビットフラグ ※「on」の場合の状態について Invisible : 表示 Hidden : 注釈はレンダリングされない Print : 印刷対象 NoZoom : 注釈の外観はページ倍率に合わせて拡大・縮小されない NoRotate : 注釈の外観はページに合わせて回転しない NoView : 非表示 ReadOnly : 読み取り専用 Locked : 注釈の削除やプロパティの変更はできない ToggleNoView : マウスが注釈の上にあるとき、または注釈が選択されたときに注釈が表示されるようにする LockedContents : 注釈の内容を変更できない
Color	注釈の色 None : なし RGB CMYK
Border Style	境界線の種類 Solid : 実線 Dashed : 破線 Beveled : 浮き上がって見えるような境界線 Inset : 凹んで見えるような境界線 Underline : 下線 None : なし
Border Width	境界線の太さ Thin : 細 Middle : 中 Thick : 太 None : なし

## 【マークアップ注釈情報】

項目名	内容
Markup Title	注釈の「作成者」
Markup Subject	注釈の「タイトル」
CreationDate	注釈の作成日
Markup CA	不透明度 値の範囲 : 0.0~1.0。1.0 が不透明
Popup	exist : ポップアップ注釈を持つ not exist : ポップアップ注釈を持たない
※ポップアップ注釈の基本情報	ポップアップ注釈を持つ場合に出力される。 「【注釈基本情報】」参照

#### 11.3.37.4 Action 情報・Destination 情報

##### 【Action 情報】

「Action」の値	内容
GoTo	文書内のページを表示する。 「【Destination 情報】」参照
GoToR	別の PDF のページを表示する。 「【Destination 情報】」参照
Launch	PDF 以外のファイルに関連付けられているアプリケーションで開く。 file name : 表示するファイル名 new window true : ファイルは新しいウィンドウに表示される。 false : ファイルは同値字のウィンドウに表示される。 ※表示するファイルが PDF でない場合、new windows の設定は無視される。
URI	web リンクを開く。 uri : 表示するアドレス
none	動作はない（なにも行われない）
unknown	動作の種類が不明

##### 【Destination 情報】

項目名	内容
Markup Title	注釈の「作成者」
Markup Subject	注釈の「タイトル」
CreationDate	注釈の作成日
Markup CA	不透明度 値の範囲：0.0～1.0。1.0 が不透明
Popup	exist : ポップアップ注釈を持つ not exist : ポップアップ注釈を持たない
※ポップアップ注釈の基本情報	ポップアップ注釈を持つ場合に出力される。 「【注釈基本情報】」参照

## 11.4 コマンドの複数指定

コマンドラインにおいて、複数のコマンドを同時に指定することができます。

ただし、情報取得を行うコマンド、画像ファイルを PDF 化するコマンド、PDF/A 変換コマンド、PDF/A 準拠確認コマンド、ファイル分割コマンドは指定しても動作しません。

透かしコマンドについては、複数の種類を同時に指定した場合、コマンド上で最初に出現した透かし処理のみ実行されません。

処理対象をフォルダ指定した場合は、複数のコマンド指定はできません。

コマンドの指定順は任意ですが、実行される順番は以下の通りです。

- 1) -decrypt
- 2) -mergeFiles
- 3) -changeDocInfo
- 4) -pageRotate
- 5) -pageDelete
- 6) -insertEmptyPage
- 7) -changeViewInfo
- 8) -setPdfWatermark/-setImageWatermark/-setTextWatermark/-setColorWatermark
- 9) -writeText
- 10) -addPageNumber
- 11) -addEmbeddedFile
- 12) -exportAnnotsToFDF
- 13) -importAnnotsFromPDF
- 14) -optimize
- 15) -encrypt
- 16) -pageExtract
- 17) -embedFonts
- 18) -extractImage
- 19) -extractText
- 20) -extractEmbeddeFile
- 21) -importAnnnotsFormFDF

エラーが発生した場合、コマンドラインは保存を行わずに終了します。

「-pageDelete」、「-insertEmptyPage」は、1 回のコマンド指定内で複数回呼び出すことはできません。

### ■コマンド例

[Windows]

```
AHPDFToolCmd70.exe -encrypt -ownerPass test -filterType 4 -perms2 -print 1 -modify 0 -changeViewInfo -pageLayout 2
-setTextWatermark -text サンプル -colorText 1 0 0 -diagonal -font -name メイリオ -size 72 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -encrypt -ownerPass test -filterType 4 -perms2 -print 1 -modify 0 -changeViewInfo -pageLayout 2
-setTextWatermark -text サンプル -colorText 1 0 0 -diagonal -font -name ipa-gothic -size 72 -d /home/antenna/test/test.pdf -o /home/
```

antenna/sav/out.pdf

この例を実行した場合、-changeViewInfo、-setTextWatermark、-encryptの順に処理が行われます。

## 11.5 マクロ文字列

「-setTextWatermark」（テキスト透かし）と「-writeText」（テキスト追加）では、日付やファイル名、ページ番号などに置き換える「マクロ文字列」が使用できます。

マクロ文字列	内容	
#DATE()	「#DATE(」と「)」の間に以下の日付文字を設定します。この間にある他のマクロ文字列は無効となります。日付文字以外はそのまま出力されます。	
	[日付文字]	
	YYYY または yyyy	西暦
	MM または mm	月 (2桁表示)
	M または m	月
	DD または dd	日 (2桁表示)
	D または d	日
#PAGE_NUM	ページ番号	
#TOTAL_PAGES	総ページ数	
#FILE_NAME	ファイル名	
#FULL_PATH	ファイルのフルパス名	
#TITLE	文書情報：タイトル	
#SUB_TITLE	文書情報：サブタイトル	
#AUTHOR	文書情報：作成者	

### ■#DATE() の置換例

2021年5月31日の場合、

- #DATE(YYYY年MM月DD日) → 「2021年05月31日」
- #DATE(M/D) → 「5/31」

### ■コマンド例

[Windows]

```
AHPDFToolCmd70.exe -setTextWatermark -text "今日は何日? #DATE(YYYY/M/D)" -colorText 1 0 0 -diagonal -font -name メイリオ -size 52 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux / Amazon Linux2 X86]

```
AHPDFToolCmd70 -setTextWatermark -text "今日は何日? #DATE(YYYY/M/D)" -colorText 1 0 0 -diagonal -font -name ipa-gothic -size 52 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

## 11.6 エラー処理について

### 11.6.1 コマンドラインの戻り値

#### Windows

状態	戻り値
成功	0
パラメータエラー	-1
PdfTk (PDF Tool API の内部ライブラリ) のエラー	PdfTk のエラーコード 「PdfTk のエラーコード一覧 (p. 69)」参照
その他のエラー	-2

#### Linux / Amazon Linux2 X86

状態	戻り値
成功	0
エラー	1

### 11.6.2 エラーの内容について

コマンドラインの実行中にエラーが発生した場合は、標準出力にエラー内容を示すメッセージ文字列が出力されます。  
(「PdfTk のエラーコード一覧 (p. 69)」参照)

## 11.6.3 Pdftk のエラーコード一覧

エラーコード	エラーメッセージ
0	正常終了
10	Cannot find license file. ライセンスファイルが見つかりません。
11	License file is expired. 評価版ライセンスの有効期限が切れています。
12	License file is invalid. ライセンスファイルが無効です。
13	License file is for other platform. ライセンスファイルが他プラットフォーム用です。
14	License file is for other product. ライセンスファイルが他製品用です。
100	Invalid PDF file. PDF が異常です。
101	Cannot read PDF. PDF の読み込みができません。
102	Cannot write PDF. PDF の書き出しができません。
103	Cannot write too large PDF. 大きすぎる PDF の書き出しができません。
110	Invalid user password. ユーザーパスワード不正です。
111	Invalid owner password. オーナーパスワード不正です。
112	Invalid password. パスワード不正です。
113	Has not authority. 処理権限がありません。
114	Is not encrypted. 暗号化されていません。
115	Unsupported security handler. 未対応のセキュリティハンドラです。
116	Unsupported security algorithm. 未対応のセキュリティアルゴリズムです。
117	Is signed. 電子署名されています。
118	Has XFA(XML Form). XFA(XML Form) を持っています。
120 ※V7.0 追加	Can not certificate. 認証できません。
200	Invalid parameter value. パラメータに問題があります。
201	Invalid page number. ページの指定が間違っています。
202	Has no text. テキストの設定がありません。
203	Has no font. フォントの設定がありません。
204	Has no valid data. 有効なデータの設定がありません。
205 ※V7.0 追加	Cannot use this function. この関数は使えません。
210 *1	Need password. パスワードが必要です。

211 *2	Need user password. ユーザーパスワードが必要です。
212 *3	Need owner password. オーナーパスワードが必要です。
213	Invalid encrypt key length. 暗号化キー長が間違っています。
214	Invalid encrypt permission. 権限が間違っています。
215	Invalid encrypt component. 暗号化する文書コンポーネントに誤りがあります。
216	Invalid encrypt method. 暗号化メソッドが間違っています。
217 ※V7.0 追加	Need PKCS12. PKCS12 が必要です。
220	Can not read attached file. 添付ファイルの読み込みが出来ません。
221	Can not write attached file. 添付ファイルの書き出しが出来ません。
222	No attached file. 添付ファイルがありません。
223	Attached file has no name. 添付ファイルに名前がありません。
230	Can not read image file. 画像ファイルの読み込みが出来ません。
231	Can not write image file. 画像ファイルの書き出しが出来ません。
232	Unsupported image. 未対応の画像です。
233	Unsupported image for stencil mask. ステンシルマスクとしてサポートしていない画像です。
234	Image is not single. ステンシルマスクがモノクロ画像ではありません。
235	Unsupported image for color key mask. カラーキーマスクとしてサポートしていない画像です。
236	Unsupported image for explicit mask. 明示マスクとしてサポートしていない画像です。
237	Image is not single. 明示マスクがモノクロ画像ではありません。
238	Image is not gray scale. ソフトマスクとしてサポートしていない画像です。
240	Image processing error. イメージ処理で問題が発生しました。
241 ※V7.0 追加	Can not read ICC Profile. ICC プロファイルの読み込みが出来ません。
245	Font processing error. フォント処理で問題が発生しました。
250	Can not insert page. ページの挿入が出来ません。
251	Can not delete page. ページの削除が出来ません。
252	Has no pages. ページが存在しません。
260 *4	Free docproperty error. DocProperty がドキュメントからフリーです。
261 *4	Free openmode error. OpenMode がドキュメントからフリーです。
262 *4	Free embeddedfiles error. EmbeddedFiles がドキュメントからフリーです。

263 *4	Free pages error. Pages がドキュメントからフリーです。
264 *4	Free page error. Page がドキュメントからフリーです。
270	Can not set to root outline. ルートアウトラインには設定出来ません。
271 *5	Can not set to free outline. フリーアウトラインには設定出来ません。
280	Invalid FDF file. FDF が異常です。
281	Cannot read FDF. FDF の読み込みができません。
282	Cannot write FDF. FDF の書き出しができません。
290 ※V7.0 追加	Cannot read PKCS12. PKCS12 の読み込みができません。
291 ※V7.0 追加	Cannot read X509. X509 の読み込みができません。
500 *6	Linearize processing error. 線形化処理で問題が発生しました。
700	null value. null 値です。
800	No Object. オブジェクトが存在しません。
900	Not enough memory. メモリが不足しています。
901	Internal error. 内部エラーです。
902	Other error. その他のエラーです。
999	Sorry, not implemented. 未実装です。

## ※備考

\*1[210] PDF の暗号化設定処理においてパスワードの設定が行われていない場合に発生します。

\*2[211] 添付ファイルのみの暗号化設定処理において添付ファイルを開くためのパスワードが設定されていない場合に発生します。

\*3[212] PDF のセキュリティ権限フラグ設定処理において権限パスワードの設定が行われていない場合に発生します。

\*4[260/261/262/263/264] 取得されたオブジェクトが PDF とは紐づいていない場合に発生します。

\*5[271] しおりが作成できないことを示します。

\*6[500] 線形化処理 = Web 表示用に最適化する処理で問題が発生したことを意味します。

## 第 12 章 V4.0 / V5.0 / V6.0 との相違点

### 12.1 対応コマンド

コマンド名	V7.0	V6.0	V5.0	V4.0
-getFileInfo	○	○	○	○
-getFileInfo	○	-	-	-
-changeDocInfo	○	○	○	○
-changeViewInfo	○	○	○	○
-mergeFiles	○	○	○	○
-getOutlineInfo	○	○	○	○
-getAnnotInfo	○	○	○	○
-exportAnnotsToFDF	○	○	-	-
-importAnnotsFromPDF	○	○	-	-
-setPdfWatermark	○	○	○	○
-setImageWatermark	○	○	○	○
-setTextWatermark	○	○	○	○
-setColorWatermark	○	○	○	-
-encrypt	○	○	○	○
-decrypt	○	○	○	○
-optimize	○	○	○	-
-optimize -mergeFonts	○	○	-	-
-addEmbeddedFile	○	○	-	-
-pageExtract	○	○	-	-
-pageDelete	○	○	-	-
-insertEmptyPage	○	○	-	-
-pageRotate	○	○	-	-
-addPageNumber	○	○	-	-
-writeText	○	○	-	-
-embedFonts	○	○	-	-
-extractImage	○	-	-	-
-extractText	○	-	-	-
-imageToPdf	○	-	-	-
-extractEmbeddedFile	○	-	-	-
-importAnnotsFromFDF	○	-	-	-
-fixedUpPDFA	○	-	-	-
-validatePDFA	○	-	-	-
-dividePdf	○	-	-	-
-linearize	○	○	○	○
-lic	○	-	-	-

## 12.2 PDFの情報取得 (-getFileInfo) の変更点

V7.0では、これまでの取得情報に加え、

- 署名の有無
  - PDF/A か否か
  - PDF/X か否か
- を取得します。

## 12.3 PDFセキュリティ設定 (-encrypt) の変更点

- V7.0では、40bit RC4の暗号化処理を廃止しました
- V7.0の256bit AES暗号化処理は「R (リビジョン) 6」仕様です。V6.0以下では「R5」仕様です。

## 12.4 フォルダ指定

V7.0では一部のコマンドを除き、フォルダ指定が可能です。指定されたフォルダにあるPDFファイルあるいは画像ファイルに対し処理を実行します。

## 12.5 マクロ文字列

V7.0では、日付やページ番号などに置換するマクロ文字列を定義しました。テキスト透かし設定 (-setTextWatermark)、文字列追加 (-writeText) の各コマンドで使用できます。

## 第 13 章 商標／著作権情報

### 13.1 商 標

Microsoft、Windows、OpenType は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Adobe、Acrobat、および Reader は、アドビ システムズ社の米国ならびに他の国における登録商標または商標です。

TrueType は米国その他の国で登録された米国アップルコンピュータ社の商標です。

その他記載されている全ての会社名および製品名は、個々の所有者の登録商標または商標です。

### 13.2 第三者ライブラリー著作権情報

#### \* jpeg

This product includes software developed by Independent JPEG Group

(<http://www.ijg.org/>).

Copyright (c) 1991-1998 Thomas G. Lane.

#### \* libtiff

This product includes software developed by Sam Leffler and Silicon Graphics, Inc (<http://www.libtiff.org/>).

Copyright (c) 1988-1997 Sam Leffler

Copyright (c) 1991-1997 Silicon Graphics, Inc.

#### \* zlib

This product includes software developed by Jean-loup Gailly and Mark Adler

(<https://zlib.net/>).

Copyright (C) 1995-2002 Jean-loup Gailly and Mark Adler

#### \* libpng

This product includes software developed by Glenn Randers-Pehrson and many other contributors (<http://www.libpng.org/pub/png/libpng.html>).

Copyright (c) 2004, 2006-2011 Glenn Randers-Pehrson

Copyright (c) 2000-2002 Glenn Randers-Pehrson

Copyright (c) 1998, 1999 Glenn Randers-Pehrson

Copyright (c) 1996, 1997 Andreas Dilger

Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

#### \* j2k(jasper)

JasPer License Version 2.0

Copyright (c) 1999-2000 Image Power, Inc.

Copyright (c) 1999-2000 The University of British Columbia

Copyright (c) 2001-2003 Michael David Adams

All rights reserved.

\* LittleCMS

Little Color Management System

Copyright (c) 1998-2016 Marti Maria Saguer

\* ICU

This product includes softwares developed by:

International Business Machines Corporation

International Components for Unicode (ICU) libraries

(<https://icu.unicode.org/>)

Copyright (c) 1995-2010 International Business Machines Corporation and others All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

\* FreeType

This product includes softwares developed by:

Portions of this software are copyright 2017 The FreeType Project ([www.freetype.org](http://www.freetype.org)).

All rights reserved.

\* OpenSSL

Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.



## 履歴

日付	内容
2021.4.27	初版
2021.6.29	・Linux 版の記述を追加しました。 ・「取得情報の出力内容 (p. 59)」を追加しました。
2021.9.16	・「コマンドの複数指定」の説明を修正しました。
2021.11.15	・「-linearize : Web 表示用に最適化 (p. 57)」の説明を追加しました。
2021.12.16	・「-extractText : テキスト抽出」コマンドに「-sort」コマンドを追加しました。
2022.6.29	・「13.2」の項目名を「第三者ライブラリー著作権情報」とし、一部修正しました。
2023.2.28	・「11.1.1 ファイル指定」と「11.1.2 フォルダ指定」の説明を修正しました。
2023.3.15	・「動作環境」に「Amazon Linux2 X86」を追加しました。
2023.8.30	・「11.1.1 ファイル指定」と「11.1.2 フォルダ指定」の説明を修正しました。 ・「11.3.7 -setPdfWatermark」「11.3.8 -setImamgeWatermark」「11.3.9 -setTextWatermark」の「-tiling」の説明を修正しました。
2023.9.29	・コマンド「-fixedUpPDFA : PDF/A-1b,2b 変換 (p. 54)」「-validatePDFA : PDF/A-1b,2b 準拠確認 (p. 55)」「-dividePdf : ファイル分割 (p. 56)」を追加しました。(改訂 7a)
2024.1.25	・コマンド「-mergeFiles : PDF ファイル結合 (p. 22)」に3つのパラメータ「-dmerge」、「-copyAnnots」、「-copyOutlines」を追加しました。(改訂 8) ・「2 動作環境」-「2.1 Windows」を修正しました。
2024.3.27	・「第三者ライブラリー著作権情報」において一部の URL アドレスを修正しました。

