

Antenna House PDF Tool API V8.0 コマンドライン説明書



索引

目次

ANTENNA HOUSE PDF TOOL API V8.0 コマンドライン説明書	1
第1章 概要 コマンドラインアプリケーションについて	1
第2章 動作環境	2
2-1 WINDOWS	2
2-2 LINUX	2
第3章 インストール / アンインストール	4
3-1 WINDOWS版	4
3-2 LINUX版	9
第4章 ライセンスファイルについて	13
4-1 ライセンスファイルの区分と仕様	13
4-2 ライセンスファイルの入手	14
4-3 ライセンスファイルの参照先指定	15
4-4 ライセンスファイルの入れ替えについて	17
4-5 保守契約期限内における改訂版アップデートとバージョンアップについて	17
4-6 保守契約期限と改訂版アップデートにおけるコマンドラインアプリケーションの動作について	18
4-7 ライセンスファイルの情報確認について	19
第5章 製品仕様	21
5-1 処理対象 PDF ファイルについて	21
5-2 フォントについて	21
5-3 カラープロファイルの扱い	26
5-4 読み書きの制限サイズ	28
5-5 PDF/A、PDF/X および PDF/E に対する編集について	29
5-6 PDF の最適化について	29
5-7 セキュリティ情報の保持について	29
5-8 複数スレッドの利用について	30
5-9 文字の抽出・検索における「-SORT」指定について	30
5-10 コマンドプロンプトにおける字数制限	30
第6章 評価版動作仕様	31
第7章 コマンドラインアプリケーションの使い方	32
7-1 実行方法	32
7-2 入出力ファイル・フォルダの指定	36

7-3	コマンド一覧.....	40
第8章	各種コマンド	43
8-1	-GETFILEINFO : PDF の情報取得.....	44
8-2	-CHANGEDOCINFO : 文書プロパティ設定.....	47
8-3	-CHANGEVIEWINFO : 開き方設定	49
8-4	-MERGEFILES : PDF ファイル結合.....	53
8-5	-GETOUTLINEINFO : しおり情報取得	56
8-6	-GETANNOTINFO : 注釈情報取得	58
8-7	-SETPDFWATERMARK : PDF ファイル透かしの挿入	60
8-8	-SETIMAGEWATERMARK : 画像ファイル透かしの挿入	65
8-9	-SETTEXTWATERMARK : テキスト透かしの挿入.....	70
8-10	-SETCOLORWATERMARK : 色透かしの挿入	76
8-11	-ENCRYPT : PDF セキュリティ設定.....	80
8-12	-DECRYPT : PDF セキュリティ削除.....	84
8-13	-OPTIMIZE : PDF の最適化.....	86
8-14	-ADDEMBEDDEDFILE : ファイル添付	89
8-15	-PAGERotate : ページ回転.....	91
8-16	-PAGEDELETE : ページ削除.....	93
8-17	-WRITETEXT : 文字列追加.....	95
8-18	-EXPORTANNOTSToFDF : 注釈情報を FDF へエクスポート	99
8-19	-IMPORTANNOTSFROMPDF : 他 PDF からの注釈のインポート	101
8-20	-INSERTEMPTYPAGE : 空白ページ挿入	103
8-21	-ADDPAGENUMBER : ページ番号挿入	105
8-22	-PAGEEXTRACT : ページ抽出	108
8-23	-EMBEDFONTS : フォント埋め込み	110
8-24	-EXTRACTIMAGE : 画像抽出	112
8-25	-EXTRACTTEXT : テキスト抽出	116
8-26	-IMAGETOPDF : 画像ファイルの PDF 化.....	119
8-27	-EXTRACTEMBEDDEDFILE : 添付ファイルの抽出	122
8-28	-IMPORTANNOTSFROMFDF : FDF からの注釈のインポート	124
8-29	-FIXEDUPPDFA : PDF/A 変換.....	126
8-30	-VALIDATEPDFA : PDF/A 準拠確認.....	130
8-31	-DIVIDEPDF : ファイル分割.....	133
8-32	-GETPAGESIZE : ページサイズの取得.....	136
8-33	-SEARCHTEXTANDHIGHLIGHT : テキスト検索後にハイライト注釈挿入	138
8-34	-SEARCHTEXTANDREDACT : テキスト検索後に注釈挿入	141

8-35	-SEARCHTEXTANDSETMASK : テキスト検索後にマスクをかける	144
8-36	-ZOOM : ページを拡大・縮小する	147
8-37	-SETCUSTOMPROPERTY : カスタムプロパティの設定	149
8-38	-SETMASKUSINGREDACTION : REDACT 注釈を適用しマスクをかける	151
8-39	-LINEARIZE : WEB 表示用に最適化	153
8-40	-LIC : ライセンス情報の取得	155
8-41	-V : バージョン情報・著作権情報・ライセンス情報の表示	157
8-42	-H : コマンド名一覧表示	159
8-43	-H -ALL : 全コマンドオプション説明の表示	161
8-44	-H <コマンド名> : 個別コマンドオプション説明の表示	163
第 9 章	取得情報の出力内容	164
9-1	-GETFILEINFO : PDF の情報取得	164
9-2	-GETOUTLINEINFO : しおり情報取得	168
9-3	-GETANNOTINFO : 注釈情報取得	169
9-4	-GETPAGESIZE : ページサイズ情報取得	173
9-5	ACTION 情報・DESTINATION 情報	174
第 10 章	コマンドラインアプリケーション操作の仕様	177
10-1	コマンドの複数指定	177
10-2	マクロ文字列	179
10-3	エラー処理について	181
第 11 章	V4.0 / V5.0 / V6.0 / V7.0 との相違点	188
11-1	コマンド対応バージョン一覧	188
11-2	各コマンドの変更点	189
第 12 章	バージョンアップについて	193
12-1	保守契約期限内のバージョンアップについて	193
12-2	バージョンアップとライセンスに関する制限事項	194
12-3	複数バージョンのインストールについて	194
第 13 章	改訂版について	196
13-1	保守契約期限内の改訂版へのアップデートについて	196
13-2	改訂版のインストール方法について	196
第 14 章	よくいただくご質問	198
第 15 章	商標／著作権情報	199
15-1	商標	199
15-2	第三者ライブラリー著作権情報	199

索引	205
改訂履歴	210
奥付	211

第1章 概要 コマンドラインアプリケーションについて

『Antenna House PDF Tool API V8.0』のコマンドラインアプリケーションは、コンソール上で実行ファイルを呼び出し、コマンドとオプションを組み合わせて指定して実行されます。このコマンドラインアプリケーションはコマンドライン版、ライブラリ版双方の『PDF Tool API』に同梱されます。

ライブラリ版と比べて処理内容は限定されますが、コマンドラインアプリケーションを使えば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。特に、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利用できることはコマンドラインアプリケーションの大きな特徴です。

コマンドラインアプリケーションは 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簡単に利用できます。

本マニュアルでは動作環境、インストール方法の他、各コマンドの仕様について解説します。

『PDF Tool API』のコマンドラインアプリケーションが利用しているのは、ライブラリ版が持つ機能の一部です。複雑な機能を利用したい場合ライブラリ版の利用を検討してください。

第2章 動作環境

『PDF Tool API』コマンドラインアプリケーションの動作環境は以下の通りです。

2-1 Windows

OS の種類	Microsoft Windows 11 日本語版 Microsoft Windows Server 2022 / 2025 日本語版 注：ARM 版 Windows11 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++ 2022 ランタイムライブラリ (※1)

(※1) ランタイムライブラリの入手先：

『サポートされている最新の Visual C++ 再頒布可能パッケージのダウンロード』—Microsoft

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

2-2 Linux

OS の種類	X86 64bit (※2)
必要なランタイムライブラリ	glibc 2.34 libstdc++.so.6.0.29 もしくはこれらと互換のあるライブラリ 注：GCC 11.4 でビルドされています。

(※2) Rocky Linux 9、Amazon Linux 2023 で動作確認を行っています。

第3章 インストール / アンインストール

『PDF Tool API』のコマンドラインアプリケーションはコマンドライン版、ライブラリ版、双方の『PDF Tool API』に同梱されています。

そのため、本章ではコマンドライン版、ライブラリ版それぞれのインストール方法・アンインストール方法を解説します。

そのほか、インストールフォルダの構成やインストーラにより設定される環境変数などの情報も解説しています。

本書における「フォルダ」表記について

本マニュアルの本項以降ではフォルダ／ディレクトリのことを「フォルダ」と総称して記述します。たとえば、フォルダ／ディレクトリパスのことは「フォルダパス」と表記します。

Linuxで操作される方は以下項目では「フォルダ」を「ディレクトリ」と読み替えてお読みください。

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]

- [AHPDFToolCmd_V80_***_x64.exe]がインストーラです。（ファイルをダブルクリックするなどして起動してください）
「***」にはR1、MR1などの改訂バージョン名が入ります。
- [AHPDFToolCmd_V80_***_x64.exe]の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}:¥AHPDFToolCmd_80

新規インストールが完了すると、インストール後 30 日間有効な評価版ライセンスが配置されます。評価版ライセンスの期限切れ、もしくはライセンス正規版に入れ替えるまでコマンドラインアプリケーションは評価版として動作します。評価版における制限などは『第 6 章 評価版動作仕様』を参照してください。

[ライセンス版の場合]

1. パッケージ内の[AHPDFTool_V80_***_x64.exe]がインストーラです。（ファイルをダブルクリックするなどして起動してください）
「***」には R1、MR1 などの改訂バージョン名が入ります
2. [AHPDFTool_V80_***_x64.exe]の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}:¥AHPDFToolLib_80

新規インストールが完了すると、インストール後 30 日間有効な評価版ライセンスが配置されます。評価版ライセンスの期限切れ、もしくはライセンス正規版に入れ替えるまでコマンドラインアプリケーションは評価版として動作します。評価版における制限などは『第 6 章 評価版動作仕様』を参照してください。

Windows 版の上書きインストールについて

Windows 版では上書きインストールが可能です。改訂版の『PDF Tool API』をインストールする際は旧バージョンが残ったままインストーラを実行することができます。

上書きインストールを実行した場合の各ファイルは以下のように置換されます。

- 実行バイナリ・ファイル：日付とファイルバージョンを確認して新しいものに置換
- 非実行ファイル（ヘッダファイルなど）：日付を確認して新しいものに置換
- ライセンスファイル：置換しない（入替えの必要がある場合は手動での置換が必要）

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」を開く
2. プログラム一覧にある「Antenna House PDF Tool API V8.0 Windows コマンドライン」を選択
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

[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」を開く
2. プログラム一覧にある「Antenna House PDF Tool API V8.0 Windows ライブラリ」を選択
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

3-1-3 Microsoft Visual C++ランタイムライブラリについて

『PDF Tool API V8.0』の動作には、Microsoft Visual C++ 2022 ランタイムライブラリが必要です。

インストール時に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、自動的に「Microsoft Visual C++ 2022 再頒布パッケージ」のインストーラが起動します。

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してください。

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
|-- bin : コマンドラインアプリケーションモジュール
|-- CopyRightFiles : 第三者ライブラリの著作権情報
|-- fontconfig : フォント構築ファイル
|-- icc : カラープロファイル用 icc ファイル格納フォルダ
|-- License : ライセンスファイル
|-- 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
|-- bin : ライブラリモジュール : コマンドラインアプリケーションを含みます
|-- CopyRightFiles : 第三者ライブラリの著作権情報
|-- Include : ヘッダファイル : コマンドラインアプリケーションでは使用しません
|-- fontconfig : フォント構築ファイル
|-- icc : カラープロファイル用 icc ファイル格納フォルダ
|-- lib : lib ファイル : コマンドラインアプリケーションでは使用しません
|-- License : ライセンスファイル
|-- 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

インストーラが値を設定する環境変数

インストール時、以下の環境変数について、インストーラにより値が設定されます。環境変数が無かった場合は環境変数が作成されます。

- 環境変数名 : PTL80_LIC_PATH
設定値 : {インストールフォルダ}\License

- 環境変数名 : PTL80_FONT_CONFIGFILE
設定値 : {インストールフォルダ}¥fontconfig¥font-config.xml
- 環境変数名 : PTL80_ICCPROFILE_PATH
設定値 : {インストールフォルダ}¥icc

ダイアログで指定時にインストーラが値を設定する環境変数

以下の環境変数は、インストーラ実行時にダイアログで「環境変数に追加する」を指定した場合にインストーラにより値が設定・追加されます。

- 環境変数名 : PATH
設定値 : 「{インストールフォルダ}¥bin」を追加する

[PTL80_LIC_PATH]に関しては『第4章 ライセンスファイルについて』を参照してください。

[PTL80_FONT_CONFIGFILE]に関しては 『5-2-2 描画とフォント埋め込みに使用するフォントの参照先について』を参照してください。

[PTL80_ICCPROFILE_PATH]に関しては 『5-3 カラープロファイルの扱い』を参照してください。

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

インストールされるライセンスファイルに関しては以下の注意事項があります。

- 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日間有効の評価版用ライセンスです。
- 期限を過ぎると、『PDF Tool API』は利用不可になります。

- 評価版ライセンスで実行される『PDF Tool API』では、出力される PDF ファイルの各ページに透かし文字列が挿入されます。
 - その他、評価版における制限などは『第 6 章 評価版動作仕様』を参照してください。
-

3-2 Linux 版

3-2-1 インストール方法

[コマンドライン版の場合]

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 AHPDFToolCmd80-8.0-***.x86_64.rpm [--prefix インストールパス]
```

「***」には R1、MR1 などの改訂バージョン名が入ります。

3. インストールパスを指定しなかった場合、「/usr/AHPDFToolCmd80」にインストールされます。
4. コマンドラインアプリケーションは、『PDF Tool API』を初めて使用した時点から 30 日間有効な評価版としてインストールされます。評価版における制限などは『第 6 章 評価版動作仕様』を参照してください。

[ライブラリ版の場合]

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 AHPDFToolLib80-8.0-***.x86_64.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/AHPDFToolLib80」にインストールされます。
4. コマンドラインアプリケーションは、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。評価版における制限などは『第 6 章 評価版動作仕様』を参照してください。

Linux 版は上書きインストールできない

Linux 版では『PDF Tool API』を上書きインストールすることができません。そのため、再インストールをする際や改訂版の『PDF Tool API』をインストールする際は、その前に古い『PDF Tool API』をアンインストールしてください。

3-2-2 アンインストール方法

[コマンドライン版の場合]

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 AHPDFToolCmd80
```

3. シェルスクリプトなどで環境変数 [PTL80_LIC_PATH]、[PTL80_FONT_CONFIGFILE]、[PTL80_ICCPROFILE_PATH] を設定している場合は元に戻します。その他の環境変数を設定していた場合も同様に記述を削除するなどしてください。

[PTL80_LIC_PATH] に関しては『第 4 章 ライセンスファイルについて』を参照してください。

[PTL80_FONT_CONFIGFILE] に関しては『5-2-2 描画とフォント埋め込みに使用するフォントの参照先について』を参照してください。

[PTL80_ICCPROFILE_PATH] に関しては『5-3 カラープロファイルの扱い』を参照してください。

[ライブラリ版の場合]

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 AHPDFToolLib80
```

3. シェルスクリプトなどで環境変数 [PTL80_LIC_PATH]、[PTL80_FONT_CONFIGFILE]、

[PTL80_ICCP PROFILE_PATH]を設定している場合は元に戻します。その他の環境変数を設定していた場合も同様に記述を削除するなどしてください。

[PTL80_LIC_PATH]に関しては『第4章 ライセンスファイルについて』を参照してください。

[PTL80_FONT_CONFIGFILE]に関しては『5-2-2 描画とフォント埋め込みに使用するフォントの参照先について』を参照してください。

[PTL80_ICCP PROFILE_PATH]に関しては『5-3 カラープロファイルの扱い』を参照してください。

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
|-- CopyRightFiles : 第三者製ライブラリの著作権情報ファイル
|-- fontconfig : フォント構築ファイル
|-- icc : カラープロファイル用 icc ファイル格納フォルダ
|-- lib : PDF Tool API モジュール
|-- License : ライセンスファイル
|-- ToolCmd : コマンドラインアプリケーション実行ファイル
|-- EULA.txt : 使用許諾契約書
|-- run.sh : コマンドラインアプリケーション実行スクリプト
```

[ライブラリ版]

```
インストールフォルダ
|-- bin : Java API 用ファイル : コマンドラインアプリケーションでは使用しません
|-- CopyRightFiles : 第三者製ライブラリの著作権情報ファイル
|-- fontconfig : フォント構築ファイル
|-- icc : カラープロファイル用 icc ファイル格納フォルダ
|-- lib : PDF Tool API モジュール
```

```
|-- License : ライセンスファイル  
|-- ToolCmd : コマンドラインアプリケーション実行ファイル  
|-- EULA.txt : 使用許諾契約書  
|-- run.sh : コマンドラインアプリケーション実行スクリプト
```

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

インストールされるライセンスファイルに関しては以下の注意事項があります。

- 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、30日間有効の評価版用ライセンスです。Linux版の場合、最初に『PDF Tool API』を使用した日から30日間使用可能です。
 - 期限を過ぎると、『PDF Tool API』は利用不可になります。
 - 評価版ライセンスで実行される『PDF Tool API』では、出力されるPDFファイルの各ページに透かし文字列が挿入されます。
 - その他、評価版における制限などは『第6章 評価版動作仕様』を参照してください。
-

第4章 ライセンスファイルについて

『PDF Tool API』を実行する際にはライセンスファイルが必要です。

ライセンスファイルは[ptalic.dat]の名前を持つファイルです。『PDF Tool API』が参照した先にライセンスファイルが存在しない場合、『PDF Tool API』はエラーを出力して動作を停止します。

動作停止の際のエラーコード、及びエラーに関しては『10-3 エラー処理について』を参照してください。

本章ではライセンスファイルの仕様・入手・参照先指定・入れ替えについて説明します。

4-1 ライセンスファイルの区分と仕様

ライセンスファイルは、評価版ライセンスと正規版ライセンスの2種類に大別されます。それぞれの仕様は以下の通りです。

評価版ライセンス：

30日の期限付きの評価用のライセンスです。

インストーラによって後述のインストールフォルダ内に自動的に配置されます。その他、透かしが入るなど、評価版の動作には制限事項があります。

評価版の制限事項についての詳細は『第6章 評価版動作仕様』を参照してください。

評価版ライセンスの正確な動作期限は、以下の通りです。

- Windows版 : インストール日から30日
- Linux版 : 最初に利用した日から30日

インストーラによって配置される評価版ライセンスファイル[ptalic.dat]のフォルダパスは以下です。

ライセンスファイルの配置先:{インストールフォルダ}\License

正規版ライセンス：

『PDF Tool API』を正規購入した際に得られるライセンスです。ファイル名は評価版と同じく[ptalic.dat]です。

正規版ライセンスには動作期限はないので無期限に使用できます。ライセンスファイルには保守期限が設けられています。保守契約期限内であれば製品をアップデートすることができます。

- 保守期限に関する詳細は『4-6 保守契約期限と改訂版アップデートにおけるコマンドラインアプリケーションの動作について』を参照してください。
- 『PDF Tool API』がメジャーバージョンアップ及びマイナーバージョンアップをする場合はライセンスファイルの入れ替えが必要です。詳細は『12-1 保守契約期限内のバージョンアップについて』を参照してください。
「メジャーバージョンアップ」及び「マイナーバージョンアップ」の定義については『第12章 バージョンアップについて』を参照してください。

注意：

- ・ 1つの正規版ライセンスで運用可能な『PDF Tool API』のバージョンは1つだけです。詳細は『12-2 バージョンアップとライセンスに関する制限事項』を参照してください。

4-2 ライセンスファイルの入手

- 『PDF Tool API』を新規にご購入の場合正規版のライセンスファイルはインストーラとは別途で弊社よりお送りします。

- 保守契約を更新された際には、その都度弊社より新しいライセンスファイルを送付いたします。
 - バージョンアップをご希望の場合、該当バージョン用のライセンスファイルが必要です。その他バージョンアップ手続きに関する詳細は『12-1 保守契約期限内のバージョンアップについて』を参照してください。
-

4-3 ライセンスファイルの参照先指定

『PDF Tool API』は主に環境変数[PTL80_LIC_PATH]を用いてライセンスファイルを参照します。

この参照先の指定方法は大きく3つあり、以下の方法となります。

- インストーラによる参照先の指定：
これが最も簡単な方法です。詳細は『4-3-1 インストーラによる参照先指定』を参照してください。
- 環境変数による任意の位置の指定：
インストールフォルダ以外の位置を指定したい場合に用います。詳細は『4-3-2 任意の参照先指定』を参照してください。
- 環境変数を用いないライセンスファイル参照 (Windows版のみ)
モジュールファイルと同じフォルダにライセンスファイルを配置することで環境変数を用いずに参照することができます。詳細は『4-3-3 環境変数を用いない参照先 (Windows版のみ)』を参照してください。

正規版ライセンスを購入した場合などはライセンスファイルの更新が必要です。「4-4 ライセンスファイルの入れ替えについて」を参照して参照先のライセンスファイルを入れ替えてください。

4-3-1 インストーラによる参照先指定

インストーラを用いてインストールした場合、評価版のライセンスファイル[ptalic.dat]はインストーラによってインストールフォルダ内に配置されます。

評価版ライセンスファイルが配置されるのは、具体的には以下のフォルダパスです。

ライセンスファイルの配置先：{インストールフォルダ}\License

Windows 版の場合はインストーラ実行時に環境変数[PTL80_LIC_PATH]も同時に設定されます。

そのため、例えばインストール後すぐに評価版ライセンスを使いたい場合はコマンドラインアプリケーションをそのまま使い始めることができます。

Linux 版では Windows 版とは異なりインストーラが環境変数の設定を行いません。

その代わりに、インストール先情報が設定されたファイル[run.sh]がインストーラによって配置されます。

[run.sh]はインストール先情報を持った環境変数を設定した上で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です。詳細は『シェルスクリプト[run.sh]について』を参照してください。

[run.sh]を使用しない場合は後記の『4-3-2 任意の参照指定』を参考に、ライセンスファイルの配置先を環境変数[PTL80_LIC_PATH]に指定してください。

4-3-2 任意の参照先指定

任意の位置にあるライセンスファイルを参照したい場合、環境変数[PTL80_LIC_PATH]の値を指定する必要があります。

具体的な手順は以下の方法です。

1. ライセンスファイル[ptalic.dat]を任意の場所に配置します。
 2. 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：PTL80_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

4-3-3 環境変数を用いない参照先 (Windows 版のみ)

Windows 版の場合は、『PDF Tool API V8.0』のモジュールファイル[PdfTk80.dll]と同一の場所にライセン

スファイルを配置して使用することも可能です。

このとき、環境変数「PTL80_LIC_PATH」は必要ありません。

4-4 ライセンスファイルの入れ替えについて

新規ご購入や保守契約の延長に伴い保守窓口から入手したライセンスファイルを有効にするには、古いライセンスファイルを新しいライセンスファイルに入れ替える必要があります。

入れ替え方法は以下のいずれかの方法から選択できます。

- 古いファイルを新しいライセンスファイルで上書きする方法：
これが最も簡単な方法です。(各ライセンスファイルは[ptalic.dat]で同一名です)
- 任意の位置に新しいライセンスファイルを配置する方法。
この場合は、環境変数で指定するフォルダパスを新しいものに変更してください。
指定する環境変数に関する詳細は『4-3-2 任意の参照先指定』を参照してください。
この場合、古いライセンスファイルは元のフォルダに残されます。

補足事項：

- 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の入れ替えは、ライセンスファイルの入れ替えた次の実行時から反映されます。マシンの再起動の必要はありません。

4-5 保守契約期限内における改訂版アップデートとバージョンアップについて

新規ご購入頂いたライセンスファイル及び保守契約を延長されたライセンスファイルには、保守契約期間情報が記入されています[*1]。

保守契約期間内に改訂版や新バージョンがリリースされた場合、改訂版へのアップデートやバージョンアップを行うことができます。改訂版へのアップデートとバージョン変更を伴うバージョンアップでは、ラ

イセンスファイルの扱いに違いがあります。

- 保守契約期限内にリリースされた『PDF Tool API』の改訂版[*2]へアップデートする場合、現在お使いのライセンスファイルをご利用いただけます。詳細は『13-1 保守契約期限内の改訂版へのアップデートについて』を参照してください。
- 保守契約期限内に『PDF Tool API』の新バージョンがリリースされた場合のバージョンアップ[*3]に関しては、該当バージョンに合わせたライセンスファイルの入手が必要です。バージョンアップに関する製品仕様などの詳細は『12-1 保守契約期限内のバージョンアップについて』を参照してください。
- 改訂版へのアップデートやバージョンアップは必須ではありません。古いバージョンのままの使用も可能です。

[*1] :

ライセンスファイルの保守契約期間はライセンスファイルから読み取ることが可能です。具体的な操作については『4-7 ライセンスファイルの情報確認について』を参照してください。

[*2] :

「改訂版」の定義については『第 13 章 改訂版について』を参照してください。

[*3] :

バージョンアップの定義や詳細については『第 12 章 バージョンアップについて』を参照してください。

4-6 保守契約期限と改訂版アップデートにおけるコマンドラインアプリケーションの動作について

改訂版へのアップデートにおいて、現在ご利用中のライセンスファイルをそのままご利用頂けます。

しかし、アップデート先の改訂版が保守契約期間内にリリースされたものか否かでコマンドラインアプリケーションの動作が異なります。

- 保守契約期限内にリリースされた『PDF Tool API』の改訂版に関してはアップデートしてそのままご利用いただけます。
- 保守契約期限後にリリースされた改訂版にアップデートした場合、コマンドラインアプリケーションは評価版[*1]として動作します。評価版では出力される評価用の透かしが挿入されるなどの制限事項

が発生します。

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。

- 保守契約期限後も『PDF Tool API』をアップデートしない場合は引き続き正規版としてご利用いただけます。

[*1] :

評価版の制限事項についての詳細は『第6章 評価版動作仕様』を参照してください。

4-7 ライセンスファイルの情報確認について

コマンドラインアプリケーションを利用して、ライセンスファイル情報と使用中の『PDF Tool API』のバージョン情報・著作権情報が確認できます。指定したファイルパスのライセンスファイルの情報を取得することも可能です。

コマンド操作の詳細は『8-41 _v：バージョン情報・著作権情報・ライセンス情報の表示』を参照してください。

コマンドラインアプリケーションと同じ場所、あるいは環境変数「PTL80_LIC_PATH」にあるライセンスファイルの情報と、『PDF Tool API』のバージョン確認を行う場合

Windows版

```
AHPDFToolCmd80.exe -v
```

Linux版

```
AHPDFToolCmd80 -v
```

指定したファイルパスにあるライセンスファイルの情報と、

『PDF Tool API』 のバージョン確認を行う場合

Windows 版

```
AHPDFToolCmd80.exe -v C:\ahlic\ptalic.dat
```

Linux 版

```
AHPDFToolCmd80 -v /home/ahlic/ptalic.dat
```

第5章 製品仕様

本章では『PDF Tool API』を利用した加工・編集における製品仕様を扱います。

5-1 処理対象 PDF ファイルについて

処理可能な PDF バージョン

- 『PDF Tool API』で取扱い可能な PDF ファイルは、「ISO 32000 (32000-1, 32000-2)」の仕様に準拠したファイルです。
ISO に準拠していない PDF の場合、『PDF Tool API』が正常に動作しない場合があります。

入出力ファイルの PDF バージョンについて

- 出力 PDF に任意の PDF バージョン番号を指定することはできません。
 - 出力 PDF のバージョンは原則、入力 PDF と同じものになります。
例えば PDF 同士の結合など複数のファイルを入力 PDF として処理した場合、出力ファイルの PDF バージョンは入力ファイルの中で一番大きなバージョン番号と同じになります。
 - 暗号化セキュリティを設定して保存する際に PDF のバージョンが大きくなる場合があります。詳細は『[8-11 -encrypt : PDF セキュリティ設定](#)』を参照してください。
-

5-2 フォントについて

『PDF Tool API』で扱うフォントの仕様について説明します。

5-2-1 『PDF Tool API』におけるフォントを扱う機能

『PDF Tool API』におけるフォントを扱う機能は以下の通りです。

- PDF に文字列を描画する：
 - 8-9 -setTextWatermark : テキスト透かしの挿入
 - 8-17 -writeText : 文字列追加
 - 8-21 -addPageNumber : ページ番号挿入
- PDF 上に存在するフォントを埋め込み状態にする：
 - -embedFonts : フォント埋め込み
- PDF 上に存在するフォントを統合する：
 - -optimize : PDF の最適化において、パラメータ「-mergeFonts」を有効にする

5-2-2 描画とフォント埋め込みに使用するフォントの参照先

について

文字列の描画処理とフォントの埋め込み処理では、動作環境に存在するフォントを使用します。

フォント情報の参照先や、参照するための設定について説明します。

[フォントの参照先]

<Windows>

動作環境のフォントフォルダに存在するフォントを使用します。

フォントフォルダは、以下のパスに存在します。

{システムドライブ}:¥WINDOWS¥Fonts

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」の設定が必要です。

<Linux(X86)>

使用するフォントの配置フォルダを「フォント構築ファイル」で設定する必要があります。

[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したい場合、フォルダパスをフォント構築ファイル内に記入・設定する必要があります。

フォント構築ファイルは「font-config.xml」の名前を持つ XML ファイルです。Windows 版、Linux 版を問わず、インストール時にインストーラによって以下のフォルダに配置されます。

```
{インストールフォルダ}\fontconfig\font-config.xml
```

フォルダパスは[font-config.xml]内の「font-folder」要素の path 属性の値として記述します。

例えば Windows で「C:\TestFont」フォルダを指定したい場合、以下のように指定します。

font-config.xml 記述例 (Windows の場合)

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

[フォント構築ファイル配置先の指定]

任意のフォルダに配置したフォント構築ファイルを反映させたい場合、環境変数を次のように指定します。

- 環境変数名 : PTL80_FONT_CONFIGFILE

設定する値 : font-config.xml のフルパス

Windows 版では、インストール時に環境変数を指定するオプションが選択可能です。その場合、インストーラによって配置される[font-config.xml]のフォルダパスが環境変数に指定されます。

注意 :

Linux 版では Windows 版とは異なりインストーラが環境変数の設定を行いません。その代わりに、環境変数が設定されたシェルスクリプト [run.sh] がインストーラによって配置されます。[run.sh] に関する詳細は [『7-1-4 Linux 版の実行方法』](#) を参照してください。

5-2-3 PDF に文字列を描画する

PDF のページコンテンツに文字列を描画したり、テキスト透かしを挿入する処理を行う場合、動作環境に存在するフォント情報を取得しこれを利用します。

フォント情報取得のために参照する場所などについては、[『5-2-2 描画とフォント埋め込みに使用するフォントの参照先について』](#) をご参照ください。

[描画に使用するフォントについて]

以下の種類のフォントを用いて描画処理が可能です。

- 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

補足：描画処理におけるフォントの例外について

- ビットマップフォント、Type3 フォントでの描画には対応していません。
- 縦書き処理の場合、イタリック（斜体）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では描画されません。

5-2-4 PDF 上に存在するフォントを埋め込み状態にする

PDF 上のテキストには、PDF 内にフォント情報が埋め込まれていないことがあります。そのとき、動作

環境に存在するフォント情報を使用して PDF への埋め込みが可能です。

使用するフォント情報の参照先については、『5-2-2 描画とフォント埋め込みに使用するフォントの参照先について』を参照してください。

補足：埋め込み処理におけるフォントの例外について

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。

- 動作環境に存在しないフォント
- TrueType：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

5-2-5 PDF 上に存在するフォントを統合する

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。

フォント統合の製品仕様には以下のものがあります。

- 次のフォントが統合対象です。
 - TrueType、Type1(CFF 形式のみ)、CIDFontType2、CIDFontType0
- 統合対象は、埋め込まれたフォントファイルとフォント辞書です。
- フォント辞書の統合について、
 - TrueType、Type1(CFF 形式のみ)の場合、FontDescriptor 辞書が統合されます。
 - CIDFontType2、CIDFontType0 の場合、CIDFont 辞書が統合されます。
- ページと FormXObject のフォントが統合対象です。

補足：フォント統合における例外について

- 注釈とフォームフィールドのフォントは統合の対象となりません。
- 同じ文字で描画命令が違うものは統合の対象となりません。
- 埋め込まれたフォントファイルの内容をチェックし以下の条件の場合は統合されません。

- TrueType フォントの場合、cmap、head,hhea,hmtx,maxp,cvt,fpgm,prep の各テーブルで違うものがあれば統合の対象となりません。
- CIDFontType2 フォントの場合、head,hhea,hmtx,maxp,cvt,fpgm,prep の各テーブルで違うものがあれば統合の対象となりません。
- Type1(CFF 形式のみ)、CIDFontType0 において、メジャー・バージョン、マイナー・バージョンが異なる場合、統合の対象となりません。
- フォント辞書の内容をチェックし以下の条件の場合は統合されません。
 - CIDFontType2、CIDFontType0 の場合：
 - ✧ 「CIDSystemInfo」辞書の「Registry」キー(文字コレクションの発行者を識別する文字列)、「Ordering」キー(「Registry」の文字コレクションに一意の名前を付ける文字列)が違うものは統合対象外
 - ✧ 「DW」キー(グリフのデフォルトの幅)が違うものは統合対象外
 - ✧ 「DW2」キー(縦書き用のグリフのメトリック)が違うものは統合対象外
 - ✧ 「CIDToGIDMap」キーが「Identity」という値以外の名前、もしくは型がストリームの場合は統合対象外
 - ✧ 同じ文字で「W」キー(「DW」の説明)の値が違うものがある場合は統合対象外
 - ✧ 同じ文字で「W2」キー(「DW2」の説明)の値が違うものがある場合は統合対象外
 - ✧ 「Weight」キー(太さ)が違う場合は統合対象外
 - TrueType、Type1(CFF 形式のみ)の場合：
 - ✧ 「Weight」キー(太さ)が違う場合は統合対象外

5-3 カラープロファイルの扱い

本章では『PDF Tool API』でのカラープロファイルの扱いについて説明します。

具体的には、PDF/A 変換の際にカラープロファイルを扱います。

5-3-1 カラープロファイルの検索

『PDF Tool API』は PDF/A 変換のコマンドである「-fixedUpPDFA」でカラープロファイルを扱います。その際、2 種のパラメータ「-RGBProfile」、「-CMYKProfile」でプロファイル名を指定できます。

指定しなかった場合、『PDF Tool API』は sRGB2014.icc または JapanColor2001Coated.icc のカラープロファイルを検索します。各カラープロファイルの検索順は以下となり、Windows 版と Linux 版で検索順序

が異なります。

[Windows 版の場合]

1. dll ファイル[PdfTk80.dll]があるフォルダ内
2. 環境変数[PTL80_ICCPROFILE_PATH]で指定されたフォルダ内

[Linux 版の場合]

1. 環境変数[PTL80_ICCPROFILE_PATH]で指定されたフォルダ内

注意：

Linux 版においては、カラープロファイルが未指定でなおかつ検索先にカラープロファイルが無かった場合はエラーとなります。

Windows 版の場合はインストール時に環境変数を設定するオプションが選択可能で。詳細は『5-3-2 環境変数の作成』を参照してください。

5-3-2 環境変数の作成

カラープロファイルの検索先を環境変数で指定することができます。

- 環境変数名：PTL80_ICCPROFILE_PATH
設定する値：カラープロファイル用 icc ファイルの配置パス

Windows では、インストールの際にオプション指定することでインストールフォルダ内の icc ファイル配置フォルダを環境変数に設定可能です。詳細は『3-1-5 インストーラによりシステムに設定される内容』を参照してください。

Linux でコマンドによる指定なしでカラープロファイルを扱う場合、この設定は必須です。

注意：

Linux 版では Windows 版とは異なりインストーラが環境変数の設定を行うオプションはありません。その代わりに、環境変数が設定されたシェルスクリプト [run.sh] がインストーラによって配置されます。[run.sh] に関する詳細は『[7-1-4 Linux 版の実行方法](#)』を参照してください。

5-3-3 カラープロファイルを扱う処理

『PDF Tool API』のコマンドラインアプリケーションでカラープロファイルを扱うコマンドは『[第 8 章 各種コマンド](#)』で扱います。具体的なコマンドは以下の通りです。

- [8-29 -fixedUpPDFA : PDF/A 変換](#)
-

5-4 読み書きの制限サイズ

読み書き可能な PDF ファイルサイズは 2GB までです。具体的な製品仕様は以下の通りです。

- 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 [*2] は保存時にエラーになります。

[*2] :

ここで出力ファイルサイズの上限は、『PDF Tool API』で PDF の加工や編集を行った結果の出力 PDF が 2GB を超えるかで判断します。

これが具体例に問題になるのは例えば PDF ファイルを結合する場合です。

『PDF Tool API』で行う PDF ファイルの結合は、物理的に PDF ファイルを結合する処理ではありません。そのため、結合結果の出力 PDF が 2GB を超えてしまった場合、たとえ結合前の PDF ファイルサイズの合計が 2GB 未満であってもエラーになることがあります。

具体的な PDF ファイルの結合操作に関しては『[8-4 -mergeFiles : PDF ファイル結合](#)』を参照してください。

5-5 PDF/A、PDF/X および PDF/E に対する編集について

PDF/A、PDF/X、および PDF/E 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。

5-6 PDF の最適化について

PDF の最適化において、以下の操作を行ったとき制限事項があります。

- 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。
- フォントの統合を行ったとき、統合されるのは埋め込まれているフォントのみです。統合可能なフォントについての詳細は [『5-2-5 PDF 上に存在するフォントを統合する』](#) を参照してください。

PDF のダウンサンプリングやフォント統合といった最適化の操作に関する詳細は [『8-13 -optimize : PDF の最適化』](#) を参照してください。

5-7 セキュリティ情報の保持について

入力 PDF ファイルのセキュリティ情報は基本的に出力 PDF ファイルに引き継がれます。

しかし、以下の例外コマンドでは引き継がれません。

- 例外コマンド：
 - セキュリティ情報が新規に設定される
「[-encrypt : PDF セキュリティ設定](#)」
 - セキュリティ情報が削除される
「[-decrypt : PDF セキュリティ削除](#)」
 - 出力 PDF ファイルにセキュリティが引き継がれない

「-mergeFiles : PDF ファイル結合」

「-pageExtract : ページ抽出」

各例外コマンドの詳細に関しては『第 8 章 各種コマンド』内の解説を参照してください。

5-8 複数スレッドの利用について

コマンドラインアプリケーションを複数のスレッドで同時利用することはできません。

5-9 文字の抽出・検索における「-sort」指定について

「-sort」は文字を元データを座標順に並べ替えた上で抽出するよう指定するパラメータです。

ただし、以下のようなケースでは、並べ替えが正常に動作しない場合があります。

- 縦書きか横書きかを『PDF Tool API』が判別できないレイアウトであった場合
-

5-10 コマンドプロンプトにおける字数制限

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。コマンドラインアプリケーションの利用時はご注意ください。

参照（外部サイト）：コマンド プロンプト(Cmd.exe)コマンドライン文字列の制限 —Microsoft

第6章 評価版動作仕様

Windows 版ではインストール後 30 日間、Linux 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、『PDF Tool API』を評価版として使用できます。期間を過ぎると使用できません。

評価版は以下の動作仕様があります。

- 透かし文字列：「Antenna House PDF Tool」を各ページに挿入
- 画像抽出コマンド `[-extractImage]`： 1 個のみ抽出
- 文字抽出コマンド `[-extractText]`： 1 ページのみ抽出
- 添付ファイル抽出コマンド `[-extractEmbeddedFile]`： 1 個のみ抽出

第7章 コマンドラインアプリケーションの使い方

本章ではコマンドラインアプリケーション実行の基本的な操作を解説します。

7-1 実行方法

7-1-1 コマンドラインアプリケーションのコマンド表記に関する概要

本マニュアルにおいて、コマンドラインアプリケーションのコマンド表記には以下の記法を用います。

表記	説明
無地のテキスト (角かっこ([])または波かっこ({ }) がないもの)	表記の文字列をそのまま入力するが必要な項目
<山かっこで囲われたテキスト>	値の指定が必要な項目。テキスト部分は項目名
[角かっこで囲われたテキスト]	省略可能な項目。テキスト部分は項目名
{波かっこで囲われたテキスト}	必須の項目セット。項目の中から1つを選択する必要がある
縦棒()	項目セットに対する区切り記号。1つを選択する必要があり、同時に複数指定できない

7-1-2 実行方法の基礎

『PDF Tool API』のコマンドラインアプリケーションは、次のような構成で引数を指定して実行します。

```
<コマンドラインアプリケーション名> <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}
```

引数などの各項目の解説は以下の通りです。

- <コマンドラインアプリケーション名> :

『PDF Tool API』のコマンドラインアプリケーション名です。OSにより表記が異なります。
(何らかの理由でアプリケーションのファイル名を変更している場合はそちらを指定してください)
➤ Windows 版: 「AHPDFToolCmd80.exe」
➤ Linux 版: 「AHPDFToolCmd80」

- <コマンド> :

『PDF Tool API』の処理を指定するコマンドです。パラメータと組み合わせて主に PDF を処理します。詳細は『第 8 章 各種コマンド』の各コマンドの説明を参照してください。
コマンドにより、複数コマンドを組み合わせて指定することも可能です。その場合はコマンドに付随する[パラメータ]を指定した後に次のコマンドを指定する必要があります。詳細は『10-1 コマンドの複数指定』を参照してください。

例: -changeDocInfo

- [パラメータ] :

コマンドと対になって処理を行う項目です。コマンドによっては指定必須のパラメータがある場合や、パラメータを必要としない場合があります。細かく設定項目が分かれる内容などで、パラメータに別のパラメータが連なることがあります。

詳細は『第 8 章 各種コマンド』における各コマンドの説明を参照してください。

例: -title Val

- -d {入力ファイル|フォルダパス} :

入力ファイル、または入力フォルダを指定する特殊なパラメータです。PDF 以外が入力ファイルになることもあります。コマンドによっては入力ファイルが不要なものもあります。

指定方法の詳細は『詳細情報: -d』を参照してください。

例 : -d C:\test\test.pdf

- -o {出力ファイル|フォルダパス} :

出力ファイル、または出力フォルダを指定する特殊なパラメータです。PDF 以外が出力ファイルになることもあります。コマンドによっては出力ファイルが不要なものもあります。

指定方法の詳細は『詳細情報 : -o』を参照してください。

例 : -o C:\sav\out.pdf

7-1-3 Windows 版の実行方法

Windows 版のコマンドラインアプリケーションでは、次のように引数を指定して実行します。

```
AHPDFToolCmd80.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}
```

補足事項 [Windows 版] :

- パラメータを必要とするコマンドの場合、コマンド指定後に必ずパラメータを指定してください。パラメータを指定しない場合はエラーになります。
- 環境変数[ERRORLEVEL]で取得する戻り値で実行の成否とエラーコードを取得可能です。詳細は『10-3-1 コマンドラインアプリケーションの戻り値[Windows]』を参照してください。
- コマンドラインアプリケーショ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表示されます。エラー出力やエラー内容の詳細は『10-3 エラー処理について』を参照してください。
- 入出力におけるファイル指定またはフォルダ指定についての詳細は『7-2 入出力ファイル・フォルダの指定』を参照してください。

7-1-4 Linux 版の実行方法

Linux 版の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。

```
AHPDFToolCmd80 <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}
```

シェルスクリプト[run.sh]について

Linux 版ではインストーラによってシェルスクリプト[run.sh]が配置されます。[run.sh]はコマンドラインアプリケーションやライセンスファイルの配置情報を環境変数に設定して実行できるシェルスクリプトです。

Linux 版では Windows 版とは異なりインストーラが環境変数の設定を行うオプションがありません。インストール先情報が設定された[run.sh]はその代わりとして同梱されるファイルです。

(より正確には、[run.sh]はインストーラにより配置フォルダなどの情報を記入してインストール時に生成されるファイルです)

シェルスクリプトを通じて『PDF Tool API』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}
```

シェルスクリプト[run.sh]を一旦実行した後はターミナルを閉じるまで、環境変数の設定なしで「AHPDFToolCmd80」をコマンドラインアプリケーション上で実行することが可能になります。

補足事項 [Linux 版] :

- パラメータを必要とするコマンドの場合、「コマンド」指定後に必ず「パラメータ」を指定してください。必要とする「パラメータ」を指定しない場合はエラーになります。
- 変数「\$?」で取得する戻り値で実行の成否を取得可能です。詳細は『[10-3-1 コマンドラインの戻り値 \[Linux\]](#)』を参照してください。
- コマンドラインアプリケーショ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表示されます。エラー出力やエラー内容の詳細は『[10-3 エラー処理について](#)』を参照してください。
- シェルスクリプト[run.sh]による環境変数設定はターミナルを閉じるまでしか保持されません。Windows のシステム環境変数のように、ログイン時に自動的に読み込まれる環境変数を設定したい場合は、[`~/.profile`]や[`~.bashrc`]、[`~/.bash_profile`]を用いた実行環境の環境変数の

設定をお試しください。

- 入出力におけるファイル指定またはフォルダ指定についての詳細は「7-2 入出力ファイル・フォルダの指定」を参照してください。
-

7-2 入出力ファイル・フォルダの指定

『PDF Tool API』のコマンドラインアプリケーションの多くは入出力ファイルを指定する必要があります。本節では1つ1つファイルを指定する方法と、フォルダをまとめて指定する方法をそれぞれ解説します。

7-2-1 ファイル指定

入出力ファイルをファイルパスで指定します。すべてのコマンドで使用可能な指定方法です。

「-d」で入力ファイルパスを指定し、「-o」で出力ファイルパスを指定します。

[ファイル指定方法]

```
-d <入力ファイルパス> [パスワード] -o <出力ファイルパス>
```

[詳細情報 : -d]

```
-d <入力ファイルパス> [パスワード]
```

「-d」で入力ファイルパスを指定します。

パスワードセキュリティが施された入力ファイルの処理も可能です。パスワードセキュリティが施された入力ファイルの場合はパスワード文字列を指定してください。文書を開くパスワードと権限パスワードの2種類が入力ファイルに付与されている場合は、権限パスワードを指定してください。

「-d」についての補足事項：

- ファイルパスにスペースが含まれる場合は、ファイルパスをダブルクオーテーション ("") で囲んでください。
- 入力 PDF ファイルのセキュリティ情報は、以下の例外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。

➤ 例外コマンド：

- ❖ セキュリティ情報が新規に設定される
「-encrypt : PDF セキュリティ設定」
- ❖ セキュリティ情報が削除される
「-decrypt : PDF セキュリティ削除」
- ❖ 出力 PDF ファイルにセキュリティが引き継がれない
「-mergeFiles : PDF ファイル結合」
「pageExtract : ページ抽出」

➤ 各例外コマンドに関しては『第8章 各種コマンド』内の各コマンドの仕様を参照してください。

[詳細情報 : -o]

-o <出力ファイルパス>

「-o」で出力ファイルパスを指定します。

「-o」についての補足事項：

- ファイルパスにスペースが含まれる場合は、ファイルパスをダブルクオーテーション ("") で囲んでください。
- 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。(例 : -getFileInfo : PDF の情報取得)
出力ファイルパスの指定の要・不要は『第8章 各種コマンド』に記載の各コマンドの仕様を参照してください。
- 出力先には、存在するフォルダ上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。

[具体的な入出力ファイル指定例]

test 1.pdf を読み込み、<コマンド>で指定した処理内容を実行して out.pdf として出力する。

このとき入力ファイルのパスワードは「ownerpass」である。

<Windows>

```
AHPDFToolCmd80.exe <コマンド> [パラメータ] -d "c:\test\test 1.pdf" ownerpass -o c:\$sav\$out.pdf
```

<Linux>

```
AHPDFToolCmd80 [コマンド] [パラメータ] -d "/home/antenna/test/test 1.pdf" ownerpass -o /home/antenna/sav/out.pdf
```

7-2-2 フォルダ指定

入出力先をフォルダパスで指定できます。フォルダ指定の可否や動作内容はコマンドごとに異なります。通常、入力時にフォルダパスを指定した場合は出力時にもフォルダパスを指定する必要があります。

「-d」で入力フォルダパスを指定し、「-o」で出力フォルダパスを指定します。「-d」「-o」のどちらも、基本的な製品仕様はファイル指定と同じです。

本項ではフォルダ指定時の扱いについて解説します。

[フォルダ指定方法]

```
-d <入力フォルダパス> [パスワード] -o <出力フォルダパス>
```

フォルダ指定時の「-d」の扱い

```
-d <入力フォルダパス> [パスワード]
```

「-d」で入力フォルダパスを指定します。

指定フォルダにパスワードセキュリティが施された入力ファイルがある場合も処理できます。その場合はパスワード文字列を指定してください。文書を開くパスワードと権限パスワードの2種類が入力ファイルに付与されている場合は、権限パスワードを指定してください。

補足事項：

- 基本的な製品仕様はファイル指定と同様です。
- フォルダパスにスペースが含まれる場合は、ダブルクオーテーション ("") で囲んでください。
- 指定された入力フォルダパス直下にあるファイルが処理対象となります。サブフォルダ内のファイルは処理対象にはなりません。
- 異なるパスワードでセキュリティが施された入力ファイルがフォルダ内に含まれていた場合、指定したパスワード以外の入力 PDF に関してはエラーとなります。
- フォルダ指定で複数のパスワードを指定することはできません。
- フォルダ指定時の動作や入力ファイルの種別 (PDF・画像など) はコマンドごとに異なります。
フォルダ指定時の詳細な動作内容は『第 8 章 各種コマンド』に記載の各コマンドの仕様を参照してください。
- 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。
複数のコマンド指定に関する詳細は『10-1 コマンドの複数指定』を参照してください。
- 一部のコマンドはフォルダ指定に対応していません。
『7-3 コマンド一覧』にフォルダ指定の可否を記した一覧がありますので参照してください。または、『第 8 章 各種コマンド』に記載の各コマンドの仕様を参照してください。

フォルダ指定時の「-o」の扱い

-o (出力フォルダパス)

「-o」で出力フォルダパスを指定します。

補足事項：

- 基本的な製品仕様はファイル指定と同様です。
- フォルダパスにスペースが含まれる場合は、ダブルクオーテーション ("") で囲んでください。
- 出力を伴わないコマンドでは、出力フォルダパスの指定は不要です。(例：-getFileInfo : PDF の情報取得)
出力フォルダパスの指定の要・不要は『第 8 章 各種コマンド』に記載の各コマンドの仕様を参照してください。
- 出力先には、存在するフォルダ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。
- フォルダ指定時の動作や出力ファイルの種別 (PDF・画像など) はコマンドごとに異なります。

フォルダ指定時の詳細な動作内容は『第 8 章 各種コマンド』に記載の各コマンドの仕様を参照してください。

- 处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。
複数のコマンド指定に関する詳細は『10-1 コマンドの複数指定』を参照してください。
- 一部のコマンドはフォルダ指定に対応していません。
『7-3 コマンド一覧』にフォルダ指定の可否を記した一覧がありますので参照してください。または、『第 8 章 各種コマンド』に記載の各コマンドの仕様を参照してください。

[具体的な入出力フォルダ指定例]

「in 2」 フォルダ内のファイルを処理対象として<コマンド> [パラメータ]で指定した処理内容を実行し、
「out」 フォルダにファイルを出力する。
このとき全入力ファイルのパスワードは「ownerpass」である。

<Windows>

```
AHPDFToolCmd80.exe <コマンド> [パラメータ] -d "c:$in 2" ownerpass -o c:$out
```

<Linux>

```
AHPDFToolCmd80 <コマンド> [パラメータ] -d "/home/antenna/in 2" ownerpass -o /home/antenna/out
```

7-3 コマンド一覧

各種コマンド一覧です。

フォルダ指定対応の有無、コマンドの複数指定の可否が確認できます。

	コマンド	機能	フォルダ指定対応の有無	コマンドの複数指定の可否
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 ファイル透かしの挿入	○	○
8	-setImageWatermark	画像ファイル透かしの挿入	○	○
9	-setTextWatermark	テキスト透かしの挿入	○	○
10	-setColorWatermark	色透かしの挿入	○	○
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	注釈を FDF へエクスポート	○	○
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	画像抽出	○	○
25	-extractText	テキスト抽出	○	○
26	-imageToPdf	画像ファイルの PDF 化	○	×
27	-extractEmbeddedFile	添付ファイルの抽出	○	○
28	-importAnnotsFromFDF	FDF からの注釈の取り込み	×	○
29	-fixedUpPDFA	PDF/A 変換	○	×
30	-validatePDFA	PDF/A 準拠確認	○	×
31	-dividePdf	ファイル分割	×	×
32	-getPageSize	ページサイズの取得	×	×
33	-searchTextAndHighlight	テキスト検索後にハイライト注釈挿入	×	×

34	-searchTextAndRedact	テキスト検索後に Redact 注釈挿入	×	×
35	-searchTextAndSetMask	テキスト検索後にマスクをかける	×	×
36	-zoom	ページを拡大・縮小する	×	×
37	-setCustomProperty	カスタムプロパティの設定	○	○
38	-setMaskUsingRedaction	Redact 注釈の適用	×	×
39	-linearize	Web 表示用に最適化	○	○
40	-lic	ライセンス情報の取得	×	×
41	-v	バージョン情報・著作権情報・ライセンス情報の表示	-	-
42	-h	コマンド名一覧表示	-	-
43	-h-all	全コマンドオプション説明	-	-
44	-h{コマンド名}	個別コマンドオプション説明	-	-

第8章 各種コマンド

本章では『PDF Tool API』が持つ具体的なコマンドについて解説していきます。

8-1 -getFileInfo : PDF の情報取得

処理内容

PDF ファイルから以下の情報を取得し、それらを標準出力に表示します。

どの項目の情報を取得するかはパラメータで指定可能です。パラメータの指定が無い場合はすべての情報を取得します。

- PDF バージョン情報
- 文書情報と総ページ数
- 開き方情報
- カスタムプロパティ情報
- セキュリティ情報
- 署名の有無
- PDF/A かどうか
- PDF/X かどうか

このコマンドを実行するのに出力ファイル指定「-o」は必要ありません。

コマンド例

[コマンド例の実行内容]

test.pdf が持つ PDF ファイル情報のうち、文書情報と総ページ数、セキュリティ情報、PDF/A かどうかを標準出力に出力する。

[Windows]

```
AHPDFToolCmd80.exe -getFileInfo -docInfo -security -pdfa -d C:\test\test.pdf
```

[Linux]

```
AHPDFToolCmd80 -getFileInfo -docInfo -security -pdfa -d /home/antenna/test/test.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダの 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	内容
-version	省略可。 PDF バージョンを出力する。
-docInfo	省略可。 文書情報と総ページ数を出力する。
-viewInfo	省略可。 開き方設定を出力する。
-customProp	省略可。 カスタムプロパティ情報を出力する。
-security	省略可。 セキュリティ情報を出力する。 具体的にはセキュリティ設定の有無、およびセキュリティ設定がある場合はその詳細情報を出力する。
-signature	省略可。 署名の有無を出力する。
-pdfa	省略可。 PDF/A かどうかを出力する。

-pdfx	省略可。 PDF/X かどうかを出力する。
-------	--------------------------

出力内容

詳細な出力内容は『取得情報の出力内容』内の『9-1 -getFileInfo : PDF の情報取得』の項を参照してください。

8-2 -changeDocInfo : 文書プロパティ設定

処理内容

文書プロパティ情報を出力 PDF に設定します。

コマンド例

[コマンド例の実行内容]

test.pdf の文書プロパティ情報を指定し、out.pdf として出力する。

パラメータで文書プロパティは「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」と指定する。

[Windows]

```
AHPDFToolCmd80.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]

```
AHPDFToolCmd80 -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
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括設定することができます。

フォルダ指定をされた場合はフォルダ内の全てのファイルに同じ文書プロパティ情報が設定されます。
なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

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

8-3 -changeViewInfo : 開き方設定

処理内容

PDF の開き方を設定します。

「連続ページ/単一ページ」の指定の他、ページの並び順やツールバーの表示状態、フルスクリーンモードなどを切り替えて指定することができます。

コマンド例

[コマンド例の実行内容]

test.pdf の開き方を指定し、out.pdf として出力する。

パラメータで PDF ファイルを開いたときの動作を、以下の状態で表示されるよう指定する。

- 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

[Windows]

```
AHPDFToolCmd80.exe -changeViewInfo -pageLayout 2 -pageMode 3 -hideToolbar true -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -changeViewInfo -pageLayout 2 -pageMode 3 -hideToolbar true -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括設定することができます。

フォルダ指定をされた場合は入力フォルダの PDF ファイルに開き方を設定します。なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-direction {1 2}	省略可。 読み上げオプション（綴じ方）の設定。 1: 左から右（左綴じ） 2: 右から左（右綴じ） 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 {1 2 3 4 5 6}	省略可。 ページレイアウトの設定。 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 {1 2 3 5 6}	省略可。 表示状態の設定。指定がない場合、変更なし。

	<p>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</p> <p>1～3,5,6以外が指定された場合、エラーとなる。</p> <p>「-fullScreen true」指定時「-pageMode」は無視される。</p>
-fitWindow {true false}	<p>省略可。</p> <p>ページにウィンドウサイズを合わせる。指定がない場合、変更なし。</p> <p>true: 合わせる false: 合わせない</p>
-centerWindow {true false}	<p>省略可。</p> <p>ウィンドウを画面中央に配置。指定がない場合、変更なし。</p> <p>true: 配置する false: 配置しない</p>
-fullScreen {true false}	<p>省略可。</p> <p>フルスクリーンモードで開く。指定がない場合、変更なし。</p> <p>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</p> <p>「-fullScreen true」指定時「-pageMode」が無視される。</p>
-displayTitle {true false}	<p>省略可。</p> <p>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。指定がない場合、変更なし。</p> <p>true: 表示する false: 表示しない</p>
-hideToolbar {true false}	<p>省略可。</p> <p>ツールバーを非表示。指定がない場合、変更なし。</p> <p>true: 非表示 false: 表示</p>

<p>-hideMenubar {true false} [*1]</p>	<p>省略可。 メニューを非表示。指定がない場合、変更なし。</p> <p>true: 非表示 false: 表示</p>
<p>-hideWindowUI {true false} [*1]</p>	<p>省略可。 ウィンドウコントロールを非表示。指定がない場合、変更なし。</p> <p>true: 非表示 false: 表示</p>
<p>-viewFit {1 2 3 4 5}</p>	<p>省略可。 倍率設定。指定がない場合、変更なし。</p> <p>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</p> <p>1~5 以外が指定された場合、エラーとなる。</p>
<p>-defaultPageNo <Val></p>	<p>省略可。 最初に開くページ番号設定。指定がない場合、変更なし。</p> <p>1 ページ目を「0」とする。</p>

[*1] :

「-hideMenubar true」「-hideWindowUI true」を同時に設定した PDF ファイルを Acrobat/Adobe Reader で表示した場合、メニューが表示されますが、これは Acrobat Reader の仕様です。

8-4 -mergeFiles : PDF ファイル結合

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。

それぞれの入力 PDF ファイルは-d で複数指定します。

- 「-d」でファイルを指定する場合、各入力 PDF の任意のページを選んで結合することができます。
その場合は「-d」で入力ファイルを指定した後にページ番号やページ範囲（ページ）をさらに指定してください。指定するページ番号は 0 オリジンとなっています。そのため、1 ページ目を「0」から数えます。
- 結合する PDF ファイルが多数ある場合などは、テキストファイルでまとめて指定することができます。
「-d」の代わりに「-dmerge」を使用し、テキストファイル内でファイルパスを記述してください。
「-d」と「-dmerge」を併用することはできません。

補足：

- ファイル全体を結合させた場合、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 PDF があった場合、その入力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 output 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。
- 出力 PDF のファイルサイズが 2GB を超える場合は保存時にエラーになります。PDF 前のファイルサイズの合計が 2GB 未満であった場合にもエラーとなることがあります。『PDF Tool API』の結合処理は物理的に PDF ファイルを結合する処理ではないため、結合前のファイルサイズの合計が 2GB 未満であった場合にもエラーとなることがあります。

コマンド例

[コマンド例の実行内容]

2つの入力ファイル test1.pdf、test2.pdf を結合し、出力する。

出力 PDF は Web 表示用に最適化した上で out.pdf として出力する。

入力ファイル指定の際、test1.pdf はファイル全体を、test2.pdf からは 1 ページ目、そして 10~15 ページ目を追加し、出力 PDF とする。

test2.pdf は開く際にユーザーパスワード"userpass"を入力している。

[Windows]

```
AHPDFToolCmd80.exe -mergeFiles -d C:\test\test1.pdf -d C:\test\test2.pdf userpass "0,9-14" -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -mergeFiles -d /home/antenna/test/test1.pdf -d /home/antenna/test/test2.pdf userpass "0,9-14" -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
-dmerge <path>	省略可。 <path>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

	<p>頭行から順に結合する。</p> <p><path>にはフルパス名を指定する。</p> <p>「-d」と一緒に指定された場合、エラーになる。</p> <p>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。文字エンコーディングは「UTF-8」を使用する。</p>
-copyAnnots {true false}	<p>省略可。</p> <p>ファイル結合時に注釈をコピーするか否かを指定する。</p> <p>true: コピーする false: コピーしない</p> <p>指定がない場合はコピーする。</p>
-copyOutlines {true false}	<p>省略可。</p> <p>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。</p> <p>true: コピーする false: コピーしない</p> <p>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。</p>

8-5 -getOutlineInfo：しおり情報取得

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。

コマンド例

[コマンド例の実行内容]

test.pdf が持つしおり情報を標準出力に出力する。

[Windows]

```
AHPDFToolCmd80.exe -getOutlineInfo -d C:\test\test.pdf
```

[Linux]

```
AHPDFToolCmd80 -getOutlineInfo -d /home/antenna/test/test.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルから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

詳細な出力内容は『取得情報の出力内容』内の『9-2 -getOutlineInfo : しおり情報取得』の項を参照してください。

8-6 -getAnnotInfo : 注釈情報取得

処理内容

入力 PDF 内の注釈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。

「-getAnnotInfo」で読み取れる注釈情報は以下の各注釈の情報です。

- テキスト注釈[*1]
- リンク注釈
- スタンプ注釈
- ファイル添付注釈

[*1] :

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ることがあります。

コマンド例

[コマンド例の実行内容]

test.pdf が持つ注釈情報を標準出力に出力する。

[Windows]

```
AHPDFToolCmd80.exe -getAnnotInfo -d C:\test\test.pdf
```

[Linux]

```
AHPDFToolCmd80 -getAnnotInfo -d /home/antenna/test/test.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルから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

詳細な出力内容は『取得情報の出力内容』内の 『9-3 -getAnnotInfo：注釈情報取得』 の項を参照してください。

8-7 -setPdfWatermark : PDF ファイル透かしの挿入

処理内容

-pdfPath で指定した PDF ファイルを透かしにして入力 PDF に挿入します。

- 指定した PDF が複数ページを持っていた場合、1 ページ目を透かしにします。
- 入力 PDF のどのページに透かしを挿入するかはパラメータで指定します。
- 挿入範囲の矩形を「-rect」と「-margin」で指定できます。両方指定した場合は、2 つを重ねた矩形の内側が指定矩形になります。

コマンド例

[コマンド例の実行内容]

mark.pdf の 1 ページ目を PDF 透かしとして test.pdf の全ページに挿入し、out.pdf として出力する。

透かしは Acrobat 互換に指定する。大きさは mark.pdf の 50%、透明度は 0.7 で指定する。

[Windows]

```
AHPDFToolCmd80.exe -setPdfWatermark -pdfPath C:\test\mark.pdf -scale 0.5 -opacity 0.7 -acrobat -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -setPdfWatermark -pdfPath /home/antenna/test/mark.pdf -scale 0.5 -opacity 0.7 -acrobat -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し、指定した PDF ファイルを透かしとして挿入します。

なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-pdfPath <path> [password]	[必須] 透かしにする PDF ファイルパスの指定。PDF ファイルを開くのにパスワードが必要な場合はパスワードが必要となる。
-scale <Val>	省略可。 挿入する透かしの倍率。 <Val>の設定可能範囲：0.0～50.0 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>[*1]	省略可。 透かしを配置する矩形の指定（mm 単位）。 各要素は左下を(0, 0)とした距離で指定する。指定がない場合はページサイズの矩形。 注意： <left>に<right>より大きい数値を指定することはできない。<bottom>に<top>より大きい数値を指定することはできない。どちらかで違反した場合はエラーとなる。

	[※ 「-acrobat」と併用不可]
-margin <top> <left> <bottom> <right>[*1]	<p>省略可。</p> <p>透かしを配置するときの余白の設定（mm 単位）。各要素は左下を(0, 0)とした距離で指定する。ページ全体を基準に上下左右それぞれの余白を引いた範囲が指定矩形となる。</p> <p>注意：</p> <p>left と right の合計値が PDF の幅より大きい場合や、top と bottom の合計値が PDF の高さより大きい場合はエラーとなる。</p>
	[※ 「-acrobat」と併用不可]
-align {1 2 3 4 5 6 7 8 9}	<p>省略可。</p> <p>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。</p> <p>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</p> <p>1～9 以外が指定された場合、エラーとなる。</p>
-zorder {1 2}	<p>省略可。</p> <p>透かしの Z オーダーの設定。</p> <p>1: 前面 2: 背面</p> <p>指定がない場合、「1: 前面」に配置。</p> <p>1,2 以外が指定された場合、エラーとなる。</p>
-pageRange {1 2 3 4 5}	<p>省略可。</p> <p>透かしを挿入する先のページ範囲の設定。</p> <p>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</p> <p>指定がない場合、「0: 全ページ」に挿入する。</p> <p>1～5 以外が指定された場合、エラーとなる。</p> <p>1 と-notInFirst、2 と-notInLast を指定するとエラーとなる。</p>

	「-pageRange 5」を指定する場合は「-pageRangeCustom」が指定必須となる。
-pageRangeCustom <Val>[*2]	<p>[「-pageRange 5」を指定した場合：設定必須]</p> <p><Val>には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。</p> <p>ページ番号は 0 オリジン。そのため 1 ページ目を「0」から数える。複数指定する場合はカンマで区切る。</p> <p>例) -pageRangeCustom “0,2-4” 1 ページと 3～5 ページに挿入される。</p> <p>[※ 複数範囲を指定する場合は「-acrobat」と併用不可]</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。</p> <p>指定がない場合、先頭ページに透かしを挿入できる。</p> <p>「-pageRange 1」と一緒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。</p> <p>指定がない場合、最終ページにも透かしを挿入できる。</p> <p>「-pageRange 2」と一緒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。</p> <p>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。</p> <p>指定がない場合、透かしを印刷対象とする。</p>
-opacity <Val>	<p>省略可。</p> <p>透かしの透明度の設定。指定がない場合、1.0（不透明）。</p> <p><Val>の範囲：0.0～1.0。</p> <p>0.0 が透明、1.0 が不透明。</p>
-tiling [*1]	<p>指定がある場合、透かしをタイリング挿入する。</p> <p>-rect 指定があるときは rect 内でタイリングされる。</p> <p>指定がない場合、透かしをタイリングで挿入しない。</p> <p>[※ 「-acrobat」と併用不可]</p>

-acrobat	<p>省略可。</p> <p>このパラメータが指定された場合、透かしを Acrobat 互換のものとして追加する。</p> <p>指定が無い場合、『PDF Tool API』独自の透かしとして追加する。</p> <p>[*1]が付いたパラメータと同時指定した場合、および[*2]の条件と合致する場合はエラーとなる。</p>
----------	---

[*1] :

「-acrobat」と一緒に指定するとエラーとなる。『PDF Tool API』独自の透かし専用パラメータ。

[*2] :

「-pageRangeCustom」でカッコ区切りで複数範囲を指定した場合、「-acrobat」と一緒に指定するとエラーとなる。

8-8 -setImageWatermark : 画像ファイル透かしの挿入

処理内容

画像ファイルを透かしとして入力 PDF に挿入します。

- どのページに透かしを挿入するかはパラメータで指定します。
- 挿入範囲の矩形を「-rect」と「-margin」で指定できます。両方指定した場合は、2つを重ねた矩形の内側が指定矩形になります。

コマンド例

[コマンド例の実行内容]

mark.bmp を画像透かしとして test.pdf に挿入し、out.pdf として出力する。

画像透かしは入力 PDF の 1,4,11~13 ページ目に挿入する。

パラメータで画像透かしは透明度 0.8 のタイリング状態に指定する。

[Windows]

```
AHPDFToolCmd80.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]

```
AHPDFToolCmd80 -setImageWatermark -imagePath /home/antenna/test/mark.bmp -page  
Range 5 -pageRangeCustom "0,3,10-12" -opacity 0.8 -tiling -d /home/antenna/test/test.p
```

```
df -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の 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。

なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-imagePath	[必須] 透かしにする画像ファイルパスの指定。 対応形式：bitmap、jpeg、png、tiff、gif
-scale <Val>	省略可。 挿入する透かしの倍率。 <Val>の指定可能範囲：0.0～50.0 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> [*1]	省略可。 透かしを配置する矩形の指定（mm 単位）。各要素は左下を(0, 0)とした距離で指定する。指定がない場合はページサイズの矩形。 注意： <left>に<right>より大きい数値を指定することはできない。<bottom>に<top>

	<p>より大きい数値を指定することはできない。どちらかで違反した場合はエラーとなる。</p> <p>[※ 「-acrobat」と併用不可]</p>
-margin <top> <left> <bottom> <right> [*1]	<p>省略可。</p> <p>透かしを配置するときの余白の設定（mm 単位）。各要素は左下を(0, 0)とした距離で指定する。ページ全体を基準に上下左右それぞれの余白を引いた範囲が指定矩形となる。</p> <p>注意：</p> <p>left と right の合計値が PDF の幅より大きい場合や、top と bottom の合計値が PDF の高さより大きい場合はエラーとなる。</p> <p>[※ 「-acrobat」と併用不可]</p>
-align {1 2 3 4 5 6 7 8 9}	<p>省略可。</p> <p>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。</p> <p>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</p> <p>1~9 以外が指定された場合、エラーとなる。</p>
-zorder {1 2}	<p>省略可。</p> <p>透かしの Z オーダーの設定。</p> <p>1: 前面 2: 背面</p> <p>指定がない場合、「1:前面」に配置。</p> <p>1,2 以外が指定された場合、エラーとなる。</p>
-pageRange {1 2 3 4 5}	<p>透かしを挿入するページ範囲の設定。省略可。</p> <p>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</p> <p>指定がない場合、「0:全ページ」に挿入する。</p> <p>1~5 以外が指定された場合、エラーとなる。</p>

	<p>1 と -notInFirst、2 と -notInLast を指定するとエラーとなる。 「-pageRange 5」を指定した場合は「-pageRangeCustom」が指定必須となる。</p>
-pageRangeCustom <Val> [*2]	<p>[「-pageRange 5」の場合：指定必須]</p> <p><Val>には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。ページ番号は0オリジン。そのため、1ページ目を「0」から数える。</p> <p>複数指定する場合はカンマで区切る。 例) -pageRangeCustom “0,2-4” 1ページと3~5ページに挿入される。</p> <p>[※複数範囲を指定する場合は「-acrobot」と併用不可]</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。</p> <p>「-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。</p> <p>「-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 <rate>	<p>省略可。 透かしの透明度の設定。指定がない場合、1.0（不透明）。</p> <p><rate>の範囲：0.0~1.0。 0.0が透明、1.0が不透明。</p>
-tiling [*1]	<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。</p> <p>指定がない場合、透かしをタイリングで挿入しない。</p> <p>[※「-acrobot」と併用不可]</p>

<p>-acrobat</p>	<p>省略可。</p> <p>このパラメータが指定された場合、透かしを Acrobat 互換のものとして追加する。</p> <p>指定が無い場合、『PDF Tool API』独自の透かしとして追加する。</p> <p>[*1]が付いたパラメータと同時指定した場合、および[*2]の条件と合致する場合はエラーとなる。</p>
-----------------	---

[*1] :

「-acrobat」と一緒に指定するとエラーとなる。『PDF Tool API』独自の透かし専用パラメータ。

[*2] :

「-pageRangeCustom」でカンマ区切りで複数範囲を指定した場合、「-acrobat」と一緒に指定するとエラーとなる。

8-9 -setTextWatermark : テキスト透かしの挿入

処理内容

文字列を透かしとして挿入します。

- Linux 上で実行する場合、フォント名の指定が必須です。「-font -name」でフォント名を指定してください。Windows で実行する場合、フォントが指定されない場合は日本語環境では「MS-Gothic」が、それ以外の環境では「Arial」がフォント名に指定されます。
- 挿入する文字列には、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。詳細は『10-2 マクロ文字列』を参照してください。
- 挿入範囲の矩形を「-rect」と「-margin」で指定できます。両方指定した場合は、2つを重ねた矩形の内側が指定矩形になります。

コマンド例

[コマンド例の実行内容]

test.pdf に「PDF Tool API V8.0 テスト」という文字列をテキスト透かしとして全ページに挿入し out.pdf として出力する。

テキスト透かしは以下の設定・配置で挿入される。

- フォント名：メイリオフォント(Windows) / IPA ゴシック(Linux)
- フォントサイズ：52
- 文字色：{R G B}={1.0, 0.0, 0.0}
- 透明度 0.9
- 挿入角度 20 度
- 配置：ページ中央

[Windows]

```
AHPDFToolCmd80.exe -setTextWatermark -text "PDF Tool API V8.0 テスト" -colorText 1.
```

```
0 0.0 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]

```
AHPDFToolCmd80 -setTextWatermark -text "PDF Tool API V8.0 テスト" -colorText 1.0 0.0 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
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し、文字列を透かしとして挿入します。

なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-text <Val>	[必須] 透かしとして挿入する文字列の設定。 <Val>に設定する文字列にはマクロ文字列も利用可能です。詳細は『 10-2 マクロ文字列 』を参照してください。
-colorText <r> <g> 	省略可。 文字の塗りつぶし色の設定。 設定値の範囲：<r>, <g>, それぞれ 0.0~1.0 指定がない場合、黒色(0.0 0.0 0.0)

<p>-colorOutline <r> <g> [*1]</p>	<p>省略可。 文字の輪郭の色の設定。 設定値の範囲 : <r>, <g>, それぞれ 0.0~1.0 指定がない場合、黒色(0.0 0.0 0.0) [※ 「-acrobat」 と併用不可]</p>
<p>-diagonal [*1]</p>	<p>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 [※ 「-acrobat」 と併用不可]</p>
<p>-angle <Val></p>	<p>省略可。 透かしの挿入角度を<Val>で設定する。反時計周りが正の値。 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。</p>
<p>-rect <left> <bottom> <right> <top> [*1]</p>	<p>省略可。 透かしを配置する矩形の指定 (mm 単位)。 各要素は左下を(0, 0)とした距離で指定する。指定がない場合はページサイズの矩形。 注意： <left>に<right>より大きい数値を指定することはできない。<bottom>に<top>より大きい数値を指定することはできない。どちらかで違反した場合はエラーとなる。 [※ 「-acrobat」 と併用不可]</p>
<p>-margin <top> <left> <bottom> <right> [*1]</p>	<p>省略可。 透かしを配置するときの余白の設定 (mm 単位)。各要素は左下を(0, 0)とした距離で指定する。ページ全体を基準に上下左右それぞれの余白を引いた範囲が指定矩形となる。 注意： left と right の合計値が PDF の幅より大きい場合や、top と bottom の合計値が PDF の高さより大きい場合はエラーとなる。 [※ 「-acrobat」 と併用不可]</p>

<p>-align {1 2 3 4 5 6 7 8 9}</p>	<p>省略可。 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。</p>
<p>-zorder {1 2}</p>	<p>省略可。 透かしの Z オーダーの設定。 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。</p>
<p>-pageRange {1 2 3 4 5}</p>	<p>省略可。 透かしを挿入するページ範囲の設定。指定がない場合、「0:全ページ」に挿入する。 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 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。 「-pageRange 5」を設定した場合は「-pageRangeCustom」が指定必須となる。無い場合はエラーとなる。</p>
<p>-pageRangeCustom <Val> [*2]</p>	<p>[「-pageRange 5」の場合：指定必須] <Val>には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。ページ番号は0 オリジン。そのため、1 ページ目を「0」から数える。 複数指定する場合はカンマで区切る。 例) -pageRangeCustom “0,2-4” 1 ページと 3~5 ページに挿入される。 [※複数範囲を指定する場合は「-acrobat」と併用不可]</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>省略可。 透かしの透明度の設定。指定がない場合、1.0（不透明）。 <Val>の範囲：0.0～1.0。 0.0 が透明、1.0 が不透明。</p>
-tiling [*1]	<p>指定がある場合、透かしをタイリング挿入する。「-rect」指定があるときは指定した範囲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。 [※「-acrobat」と併用不可]</p>
-acrobat	<p>省略可。 このパラメータが指定された場合、透かしを Acrobat 互換のものとして追加する。 指定が無い場合、『PDF Tool API』独自の透かしとして追加する。 [*1]が付いたパラメータと同時指定した場合、および[*2]の条件と合致する場合はエラーとなる。</p>
-font	<p>[Linux の場合は指定必須] 透かし文字列のフォント設定。[*3]</p>
	<p>-name <Val></p> <p>[Linux の場合は指定必須] フォント名。 Windows で省略された場合は日本語環境では「MS-Gothic」、それ以外の環境では「Arial」が設定される</p>

	-size <Val>	省略可。 フォントサイズ。 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 {true false}	省略可。 フォント埋め込み。指定がない場合、「true: 埋め込む」 true: 埋め込む false: 埋め込まない

[*1] :

「-acrobat」と同時に指定するとエラーとなる。『PDF Tool API』独自の透かし専用パラメータ。

[*2] :

「-pageRangeCustom」でカンマ区切りで複数範囲を指定した場合、「-acrobat」と同時に指定するとエラーとなる。

[*3] :

『PDF Tool API』のコマンドラインアプリケーションにおけるフォント情報の利用や製品仕様に関しては『5-2 フォントについて』を参照してください

8-10 -setColorWatermark : 色透かしの挿入

処理内容

指定した色で塗りつぶす色透かしを挿入します。

- 挿入範囲の矩形を「-rect」と「-margin」で指定できます。両方指定した場合は、2つを重ねた矩形の内側が指定矩形になります。

コマンド例

[コマンド例の実行内容]

test.pdf の先頭ページ以外のページに色透かしを挿入し、out.pdf として出力する。

色透かしは以下の設定で挿入される。

- 塗りつぶし色 : {R G B}={0.0, 1.0, 0.0}
- 透明度 : 0.7
- 指定 : 印刷非対象

[Windows]

```
AHPDFToolCmd80.exe -setColorWatermark -color 0.0 1.0 0.0 -notInFirst -notPrint -opacity 0.7 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -setColorWatermark -color 0.0 1.0 0.0 -notInFirst -notPrint -opacity 0.7 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し、色の透かしを挿入します。
なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-color <r> <g> 	<p>[指定必須]</p> <p>挿入する色の設定。</p> <p>設定値の範囲：$<r>$, $<g>$, $$ それぞれ 0.0~1.0</p>
-rect <left> <bottom> <right> <top>	<p>省略可。</p> <p>透かしを配置する矩形の指定 (mm 単位)。各要素は左下を(0, 0)とした距離で指定する。指定がない場合はページサイズの矩形。</p> <p>注意：</p> <p>$<left>$ に $<right>$ より大きい数値を指定することはできない。$<bottom>$ に $<top>$ より大きい数値を指定することはできない。どちらかで違反した場合はエラーとなる。</p>
-margin <top> <left> <bottom> <right>	<p>省略可。</p> <p>透かしを配置するときの余白の設定 (mm 単位)。各要素は左下原点からの距離で指定する。ページ全体を基準に上下左右それぞれの余白を引いた範囲が指定矩形となる。</p> <p>注意：</p> <p>$left$ と $right$ の合計値が PDF の幅より大きい場合や、top と $bottom$ の合計値が PDF の高さより大きい場合はエラーとなる。</p>

-zorder {1 2}	<p>省略可。</p> <p>透かしの Z オーダーの設定。</p> <p>1: 前面 2: 背面</p> <p>指定がない場合、「1: 前面」に配置。</p> <p>1,2 以外が指定された場合、エラーとなる。</p>
-pageRange {0 1 2 3 4 5}	<p>省略可。</p> <p>透かしを挿入する先のページ範囲の設定。</p> <p>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</p> <p>指定がない場合、「0: 全ページ」に挿入する。</p> <p>1~5 以外が指定された場合、エラーとなる。</p> <p>1 と「-notInFirst」、2 と「-notInLast」を指定するとエラーとなる。</p> <p>「-pageRange 5」を指定する場合は「-pageRangeCustom」が指定必須となる。</p>
-pageRangeCustom <Val>	<p>[「-pageRange 5」を指定した場合：設定必須]</p> <p><Val>に透かしを挿入する先のページ番号あるいはページ範囲を指定する。</p> <p>ページ番号は 0 オリジン。そのため 1 ページ目を「0」から数える。</p> <p>複数指定する場合はカンマで区切る。</p> <p>例) -pageRangeCustom “0,2-4”</p> <p>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。</p> <p>指定がない場合、先頭ページに透かしを挿入できる。</p> <p>「-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。</p> <p>指定がない場合、最終ページにも透かしを挿入できる。</p>

	「-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>	省略可。 透かしの透明度の設定。指定がない場合、1.0（不透明）。 <Val>の範囲：0.0～1.0。 0.0 が透明、1.0 が不透明。

8-11 -encrypt : PDF セキュリティ設定

処理内容

入力ファイルにパスワードセキュリティを設定します。

補足：セキュリティ設定における PDF バージョンについて：

- 入力ファイルが持っていた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。

入力ファイルの PDF バージョンが小さかった場合、出力ファイルの暗号化種別に応じた PDF バージョンにバージョンが上がります。各暗号化種別に対応した PDF バージョンは以下の表の通りです。

表：暗号化種別に応じた PDF バージョン

暗号化種別	PDF バージョン
128bit RC4	1.5
128bit AES	1.6
256bit AES	1.7

コマンド例

[コマンド例の実行内容]

test.pdf に対しパスワードセキュリティを設定し、out.pdf として出力する。

今回のパスワードセキュリティは以下内容で設定される。

- 暗号化種別：256bit AES 暗号化
- 開くためのパスワード：なし
- 権限パスワード文字列：「pass123」
- 印刷権限：「低解像度」

- 内容のコピー：不許可

[Windows]

```
AHPDFToolCmd80.exe -encrypt -ownerPass pass123 -filterType 4 -perms2 -print 1 -copy
false -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -encrypt -ownerPass pass123 -filterType 4 -perms2 -print 1 -copy fal
se -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

入力フォルダ内の PDF ファイルに対し、セキュリティを設定します。なお、出力ファイルは入力ファイル名と同名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-userPass <Val>	[-userPass と -ownerPass のどちらかは指定必須] <Val>で文書を開くパスワード（別名ユーザーパスワード）を設定。
-ownerPass <Val>	[-userPass と -ownerPass のどちらかは指定必須] [-filterType を指定する場合 : -ownerPass の指定必須] <Val>で権限パスワードを設定。
-filterType {2 3 4}	省略可。 暗号化種別の設定。指定がない場合、「4: 256bit AES」

	2: 128bit RC4 3: 128bit AES 4: 256bit AES 2~4 以外が指定された場合エラーとなる。 「-filterType」を指定した場合は「-ownerPass」の指定が必須。無い場合はエラーとなる。	
-perms2	「-filterType = 2/3/4」の場合のセキュリティ設定。省略可。	
	-print {0 1 2}	省略可。 印刷の可否。指定がない場合、「0: 許可しない」 0: 許可しない 1: 低解像度 2: 高解像度 0~2 以外が指定された場合エラーとなる。
	-modify {0 1 2 3 4}	省略可。 文書の編集。指定がない場合、「0: 許可しない」 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 0~4 以外が指定された場合エラーとなる。
	-copy {true false}	省略可。 内容のコピー。指定がない場合、「false: 不許可」 true: 許可 false: 不許可
	-accessibility {true false}	省略可。 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。指定がない場合、「true: 有効」 true: 有効 false: 無効

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

8-12 -decrypt : PDF セキュリティ削除

処理内容

入力ファイルのパスワードセキュリティを削除します。

このとき、入力ファイルが持っていた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。

コマンド例

[コマンド例の実行内容]

test.pdf のセキュリティを削除し、out.pdf として出力する。

このときの入力ファイルの権限パスワードは pass123 である。

[Windows]

```
AHPDFToolCmd80.exe -decrypt -d C:\test\test.pdf pass123 -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -decrypt -d /home/antenna/test/test.pdf pass123 -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

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。なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータはありません。

8-13 -optimize : PDF の最適化

処理内容

PDF を最適化します。

ここでの「最適化」とは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・サムネイルといった不要なデータの削除、フォントの統合を行うことを指します。

コマンド例

[コマンド例の実行内容]

test.pdf を以下の条件で最適化し、out.pdf として出力する。

- PDF 内の画像をダウンサンプリングする
- すべての注釈を削除する
- 添付ファイルを削除する
- PDF 内のタグを削除する

[Windows]

```
AHPDFToolCmd80.exe -optimize -optimizelImage true -removeAnnots true -removeAttachment true -removeMetadata true -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -optimize -optimizelImage true -removeAnnots true -removeAttachment true -removeTag true -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対して最適化処理を行います。なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-optimizeImage {true false}	省略可。 画像のダウンサンプリング処理 [*1]。省略した場合ダウンサンプリングは行いません。 true : する false : しない ダウンサンプリングは次の設定で行われます。 <ul style="list-style-type: none">バイキュービック法カラー/グレースケール画像：解像度が 225dpi 以上の場合 150dpi まで縮小する。白黒画像：解像度が 450dpi 以上の場合 300dpi まで縮小する。
-removeOpenAction {true false}	省略可。 開き方設定の削除。省略した場合は削除されない。 true : する false : しない
-removeOutlines {true false}	省略可。 しおりの削除。省略した場合は削除されない。 true : する false : しない
-removeAnnots {true false}	省略可。 注釈の削除。省略した場合は削除されない。

	true : する false : しない
-removeArticles {true false}	省略可。 アーティクルの削除。省略した場合は削除されない。
	true : する false : しない
-removeThumbnails {true false}	省略可。 サムネイルの削除。省略した場合は削除されない。
	true : する false : しない
-mergeFonts {true false}	省略可。 フォントの統合 [*2]。省略した場合、統合は行いません。
	true : 統合する false : 統合しない
-removeAttachment {true false}	省略可。 添付ファイルの削除。省略した場合、削除は行わない。
	true : する false : しない
-removeTag {true false}	省略可。 タグの削除。省略した場合、削除は行わない。
	true : する false : しない
-removePieceInfo {true false}	省略可。 PieceInfo の削除。省略した場合、削除は行わない。
	true : する false : しない

[*1] :

ダウンサンプリングを行ったとき、カラースペースが元 PDF のデータとは異なる場合があります。

[*2] :

フォントの統合を行ったとき、統合されるのは埋め込まれているフォントのみです。その他のフォント統合に関する製品仕様は『5-2-5 PDF 上に存在するフォントを統合する』を参照してください。

[*3]

具体的には、文書情報 (document information dictionary) 以外のメタデータが削除されます。したがって、例えば PDF/A や PDF/X の種類の情報が削除されることになります。

8-14 -addEmbeddedFile : ファイル添付

処理内容

入力 PDF にファイルを添付します。

- 添付ファイルは「-addEmbeddedFile」のパラメータ<Path>で指定します。
- 添付されたファイルは出力 PDF 内の catalog 辞書にエントリーされます。
- 同一 PDF に複数のファイル添付をしたい場合は、実行時に「-addEmbeddedFile」を複数回指定することで実現できます。

コマンド例

[コマンド例の実行内容]

test.pdf に対し sample.docx を添付し、out.pdf として出力する。

[Windows]

```
AHPDFToolCmd80.exe -addEmbeddedFile C:\test\sample.docx -d C:\test\test.pdf -o C:  
\$sav\$out.pdf
```

[Linux]

```
AHPDFToolCmd80 -addEmbeddedFile C:\test\sample.docx -d /home/antenna/test/test.p  
df -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し、ファイル添付を行います。なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
<Path>	[指定必須] 添付するファイルのフルパス名。

8-15 -pageRotate：ページ回転

処理内容

入力 PDF の指定したページを回転します。

- 指定可能な回転角度は右回りに 0 度、90 度、180 度、270 度です。
- ページ番号は 0 オリジンで指定します。そのため、1 ページ目を「0」から数えます。
- 複数ページの回転をしたい場合は複数回指定してください。

コマンド例

[コマンド例の実行内容]

test.pdf の 3 ページ目を 90 度、12 ページ目を 180 度回転し、out.pdf として出力する。

[Windows]

```
AHPDFToolCmd80.exe -pageRotate 2 90 -pageRotate 11 180 -d C:\test\test.pdf -o C:\$sav\$out.pdf
```

[Linux]

```
AHPDFToolCmd80 -pageRotate 2 90 -pageRotate 11 180 -d /home/antenna/test/test.pdf  
-o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
<page>	<p>[指定必須]</p> <p>回転するページ番号。</p> <p>ページ番号は 0 オリジン。そのため、1 ページ目を「0」から数える。</p>
<rotation>	<p>[指定必須]</p> <p>回転角度。指定したページを右回りに回転する。</p> <p>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。</p>

8-16 -pageDelete：ページ削除

処理内容

入力 PDF から指定したページ数をまとめて削除します。

- ページの指定は削除する始点のページ[Index]と、削除するページ数[numPages]をそれぞれ指定して行います。
- ページ番号は 0 オリジンで指定します。そのため、1 ページ目を「0」から数えます。

補足：削除箇所の指定について

- 1 回のコマンド指定で異なる箇所を複数回指定することはできません。

コマンド例

[コマンド例の実行内容]

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、out.pdf として出力する。

[Windows]

```
AHPDFToolCmd80.exe -pageDelete 5 4 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -pageDelete 5 4 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
<index>	[指定必須] 削除を開始するページ位置。 ページ番号は 0 オリジン。そのため、1 ページ目を「0」から数える。
[numPages]	省略可。 削除するページ数。指定がない場合、1 ページ削除する。

8-17 -writeText：文字列追加

処理内容

入力 PDF 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。
- 文字列を挿入するページを指定することができます。指定しない場合は全ページに文字列が挿入されます。
- ページ番号は 0 オリジンで指定します。そのため、1 ページ目を「0」から数えます。
- 挿入する文字列には、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。詳細は『10-2 マクロ文字列』を参照してください。

補足：フォント名の指定について：

- Linux 上で実行する場合、フォント名の指定が必須です。「-font -name」でフォント名を指定してください。Windows で実行する場合、フォントが指定されない場合は日本語環境では「MS-Gothic」が、それ以外の環境では「Arial」がフォント名に指定されます。

コマンド例

[コマンド例の実行内容]

「アンテナハウス株式会社」という文字列を test.pdf の 1 ページ目と 3~5 ページ目に挿入し、out.pdf として出力する。

文字列は以下の設定・配置で挿入される。

- フォント：メイリオフォント(Windows) / IPA ゴシック(Linux)
- フォントサイズ：10 ポイント
- 文字色：{R G B}={1.0 0.0 0.0}
- 透明度：0.8

- 配置：右上

[Windows]

```
AHPDFToolCmd80.exe -writeText -text "アンテナハウス株式会社" -align 3 -font -name Me
iryo -size 10 -colorText 1.0 0.0 0.0 -opacity 0.8 -pageNo "0,2-4" -d C:\test\test.pdf -o
C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -writeText -text "アンテナハウス株式会社" -align 3 -font -name ipa-got
hic -size 10 -colorText 1.0 0.0 0.0 -opacity 0.8 -pageNo "0,2-4" -d /home/antenna/test/t
est.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し、文字列追加を行います。なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-text <Val>	[必須] 挿入する文字列。 <Val>に設定する文字列にはマクロ文字列も利用可能です。詳細は『 10-2 マクロ文字列 』を参照してください。

-pageNo <Val>	<p>省略可。</p> <p><Val>で文字列を挿入するページ番号を設定する。指定がない場合、全ページに挿入する。</p> <p>ページ番号は0オリジン。そのため、1ページ目を「0」から数える。</p> <p>複数指定する場合はカンマで区切る。(例) "0,2-4"</p>	
-vertical	<p>省略可。</p> <p>縦書き指定。</p> <p>指定がある場合、縦書きする。</p> <p>指定がない場合、横書きする。</p>	
-align {1 2 3 4 5 6 7 8 9}	<p>省略可。</p> <p>挿入位置。指定がない場合「5: 中央」に配置。</p> <p>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</p> <p>1~9以外が指定された場合エラーとなる。</p>	
-angle <Val>	<p>省略可。</p> <p><Val>で文字列の角度を指定。反時計回りが正の値。</p> <p>横書きの場合のみ有効。</p>	
-font	<p>[必須]</p> <p>フォント設定。[*1]</p>	
	-name <Val>	[指定必須] フォント名。
	-size <Val>	省略可。 フォントサイズ。 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> 	省略可。 文字の塗りつぶし色。指定がない場合、黒色 (0.0 0.0 0.0) 設定値の範囲 : <r>, <g>, それぞれ 0.0~1.0	
-colorOutline <r> <g> 	省略可。 文字の輪郭の色。指定がない場合、黒色 (0.0 0.0 0.0) 設定値の範囲 : <r>, <g>, それぞれ 0.0~1.0	
-opacity <Val>	省略可。 文字列の透明度。指定がない場合、1.0 (不透明)。 <Val>の範囲 : 0.0~1.0。 0.0 が透明、1.0 が不透明。	

[*1] :

『PDF Tool API』のコマンドラインアプリケーションにおけるフォント情報の利用や製品仕様に関しては『5-2 フォントについて』を参照してください

8-18 -exportAnnotsToFDF: 注釈情報を FDF へエクスポート

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 ファイルの形でエクスポートします。

- エクスポート先は「-exportAnnotsToFDF」自身のパラメータ<outFilePath>で指定します。出力ファイル指定「-o」は必要ありません。

補足：エクスポート対象について：

- エクスポートできるのはマークアップ注釈のみです。例えばリンク注釈などはエクスポートできません。
各注釈のタイプやマークアップ注釈か否かについての詳細は PDF Reference Version 1.7 「8.4.5 Annotation Types」、または ISO 32000-1/ 32000-2 「12.5.6 Annotation Types」を参照してください。

コマンド例

[コマンド例の実行内容]

test.pdf のページにあるマークアップ注釈を抽出して out.fdf にエクスポートする。

[Windows]

```
AHPDFToolCmd80.exe -exportAnnotsToFDF C:\$sav\$out.fdf -d C:\$test\$test.pdf
```

[Linux]

```
AHPDFToolCmd80 -exportAnnotsToFDF /home/antenna/sav/out.fdf -d /home/antenna/test/test.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、出力先パラメータに出力フォルダを指定して一括処理することができます。

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>	[指定必須] 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、コマンドラインアプリケーションが出力時に「fdf」を追加する。

8-19 -importAnnotsFromPDF：他 PDF からの注釈のインポート

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。

補足：インポート対象について

- インポートできるのはマークアップ注釈のみです。例えばリンク注釈はインポートできません。各注釈のタイプやマークアップ注釈か否かについての詳細は PDF Reference Version 1.7 「8.4.5 Annotation Types」、または ISO 32000-1/ 32000-2 「12.5.6 Annotation Types」を参照してください。

コマンド例

[コマンド例の実行内容]

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 として出力する。

[Windows]

```
AHPDFToolCmd80.exe -importAnnotsFromPDF C:\test\annot.pdf -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -importAnnotsFromPDF /home/antenna/test/annot.pdf -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
<Path>	[指定必須] インポート元の PDF ファイルのフルパス

8-20 -insertEmptyPage : 空白ページ挿入

処理内容

入力 PDF に指定したページ数の空白ページをまとめて挿入します。

- 挿入できる空白ページのサイズは「A3」「A4」「A5」で、それぞれ縦長と横長を指定可能です。
- 挿入する位置の指定は挿入する始点のページと、挿入するページ数をそれぞれ指定して行います。
- ページ番号は 0 オリジンで指定します。そのため、1 ページ目を「0」から数えます。

補足：挿入箇所の指定について

- 1 回のコマンド指定で異なる挿入箇所を複数回指定することはできません。

コマンド例

[コマンド例の実行内容]

test.pdf に対し、3 ページ目から 5 ページ分空白ページを挿入し、out.pdf として出力する。

挿入する空白ページは A4 横サイズを指定する。

[Windows]

```
AHPDFToolCmd80.exe -insertEmptyPage -index 2 -number 5 -size 1 -d C:\test\test.pdf  
-o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -insertEmptyPage -index 2 -number 5 -size 1 -d /home/antenna/test  
/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
<code>-index <Val></code>	[必須] 空白ページを挿入するページ位置。 ページ番号は0オリジン。そのため、1ページ目を「0」から数える。
<code>-number <Val></code>	省略可。 挿入するページ数。指定がない場合、1ページ挿入する。
<code>-size {0 1 2 3 4 5}</code>	省略可。 挿入するページサイズ。指定がない場合、「0:A4 縦」 0 : A4 縦 1 : A4 横 2 : A5 縦 3 : A5 横 4 : A3 縦 5 : A3 横 0~5以外が指定された場合、エラーとなる。

8-21 -addPageNumber：ページ番号挿入

処理内容

入力 PDF にページ番号を挿入します。

- 「-index」で何ページ目からページ番号を挿入し始めるかを指定可能です。
このページ指定は 0 オリジンで行います。そのため、1 ページ目を「0」から数えます。指定したページ以降の全てにページ番号が挿入されます。
- 挿入するページ番号について、数え始める番号は任意の数値を指定可能です。
残りのページについては前のページ番号に +1 されたページ番号が自動で挿入されます。
- 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。

コマンド例

[コマンド例の実行内容]

test.pdf に対し、ページ番号を挿入し out.pdf として出力する。

ページ番号は以下の設定で挿入する

- 挿入開始ページ：1 ページ目（未指定）
- 開始番号：1 ページ目をページ番号「11」とする
- 配置：ページの右下
- 文字色：{R,G,B}={0.0, 0.0, 1.0}

[Windows]

```
AHPDFToolCmd80.exe -addPageNumber -startNumber 11 -align 9 -colorText 0.0 0.0 1.0  
-d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -addPageNumber -startNumber 11 -align 9 -colorText 0.0 0.0 1.0 -d /  
home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し、ページ番号挿入を行います。なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-index <Val>	省略可。 ページ番号を挿入しはじめるページ位置。 ページ番号は 0 オリジン。そのため、1 ページ目を「0」から数える。 指定がない場合「0」。
-startNumber <Val>	省略可。 挿入するページ番号の開始番号。指定がない場合「1」。
-align {1 2 3 4 5 6 7 8 9}	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	省略可 フォント設定[*1]。	
	-name <value>	省略可。 フォント名。 指定がない場合、「Helvetica」
	-size <value>	省略可。 フォントサイズ。 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> 	省略可。 文字の塗りつぶし色。 設定値の範囲：<r>, <g>, 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> 	省略可。 文字の輪郭の色。 設定値の範囲：<r>, <g>, 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>	省略可。 文字列の透明度の設定。指定がない場合、1.0 (不透明)。 <Val>の範囲：0.0~1.0。 0.0 が透明、1.0 が不透明。	

[*1] :

『PDF Tool API』のコマンドラインアプリケーションにおけるフォント情報の利用や製品仕様に関しては『5-2 フォントについて』を参照してください

8-22 -pageExtract：ページ抽出

処理内容

指定されたページやページ範囲を抽出します。

- 抽出する際に分割指定をするかどうかで抽出されたページの出力方法が変わります。
- 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別々の 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。
- 「-copyInfo」を指定することで文書情報を出力ファイルにコピーすることができます。

補足事項：

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

コマンド例

[コマンド例の実行内容]

test.pdf から 1 ページ目と、4~7 ページ目を抽出し、出力する。出力時に分割指定をする。

そのため保存ファイル名は分割され、出力ファイルはそれぞれ「out_0.pdf」、「out_3-6.pdf」となる。

[Windows]

```
AHPDFToolCmd80.exe -pageExtract "0,3-6" -divideFile -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -pageExtract "0,3-6" -divideFile -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
<PageRange>	<p>[指定必須]</p> <p>抽出するページ番号またはページ範囲。</p> <p>ページ番号は0オリジン。そのため、1ページ目を「0」から数える。</p> <p>複数指定する場合はカンマで区切る。 (例) "0,2-4"</p>
-divideFile	<p>省略可</p> <p>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。</p> <p>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。</p> <p>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。</p>
-copyInfo	<p>省略可</p> <p>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。</p> <p>指定がない場合、文書情報はコピーしない。</p>

8-23 -embedFonts : フォント埋め込み

処理内容

入力 PDF にフォントが埋め込まれていない場合、そのフォントを PDF に埋め込みます。

補足事項：

- フォントを埋め込むことができるのは、以下の条件をすべて満たす場合だけです。
 - 動作環境に該当フォントが存在する
 - 該当フォントの埋め込みが許可されている
- フォントを埋め込んだ PDF から埋め込みフォント情報を削除することはできません。
- フォント埋め込みの製品仕様に関するその他の詳細は、『5-2-4 PDF 上に存在するフォントを埋め込み状態にする』を参照してください。

コマンド例

[コマンド例の実行内容]

test.pdf に対しフォント埋め込みを行い、out.pdf として出力する。

[Windows]

```
AHPDFToolCmd80.exe -embedFonts -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -embedFonts -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し、フォント埋め込み処理を行います。なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータはありません。

8-24 -extractImage：画像抽出

処理内容

入力 PDF のページに含まれる画像データを抽出します。

- 出力フォーマットは自動、Bitmap、JPEG、PNG から選択可能です。自動の場合は出力画像フォーマットが Bitmap、JPEG、PNG の中から選ばれます。
- 画像はパラメータで指定された出力フォルダパスに出力されます。
- このコマンドを実行するのに出力ファイル指定「-o」は必要ありません。
- 出力ファイル名は「入力ファイル名_ページ番号_連番」に拡張子を付けたものになります。この連番の開始は「0001」でページが変わることに連番はリセットされます。

補足：抽出対象について

- 抽出するのは PDF のページに配置された画像データのみです。これはページ全体を画像化する処理ではありません。

コマンド例

[コマンド例の実行内容]

test.pdf に存在する画像データのうち、40ppi 以上 100ppi 以下のものを JPEG 形式のファイルとして指定フォルダに出力する。この際、抽出時に変更を加えない。

[Windows]

```
AHPDFToolCmd80.exe -extractImage C:\out -format 2 -morePPI 40 -lessPPI 100 -passThrough true -d C:\test\test.pdf
```

[Linux]

```
AHPDFToolCmd80 -extractImage /home/antenna/sav -format 2 -morePPI 40 -lessPPI 10
```

```
0 -passThrough true -d /home/antenna/test/test.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルから画像データを抽出します。出力フォルダはパラメータ [outFolderPath] で指定してください。

出力フォルダには入力ファイル名のサブフォルダが作成されます。

出力される画像データは各サブフォルダ内に保存されます。

パラメータ

パラメータ	内容
<outFolderPath>	<p>[指定必須]</p> <p>画像ファイルの出力フォルダパスを指定する。</p> <p>出力ファイル名は「入力ファイル名_ページ番号_連番」となる。</p> <p>ページ番号の開始は「1」</p> <p>連番の開始は「0001」でページごとにリセットする。</p>
-pageNo	<p>省略可。</p> <p>画像を抽出するページ番号。指定がない場合、全ページの画像を抽出する。</p> <p>ページ番号は 0 オリジン。そのため 1 ページ目を「0」から数える。</p> <p>複数指定する場合はカンマで区切る。</p> <p>例) -pageNo “0,2-4”</p> <p>1 ページと 3~5 ページから抽出される。</p>
-format {0 1 2 3}	<p>省略可。</p> <p>画像の保存形式。指定がない場合、「0 = AUTO」</p>

	0 = AUTO 1 = Bitmap 2 = JPEG 3 = PNG
-morePPI <Val>	<p>省略可。</p> <p>このパラメータが指定された場合、<Val>で指定された解像度以上の画像を選んで抽出する。</p> <p>「-lessPPI」との同時指定が可能。その場合、「-morePPI」以上「-lessPPI」以下の画像を抽出する。どちらも指定されていない場合はすべての解像度の画像を抽出する。</p>
-lessPPI <Val>	<p>省略可。</p> <p>このパラメータが指定された場合、<Val>で指定された解像度以下の画像を抽出する。</p> <p>「-morePPI」との同時指定が可能。その場合、「-morePPI」以上「-lessPPI」以下の画像を抽出する。どちらも指定されていない場合はすべての解像度の画像を抽出する。</p>
-passThrough {true false}	<p>省略可。</p> <p>変更を加えずに抽出する[*1]かどうか。省略した場合は false が指定されたものとみなす。</p> <p>true : 変更を加えない false : 変更する</p> <p>画像の出力フォーマットに JPEG を指定した場合のみ有効。</p> <p>JPEG 以外の出力フォーマットの場合はエラー。</p>

[*1] :

PDF への画像埋め込み時に再圧縮をされていない場合にのみ、埋め込み時の画像をそのまま抽出可能。具体的には PDF への埋め込み時に以下の条件を満たしている必要がある。

- カラースペースが、DeviceRGB, DeviceGray, 無指定(=PDF_EMPTY_NAME)のいずれかで指定されていること
- マスクが指定されていないこと
- Decode の値がデフォルトであること
- トランスマスク関数を表すグラフィックステートパラメータ辞書の『ExtGState:TR』が指定されていないこと (一

般にトランスマニア関数はガムマ補正の用途で使用される機能です)

8-25 -extractText：テキスト抽出

処理内容

入力 PDF ファイルからテキストを抽出し、テキストファイルの形で出力します。

- テキストファイルはパラメータで指定された出力パスに出力されます。
- このコマンドを実行するのに出力ファイル指定「-o」は必要ありません。

コマンド例

[コマンド例の実行内容]

test.pdf のテキストを抽出して out.txt に出力する。

抽出時、ページ指定やテキスト抽出順は以下の設定に従う。

- ページ指定：1 ページ目、3~5 ページ目
- 位置：[left, bottom, right, top] がそれぞれ [0mm, 0mm, 100mm, 100mm] の矩形
- テキスト抽出順：ページ上に存在するテキストを座標順に並べ替える

[Windows]

```
AHPDFToolCmd80.exe -extractText C:\sav\out.txt -pageNo "0,2-4" -sort -rect 0 0 100 100 -d C:\test\test.pdf
```

[Linux]

```
AHPDFToolCmd80 -extractText /home/antenna/sav/out.txt -pageNo 0,2-4 -sort -rect 0 0 100 100 -d /home/antenna/test/test.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルからテキストを抽出します。出力フォルダはパラメータ [outTextFilePath] で指定してください。

出力ファイルは入力ファイル名の拡張子を「.txt」に変更したファイル名で指定したフォルダに出力されます。

パラメータ

パラメータ	内容
<outTextFilePath>	[必須] このパラメータは「-extractText」の直後に指定する必要があります。 テキスト出力先の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 <Val>	省略可。 <Val>でテキストを抽出するページ番号を指定。指定がない場合、全ページのテキストを抽出する。 ページ番号は 0 オリジン。そのため、1 ページ目を「0」から数える。 複数指定する場合はカンマで区切る。（例）"0,2-4"
-sort	元データを座標順に並べ替えた上で抽出する。
-rect <left> <bottom> <right> <top>	省略可。 テキストを取得する範囲（mm 単位）。指定が無い場合、ページ全体のテキストを取得する。 -sort 指定がある場合：指定領域内でソートする。

	<p>-rect が複数指定された場合：最初の-rect を採用し、2 個目以降の-rect は無視される。</p>
--	--

8-26 -imageToPdf：画像ファイルの PDF 化

処理内容

画像ファイルを PDF ファイルに変換して出力します。

- 対応する画像形式はそれぞれ、bitmap、jpeg、png、tiff、gif です。
- 出力される PDF ファイルのバージョンは「1.4」です。
- 画像を PDF ファイルに変換する際に dpi の指定、画像の回転をすることができます。
- 指定可能な回転角度は 0 度、90 度、180 度、270 度で、画像は右回りで回転します。

補足：「-d」パラメータの指定対象について

- 「-imageToPDF」コマンドを使用した場合、「-d」パラメータで指定するのは PDF ファイルではなく、変換元画像のファイルパスである点に注意してください。

コマンド例

[コマンド例の実行内容]

test.jpg を out.pdf に変換して出力する。

out.pdf 上の画像の設定は以下に従う。

- 解像度：300dpi
- 回転：右回りに 90 度回転させる

[Windows]

```
AHPDFToolCmd80.exe -imageToPdf -dpi 300 -rotateAngle 90 -d C:\test\test.jpg -o C:\s
av\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -imageToPdf -dpi 300 -rotateAngle 90 -d /home/antenna/test/test.jpg  
-o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の画像ファイルを全て PDF に変換します。

出力ファイルは「入力ファイル名.拡張子」の後に「.pdf」を付けたファイル名で指定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 に変換する元画像のファイルパスを指定する。 対応形式：bitmap / jpeg / png / tiff / gif
-o	[必須] PDF の出力ファイルパスを指定する。
-dpi <Val>	省略可。 <Val>で解像度を整数値で指定する。指定がない場合、画像が持つ解像度となる。
-rotateAngle <Val>	省略可。 回転角度を指定する。時計周りに回転する。指定がない場合、「0」（度） <Val>で設定可能な値：0/90/180/270（度）

8-27 -extractEmbeddedFile:添付ファイルの抽出

処理内容

PDF ファイルに埋め込まれたすべての添付ファイルを抽出します。

- 添付ファイルの出力先フォルダは<outFolderPath>で指定します。
- このコマンドを実行するのに出力ファイル指定「-o」は必要ありません。

コマンド例

[コマンド例の実行内容]

test.pdf に埋め込まれた添付ファイルを指定フォルダである C:\out に出力する。

[Windows]

```
AHPDFTToolCmd80.exe -extractEmbeddedFile C:\out -d C:\test\test.pdf
```

[Linux]

```
AHPDFTToolCmd80 -extractEmbeddedFile /home/antenna/sav -d /home/antenna/test/test.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は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。出力フォルダはパラメータ[outFolderPath]で指定してください。

出力フォルダには入力ファイル名のサブフォルダが作成されます。

抽出された添付ファイルはこの各サブフォルダ内に保存されます。

パラメータ

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

8-28 -importAnnotsFromFDF : FDF からの注釈のインポート

処理内容

FDF ファイルからマークアップ注釈をインポートします。

これは例えば「-exportAnnotsToFDF」でエクスポートされた FDF 形式の注釈データをインポートするのに役立ちます。

インポートする FDF ファイルは「-importAnnotsFromFDF」自身のパラメータ<FDFFilePath>で指定します。

補足事項 :

- インポートできるのはマークアップ注釈のみです。例えばリンク注釈などはインポートできません。
- 各注釈のタイプやマークアップ注釈か否かについての詳細は PDF Reference Version 1.7 「8.4.5 Annotation Types」、または ISO 32000-1/ 32000-2 「12.5.6 Annotation Types」を参照してください。

コマンド例

[コマンド例の実行内容]

import.fdf から test.pdf に注釈を取り込み、out.pdf として出力する。

[Windows]

```
AHPDFToolCmd80.exe -importAnnotsFromFDF C:\test\import.fdf -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -importAnnotsFromFDF /home/antenna/test/import.fdf -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
<FDFFilePath>	[必須] インポートする FDF ファイルパス

8-29 -fixedUpPDFA : PDF/A 変換

処理内容

PDF ファイルを PDF/A に変換します。

コマンド 「-fixedUpPDFA」 で変換可能な PDF/A の規格は以下となります。

- PDF/A-1a
- PDF/A-1b
- PDF/A-2a
- PDF/A-2b
- PDF/A-3a
- PDF/A-3b

[処理の詳細]

- 変換時に準拠エラーが発生した場合、エラーコードとエラーメッセージが出力されます。準拠エラーが発生しなかった場合「ok!」と表示されます。
- 準拠エラーに関する処理ログを出力することも可能です。処理ログの出力は-log で指定します。
- 処理ログは出力ファイルと同じフォルダに.log 形式のテキストファイルで出力されます。
- 処理ログは[出力ファイル名.log]のファイル名で出力されます。
- 変換の際に用いる RGB カラープロファイル、CMYK カラープロファイルをそれぞれ指定可能です。
- icc プロファイルの指定がない場合『PDF Tool API』は sRGB2014.icc または JapanColor2001Coated.icc を次の順で検索します。
検索順は OS により異なります。

[Windows]

1. PdfTk80.dll があるフォルダ内
2. 環境変数「PTL80_ICCPROFILE_PATH」で指定されたフォルダ内]

[Linux]

環境変数「PTL80_ICCPROFILE_PATH」で指定されたフォルダ内

コマンド例

[コマンド例の実行内容]

test.pdf を PDF/A-2b に変換し、out.pdf として出力する。

この際、プロファイルとして sRGB2014.icc と JapanColor2001Coated.icc を使用する。

コマンド実行時の処理ログを-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]

```
AHPDFToolCmd80.exe -fixedUpPDFA -type 2b -RGBProfile sRGB2014.icc -CMYKProfile JapanColor2001Coated.icc -log -d c:\in\test.pdf -o c:\out\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -fixedUpPDFA -type 2b -RGBProfile sRGB2014.icc -CMYKProfile JapanColor2001Coated.icc -log -d /home/antenna/in/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し、同一形式の PDF/A 変換を行います。なお、出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。処理ログファイルも「入力ファイル名.log」の形式で出力フォルダに出力されます。

パラメータ

パラメータ	内容
<code>-type {1a 1b 2a 2b 3a 3b}</code>	<p>[必須]</p> <p>PDF/A 規格の種類を指定</p> <p>1a : PDF/A-1a に変換する</p> <p>1b : PDF/A-1b に変換する</p> <p>2a : PDF/A-2a に変換する</p> <p>2b : PDF/A-2b に変換する</p> <p>3a : PDF/A-3a に変換する</p> <p>3b : PDF/A-3b に変換する</p>
<code>-RGBProfile <Val></code>	<p>省略可。</p> <p><code><Val></code>で RGB カラープロファイル名を指定。</p> <p>フルパスで指定する。ファイル名のみ指定の場合、コマンドラインアプリケーションと同階層に存在するものとする。</p> <p>未指定の場合は Windows では</p> <ol style="list-style-type: none"> 1. PdfTk80.dll があるフォルダ内 2. 環境変数「PTL80_ICCPROFILE_PATH」で指定されたフォルダ内 <p>の順で検索する。</p> <p>Linux では環境変数「PTL80_ICCPROFILE_PATH」で指定されたフォルダ内を検索する。</p> <p>未指定で検索先にカラープロファイルが無い場合はエラーとなる。</p>
<code>-CMYKProfile <Val></code>	<p>省略可。</p> <p><code><Val></code>で CMYK カラープロファイル名を指定</p> <p>フルパスで指定する。ファイル名のみ指定の場合、コマンドラインアプリケーションと同階層に存在するものとする。</p>

	<p>未指定の場合は Windows では</p> <ol style="list-style-type: none"> 1. PdfTk80.dll があるフォルダ内 2. 環境変数[PTL80_ICCPROFILE_PATH]で指定されたフォルダ内 <p>の順で検索する。</p> <p>Linux では環境変数[PTL80_ICCPROFILE_PATH]で指定されたフォルダ内を検索する。</p> <p>未指定で検索先にカラープロファイルが無い場合はエラーとなる。</p>
-log	<p>省略可。</p> <p>指定した場合、処理ログをテキストファイルに出力する。</p> <p>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。</p>

8-30 -validatePDFA : PDF/A 準拠確認

処理内容

PDF ファイルが指定した PDF/A の規格に準拠しているかどうかを確認します。

コマンド「-validatePDFA」で準拠チェック可能な PDF/A の規格は以下の通りです。

- PDF/A-1a
- PDF/A-1b
- PDF/A-2a
- PDF/A-2b
- PDF/A-3a
- PDF/A-3b

[処理の詳細]

- どのタイプの PDF/A の準拠チェックをするのかは-type で指定します。
- 入力 PDF が指定した PDF/A に準拠しておらず、準拠チェック時にエラーとなった場合はエラーコードとエラーメッセージが出力されます。
- 準拠エラーが発生しなかった場合「ok!」と表示されます。
- 処理ログを使うことで準拠チェック時のエラーコードとエラーメッセージをテキストファイルの形で残すことができます。処理ログの出力は-log で指定します。
- 処理ログを出力する場合は入力ファイルと同じフォルダに[入力ファイル名.log]のテキストファイルで出力されます。

補足事項：

- このコマンドを実行するのに出力ファイル指定「-o」は必要ありません。

コマンド例

[コマンド例の実行内容]

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 で指定した入力ファイルと同じ場所に[(入力ファイル名).log]のファイル名で出力する。

[Windows]

```
AHPDFToolCmd80.exe -validatePDFA -type 2b -log -d c:\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -validatePDFA -type 2b -log -d /home/antenna/out.pdf
```

フォルダ指定：対応有り

「-d」のパラメータで入力フォルダ指定して一括処理をすることができます。

フォルダ指定をされた場合は入力フォルダ内の PDF ファイルに対し準拠チェックを行います。-log を指定された場合、各処理ログファイルは入力ファイルと同じフォルダに[(入力ファイル名).log]の形式で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-type {1a 1b 2a 2b 3a 3b}	[必須] PDF/A 規格の種類を指定 1a : PDF/A-1a の準拠確認を行う 1b : PDF/A-1b の準拠確認を行う

	2a : PDF/A-2a の準拠確認を行う 2b : PDF/A-2b の準拠確認を行う 3a : PDF/A-3a の準拠確認を行う 3b : PDF/A-3b の準拠確認を行う
-log	省略可。 指定した場合、処理ログをテキストファイルに出力する。 出力先は「-d」で指定した入力ファイルと同じ階層。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。

8-31 -dividePdf：ファイル分割

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ます。

- 指定可能なファイル分割条件は以下の3つです。
 - 指定したページ数ごとに分割する：「-each」で指定
 - 指定したファイル数に分ける：「-into」で指定
 - 一層目のしおりごとに分割する：「-outline」で指定
- 分割したファイルに対するサフィックスは「-safix」で指定可能です。
- 「-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。

補足事項：

- 「-each」、「-into」、「-outline」のいずれかが必須です。指定されなかった場合はエラーとなります。また、同時に複数を指定した場合、最初に指定したパラメータ以外は無視されます。
- 「-o」でファイルのフルパスを指定する際は拡張子を忘れないようにしてください。拡張子を付けなかった場合はファイル名にサフィックスを付けたファイルが拡張子無しで出力されます。

コマンド例

[コマンド例の実行内容]

test.pdf を5ページずつに分割してoutフォルダに出力する。

出力するファイル名は入力ファイル名に連番を付与した形式で指定する。

[Windows]

```
AHPDFToolCmd80.exe -dividePdf -each 5 -suffix 2 -d c:\in\test.pdf -o c:\out
```

[Linux]

```
AHPDFToolCmd80 -dividePdf -each 5 -suffix 2 -d /home/antenna/in/test.pdf -o /home/antenna/out
```

フォルダ指定：対応無し

入力ファイルをフォルダ指定することはできませんが、出力ファイルを指定する-o でフォルダを指定することは可能です。その場合、入力ファイル名にサフィックスを付けた出力ファイル名で指定したフォルダに分割された PDF が output されます。

パラメータ

[-each, -into, -outline のいずれかが必須]

パラメータ	内容
-each <Val>	<Val>で指定したページ数ごとに分割
-into <Val>	<Val>で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。

- | | |
|--|---|
| | <p>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詰めする。</p> <p>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詰めする。</p> <p>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のみ有効。</p> <p>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のみ有効。</p> <p>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のみ有効。</p> |
|--|---|

注記：

「-each」、「-into」、「-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。

8-32 -getPageSize：ページサイズの取得

処理内容

指定されたページのサイズを標準出力に出力します。

パラメータにより、MediaBoxなどの詳細な表示矩形情報を出力することも可能です。

コマンド例

[コマンド例の実行内容]

test.pdf の 3 ページ目のページサイズ情報を出力する。この際、ページサイズのほかに MediaBox、CropBox、BleedBox、TrimBox、ArtBox の情報を取得する。

[Windows]

```
AHPDFToolCmd80.exe -getPageSize -pageNo 2 -detail -d C:\test\test.pdf
```

[Linux]

```
AHPDFToolCmd80 -getPageSize -pageNo 2 -detail -d /home/antenna/test/test.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
-------	----

<p>-pageNo <Val></p>	<p>省略可。 ページサイズを取得するページ番号。指定が無い場合は先頭ページのページサイズを取得する</p> <p>ページ番号は 0 オリジン。そのため、1 ページ目を「0」から数える。</p> <p>指定できるページは 1 つのみ。</p>
<p>-detail</p>	<p>このパラメータが指定された場合はページサイズのほかに MediaBox、CropBox、BleedBox、TrimBox、ArtBox の情報を取得する。</p>

出力内容

詳細な出力内容は『取得情報の出力内容』内の『9-4 -getPageSize：ページサイズ情報取得』の項を参照してください。

8-33 -searchTextAndHighlight : テキスト検索後にハイライト注釈挿入

処理内容

指定された文字列を検索し、ヒットした部分にハイライト注釈をつけます

テキスト検索時に、読み取るデータを座標順に並べ替えた上で検索するオプションが指定可能です。これにより、『文字列の形に並んで見えるが内部的には順序がばらばらのデータ』も正しく読み取ることが可能になります。

コマンド例

[コマンド例の実行内容]

test.pdf 内で「テスト」「abcd」というキーワードを検索し、ヒットした部分にハイライト注釈をつける。検索時は元データを座標順に並べ替えた上で検索する。また、アルファベットの大文字・小文字を区別して検索する。

ハイライト注釈の設定は以下の通り。

- 注釈の色 : {R G B}={0.5, 0.0, 0.7}
- 不透明度 0.5

[Windows]

```
AHPDFTCmd80.exe -searchTextAndHighlight -sort -pageNo 2 -keyword "テスト" "abcd"
" -caseSensitive -color 0.5 0.0 0.7 -opacity 0.5 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFTCmd80 -searchTextAndHighlight -sort -pageNo 2 -keyword "テスト" "abcd" -c
```

```
aseSensitive -color 0.5 0.0 0.7 -opacity 0.5 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
-pageNo <Val>	省略可。 テキストを検索するページ番号。指定がない場合、全ページのテキストを検索する。 ページ番号は 0 オリジン。そのため、1 ページ目を「0」から数える。 複数指定する場合はカンマで区切る。（例）"0,2-4"
-keyword <Val>	検索するテキスト。-keywordRegEx と同時指定可。 文字列キーワードは""（ダブルクオーテーション）で囲む。 検索する要素は 3 個まで指定できる。4 個以上指定した場合、エラーとなる。 「-keyword」と「-keywordRegEx」のどちらも指定されていない場合はエラーとなる。
-keywordRegEx <Val>	正規表現の検索キーワード。-keyword と同時指定可。 文字字列キーワードは""（ダブルクオーテーション）で囲む。 検索する要素は 3 個まで指定できる。4 個以上指定した場合、エラーとなる。 「-keyword」と「-keywordRegEx」のどちらも指定されていない場合はエラーとなる。
-sort	省略可。 指定した場合、検索元データを座標順に並べた上で検索する。 指定しない場合、内部データの並び順で文字列を検索する。 特に英文の場合は、指定時、文字列同士の間隔をもとに単語のかたまりを認識し

	た状態での文字検索が可能になる。
-caseSensitive	<p>省略可</p> <p>指定した場合、検索文字列のアルファベットについて、大文字・小文字を区別して検索する。</p> <p>指定しない場合、大文字・小文字を区別しないで検索する。</p>
-color <r> <g> 	<p>省略可。</p> <p>注釈の塗りつぶしの色。指定がない場合、黒色 (0.0 0.0 0.0)</p> <p>設定値の範囲 : <r>, <g>, それぞれ 0.0~1.0</p>
-opacity <Val>	<p>省略可。</p> <p>注釈の不透明度。指定がない場合、1.0 (不透明)。</p> <p>指定時は「-color」と併用しての指定が必要。</p> <p>設定値<Val>の範囲 : 0.0~1.0。</p> <p>0.0 が透明、1.0 が不透明。</p>

8-34 -searchTextAndRedact : テキスト検索後に Redact 注釈挿入

処理内容

指定された文字列を検索し、ヒットした部分に Redact 注釈をつけます。

Redact 注釈は適用箇所に墨消しをする注釈です。Redact 注釈の状態では該当箇所のデータを閲覧可能ですが、適用操作をすることで注釈の部分はマスクの色で塗りつぶされ、その下にある文字データは削除されます。

Redact 注釈はソフトウェアや仕様書によっては、「Redaction 注釈」、「墨消し」と表現されている場合もあります。

- テキスト検索時に、読み取るデータを座標順に並べ替えた上で検索するオプションが指定可能です。
これにより、『文字列の形に並んで見えるが内部的には順序がばらばらのデータ』も正しく読み取ることが可能になります。

コマンド例

[コマンド例の実行内容]

test.pdf 内で「テスト」「abcd」というキーワードを検索し、ヒットした部分に Redact 注釈をつける。検索時は元データを座標順に並べ替えた上で検索する。また、アルファベットの大文字・小文字を区別して検索する。

Redact 注釈の設定は以下の通り。

- 注釈の色 : {R G B}={0.5, 0.0, 0.7}
- アウトラインの色 : {R G B}={0.0, 0.0, 0.1}
- マスクの色 : {R G B}={1.0, 0.0, 0.0}

- 不透明度 0.5

[Windows]

```
AHPDFToolCmd80.exe -searchText -sort -pageNo 2 -keyword "テスト" "abcd" -caseSensitive -color 0.5 0.0 0.7 -outlineColor 0.0 0.0 0.1 -maskColor 1.0 0.0 0.0 -opacity 0.5 -d C:¥test¥test.pdf -o C:¥sav¥out.pdf
```

[Linux]

```
AHPDFToolCmd80 -searchText -sort -pageNo 2 -keyword "テスト" "abcd" -caseSensitive -color 0.5 0.0 0.7 -outlineColor 0.0 0.0 0.1 -maskColor 1.0 0.0 0.0 -opacity 0.5 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
-pageNo <Val>	省略可。 テキストを検索するページ番号。指定がない場合、全ページのテキストを検索する。 ページ番号は 0 オリジン。そのため、1 ページ目を「0」から数える。 複数指定する場合はカンマで区切る。（例）"0,2-4"
-keyword <Val>	検索するテキスト。-keywordRegEx と同時指定可。 文字列キーワードは""（ダブルクオーテーション）で囲む。 検索する要素は 3 個まで指定できる。4 個以上指定した場合、エラーとなる。 「-keyword」と「-keywordRegEx」のどちらも指定されていない場合はエラーとなる。

-keywordRegEx <Val>	正規表現の検索キーワード。-keyword と同時指定可。 文字字列キーワードは"" (ダブルクオーテーション) で囲む。 検索する要素は 3 個まで指定できる。4 個以上指定した場合、エラーとなる。 「-keyword」と「-keywordRegEx」のどちらも指定されていない場合はエラーとなる。
-sort	省略可。 指定した場合、検索元データを座標順に並べた上で検索する。 指定しない場合、内部データの並び順で文字列を検索する。 特に英文の場合は、指定時、文字列同士の間隔をもとに単語のかたまりを認識した状態での文字検索が可能になる。
-caseSensitive	省略可 指定した場合、検索文字列のアルファベットについて、大文字・小文字を区別して検索する。 指定しない場合、大文字・小文字を区別しないで検索する。
-color <r> <g> 	省略可。 Redact 注釈本体の塗りつぶし色。指定がない場合、黒色 (0.0 0.0 0.0) 設定値の範囲 : <r>, <g>, それぞれ 0.0~1.0
-outlineColor <r> <g> 	省略可。 Redact 注釈のアウトラインの色。指定がない場合、黒色 (0.0 0.0 0.0) 設定値の範囲 : <r>, <g>, それぞれ 0.0~1.0
-maskColor <r> <g> 	省略可。 Redact 注釈適用後のマスクの色。指定がない場合、黒色 (0.0 0.0 0.0) 設定値の範囲 : <r>, <g>, それぞれ 0.0~1.0
-opacity <Val>	省略可。 注釈の不透明度。指定がない場合、1.0 (不透明)。 指定時は「-color」と併用しての指定が必要。 設定値<Val>の範囲 : 0.0~1.0。 0.0 が透明、1.0 が不透明。

8-35 -searchTextAndSetMask : テキスト検索後にマスクをかける

処理内容

指定された文字列を検索し、ヒットした部分をマスクの色で塗りつぶし、該当箇所の文字データを削除します。

- テキスト検索時に、読み取るデータを座標順に並べ替えた上で検索するオプションが指定可能です。
これにより、『文字列の形に並んで見えるが内部的には順序がばらばらのデータ』も正しく読み取ることが可能になります。

コマンド例

[コマンド例の実行内容]

test.pdf 内で「テスト」「abcd」というキーワードを検索し、ヒットした部分にマスクをかける。検索時は元データを座標順に並べ替えた上で検索する。また、アルファベットの大文字・小文字を区別して検索する。

マスクの設定は以下の通り。

- マスクの色 : {R G B}={1.0, 0.0, 0.0}
- 不透明度 0.5

[Windows]

```
AHPDFToolCmd80.exe -searchText -sort -pageNo 2 -keyword "テスト" "abcd" -caseSensitive -maskColor 1.0 0.0 0.0 -opacity 0.5 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -searchText -sort -pageNo 2 -keyword "テスト" "abcd" -caseSensitive  
-maskColor 1.0 0.0 0.0 -opacity 0.5 -d /home/antenna/test/test.pdf -o /home/antenna/s  
av/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
-pageNo <Val>	省略可。 テキストを検索するページ番号。指定がない場合、全ページのテキストを検索する。 ページ番号は0オリジン。そのため、1ページ目を「0」から数える。 複数指定する場合はカンマで区切る。（例）"0,2-4"
-keyword <Val>	検索するテキスト。-keywordRegExと同時指定可。 文字列キーワードは""（ダブルクオーテーション）で囲む。 検索する要素は3個まで指定できる。4個以上指定した場合、エラーとなる。 「-keyword」と「-keywordRegEx」のどちらも指定されていない場合はエラーとなる。
-keywordRegEx <Val>	正規表現の検索キーワード。-keywordと同時指定可。 文字列キーワードは""（ダブルクオーテーション）で囲む。 検索する要素は3個まで指定できる。4個以上指定した場合、エラーとなる。 「-keyword」と「-keywordRegEx」のどちらも指定されていない場合はエラーとなる。
-sort	省略可。 指定した場合、検索元データを座標順に並べた上で検索する。

	<p>指定しない場合、内部データの並び順で文字列を検索する。</p> <p>特に英文の場合は、指定時、文字列同士の間隔をもとに単語のかたまりを認識した状態での文字検索が可能になる。</p>
-caseSensitive	<p>省略可</p> <p>指定した場合、検索文字列のアルファベットについて、大文字・小文字を区別して検索する。</p> <p>指定しない場合、大文字・小文字を区別しないで検索する。</p>
-maskColor <r> <g> 	<p>省略可。</p> <p>マスクの色。指定がない場合、黒色 (0.0 0.0 0.0)</p> <p>設定値の範囲 : <r>, <g>, それぞれ 0.0~1.0</p>
-opacity <Val>	<p>省略可。</p> <p>注釈の不透明度。指定がない場合、1.0 (不透明)。</p> <p>指定時は「-color」と併用しての指定が必要。</p> <p>設定値<Val>の範囲 : 0.0~1.0。</p> <p>0.0 が透明、1.0 が不透明。</p>

8-36 -zoom : ページを拡大・縮小する

処理内容

ページサイズを指定された用紙サイズに拡大・縮小します。

縦横比は元 PDF のものが維持されます。

- 「-h」以外の他のコマンドとの併用はできません。

コマンド例

[コマンド例の実行内容]

test.pdf の 1 ページ目と 3~5 ページ目のサイズを A3 横サイズに変更する。

[Windows]

```
AHPDFToolCmd80.exe -zoom -pageNo "0,2-4" -size 3 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -zoom -pageNo "0,2-4" -size 3 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
-pageNo <Val>	省略可。 サイズ変更するページ番号を指定する。省略した場合、すべてのページが対象となる。 ページ番号は0オリジン。そのため、1ページ目を「0」から数える。 複数指定する場合はカンマで区切る。（例）"0,2-4"
-size {0 1 2 3 4 5 6 7 8 9}	省略可。 用紙サイズ。指定が無い場合は「4:A4縦」が指定されたものとみなす。 0 : A2 縦 1 : A2 横 2 : A3 縦 3 : A3 横 4 : A4 縦 5 : A4 横 6 : B4 縦 7 : B4 横 8 : B5 縦 9 : B5 横 元の縦横比は維持される。

8-37 -setCustomProperty : カスタムプロパティの設定

処理内容

文書情報にカスタムプロパティを設定します。

- 1回の実行で設定できるカスタムプロパティは名前・値が一組です。
- 複数のカスタムプロパティを設定することも可能です。設定したいカスタムプロパティが複数ある場合は「-name」と「-value」を複数回指定してください。
「-name」と「-value」の数が一致しない場合はエラーとなります。

コマンド例

[コマンド例の実行内容]

test.pdf に以下の名前・値を持ったカスタムプロパティを設定し、出力する。

- 名前：商品コード
- 値：0001

[Windows]

```
AHPDFToolCmd80.exe -setCustomProperty -name "商品コード" -value "0001" -d C:\test
\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -setCustomProperty -name "商品コード" -value "0001" -d /home/ante
nna/test/test.pdf -o /home/antenna/sav/out.pdf
```

フォルダ指定：対応有り

-d のパラメータに入力フォルダを、-o のパラメータに出力フォルダを指定して一括設定することができます。

フォルダ指定をされた場合はフォルダ内の全てのファイルに同じカスタムプロパティが設定されます。出力ファイルは入力ファイル名と同一名で出力フォルダに出力されます。

パラメータ

パラメータ	内容
-name <Val>	[必須] カスタムプロパティの名前。
-value <Val>	[必須] カスタムプロパティの値。

8-38 -setMaskUsingRedaction : Redact 注釈を適用しマスクをかける

処理内容

文書内のすべての Redact 注釈を適用します。適用操作で Redact 注釈の部分はマスクの色で塗りつぶされ、その下にある文字データは削除されます。

適用時にマスクする範囲や塗りつぶす色は、Redact 注釈の挿入時に設定されたものが反映されます。Redact 注釈はソフトウェアや仕様書によっては、「Redaction 注釈」、「墨消し」と表現されている場合もあります。

コマンド例

[コマンド例の実行内容]

test.pdf 内にある Redact 注釈を適用し、出力する。

[Windows]

```
AHPDFToolCmd80.exe -setMaskUsingRedaction -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -setMaskUsingRedaction -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf
```

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。

8-39 -linearize : Web 表示用に最適化

処理内容

出力 PDF を Web 表示用に最適化します。この最適化処理は「リニアライズ」と呼ばれます。

リニアライズされた PDF は、Web ブラウザで開く際にできるだけ早く表示することができるデータ形式になります。

- 「-linearize」は指定位置が特殊なコマンドです。「-o」で出力先ファイル／フォルダを指定した後ろで「-linearize」を指定することで有効化されます
- PDF を出力するコマンドと組み合わせた複数指定が可能です。
出力するファイルが PDF でないコマンドは組み合わせることができません。
- 一度リニアライズされて保存されたファイルを読み込んだ場合でも、増分を追加して保存などの処理をすると出力時にリニアライズが解除されてしまいます。その場合は出力時に再度リニアライズを指定する必要があります。

コマンド例

[コマンド例の実行内容]

入力 PDF 「test.pdf」をリニアライズし、「out.pdf」として出力する。

[Windows]

```
AHPDFToolCmd80.exe -d C:\test\test.pdf -o C:\sav\out.pdf -linearize
```

[Linux]

```
AHPDFToolCmd80 -d /home/antenna/test/test.pdf -o /home/antenna/sav/out.pdf -linearize
```

フォルダ指定：対応有り

「-d」のパラメータに入力フォルダを、「-o」のパラメータに出力フォルダを指定して一括処理することができます。

「-linearize」は特殊なコマンドであり、他のコマンド1つに限り組み合わせてフォルダ指定することが可能です。

[コマンド例：Windows]

「test」フォルダ内のPDFすべてに対し、タイトルを「PDF Tool API テスト」に変更しリニアライズして「out」フォルダに出力する。

```
AHPDFToolCmd80.exe -changeDocInfo -title "PDF Tool API テスト" -d C:\test\test -o C:  
\$sav\$out -linearize
```

注意：

- ・ フォルダ指定をされた場合は出力されるすべてのPDFファイルをリニアライズします。
- ・ 組み合わせるコマンドもフォルダ指定可能である必要があります。
- ・ 出力するファイルがPDFでないコマンドは組み合わせることができません。
- ・ 出力されるPDFの名前について、他コマンドと組み合わせた場合はそのコマンドの仕様に準じます。

パラメータ

パラメータはありません。

8-40 -lic：ライセンス情報の取得

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。

入力ファイル指定「-d」や出力ファイル指定「-o」は必要ありません。

コマンド例

[コマンド例の実行内容]

『PDF Tool API』のライセンス情報を標準出力に表示する。

[Windows]

```
AHPDFToolCmd80.exe -lic
```

[Linux]

```
AHPDFToolCmd80 -lic
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
-------	----

<license file path>	省略可 ライセンス情報を表示したいライセンスファイルのパスを指定する。 指定されない場合はコマンドラインアプリケーションがデフォルトで読み込むライセンスファイル情報を表示する。
---------------------	--

8-41 -v：バージョン情報・著作権情報・ライセンス情報の表示

処理内容

『PDF Tool API』のバージョン情報、著作権情報、ライセンス情報を標準出力に出力します。入力ファイル指定「-d」や出力ファイル指定「-o」は必要ありません。

ファイルのパスを指定することで、特定の位置にあるライセンスファイルのライセンス情報を読み取ることも可能です。

パスを指定しない場合に表示されるのは『PDF Tool API』の参照先のライセンス情報です。

コマンド例

[コマンド例の実行内容]

ahlic フォルダ内の「ptalic.dat」のライセンス情報が表示される。

[Windows]

```
AHPDFToolCmd80.exe -v C:\ahlic\ptalic.dat
```

[Linux]

```
AHPDFToolCmd80 -v /home/ahlic/ptalic.dat
```

フォルダ指定：対応無し

パラメータ

パラメータ	内容
<license file path>	省略可 ライセンス情報を表示したいライセンスファイルのパスを指定する。 指定されない場合はコマンドラインアプリケーションがデフォルトで読み込むライセンスファイル情報を表示する。

8-42 -h : コマンド名一覧表示

処理内容

使用可能なすべてのコマンド名を標準出力に出力します。入力ファイル指定「-d」や出力ファイル指定「-o」は必要ありません。

「-h」を省略し、コマンドラインアプリケーションの実行名 (AHPDFToolCmd80) 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。

コマンド例

[コマンド例の実行内容]

『PDF Tool API』のコマンドラインアプリケーションで使用可能なすべてのコマンド名を表示する。

[Windows]

```
AHPDFToolCmd80.exe -h
```

[Linux]

```
AHPDFToolCmd80 -h
```

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。

8-43 -h -all：全コマンドオプション説明の表示

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。入力ファイル指定「-d」や出力ファイル指定「-o」は必要ありません。

「-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。

コマンド例

[コマンド例の実行内容]

『PDF Tool API』のコマンドラインアプリケーションで使用可能なすべてのコマンド名及びそのオプション説明を表示する。

[Windows]

```
AHPDFToolCmd80.exe -h -all
```

[Linux]

```
AHPDFToolCmd80 -h -all
```

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。

8-44 -h <コマンド名>：個別コマンドオプション説明の表示

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。入力ファイル指定「-d」や出力ファイル指定「-o」は必要ありません。

「-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。

コマンド例

[コマンド例の実行内容]

「-extractText」のオプションの説明を標準出力に出力する。

[Windows]

```
AHPDFToolCmd80.exe -h -extractText
```

[Linux]

```
AHPDFToolCmd80 -h -extractText
```

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。

第9章 取得情報の出力内容

コマンドラインアプリケーションで取得される各情報の出力内容は、次の通りです。

9-1 -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	PDF 生成アプリケーション名
	Producer	PDF 変換ライブラリ名
	CreateDate	作成日
	ModifyDate	更新日
	Max Page	総ページ数
View Information		PDF の開き方設定内容
	Direction	読み上げオプションの方向 (ページの綴じ方) <ul style="list-style-type: none">Left to Right: 左から右 (左綴じ)Right to Left: 右から左 (右綴じ)
	PageLayout	ページレイアウト表示 <ul style="list-style-type: none">Single Page: 単一ページ

		<ul style="list-style-type: none"> ● 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	<p>ページ・モード (表示される UI)</p> <ul style="list-style-type: none"> ● 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	<p>ファイルを開いたときの動作</p> <p>詳しくは <u>『9-5-1 【Action 情報】』</u> を参照</p>
Custom Property		カスタムプロパティの内容
	Name	カスタムプロパティの名前
	Value	カスタムプロパティの値
Security Information		<p>セキュリティ設定内容</p> <p>Not Encrypted.: セキュリティ設定なし。以降のセキュリティ関連出力もなし</p>
	Security Handler	<p>セキュリティの扱い方</p> <p>Standard Security: 標準セキュリティ</p>

	Key Length	暗号化キーの長さ（数値）
	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
		<p>Print: 印刷許可権限</p> <ul style="list-style-type: none"> ● not allowed.: 許可しない ● Low resolution.: 低解像度 ● High resolution.: 高解像度
		<p>Modify: 変更許可権限</p> <ul style="list-style-type: none"> ● 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
		<p>Copy: 内容のコピー権限</p> <ul style="list-style-type: none"> ● 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

		<p>Accessibility : アクセシビリティのアクセス許可</p> <ul style="list-style-type: none"> ● 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
Standard Information		
	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)

9-2 -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階層目を「0」から数えます。
Style	しおり文字列の装飾の種類 <ul style="list-style-type: none">● 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	<ul style="list-style-type: none">● 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 詳しくは 『9-5-1 【Action 情報】』 を参照

9-3 -getAnnotInfo : 注釈情報取得

項目名		内容
Annot Type		<p>注釈の種類</p> <ul style="list-style-type: none"> ● Link: リンク注釈 ● Text: テキスト注釈 ● Stamp: スタンプ注釈 ● File Attachment: ファイル添付注釈 ● Popup: ポップアップ注釈
Link の場合	Highlighting mode	<p>リンク注釈がクリックされたときにハイライトされる表示モード</p> <ul style="list-style-type: none"> ● None: 変化なし ● Invert: リンク注釈の矩形全体が反転する ● Outline: リンク注釈矩形の枠だけが反転する ● Push: リンク注釈の矩形が押し出されるように表示される
	Action	<p>リンクがクリックされたときの動作</p> <p>詳しくは 『9-5-1 【Action 情報】』 を参照</p>
	※注釈基本情報	後記 『9-5-1 【注釈基本情報】』 を参照
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	後記 『9-5-1 【注釈基本情報】』 を参照
	※マークアップ注釈情報	後記 『9-3-2 【マークアップ注釈情報】』 を参照
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	後記 『9-5-1 【注釈基本情報】』 を参照

※マークアップ注釈 情報	後記『9-3-2 【マークアップ注釈情報】』を参照
-----------------	---------------------------

9-3-1 【注釈基本情報】

項目名	内容
Rect	注釈矩形の位置情報
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	<p>注釈の表示状態を表すビットフラグ。具体的には下記。</p> <p>Invisible :</p> <p>on の場合、注釈が標準タイプに属さずなおかつ注釈ハンドラがない場合に表示しない。off の場合は未知の注釈だった場合でも PDF 内で外観ストリームが指定されていた場合はその外観で注釈が表示される。</p> <p>Hidden :</p> <p>on の場合、注釈タイプあるいはハンドラの有無にかかわらず、この注釈の表示、印刷、およびユーザによるインタラクトをできなくなる。</p> <p>Print :</p> <p>印刷。on の場合、ページが印刷される際に注釈も印刷される。off になっている注釈は印刷されない。例えば印刷ページ上では意味のない種類の注釈などではこのフラグが off にされている。</p> <p>NoZoom :</p> <p>倍率固定。on の場合、ページ表示倍率を変更しても注釈の外観上の位置は維持される。</p> <p>NoRotate :</p> <p>回転固定。on の場合、ページを回転させても注釈の外観上の位置は維持される。</p> <p>NoView :</p>

	<p>非表示。on の場合、スクリーン上に表示されず、ユーザによるインタラクトも不可になる。例えば NoView が on で Print フラグも on の注釈があった場合、スクリーン上では非表示だが印刷すると現れる挙動となる。</p> <p>ReadOnly :</p> <p>読み取り専用。on の場合、この注釈はユーザによるインタラクトが不可になる。この場合、表示・印刷は可能だが、マウスのクリックに応答すること、及びマウスの移動に対応して位置やサイズの外観を変更することを含めた変更が不可になる。</p> <p>Locked :</p> <p>on の場合、注釈の削除及び位置やサイズといったプロパティが変更不可になる。なお、on の場合でも注釈のコンテンツ内容の変更は禁止されない。</p> <p>ToggleNoView :</p> <p>on の場合、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 NoView フラグの解釈を反転させる。例えば NoView フラグの on と組み合わせることで選択されたときだけ注釈が表示されるようにする。</p> <p>LockedContents :</p> <p>on の場合、注釈のコンテンツ内容が変更不可になる。</p>
Color	<p>注釈の色</p> <ul style="list-style-type: none"> ● None : なし ● RGB (r, g, b) : RGB 色。r が赤、g が緑、b が青の値を表す。それぞれ 1.0 が最大値、0.0 が最小値。 ● CMYK (c, m, y, k) : CMYK 色。c がシアン、m がマゼンタ、y が黄、k が黒の値を表す。それぞれ 1.0 が最大値、0.0 が最小値。
Border Style	<p>注釈の境界線の種類</p> <ul style="list-style-type: none"> ● Solid : 実線 ● Dashed : 破線 ● Beveled : 浮き上がって見える立体効果付きの線 ● Inset : 凹んで見えるような立体効果付きの線 ● Underline : 下線

	<ul style="list-style-type: none"> ● None: なし
Border Width	<p>注釈の境界線の太さ</p> <ul style="list-style-type: none"> ● Thin: 細 ● Middle: 中 ● Thick: 太 ● None: なし

9-3-2 【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	<p>不透明度</p> <p>値の範囲: 0.0~1.0。1.0 が不透明</p>
Popup	<p>exist: ポップアップ注釈を持つ</p> <p>not exist: ポップアップ注釈を持たない</p>
※ポップアップ注釈の基本情報	<p>ポップアップ注釈を持つ場合に出力される。</p> <p>前記『9-5-1 【注釈基本情報】』を参照</p>

9-4 -getPageSize：ページサイズ情報取得

項目名	内容
ページサイズ	表示される領域の「幅」と「高さ」(mm 単位)
MediaBox	left,bottom,right,top の順に出力 (mm 単位)
CropBox	left,bottom,right,top の順に出力 (mm 単位)
BleedBox	left,bottom,right,top の順に出力 (mm 単位)
TrimBox	left,bottom,right,top の順に出力 (mm 単位)
ArtBox	left,bottom,right,top の順に出力 (mm 単位)

9-5 Action 情報・Destination 情報

9-5-1 【Action 情報】

「Action」の値	内容
GoTo	文書内の指定ページを表示する。 <u>後記『9-5-2 【Destination 情報】』を参照</u>
GoToR	別の PDF の指定ページを表示する。 <u>後記『9-5-2 【Destination 情報】』を参照</u>
Launch	ファイル名を指定して開く。PDF 以外のファイルの場合は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 [*1] : · true : ファイルは新しいウィンドウに表示される。 · false : ファイルは同じウィンドウに表示される。 [*1] : 表示するファイルが PDF でない場合、new window 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

9-5-2 【Destination 情報】

項目名		内容
Dest page		<p>移動先のページ。</p> <p>ページ番号は 0 オリジン。そのため、1 ページ目を「0」から数える。</p>
Dest type		<p>PDF の宛先の指定 / 表示種別。以下の種類に分かれる</p> <p>各 DestType が持つ値は以下の通り。それぞれ左下を(0.0)とする直交座標で表現される。単位は mm でマイナスの値を取ることもある。</p> <ul style="list-style-type: none"> ● left : 移動先の左端を表す座標 ● bottom : 移動先の下端を表す座標 ● right : 移動先の右端を表す座標 ● top : 移動先の上端を表す座標 ● zoom : ズーム率
	Fit	<p>Fit 型</p> <p>ページの全体が水平方向と垂直方向の両方ともウィンドウ内にちょうど収まる倍率で表示する</p>
	FitB	<p>FitB 型</p> <p>ページの描画領域全体が水平方向と垂直方向の両方ともウィンドウ内にちょうど収まる倍率で表示する。</p>
	FitBH	<p>FitBH 型</p> <p>ページの垂直座標 top をウィンドウの上端に配置し、ページの境界ボックス(CropBox)の幅全体がウィンドウ内にちょうど収まる倍率で表示する。</p>
	FitBV	<p>FitBV 型</p> <p>ページの水平座標 left をウィンドウの左端に配置し、ページの境界ボックス(CropBox)の高さ全体がウィンドウ内にちょうど収まる倍率で表示する。</p>
	FitH	<p>FitH 型</p> <p>ページの垂直座標 top をウィンドウの上端に配置し、ページの幅全体がウィンドウ内にちょうど収まる倍率で表示する。</p>
	FitR	<p>FitR 型</p> <p>ページが座標 left, bottom, right, top で指定される矩形が水平方向と垂</p>

		直方向の両方ともウィンドウ内にちょうど収まる倍率で表示する。
		必要となる倍率が水平方向、垂直方向で異なる場合、小さいほうの倍率を使用し、他方はウィンドウ中央となるように配置する
	FitV	FitV 型 ページの水平座標 left をウィンドウの左端に配置し、ページの高さ全体がウィンドウ内にちょうど収まる倍率で表示する。
	XYZ	XYZ 型 ページの座標(left,top)をウィンドウ左上隅に配置し、ページの内容を倍率 zoom で拡大して表示する。
	none	宛先なし

第10章 コマンドラインアプリケーション操作の仕様

本章ではコマンドラインアプリケーション操作の細かい仕様を解説します。

解説する具体的な内容は以下の項目です。

- コマンドの複数指定
 - マクロ文字列
 - エラーやコマンドラインアプリケーションの戻り値
-

10-1 コマンドの複数指定

コマンドラインアプリケーションにおいて、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、コマンドの複数同時指定には以下の注意点があります。

- コマンドとそれに付随するパラメータを指定した後、次のコマンドを指定してください。
- 処理対象をフォルダ指定した場合は、基本的に複数のコマンド指定ができません。[*1]
- 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。
- エラーが発生した場合、コマンドラインアプリケーションは保存を行わずに終了します。
- 透かしコマンド「-setPdfWatermark/-setImageWatermark/-setTextWatermark/-setColorWatermark」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。

[*1] :

例外に「-linearize」があります。詳細は『8-39 -linearize : Web表示用に最適化』を参照してください。

コマンドの実行順について

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. -extractEmbeddedFile
21. -importAnnnotsFormFDF
22. -linearize

具体的なコマンド例

[コマンド例の処理内容]

以下の例を実行した場合、記述順序は「-encrypt」、「-changeViewInfo」、「-setTextWatermark」の順になっていますが、

実際に処理される順序は「-changeViewInfo」、「-setTextWatermark」、「-encrypt」の順になります。

[Windows]

```
AHPDFToolCmd80.exe -encrypt -ownerPass test -filterType 4 -perms2 -print 1 -modify 0  
-changeViewInfo -pageLayout 2 -setTextWatermark -text サンプル -colorText 1 0 0 -dia  
gonal -font -name メイリオ -size 72 -d C:\test\test.pdf -o C:\sav\out.pdf
```

[Linux]

```
AHPDFToolCmd80 -encrypt -ownerPass test -filterType 4 -perms2 -print 1 -modify 0 -c  
hangeViewInfo -pageLayout 2 -setTextWatermark -text サンプル -colorText 1 0 0 -diago  
nal -font -name ipa-gothic -size 72 -d /home/antenna/test/test.pdf -o /home/antenna/s  
av/out.pdf
```

10-2 マクロ文字列

一部コマンドにおいて、コマンドラインアプリケーションの実行日付やファイル名、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。

マクロ文字列が使用可能な具体的なコマンドは以下です。

- 8-9 -setTextWatermark : テキスト透かしの挿入
- 8-17 -writeText : 文字列追加

マクロ文字列の記述法

マクロ文字列	内容
#DATE()	実行日時。 「#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」

具体的なマクロ文字列の指定例

[実行例の処理内容]

- test.pdf に「今日は何日？[#日付]」という文字列をテキスト透かしとして全ページに挿入し、out.pdf として出力する。
- 文字列の[#日付]の部分は実際にはマクロ文字列であり、実行した時点の[西暦/月/日]が挿入される。
- テキスト透かしは以下の設定・配置で挿入される。
 - フォント名：メイリオフォント(Windows) / IPA ゴシック(Linux)
 - フォントサイズ：52

➤ 文字色 : {R G B}={1.0, 0.0, 0.0}

[Windows]

```
AHPDFToolCmd80.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]

```
AHPDFToolCmd80 -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
```

10-3 エラー処理について

コマンドラインアプリケーションの戻り値でエラーの有無やコマンド処理の成否を確認することができます。

また、エラーが発生した場合は標準出力に表示されるエラーコードでエラーの内容を確認することができます。

本節はエラー処理について扱います。

10-3-1 コマンドラインアプリケーションの戻り値

コマンドラインアプリケーションの戻り値でコマンド処理の成否やエラーの有無を確認できます。

- コマンドラインアプリケーションの戻り値の確認方法や戻り値の種類は『PDF Tool API』のコマンドラインアプリケーションを実行するマシンのOSによって異なります。

[Windows]

戻り値は環境変数[ERRORLEVEL]で判定可能です。

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 『10-3-3 PdfTk のエラーコード一覧』参照
その他のエラー	-2

[Linux]

コマンドラインアプリケーションの戻り値は変数[\$?]で判定可能です。

状態	戻り値
成功	0
エラー	1

10-3-2 エラーの内容について

実行中にコマンドラインアプリケーションで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表示されます。

- コマンドラインアプリケーションのエラー詳細は標準出力にのみ出力されます。
- 『PDF Tool API』の内部ライブラリ「PdfTk」のエラーだった場合はメッセージと同時にエラーコードが表示されます。
- 各エラーコードの詳細は [『10-3-3 PdfTk のエラーコード一覧』](#) を参照してください。

10-3-3 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 the 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 signatured. 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	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	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	Need PKCS12. PKCS#12 ファイルが必要です。
220	Can not read attached file. 添付ファイルの読み込みができません。
221	Can not write attached file. 添付ファイルの書き出しができません。
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できません。
231	Can not write image file. 画像ファイルの書き出しができません。
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 2 値画像でありません。
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 2 値画像でありません。
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	Can not read ICC Profile. ICC プロファイルの読み込みができません。
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. ページの挿入ができません。
251	Can not delete page. ページの削除ができません。
252	Has no pages. ページが存在しません。
260 [*4]	Free docproperty error. DocProperty が PDF ドキュメントからフリーです。
261 [*4]	Free openmode error. OpenMode が PDF ドキュメントからフリーです。
262 [*4]	Free embeddedfiles error. EmbeddedFiles が PDF ドキュメントからフリーです。
263 [*4]	Free pages error. Pages が PDF ドキュメントからフリーです。
264 [*4]	Free page error. Page が PDF 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できません。
271 [*5]	Can not set to free outline. フリーアウトラインには設定できません。
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	Cannot read PKCS12. PKCS#12 ファイルの読み込みができません。
291	Cannot read X509. X.509 ファイル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11章 V4.0 / V5.0 / V6.0 / V7.0 との 相違点

本章では、過去バージョンの『PDF Tool API』との相違点について説明します。

11-1 コマンド対応バージョン一覧

コマンド名	V8.0	V7.0	V6.0	V5.0	V4.0
-getFileInfo	○	○	○	○	○
-changeDocInfo	○	○	○	○	○
-changeViewInfo	○	○	○	○	○
-mergeFiles	○	○	○	○	○
-getOutlineInfo	○	○	○	○	○
-getAnnotInfo	○	○	○	○	○
-exportAnnotsToFDF	○	○	○	-	-
-importAnnotsFromPDF	○	○	○	-	-
-setPdfWatermark	○	○	○	○	○
-setImageWatermark	○	○	○	○	○
-setTextWatermark	○	○	○	○	○
-setColorWatermark	○	○	○	○	-
-encrypt	○	○	○	○	○
-decrypt	○	○	○	○	○
-optimize	○	○	○	○	-
-addEmbeddedFile	○	○	○	-	-
-pageExtract	○	○	○	-	-
-pageDelete	○	○	○	-	-
-insertEmptyPage	○	○	○	-	-
-pageRotate	○	○	○	-	-

-addPageNumber	○	○	○	-	-
-writeText	○	○	○	-	-
-embedFonts	○	○	○	-	-
-extractImage	○	○	○	-	-
-extractText	○	○	○	-	-
-imageToPdf	○	○	○	-	-
-extractEmbeddedFile	○	○	○	-	-
-importAnnotsFromFDF	○	○	○	-	-
-fixedUpPDFA	○	○	-	-	-
-validatePDFA	○	○	-	-	-
-dividePdf	○	○	-	-	-
-getPageSize	○	-	-	-	-
-searchText	○	-	-	-	-
-zoom	○	-	-	-	-
(カスタムプロパティ追加)	○	-	-	-	-
(カスタムプロパティ取得)	○	-	-	-	-
-linearize	○	○	○	○	○
-lic	○	○	○	-	-
-v	○	-	-	-	-

11-2 各コマンドの変更点

11-2-1 テキスト抽出「-extractText」の変更点

V8.0 以降では、これまでのテキスト抽出機能に以下の機能を追加しました。

- テキストを抽出する領域を指定するパラメータの追加

11-2-2 文字列追加「-writeText」の変更点

V8.0 以降では、これまでの文字列追加に以下の機能を追加しました。

- 文字列の追加ページを複数ページ指定可能に

11-2-3 最適化処理「-optimize」の変更点

V8.0 以降では、これまでの最適化対象に加えて以下の要素が指定可能になりました。

- 添付ファイル削除
- タグの削除
- PieceInfo の削除
- メタデータの削除

11-2-4 画像抽出「-extractImage」の変更点

V8.0 以降では、画像抽出時に以下の指定が可能になりました。

- 指定解像度以上の画像を抽出
- 指定解像度以下の画像を抽出
- 変更を加えずに抽出

11-2-5 PDF ファイル情報取得「-getFileInfo」の変更点

V8.0 以降では、PDF ファイルの情報取得時に以下の情報も取得可能になりました。

- PDF バージョン
- 文書情報と総ページ数
- 開き方設定
- セキュリティ情報
- 署名の有無
- PDF/A かどうか
- PDF/X かどうか

11-2-6 新規機能：ページサイズ取得 「-getPageSize」

V8.0 にて、指定されたページのサイズを標準出力に出力する「-getPageSize」コマンドを追加しました。

11-2-7 新規機能：テキスト検索後にハイライト 「-searchTextAndHighlight」について

V8.0 MR2 にて、指定されたキーワードで検索しヒットした部分にハイライト注釈を挿入する「-searchTextAndHighlight」コマンドを追加しました。

11-2-8 新規機能：テキスト検索後に Redact 注釈挿入 「-searchTextAndRedact」について

V8.0 MR2 にて、指定されたキーワードで検索しヒットした部分に Redact 注釈を挿入する「-searchTextAndRedact」コマンドを追加しました。

11-2-9 新規機能：テキスト検索後に墨消し処理 「-searchText」について

V8.0 MR2 にて、指定されたキーワードで検索しヒットした部分に墨消し処理をする「-searchTextAndSetmask」コマンドを追加しました。

11-2-10 テキスト検索 「-searchText」について

V8.0 の初版で追加されたテキスト検索処理コマンド「-searchText」は削除され、ハイライト注釈の挿入は「-searchTextAndHighlight」、Redact 注釈の挿入は「-searchTextAndRedact」に分割されました。

11-2-1 新規機能：Redact 注釈の適用とマスク処理「-setMaskUsingRedaction」

V8.0 MR2 にて、PDF 内の Redact 注釈を適用し、該当箇所を墨消し処理する「-setMaskUsingRedaction」コマンドを追加しました。

11-2-2 新規機能：ページの拡大縮小「-zoom」

V8.0 にて、ページサイズを指定された用紙サイズに拡大・縮小する「-zoom」コマンドを追加しました。

11-2-3 新規機能：バージョン情報表示「-v」

V8.0 にて、バージョン情報と著作権情報を標準出力に表示する「-v」コマンドを追加しました。

11-2-4 PDF/A 変換と準拠確認の機能追加

V8.0 以降において、PDF/A 変換及び準拠変換（-fixedUpPDFA, -validatePDFA）で、以下の PDF/A バージョンに対応可能になりました。

- PDF/A-1a
- PDF/A-2a
- PDF/A-3a
- PDF/A-3b

11-2-5 カスタムプロパティの設定と取得

V8.0 にて、カスタムプロパティの設定をするコマンド「-setCustomProp」とカスタムプロパティの取得をするコマンド「-getFileInfo -customProp」を追加しました。

第12章 バージョンアップについて

『PDF Tool API』のバージョンは、「Vn.m」形式で表記されます。「Vn.m」のnをメジャーバージョンといい、mをマイナーバージョンといいます。

たとえば、『PDF Tool API V8.0』のメジャーバージョンは8、マイナーバージョンは0です。

「Vn.m」のnの増加をメジャーバージョンアップ、mの増加をマイナーバージョンアップといいます。

本章ではバージョンアップに関する手続きや仕様を解説します。

12-1 保守契約期限内のバージョンアップについて

保守契約期限内であれば、無償または購入価格との差額で『PDF Tool API』のバージョンアップが可能です。

- バージョンアップする場合は新しいバージョンのインストーラを入手・実行する必要があります。
ただし、1つのライセンスで運用可能なバージョン数や複数バージョンの同時インストールに関して注意点がございます。詳細は『12-2 バージョンアップとライセンスに関する制限事項』、『12-3 複数バージョンのインストールについて』を参照してください。
- バージョンアップする場合はインストーラの入手・実行の他、該当バージョンに合わせたライセンスファイルの入手と入れ替えも必要になります。
 - 保守契約期限内であれば、新しいライセンスファイルは保守契約窓口(hosyu@antenna.co.jp)に請求することで入手できます。
 - ライセンスファイルの入れ替えについては、新バージョンの参照先ライセンスファイルを入れ替えるか、新バージョンにおける任意の位置へのライセンスファイル配置方法に従ってください。
- バージョンアップに際して、バージョン間の価格改定があった場合に差額が必要となる場合があります。
具体的な差額についての詳細は弊社 web ページ『バージョンアップ』のページを参照してください。
(www.antenna.co.jp/ptl/versionup.html)

12-2 バージョンアップとライセンスに関する制限事項

バージョンアップとライセンスに関する制限事項は以下の通りです。

- 1つのライセンスで同時に2つのバージョンを運用することは許諾されていません。
ライセンスの新規購入をせずにバージョンアップして新バージョンのみを運用する場合は旧バージョンのアンインストールが必要です。
 - 旧バージョンと新バージョンの『PDF Tool API』を両方運用で使う場合は旧バージョンのライセンスをそのまま保有し、新バージョンのライセンスを新規購入して頂く必要があります。
 - アップグレードを検討される際に2つのバージョンを同時に運用されたい場合は保守契約窓口(hosyu@antenna.co.jp)にご相談ください。
-

12-3 複数バージョンのインストールについて

複数バージョンの『PDF Tool API』を同一マシンへインストールすることへの対応はWindows版とLinux版で異なります。

注意：

1つのライセンスで2つのバージョンを運用することは許諾されていません。そのため、同時稼働させるバージョンの数だけライセンスを購入して頂く必要があります。詳細は前述の『[12-2 バージョンアップとライセンスに関する制限事項](#)』を参照してください。

[Windows版の場合]

Windows版の場合、異なるバージョンの『PDF Tool API』を1台のPC上にインストールすることができます。

[Linux 版の場合]

Linux 版の場合は、旧バージョンをアンインストールしてから新しいバージョンをインストールすることをお勧めします。なお、Linux 版でも旧バージョンとは別フォルダに新規インストールすること自体は可能です。

第13章 改訂版について

『PDF Tool API』はメジャーバージョン、マイナーバージョンが同一のまま更新が加えられる「改訂版」がリリースされることがあります。

より具体的には、改訂版とはバージョン番号（「V n.m」の n と m）が同一でリリース日のみが新しくなっている製品のことです。メンテナンスリリース（MR）と呼ぶこともあります。改訂版の場合、バージョン番号の後に MR 番号が表記されます。

「メジャーバージョンアップ」「マイナーバージョンアップ」に関する詳細は『第12章 バージョンアップについて』を参照してください。

本章では改訂版のアップデート方法の詳細について説明します。

13-1 保守契約期限内の改訂版へのアップデートについて

- 改訂版へアップグレードする場合は改訂版のインストーラを入手・実行する必要があります。
 - バージョンアップとは異なり、保守契約期限内にリリースされた改訂版であればアップデート後も現在のライセンスファイルをご利用いただけます。詳細は『4-6 保守契約期間と改訂版アップデートにおけるコマンドラインアプリケーションの動作について』を参照してください。
 - 上書きインストールの可否が OS により異なるため、改訂版のインストール手順も OS ごとに異なります。詳細は後述の『13-2 改訂版のインストール方法について』を参照してください。
-

13-2 改訂版のインストール方法について

『PDF Tool API』の改訂版は Windows では上書きインストールが可能ですが、Linux 版では必要な操作が異なります。

[Windows 版]

改訂版（MR）にアップデートする場合は上書きインストールが可能です。

そのため、次の手順でアップデートできます。順序は 1 と 2 のどちらが先でも構いません。

1. 旧評価版をインストールしたフォルダに改訂版（最新版）を上書きインストールする。
2. 必要性な場合はインストール先のライセンスファイルを正規版のライセンスファイルに入れ替える [*1]。

[*1] :

ライセンスファイルの入れ替え作業に関する詳細は『[4-4 ライセンスファイルの入れ替えについて](#)』を参照してください。

[Linux 版]

Linux 版インストーラは、上書きインストールはできません。

つまり、改訂前の『PDF Tool API』を一旦アンインストールしてから改訂版を改めてインストールする必要があります。

Linux 版のインストールに関する詳細は『[3-2-1 インストール方法](#)』を参照してください。

Linux 版におけるライセンス用フォルダの扱いについて：

- アンインストール実行時、インストールフォルダ内のファイルはすべて削除されます。
それに伴い、インストールフォルダ内のライセンスファイル用フォルダも削除されます。
- インストールフォルダ内にライセンスファイルを配置していた場合は、改訂版のインストール後に正規版ライセンスファイルの再配置が必要になります。

第14章 よくいただくご質問

よくいただくご質問についての情報は、弊社 Web ページ [『PDF Tool API よくいただくご質問』](#) を参照してください。

第15章 商標／著作権情報

15-1 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 Adobe Acrobat Reader は、Adobe Inc.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 Apple Inc.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。

15-2 第三者ライブラリー著作権情報

jpeg

Copyright (C) 1994-1998, Thomas G. Lane.

This file is part of the Independent JPEG Group's software.

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

libtiff

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

Copyright (c) 1988-1996 Sam Leffler

Copyright (c) 1991-1996 Silicon Graphics, Inc.

zlib

This product includes software developed by Jean-loup Gailly and Mark Adler
(<https://www.zlib.net/>).

Copyright (C) 1995-2024 Jean-loup Gailly and Mark Adler

libpng

This code is released under the libpng license.

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) 2001-2016 Michael David Adams

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

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

All rights reserved.

LittleCMS

Little Color Management System

Copyright (c) 1998-2020 Marti Maria Saguer

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, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software
is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in

all copies or substantial portions of the Software.

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. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ICU

Copyright © 1991-Present Unicode, Inc.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc. in the United States and other countries. (<https://icu.unicode.org/>)

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING DATA FILES, AND/OR SOFTWARE, YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy of data files and any associated documentation (the "Data Files") or software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify,

merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE 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 THE DATA FILES OR 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 these Data Files or Software without prior written authorization of the copyright holder.

FreeType

This product includes softwares developed by:

Portions of this software are copyright 2024 The FreeType Project (www.freetype.org).

All rights reserved.

brotli

Copyright (c) 2009, 2010, 2013-2016 by the Brotli Authors.

(<https://brotli.org/>)

OpenSSL

OpenSSL is licensed under the Apache License 2.0, which means that you are free to get and use it for commercial and non-commercial purposes as long as you fulfill its conditions.

Copyright (c) 1998-2025 The OpenSSL Project

Copyright (c) 1995-1998 Eric A. Young, Tim J. Hudson

All rights reserved.

索引

PDF/A	PDF のアクション情報
PDF/A かどうか, 44	アクション情報の内容, 175
PDF/A に準拠しているか確認, 131	移動先情報の内容, 176
PDF ファイルを PDF/A に変換, 127	PDF のしおり
PDF からデータ抽出	しおり情報の内容, 169
PDF ファイルからテキスト抽出, 117	しおり情報を出力, 56
ページの画像データを抽出, 113	しおりの削除, 86
PDF 最適化	PDF の透かし
Web 表示用に最適化, 154	PDF ファイルを透かしとして挿入, 60
画像のダウンサンプリング, 86	色透かしを挿入, 76
最適化とは, 86	画像ファイルを透かしとして挿入, 65
タグを削除, 86	文字列を透かしとして挿入, 70
リニアライズ, 154	PDF のセキュリティ, 80
PDF に文字を追加	PDF バージョン, 80
ページ番号を挿入, 106	暗号化種別, 80
文字列を挿入, 96	削除, 84

出力 PDF ファイルに引き継がれない, 37	PDF の添付ファイル
出力 PDF ファイルに引き継ぎ, 29	PDF にファイルを添付, 90
セキュリティ情報の取得, 44	PDF の添付ファイルを抽出, 123
設定, 80	添付ファイルを削除, 86
入力ファイルに施されている時, 36	PDF の開き方
入力フォルダ指定のときの扱い, 38	出力 PDF に設定, 49
PDF のマスク	開き方の削除, 86
検索でヒットした箇所にマスク, 145	PDF のページサイズ
PDF の注釈	ページサイズ情報の内容, 174
FDF ファイルから注釈をインポート, 125	ページのサイズを出力, 137
PDF ファイルから注釈をインポート, 102	PDF のページ編集
検索でヒットした箇所に Redact 注釈, 142	PDF ファイルを指定条件で分割, 134
検索でヒットした箇所にハイライト注釈, 139	指定したページを回転, 92
注釈を FDF ファイルにエクスポート, 100	指定したページを削除, 94
注釈を削除, 86	指定したページを抽出, 109
入力 PDF の注釈情報を出力, 58	指定数の空白ページを挿入, 104
PDF の注釈	ページサイズを拡大・縮小, 148
Redact 注釈を適用しマスクする, 152	PDF バージョン
PDF の注釈	PDF ファイル結合時, 21
注釈情報の内容, 170	暗号化種別, 80

出力 PDF のバージョン, 21	インストール, 197
情報取得, 44	改訂版へアップグレード, 197
処理可能な PDF バージョン, 21	定義, 197
PDF ファイルに変換	ライセンスファイルの継続利用, 197
画像ファイル, 120	コマンド
PDF ファイルの結合, 53	一覧, 40
入力 PDF ファイルの任意のページを選ぶ, 53	コマンドとパラメータ, 33
PDF ファイルのプロパティ情報	指定コマンドのオプションの表示, 164
PDF ファイルから情報取得, 44	指定順と実行順, 179
カスタムプロパティ情報の取得, 44	使用可能なコマンド名の表示, 160
カスタムプロパティの設定, 150	全てのコマンドとオプションの表示, 162
出力 PDF ファイルに設定, 47	フォルダ指定対応有無, 40
インストール	複数コマンド指定時注意事項, 178
VC++ランタイムライブラリ, 6	複数コマンドを組み合わせ, 33
上書き, 5, 10, 198	マクロ文字列が使用可能なコマンド, 180
上書きで置き換わるファイル, 5	コマンドラインアプリケーション
作成されるライセンスファイル, 8	Linux 版実行方法, 34
新規, 5	Windows 版実行方法, 34
ライセンスファイルの配置先, 14	インストール方法, 4
改訂版	エラー出力, 34

エラー詳細, 183
コマンド処理の成否確認, 182
コマンドとパラメータ, 32
操作方法, 32
動作環境, 2
ファイル名, 33
メリット, 1
ライブラリ版の機能との関係, 1
コマンドライン版, 4
出力指定
パス, 37
パラメータ, 37
正規版
運用可能なバージョン, 195
定義, 14
動作期限, 14
入手方法, 14
入力指定
パラメータ, 36
ファイル指定, 36
フォルダ指定, 38
バージョン
改訂版, 197
バージョン情報を出力, 158
表記形式, 194
複数バージョンを同一マシンにインストール, 195
バージョンアップ, 194
ライセンスファイルの扱い, 17
評価版
動作期限, 13
動作仕様, 31
評価版と正規版, 13
ファイルサイズ
出力 PDF の制限, 53
読み書き可能な PDF ファイルのサイズ制限, 28
フォルダとディレクトリ, 4
フォント
埋め込みできないフォントの種類, 25
埋め込みフォントの統合, 25, 86

環境のフォントを PDF に埋め込み, 25	ライセンス情報を出力, 158
テキスト描画に使えるフォントの種類, 24	ライセンスファイル, 13
フォント情報を特定フォルダから取得, 23	ライセンスファイルの種類, 13
フォントの参照先, 22	ライセンスファイル
フォントを扱うコマンド, 22	入れ替えが必要なとき, 14
フォントを埋め込み, 111	入れ替え操作, 17
ライセンス	参照先, 15
新しいライセンスファイル請求先, 194	情報確認, 19
バージョンアップ時に新ライセンス, 15	ライブラリ版, 5
保守更新時に新ライセンス, 15	ランタイムライブラリ, 2
ライセンス情報の取得, 156	

改訂履歴

日付	内容
2025/07/31	初版発行
2025/09/30	MR2 に改訂 ・ -searchTextAndHighlight ・ -searchTextAndRedact ・ -searchTextAndSetMask ・ -setMaskUsingRedaction を追加 ・ -searchText を削除 誤字・脱字の修正

奥付

Antenna House PDF Tool API V8.0 コマンドライン説明書 2025.9.30

©Anttена House, Inc. 2025 All Rights Reserved.