

Antenna House PDF Tool API V7.0 利用ガイド

アンテナハウス株式会社

目 次

第1章 Antenna House PDF Tool API について	1
第2章 特長とおもな機能	2
2.1 特長	2
2.2 おもな機能	4
2.2.1 V7.0の新機能	4
2.2.2 その他のおもな機能	7
第3章 対応環境とプログラム言語	11
3.1 対応プラットフォーム	11
3.1.1 Windows	11
3.1.2 Linux	11
3.1.3 Amazon Linux2 X86	11
3.2 プログラム言語	12
3.2.1 Windows	12
3.2.2 Linux	12
3.2.3 Amazon Linux2 X86	12
3.3 コマンドライン	12
3.4 仮想環境への対応	13
第4章 ライセンスファイルについて	14
4.1 ライセンスファイルの入手	14
4.2 ライセンスファイルの配置	14
4.2.1 配置方法 1	14
4.2.2 配置方法 2 (Windows 版のみ)	14
4.3 ライセンスファイルの入れ替えについて	14
4.4 保守期間と PDF Tool API の動作について	14
第5章 導入方法	15
5.1 開発のための導入	15
5.2 プログラム実行のための導入	15
5.3 環境変数について	15
5.3.1 Windows	15
5.3.2 Linux / Amazon Linux2 X86	15
5.4 APIを使用する開発のチュートリアル	16
5.5 コマンドラインの使い方	16
第6章 インストール方法	17

6.1 Windows 版	17
6.1.1 インストール方法	17
6.1.2 アンインストール方法	17
6.1.3 Microsoft Visual C++ ランタイムライブラリについて	17
6.1.4 インストール先の構成	17
6.1.5 インストーラによりシステムに設定される内容	17
6.1.6 インストールされるライセンスファイルについて	18
6.1.7 インストールされる「PdfTkJava70.jar」について	18
6.2 Linux / Amazon Linux2 X86 版	19
6.2.1 インストール方法	19
6.2.2 アンインストール方法	19
6.2.3 インストール先の構成	19
6.2.4 インストールされるライセンスファイルについて	19
6.2.5 インストールされる「PdfTkJava70.jar」について	19
 第 7 章 モジュールファイルについて	20
7.1 Windows 版	20
7.1.1 モジュールファイル名	20
7.1.2 モジュールファイルの参照について	21
7.2 Linux /Amazon Linux2 X86 版	22
7.2.1 モジュールファイル名	22
7.2.2 モジュールファイルの参照について	23
 第 8 章 バージョンの互換について	24
8.1 PDF Tool API 使用の既存プログラムの V7.0 へのバージョンアップ方法	24
 第 9 章 制限事項	25
9.1 全般	25
9.2 PDF の表示あるいは描画	25
9.3 保存	25
9.4 閲覧制限	26
9.5 開き方	26
9.6 テキスト検索	26
9.7 文書情報	27
9.8 Separation Color	27
 第 10 章 仕様について	28
10.1 フォントについて	28
10.1.1 フォント情報の取得	28
10.1.2 フォント構築ファイルの設定	28
10.1.3 環境変数の作成	28
10.1.4 サポートしているフォントの種類	28
10.1.5 フォントの埋め込み	28

10.1.6 フォントの統合	29
10.1.7 制限事項	29
10.2 原点・座標・単位	30
10.2.1 原点	30
10.2.2 座標	30
10.2.3 単位	30
10.3 画像の扱い	30
10.3.1 対応形式	30
10.3.2 画像のマスク処理	30
10.3.2.1 ステンシルマスク	30
10.3.2.2 カラーキーマスク	31
10.3.2.3 明示マスク	31
10.3.2.4 ソフトマスク	31
10.3.3 ダウンサンプリングと JPEG 圧縮	31
10.3.4 画像抽出	31
10.4 テキスト透かし	32
10.5 Acrobat の透かし機能との互換性について	32
10.6 パスワードによるセキュリティ	33
10.7 証明書セキュリティ	33
10.7.1 設定	33
10.7.2 オープン	33
10.7.3 制限事項	33
10.8 データ削除処理	34
10.9 閲覧制限設定	34
10.10 注釈の扱い	34
10.10.1 新規作成	34
10.10.2 編集	35
10.10.3 境界線の幅	35
10.11 テキスト抽出	36
10.11.1 制限事項	36
10.12 PDF フォームの値の取得	36
10.12.1 エクスポート対象となる PDF フォームの種類	37
10.12.2 XFDF の仕様について	37
10.13 PDF/A 変換	38
10.14 PDF/A の準拠確認	38
10.15 画像の入れ替え	38
10.15.1 画像の追加	38
10.15.2 PDF の追加	38
10.15.3 エレメントの削除	38
10.16 PDF2.0 対応について	39
 第 11 章 仕様変更について	 40
11.1 V7.0 におけるセキュリティ設定の仕様変更	40

11.1.1 40bit RC4 設定処理は非対応	40
11.1.2 256bit AES 設定処理の仕様変更	40
11.2 V7.0 改訂 1 におけるテキスト抽出の仕様変更	40
11.3 文書情報の Metadata 作成について	40
 第 12 章 評価版の仕様について	 41
 第 13 章 サンプルコードについて	 42
 第 14 章 エラーコード一覧	 43
 第 15 章 商標・著作権情報	 46
15.1 商標情報	46
15.2 第三者ライブラリー著作権情報	46
 履歴	 49

第 1 章 Antenna House PDF Tool API について

Antenna House PDF Tool API（PDF Tool API）は、PDF ファイルの編集処理や、PDF ファイルからの情報取得処理を支援するソフトウェアです。PDF ファイルの結合、ページ抽出、セキュリティ設定、透かしの挿入、しおりの作成・編集、テキスト抽出、テキスト検索などさまざまな機能があります。

第2章 特長とおもな機能

2.1 特 長

豊富な処理機能

- PDF ファイル加工
ファイルの結合、ページの挿入や削除、透かしの挿入、セキュリティ設定などの PDF ファイルを加工する処理
- PDF ページ編集
ページコンテンツへのテキストや画像の描画や削除、画像の最適化（ダウンサンプリング）、フォント埋め込みなどのページを編集する処理
- 情報取得
文書情報やページ数の取得、しおりや注釈情報の取得、テキストや画像の抽出などのファイルの情報を取得する処理

API とコマンドライン

PDF Tool API の利用方法は、API またはコマンドラインです。

API で高度な開発

PDF Tool API では、PDF の各オブジェクトや機能をクラスライブラリ化しています。ライブラリに用意されている API を自在に組み合わせて独自の PDF 処理プログラムの作成が可能です。

API は、少ないコード行数でプログラムを書くことができるよう設計を工夫しています。高度な開発向きですが、PDF の仕様についてあまり深い知識がなくても使用できるようにしました。

対応するプログラム言語は、C++、.NET、Java です。

- 改訂6版にて、.NET6 対応の API を追加しました。Windows 上で開発したプログラムは、Windows、Linux のそれぞれの環境で動作します。API のインターフェースに変更はありません。これまでの .NET Framework 用 API で作成したコードを .NET6 でビルドすることができます。
- .NET6 用のモジュールファイル「PdfTkNet6_70.dll」は、.NET8 で使用可能です。

API のサンプルコード

C++、.NET、Java の言語別に各機能の基本的な処理のサンプルコードを用意しました。
.NET 用は C# で記述したサンプルコードになります。

コマンドラインで手軽に実装

コマンドラインは、コマンドとオプションを指定して実行するしくみです。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くても、PDFファイルの情報取得や加工・編集処理ができます。

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

処理内容は限定されますが、PDF に対する処理として要望の多いコマンドとオプションをそろえました、開発工数の削減にもつながります。

処理対象ファイル

PDF Tool API で取り扱いが可能な PDF ファイルは、「ISO 32000-1」および「ISO 32000-2」に準拠したファイルです。PDF バージョンでは、「1.7」までと、「2.0」が対象となります。

※ 注意点

PDF は ISO によって定められた仕様はありますが、その作り方の自由度が高く、またカスタマイズできる部分も多くあります。

したがって、PDF を生成するソフトウェアにより、ファイル内部の作り方がさまざまです。

このため、PDF ファイルに対して行うことによって問題現象が発生する場合があります。

PDF ファイルに対して行うこととして、

- PDF を表示すること
- PDF から情報を取得すること
- PDF を編集すること

があります。

PDF ビューワ上で問題なく表示できるファイルであっても、情報の取得や編集の処理で問題が発生する場合があります。

また、情報の取得処理が可能なファイルに対し、編集処理では問題が発生する場合があります。

2.2 おもな機能

2.2.1 V7.0の新機能

・ PDF2.0 対応

PDF Tool API V7.0 では、PDF2.0 対応を行っています。

PDF2.0 のファイルの読み込み

セキュリティ設定対応

注釈の外観辞書（AP キー）への対応

・ PDF/A への変換・準拠確認

PDF/A への変換処理と準拠確認処理を追加しました。

対応する PDF 規格は、「PDF/A-1b」と「PDF/A-2b」です。

※「PDF/A」は、文書の長期保存を目的とした国際基準規格です。

「PDF/A-1b（PDF1.4 相当）」は、ISO 19005-1 のレベル B に準拠した規格です。

「PDF/A-2b（PDF1.7 相当）」は、ISO 19005-2 のレベル B に準拠した規格です。

表 2.1 出力仕様の違い

PDF/A-1b	PDF/A-2b
フォントはすべて埋め込まなければならない	フォントはすべて埋め込まなければならない
XMP 準拠のメタデータを含んでいなければならない	XMP 準拠のメタデータを含んでいなければならない
パスワードや印刷制限、変更制限などセキュリティを設定してはならない	パスワードや印刷制限、変更制限などセキュリティを設定してはならない
LZW 圧縮してはならない	LZW 圧縮してはならない
外部コンテンツを参照してはならない	外部コンテンツを参照してはならない
JavaScript を含めてはならない	JavaScript を含めてはならない
透明な画像を含んではならない	透明な画像を含めてよい
ファイルを添付してはならない	ファイルを添付してもよい。ただし、PDF/A に準拠したファイルのみ可
JPEG2000 圧縮してはならない	JPEG2000 圧縮してもよい
オプションコンテンツ（レイヤー）を含めてはならない	オプションコンテンツ（レイヤー）を含めてもよい

・ PDF フォームデータの FDF/XFDF のインポート・エクスポート

PDF フォーム（「テキスト」「チェックボックス」「ラジオボタン」「リストボックス」「ドロップダウン（コンボボックス）」）のデータを FDF あるは XFDF 形式ファイルへエクスポートします。

また、FDF/XFDF 形式ファイルから PDF フォームデータをインポートします。

※XFDF ： 「ML Forms Data Format」の頭文字。ISO 19444-1 で規定された文書形式。

・ テキストボックス機能

テキスト描画機能に文字装飾などの処理を追加しました。

矩形（＝テキストボックス）内のテキストに対し、次のような指定が可能です。

太字・斜体

フォントの種類、サイズ

文字の輪郭色、塗りつぶし色（透明色の指定が可能）

文字の透明度

上付き文字、下付き文字（縦書きは非対応）

下線、取り消し線

テキストの折り返し

文字と文字の間隔

行と行の間隔

化学式、イオン式

テキストボックスに対しては次の指定が可能です。

横書き、縦書き

輪郭線の色、背景色

輪郭線がテキストを囲む大きさとなるよう変更するか否か

※「テキストボックス機能」は「テキスト注釈」ではありません。

PDF Tool API の独自の機能で、文字をコンテンツに描画します。

• 証明書によるセキュリティ設定

X.509 形式の証明書を使用してセキュリティ設定を行います。

ひとつの PDF に対し複数の証明書ファイルを指定することができます。証明書ごとにセキュリティの権限設定が可能です。

暗号化されているファイルを読み込む場合、PKCS#12 形式の証明書を使用します。

• エレメント情報の取得

PDF ページのコンテンツ上にあるエレメント（テキスト、画像、図形など）の情報を取得します。

次のような情報の取得が可能です。

エレメントを囲む矩形座標

直線などの開始位置と終了位置の座標

色

フォント情報

• JavaScript アクションの破棄

PDF ファイル上のすべての JavaScript アクションを破棄します。最適化処理のオプションとして追加しました。

• セパレーションカラー対応

図形やテキストの描画にセパレーションカラーを指定できるようにしました。

• 画像の入れ替え

ページコンテンツへの画像／PDF の追加と、指定エレメントの削除に対応することにより、画像の入れ替え処理を支援します。

Pattern の画像は、入れ替え対象とはなりません。

• Amazon Linux2(X86) 対応

改訂 5 版にて、「Amazon Linux2」の X86 アーキテクチャーに対応しました。

- **.NET6 / .NET8 対応**

改訂 6 版にて、.NET6 対応の API を追加しました。API のインターフェースに変更はありません。

,NET6 用モジュールファイル「PdfTkNet6_70.dll」は、.NET8 で使用可能です。

2.2.2 その他のおもな機能

• PDF ファイルの結合

複数の PDF ファイルを結合してひとつの PDF として保存します。注釈やフォーム、しおり、添付ファイルとともに結合するかどうかのオプションがあります。

• PDF ファイルの分割（ページ抽出）

ひとつの PDF ファイルから指定のページを抽出し別の PDF として保存します。

• 透かしの挿入と削除

PDF ファイルの各ページに、テキスト、画像ファイル、PDF ファイル、色を透かしとして挿入します。マルチ TIFF ファイルや PDF ファイルの場合は透かしにするページを指定することができます。

テキスト透かしでは、挿入角度設定がタイリング時にも適用されます。また、下線付きテキスト、複数行テキストの挿入が可能です。 *V7.0 NEW

画像・PDF 透かしでは、任意の角度に回転して挿入できます。 *V7.0 NEW

PDF Tool API V7.0 では、Adobe Acrobat®の「透かし」機能で編集可能な透かしを作成するオプションを追加しました。

*V7.0 NEW

PDF Tool API により挿入した透かしは、PDF Tool API により削除可能です。

• PDF セキュリティの取得、設定、削除

PDF ファイルに付けられている、パスワードによるセキュリティの情報を取得します。

新規のセキュリティ設定、すでに設定されているセキュリティ情報の変更、セキュリティを削除してセキュリティの付いていない PDF としての保存ができます。40bit RC4 ※、128bit RC4、128bit AES、256bit AES に対応しています。

※ V7.0 では、40bit RC4 レベルのセキュリティを新規に設定する処理には対応していません。

• 文書情報の取得と設定

PDF ファイルに設定されている文書情報を取得、PDF ファイルへの文書情報の設定ができます。任意の項目名を持つカスタムプロパティの設定ができます。

• ページの削除

PDF ファイルの任意のページを削除します。

• ページの回転

PDF ファイルの任意のページを回転します。

• ページの拡大・縮小

PDF ファイルの任意のページの内容を拡大、縮小します。

• データ削除処理（墨消し）

指定された矩形内にあるテキスト、画像、パスを削除します。「削除」とは、非表示にするということではなく、PDF ファイルからデータそのものを取り除きます。矩形内のどの要素（テキスト、画像、パス）を削除するかを指定するオプションもあります。

テキストは、指定矩形内の文字が削除されます。指定矩形の線上の文字も削除対象です。画像については、指定矩形内の画像データを部分削除します。線や四角形などのパスデータについては、パスデータ全体が矩形内に指定された場合に削除

します。パスデータが指定矩形からはみ出している場合は削除されません。

・ 画像抽出

PDF ページコンテンツ上の画像をファイルに書き出します。出力画像形式は、bitmap、jpeg、png。画像の幅、高さ、PPI (Pixel Per Inch) が取得可能です。

オプションとして、変更を加えず抽出する機能、画像のマスキの種類を取得する機能があります。 *V7.0 NEW

・ テキスト抽出

PDF ページコンテンツ上のテキストを取得します。矩形を指定した場合は、矩形内のテキストを取得します。

抽出時に、テキストを座標でソートするオプションがあります。 *V7.0 NEW

・ テキスト検索

PDF ページコンテンツ上のテキストを対象に検索を行います。複数キーワード検索に対応しています。検索結果場所の矩形取得、検索結果にハイライト注釈を付与、検索結果場所のデータ削除、以上を行う API を用意しました。

・ フォントの埋め込み

フォント情報が埋め込まれていないフォントに対し、フォント情報を埋め込み状態に変更します。ただし、埋め込み対象のフォント情報は、動作環境に存在していなければなりません。

・ しおりの取得、作成、削除

しおり情報の取得、既存のしおりの編集や新規しおりの追加、しおりの削除が可能です。

・ PDF の開き方情報の取得と設定

PDF ファイルの開き方情報の取得と設定が可能です。

・ 添付ファイル情報の取得、設定、削除、添付ファイルの書き出し

PDF ファイルに対して添付されている添付ファイル情報を取得します。

添付ファイルの追加、削除、添付ファイルの書き出しが可能です。

・ 注釈の新規作成

テキスト、スタンプ、ファイル添付、リンク、ハイライトの各注釈は、PDF Tool API V7.0 で新規に作成することができます。

・ 注釈情報の取得、編集、削除

PDF ファイル上にある注釈の情報取得、編集、削除ができます。

・ カスタムスタンプ注釈作成

任意の画像ファイルや PDF ファイルからスタンプ注釈を作成します。

・ 閲覧制限設定

指定の期間、あるいは指定の場所で表示された場合にのみ閲覧可能にする制限設定をします。ページごとに閲覧制限が設定可能です。場所（フォルダパス）を指定する場合、フォルダパスの末尾や途中にワイルドカードが使用できます。

Acrobat JavaScript を利用して制限制御を行います。

・ コンテンツへの描画：テキスト

PDF のページ上にテキストを挿入します。横書き、縦書きができます。

・ コンテンツへの描画：パス

PDF のページ上に直線、矩形、角丸矩形、円を描画します。

・ コンテンツへの描画：フォーム

ページをフォームとして描画します。ページ割付処理に応用できます。

・ コンテンツへの描画：画像

画像ファイルを PDF ページ上の指定矩形内に描画します。矩形に合わせて画像を拡大・縮小して描画することができます。

任意の角度に回転して描画することも可能です。 *V7.0 NEW

・ 画像ファイルからの PDF ページ生成

画像ファイルから PDF のページを生成します。

・ リニアライズ保存 (Web 表示用に最適化する保存処理)

リニアライズ保存をするかどうかの指定ができます。

・ 画像の最適化：ダウンサンプリングと JPEG 圧縮

PDF ファイル内の画像データに対し、ダウンサンプリングと JPEG 圧縮を行います。ダウンサンプリングの種類は、「バイリニア」「バイキュービック」「ニアレストネイバー」の3種類です。指定された解像度以上の解像度を持つ画像データに対し指定の解像度となるようダウンサンプリングが行われます。ダウンサンプリング率の下限値を設定することで、必要以上に画像を小さくして劣化が発生しないようにすることができます。

JPEG 圧縮はダウンサンプリング処理と同時に行われます。また、JPEG 圧縮処理のみ行うことも可能です。

・ PDF の最適化

次の処理を行うことで、PDF ファイルを最適化します。

オープンアクションの削除

しおりの削除

注釈・フォームの削除

アートの削除

サムネイルの削除

フォントの統合

・ 注釈 FDF の読み書き

マークアップ注釈（テキスト注釈、スタンプ注釈、図形注釈など）の FDF ファイルへの書き出しと、PDF ファイルに対して FDF ファイルを取り込むことができます。

・ PDF の情報取得

PDF ファイルが持つ様々な情報を取得できます。

文書情報

総ページ数

ページサイズ

開き方

添付ファイル

セキュリティ情報

Web 表示用に最適化されているか否か

添付ファイルを持っているか否か

表示矩形の取得

注釈があるか否か

サムネイルがあるか否か

注釈の種類（全種類を判別可能）

マークアップ注釈か否か

署名付きか否か

PDF/A、PDF/X であるか否か

ページモード、ページレイアウトの有無判定

第3章 対応環境とプログラム言語

3.1 対応プラットフォーム

3.1.1 Windows

OSの種類	Microsoft Windows 11 日本語版 Microsoft Windows Server 2025 / 2022 日本語版
必要なランタイムライブラリ	Microsoft Visual C++ 2019 ランタイムライブラリ

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

3.1.2 Linux

OSの種類	Linux X86 64bit
必要なランタイムライブラリ	glibc-2.28 libstdc++.so.6.0.25 もしくはこれらと互換のあるライブラリ ※GCC8.3.1 でビルドされています。

3.1.3 Amazon Linux2 X86

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

3.2 プログラム言語

3.2.1 Windows

言語	備考
C++	Microsoft Visual C++2019 でビルドされています。
.NET	・ .NET6 ・ .NET Framework 4.5 以降対応 ※.NET インターフェースは C++ インターフェースのラッパーとして作成されています。 ※.NET6 用は、Visual Studio 2022 でビルドされています。 ※.NET6 用 API は.NET8 で動作確認済みです。
Java	・ Java 11 ・ Java 8 ※Java インターフェースは C++ インターフェースのラッパーとして作成されています。 ※Java 11 用 API は、Java 17、Java 21 で動作確認済みです。 ※OpenJDK ディストリビューションの提供元に指定はありません。弊社では Eclipse Temurin JDK を使用しています。

3.2.2 Linux

言語	備考
C++	GCC8.3.1 でビルドされています。
.NET	・ .NET6 ※.NET インターフェースは C++ インターフェースのラッパーとして作成されています。 ※.NET6 用 API は.NET8 で動作確認済みです。
Java	・ Java 11 ・ Java 8 ※Java インターフェースは C++ インターフェースのラッパーとして作成されています。 ※Java 11 用 API は、Java 17、Java 21 で動作確認済みです。 ※OpenJDK ディストリビューションの提供元に指定はありません。弊社では Eclipse Temurin JDK を使用しています。

3.2.3 Amazon Linux2 X86

言語	備考
C++	GCC 7.3.1-15.amzn2 でビルドされています。
.NET	・ .NET6 ※.NET インターフェースは C++ インターフェースのラッパーとして作成されています。 ※.NET6 用 API は.NET8 で動作確認済みです。
Java	・ Java 11 ・ Java 8 ※Java インターフェースは C++ インターフェースのラッパーとして作成されています。 ※Java 11 用 API は、Java 17、Java 21 で動作確認済みです。 ※OpenJDK ディストリビューションの提供元に指定はありません。弊社では Eclipse Temurin JDK を使用しています。

3.3 コマンドライン

コマンドラインは、PDF Tool API の C++ インターフェースを使用して作成されたコンソール型アプリケーションです。

3.4 仮想環境への対応

PDF Tool API を仮想環境（VMware や Hyper-V など）で実行する場合、実環境上と同じ動作が保証されているのであれば機能制限はありません。

第4章 ライセンスファイルについて

4.1 ライセンスファイルの入手

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

電子納品の場合、ライセンスファイルは別途、弊社よりお送りします。

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

4.2 ライセンスファイルの配置

ライセンスファイルの配置方法には2種類あります。

どちらかの方法で配置してください。

4.2.1 配置方法 1

ライセンスファイル (ptalic.dat) を任意の場所に配置します。

配置したフォルダパスを下記の環境変数に設定します。

環境変数名 : PTL70_LIC_PATH

設定値 : ライセンスファイルを配置したフォルダパス

4.2.2 配置方法 2 (Windows 版のみ)

Windows 版の場合は、ライセンスファイルを PDF Tool API V7.0 のモジュールファイル「PdfTk70.dll」と同一の場所に配置して使用することも可能です。このとき、環境変数「PTL70_LIC_PATH」は必要ありません。

4.3 ライセンスファイルの入れ替えについて

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

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

ライセンスファイルを入れ替えた後、コンピュータの再起動は必要ありません。

4.4 保守期間と PDF Tool API の動作について

ライセンスファイルには、保守契約期間情報があります。

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

保守期間終了後も、それまでと同様にご利用いただけます。

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

最新の改訂版を正規利用するには、保守契約の更新をご検討ください。

第5章 導入方法

5.1 開発のための導入

プログラム開発において PDF Tool API を利用するには、付属のインストーラでセットアップを行います。
インストール方法については、「第6章 インストール方法 (p. 17)」をご参照ください。

5.2 プログラム実行のための導入

PDF Tool API を利用して開発したプログラムを開発環境とは異なる実行環境で実行するには、プログラムの実行ファイルのほかに、PDF Tool API の実行ファイルを所定の場所に配置します。

PDF Tool API の実行ファイルは、開発を行ったプログラム言語により異なります。必要な実行ファイルについては、「第7章 モジュールファイルについて (p. 20)」をご参照ください。

PDF Tool API の実行ファイルは、32bit 用と 64bit 用に分かれています。開発したプログラムが 32bit の場合は PDF Tool API 実行ファイルは 32bit 用を、プログラムが 64bit の場合は PDF Tool API 実行ファイルは 64bit 用をご使用ください。

Linux / Amazon Linux2 X86 版の実行ファイルは 64bit 用のみです。このため、32bit のプログラムからは使用できません。

5.3 環境変数について

PDF Tool API では、以下の環境変数を使用します。

5.3.1 Windows

環境変数名	設定値	設定が必要な場合
PTL70_LIC_PATH	ライセンスファイル「ptalic.dat」が存在するフォルダパス	API のバイナリファイル「Pdftk70.dll」とは異なる場所にライセンスファイルを配置する場合
PTL70_FONT_CONFIGFILE	フォント構築ファイル「font-config.xml」のフルパス	システムのフォントフォルダとは異なる場所にあるフォントを参照する場合
PTL70_ICCPROFILE_PATH	ICC プロファイルが存在するフォルダパス	この環境変数の設定が存在しない場合、API で指定された ICC プロファイルを Pdftk70.dll があるフォルダ内で検索します。 ICC プロファイルは、PDF/A-1b、PDF/A-2b 変換処理で使います。

5.3.2 Linux / Amazon Linux2 X86

環境変数名	設定値	設定が必要な場合
PTL70_LIC_PATH	ライセンスファイル「ptalic.dat」が存在するディレクトリ	設定は必須
PTL70_FONT_CONFIGFILE	フォント構築ファイル「font-config.xml」のフルパス	設定は必須
PTL70_CODEPAGE	文字コード名	PDF Tool API を特定の文字コードで使用する場合。 PTL70_CODEPAGE が存在しない場合、文字のエンコードは環境変数「LANG」の設定値にしたがいます。 どちらも存在しない場合、「UTF-8」で動作します。 [設定例] PTL70_CODEPAGE=EUC-JP

PTL70_ICCPROFILE_PATH	ICC プロファイルが存在するディレクトリ	ICC プロファイルを使用する PDF/A-1b、PDF/A-2b 変換処理を行うときは、この環境変数の設定は必須です。
-----------------------	-----------------------	--

5.4 API を使用する開発のチュートリアル

「Antenna House PDF Tool API V7.0 サンプルコードのビルド手順」では、サンプルコードのビルドを例に開発環境やプロジェクトの準備について説明しています。ご参考ください。

5.5 コマンドラインの使い方

コマンドラインの使い方については、「Antenna House PDF Tool API V7.0 コマンドライン説明書」をご参考ください。

第6章 インストール方法

6.1 Windows 版

6.1.1 インストール方法

- 1) setup-lib.exe をダブルクリックするなどして起動します。
- 2) ダイアログが表示されますので、指示にしたがってインストールを実行します。
- 3) デフォルトのインストール先フォルダパスは下記です。

{システムドライブ}\Program Files\Antenna House\PDFToolAPI_V7_lib

6.1.2 アンインストール方法

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

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

PDF Tool API V7.0 の動作には、Microsoft Visual C++ 2019 ランタイムライブラリが必要です。setup-lib.exe は、動作環境にランタイムライブラリが存在しないと判定したとき、Microsoft Visual C++ 2019 再頒布パッケージのインストーラを起動します。再頒布パッケージのインストーラが起動した場合は、表示されるダイアログにしたがって、インストールを行ってください。

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

6.1.4 インストール先の構成

インストールフォルダ

- bin32 : ライブラリモジュール (32bit 版) : コマンドラインファイル、カラープロファイルを含みます
- lib32 : lib ファイル (32bit 版)
- bin64 : ライブラリモジュール (64bit 版) : コマンドラインファイル、カラープロファイルを含みます
- lib64 : lib ファイル (32bit 版)
- Include : ヘッダファイル
- fontconfig : フォント構築ファイル
- License : ライセンスファイル「ptalic.dat」
- AHPDFToolAPI_EULA.txt: 使用許諾契約書

6.1.5 インストーラによりシステムに設定される内容

以下の環境変数の作成と値の設定が行われます。

環境変数名 : PTL70_LIC_PATH

設定値 : {インストールフォルダ}\License

6.1.6 インストールされるライセンスファイルについて

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

6.1.7 インストールされる「PdfTkJava70.jar」について

「bin32」／「bin64」にインストールされる「PdfTkJava70.jar」は、Java11 でコンパイルされたファイルです。「Java8」をご使用になる場合は、「{製品データ}\Lib-Windows\Java8」にあるファイルに入れ替えてください。

6.2 Linux / Amazon Linux2 X86 版

6.2.1 インストール方法

- 1) スーパーユーザーでログインします。
- 2) rpm コマンドを実行します。
rpm -i setup-lib.rpm [--prefix インストールパス]
- 3) インストール先のデフォルトパス : /usr/AHPDFToolV7-lib

※ Amazon Linux2 X86 にインストールする場合は、「{製品データ}\Setup-AmazonLinux2(X86)」の「setup-lib_amzn2.rpm.rpm」をご利用ください。

6.2.2 アンインストール方法

- 1) スーパーユーザーでログインします。
- 2) rpm コマンドを実行します。
rpm -e AHPDFToolV7Lib
rpm -e AHPDFToolV7Lib_amzn2 ※Amazon Linux2 X86 版の場合

6.2.3 インストール先の構成

インストールフォルダ

- |-- ToolCmd : コマンドライン実行ファイル
- |-- lib : PDF Tool API モジュールファイル
- |-- bin : Java API 用ファイル、.NET6 API 用ファイル
- |-- Inclde : ヘッダファイル
- |-- fontconfig: フォント構築ファイル
- |-- icc : カラープロファイル
- |-- License : ライセンスファイル「ptalic.dat」
- |-- run.sh : コマンドライン実行スクリプト
- |-- AHPDFToolAPI_EULA.txt: 使用許諾契約書

6.2.4 インストールされるライセンスファイルについて

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

6.2.5 インストールされる「PdfTkJava70.jar」について

「bin」にインストールされる「PdfTkJava70.jar」は、Java11 でコンパイルされたファイルです。

Java8 をご使用になる場合は、「{製品データ}/Lib-Linux/Java8」にあるファイルに入れ替えてください。Amazon Linux2 X86 用は「{製品データ}/Lib-AmazonLinux2(X86)/Java8」にあります。

第7章 モジュールファイルについて

7.1 Windows 版

7.1.1 モジュールファイル名

APIの各プログラム言語のメインモジュールファイル名とこれと依存関係にあるファイル名、コマンドラインの実行ファイル名とこれと依存関係にあるファイル名は以下の通りです。

PDF Tool APIを使用して作成したプログラムの実行環境には、プログラムの実行ファイルとともにこれらのモジュールファイルが必要です。

PDF Tool API のインターフェース	メインモジュール	依存関係にあるファイル
C++	PdfTk70.dll	PdfTkEx70.dll PtkAHCommon15.dll PtkAHDMC15.dll PtkAHFontService17.dll PtkAHGraphicService11.dll PtkAHPDFEditLib10.dll PtkAHPDFLib40.dll PtkPDFLinearizer60.dll PtkAHCertificate.dll PtkAHPDFFixUp11.dll icudt52.dll icuin52.dll icuio52.dll icule52.dll iculx52.dll icutu52.dll icuuc52.dll
.NET Framework	PdfTkNet70.dll	PdfTk70.dll PdfTkEx70.dll PtkAHCommon15.dll PtkAHDMC15.dll PtkAHFontService17.dll PtkAHGraphicService11.dll PtkAHPDFEditLib10.dll PtkAHPDFLib40.dll PtkPDFLinearizer60.dll PtkAHCertificate.dll PtkAHPDFFixUp11.dll icudt52.dll icuin52.dll icuio52.dll icule52.dll iculx52.dll icutu52.dll icuuc52.dll
Java	PdfTkJava70.jar PdfTk7JNI.dll	PdfTk70.dll PdfTkEx70.dll PtkAHCommon15.dll PtkAHDMC15.dll PtkAHFontService17.dll PtkAHGraphicService11.dll PtkAHPDFEditLib10.dll PtkAHPDFLib40.dll PtkPDFLinearizer60.dll PtkAHCertificate.dll PtkAHPDFFixUp11.dll

		icudt52.dll icuin52.dll icuio52.dll icule52.dll iculx52.dll icutu52.dll icuuc52.dll
コマンドライン	AHPDFToolCmd70.exe	PdfTk70.dll PdfTkEx70.dll PtkAHCommon15.dll PtkAHDMC15.dll PtkAHFontService17.dll PtkAHGraphicService11.dll PtkAHPDFEditLib10.dll PtkAHPDFLib40.dll PtkPDFLinearizer60.dll PtkAHCertificate.dll PtkAHPDFFixUp11.dll icudt52.dll icuin52.dll icuio52.dll icule52.dll iculx52.dll icutu52.dll icuuc52.dll
.NET6/.NET8	PdfTkNet6_70.dll PdfTkNet6.dll	PdfTk70.dll PdfTkEx70.dll PtkAHCommon15.dll PtkAHDMC15.dll PtkAHFontService17.dll PtkAHGraphicService11.dll PtkAHPDFEditLib10.dll PtkAHPDFLib40.dll PtkPDFLinearizer60.dll PtkAHCertificate.dll PtkAHPDFFixUp11.dll icudt52.dll icuin52.dll icuio52.dll icule52.dll iculx52.dll icutu52.dll icuuc52.dll

※.NET6 Runtime 入手先：.NET6 のダウンロード

実行環境には『.NET Runtime 6.0.xx』をダウンロードしてセットアップしてください。

※.NET8 Runtime 入手先：.NET8 のダウンロード

実行環境には『.NET Runtime 8.0.xx』をダウンロードしてセットアップしてください。

7.1.2 モジュールファイルの参照について

- Internet Information Service(IIS) で利用する場合や、実行ファイルを PDF Tool API のモジュールファイルと異なる場所に配置する場合には、環境変数「PATH」に PDF Tool API のモジュールファイルが存在するフォルダパスを設定してください。
- Java 開発においては、環境変数「CLASSPATH」に「PdfTkJava70.jar」のフルパス名、「PATH」に「PdfTkJava70.jar」と依存関係にあるモジュールファイルが存在するフォルダパスを設定してください。

7.2 Linux /Amazon Linux2 X86 版

7.2.1 モジュールファイル名

API の各プログラム言語のメインモジュールファイル名とこれと依存関係にあるファイル名、コマンドラインの実行ファイル名とこれと依存関係にあるファイル名は以下の通りです。

PDF Tool API を使用して作成したプログラムの実行環境には、プログラムの実行ファイルとともにこれらのモジュールファイルが必要です。

PDF Tool API のインターフェース	メインモジュール	依存関係にあるファイル
C++	libPdfTk.so.7.0	libPdfTkEx.so.7.0 libPtkAHCertificate.so.1.0 libPtkAHCommon.so.1.5 libPtkAHDMC.so.1.5 libPtkAHFontService.so.1.7 libPtkAHGraphicService.so.1.1 libPtkAHPDFEditLib.so.1.0 libPtkAHPDFFixUpLib.so.1.1 libPtkAHPDFLib.so.4.0 libPtkPDFLinearizer.so.6.0 libcudata.so.52.1 libcui18n.so.52.1 libcui.so.52.1 libicule.so.52.1 libiculx.so.52.1 libicutu.so.52.1 libicuuc.so.52.1
Java	PdfTkJava70.jar libPdfTkJNI.so.7.0	libPdfTk.so.7.0 libPdfTkEx.so.7.0 libPtkAHCertificate.so.1.0 libPtkAHCommon.so.1.5 libPtkAHDMC.so.1.5 libPtkAHFontService.so.1.7 libPtkAHGraphicService.so.1.1 libPtkAHPDFEditLib.so.1.0 libPtkAHPDFFixUpLib.so.1.1 libPtkAHPDFLib.so.4.0 libPtkPDFLinearizer.so.6.0 libcudata.so.52.1 libcui18n.so.52.1 libcui.so.52.1 libicule.so.52.1 libiculx.so.52.1 libicutu.so.52.1 libicuuc.so.52.1
コマンドライン	AHPDFToolCmd70	libPdfTk.so.7.0 libPdfTkEx.so.7.0 libPtkAHCertificate.so.1.0 libPtkAHCommon.so.1.5 libPtkAHDMC.so.1.5 libPtkAHFontService.so.1.7 libPtkAHGraphicService.so.1.1 libPtkAHPDFEditLib.so.1.0 libPtkAHPDFFixUpLib.so.1.1 libPtkAHPDFLib.so.4.0 libPtkPDFLinearizer.so.6.0 libcudata.so.52.1 libcui18n.so.52.1 libcui.so.52.1 libicule.so.52.1 libiculx.so.52.1

		libicutu.so.52.1 libicuuc.so.52.1
.NET6/.NET8	PdfTkNet6_70.dll PdfTkNet6.so.7.0	libPdfTk.so.7.0 libPdfTkEx.so.7.0 libPtkAHCertificate.so.1.0 libPtkAHCommon.so.1.5 libPtkAHDMC.so.1.5 libPtkAHFontService.so.1.7 libPtkAHGraphicService.so.1.1 libPtkAHPDFEditLib.so.1.0 libPtkAHPDFFixUpLib.so.1.1 libPtkAHPDFLib.so.4.0 libPtkPDFLinearizer.so.6.0 libcudata.so.52.1 libcui18n.so.52.1 licuio.so.52.1 libicule.so.52.1 libiculx.so.52.1 libicutu.so.52.1 libicuuc.so.52.1

※.NET6 Runtime 入手先：.NET6 のダウンロード

実行環境には『.NET Runtime 6.0.xx』をダウンロードしてセットアップしてください。

※.NET8 Runtime 入手先：.NET8 のダウンロード

実行環境には『.NET Runtime 8.0.xx』をダウンロードしてセットアップしてください。

7.2.2 モジュールファイルの参照について

- 環境変数「LD_LIBRARY_PATH」に「{インストールフォルダ}/lib」を設定してください。
- Java 版では、環境変数「CLASSPATH」に「{インストールフォルダ}/bin/PdfTkJava70.jar」を設定してください。

第8章 バージョンの互換について

8.1 PDF Tool API 使用の既存プログラムの V7.0 へのバージョンアップ方法

PDF Tool API V7.0 は、PDF Tool API V4.0 / V5.0 / V6.0 を継承しています。V4.0 / V5.0 / V6.0 を使用しているプログラムは、ヘッダや lib ファイル、あるいは参照するバイナリファイルを V7.0 のものに置き換えてリビルドすることにより、V7.0 用のプログラムとして実行可能となります。

ただし、一部の API では名前が変更になっているものがあります。その場合は、V7.0 で修正された名称に変更してください。

なお、V7.0 は、V3.1 以前の PDF Tool API とは互換はありません。

第9章 制 限 事 項

9.1 全 般

- PDF Tool API V7.0 は、V3.1 以前の PDF Tool API とは互換はありません。
- PDF Tool API V7.0 で取扱い可能な PDF ファイルは、ISO 32000-2 の仕様に準拠したファイルです。
- 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 時エラーとなります。
- 任意の PDF バージョン番号を設定することはできません。
- 複数のスレッドでの同時利用はできません。
- PDF Tool API のメインモジュールファイルとこれと依存関係にあるモジュールファイルは、必ず、同一の改訂版をご使用ください。メインモジュールと依存関係モジュールが異なる改訂版であるファイルや、他の製品でリリースされたファイルを依存関係モジュールとして使用しないでください。

9.2 PDF の表示あるいは描画

- PDF の編集処理によって発生する表示あるいは描画問題は、PDF ファイルに依存します。確認された現象への対策は行っておりますが、PDF ファイルにより問題が発生する可能性は排除できません。
 - 表示あるいは描画問題とは：文字列や画像の挿入など PDF のページを編集する処理を行ったときに、既存の線や図がずれる、テキストが表示されないあるいは反転してしまうなど、PDF の表示あるいは描画が正しく行われない問題
- 注釈フラグの「NoRotate」が ON の状態で注釈が描画されているとき、PDF ビューアにより、表示の位置や形状が指定したものや異なったり、注釈自体の位置が異なったりする場合があります。PDF Tool API では NoRotate フラグのデフォルトは OFF です。
 - [備考]NoRotate フラグが ON の場合：ページの回転角度と関係なく、注釈の左上をページ上の固定位置として表示します。

9.3 保 存

- PDF Tool API V7.0 で出力される PDF ファイルの PDF バージョンは、原則として入力ファイルの情報が引き継がれます。
- PDF Tool API で保存される PDF のファイルサイズの上限は 2GB です。2GB を超える場合は、保存時に「103」（「大きすぎる PDF の書き出しができません。」）となります。
- 出力ファイルサイズの上限判定は、PDF Tool API で PDF の加工や編集を行い保存する結果が 2GB を超えるかどうかです。たとえば PDF ファイルを結合する場合、PDF Tool API では物理的に PDF ファイルを結合する処理ではないため、結合前の PDF ファイルサイズの合計が 2GB 未満であっても保存できない場合があります。
- PDF/A、PDF/X に対して編集を行った場合、PDF/A、PDF/X の情報は破棄されます。
- PDF Tool API V7.0 では保存処理において PDF Tool API V3.1 と異なる点があり、PDF ファイルの構造によっては V3.1 より V7.0 のほうが保存処理に時間がかかる場合があります。
 - PDF には、「オブジェクトストリーム」という構造によりオブジェクトを圧縮してファイルサイズを小さく抑えるしくみがあります。V3.1 では、オブジェクトストリームという圧縮状態を解除して保存を行うため、保存後の PDF ファイルサイズが大きくなるケースがありました。V7.0 では、オブジェクトストリームという圧縮状態のままオブジェクトを更新する保存処理となっています。保存を行っても圧縮された状態は保持されるためファイルサイズが

極端に大きくなることはありません。ですが、オブジェクトストリームの中をすべて探査して更新処理を行うため処理時間がかかる場合があります。

9.4 閲 覧 制 限

- 閲覧制限は Acrobat Java Script の機能を利用しています。Acrobat Java Script をサポートしていなかったり Acrobat Java Script が無効になっている PDF ビューア、PDF 表示機能を持つブラウザなどでは設定された制限動作は行われません。常に閲覧不可状態、もしくはページが真っ白な状態で表示されます。あるいは、ページや表示倍率の切り替え時、「レイヤー」操作が可能になっている PDF ビューアでレイヤーの有効・無効の切り替え時に、中身が一時的に表示される場合があります。
 - 次のブラウザでは閲覧制限は正しく動作しません（弊社確認）。これら以外のブラウザでの動作は未確認です。： Microsoft Edge / Google Chrome / Firefox
- 閲覧制限設定は PDF を暗号化するものではありません。PDF を扱うツールによっては、ページの内容が見えたり抽出することができる場合があります。
- 閲覧制限を設定する場合、同時にセキュリティの権限とオーナーパスワードを設定することをおすすめします。
- 閲覧制限設定を行った PDF から閲覧制限設定を取り除くことはできません。
- 閲覧制限設定された PDF を Acrobat/Adobe Reader で表示させたとき、閲覧できる状態であっても「サムネイル」ナビゲーションパネルでは内容が表示されません。
- 閲覧制限設定された PDF のしおり、添付ファイル、文書情報は、閲覧不可や Java Script が無効な場合であっても非表示にはなりません。
- 閲覧制限設定された PDF がポートフォリオの場合、ポートフォリオは添付ファイル機能であるため、閲覧不可や Java Script が無効な場合であっても非表示にはなりません。
- 閲覧制限設定された PDF を閲覧不可や Java Script が無効なときに Acrobat / Adobe Reader で表示している場合、「注釈」ナビゲーションパネル（Adobe Reader DC では「表示」 - 「ツール」 - 「コメント」メニューで表示される注釈のリスト）は非表示にはなりません。
- 閲覧制限設定を行った PDF ファイルに対して追加した注釈やフォームは、閲覧不可や Java Script が無効な場合に非表示にはなりません。
- 閲覧制限設定された PDF において閲覧不可や Java Script が無効なときにページ上の注釈あるいはフォームを選択できた場合には、注釈、フォームは表示状態になります。
- 閲覧制限設定された PDF が閲覧不可や Java Script が無効なとき、Acrobat / Adobe Reader 以外の PDF ビューアでは注釈とフォームが表示される場合があります。
- 閲覧制限設定処理において、閲覧制限時に表示する透かしは 1 つの「PtlParamWaterMark」のみ設定可能です。
- すでに閲覧制限設定された PDF に対し再度閲覧制限設定を行った場合、設定された閲覧制限の制限動作は保証されません。
- 閲覧制限設定において appendValidURL() でパスを指定する場合、ファイル名やフォルダ名に使用できない文字（¥ / : * ? " < > |）やサロゲートペア文字は指定できません。指定された場合、動作結果は不定となります。

9.5 開 き 方

- 開き方設定処理においてメニューバー非表示と Windows コントロール非表示を同時に設定した PDF ファイルを Acrobat/Acrobat Reader で開いた場合、メニューバーは表示されます。

9.6 テキスト検索

- テキスト検索処理は、ページコンテンツ全体が検索対象となります。表示領域がページコンテンツの一部である場合、

表示されていない部分のテキストも検索対象となります。

- テキスト検索処理は、クリッピングされて表示されていない部分のテキストも検索対象となります。
- PDF の表示上はテキストであっても、実体のオブジェクトの種類がテキストではなく画像やパスである場合があります。この場合、テキスト検索の対象とはなりません。
- テキスト検索処理において、キーワードがページをまたいでいる場合、その箇所は検索されません。
- 検索キーワードのスペースは、取り除いた状態で検索処理を行います。

9.7 文 書 情 報

- 文書情報にカスタムプロパティを設定する場合、次の単語はプロパティの名前として指定することはできません。エラーとなります。: Title , Author , Subject , Keywords , Creator , Producer , CreationDate , ModDate , Trapped
- 文書情報のキーワードとして設定した文字列に含まれる文字や記号の種類により、Acrobat で開いたとき、文書情報においてキーワードが二重に表示される場合があります。

9.8 Separation Color

- 作成される Separation Color は、ファンクションは「タイプ 2(指数補完) 関数」、代替カラースペースは「CMYK」のみです。

第 10 章 仕様について

10.1 フォントについて

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

10.1.1 フォント情報の取得

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

10.1.2 フォント構築ファイルの設定

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

フォント構築ファイルは、以下のファイルです。

インストールフォルダ\fontconfig\font-config.xml

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

font-config.xml 記述例（Windows 版）

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

10.1.3 環境変数の作成

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

環境変数名： PTL70_FONT_CONFIGFILE

設定する値： font-config.xml のフルパス

10.1.4 サポートしているフォントの種類

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

10.1.5 フォントの埋め込み

PDF Tool API での処理時にフォントの埋め込みを指示した場合にフォント情報が PDF ファイルに埋め込まれます。

ただし、次のフォントについては、埋め込み指示があってもフォント情報は埋め込まれません。

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

10.1.6 フォントの統合

- フォント統合は、PDF に埋め込まれているフォントに対して行われます。
- 次のフォントが統合対象です。
 - TrueType、Type1(CFF 形式のみ)、CIDFontType2、CIDFontType0
- 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 はの場合、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。
- 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

[改訂での仕様変更]

- 同じフォントタイプであってもフォント辞書の値（Ascent,CapHeight など）が異なる場合は統合の対象となりません。
→V7.0 改訂3以降、この条件はありません。

10.1.7 制 限 事 項

- ビットマップフォント、Type3 フォントには対応していません。
- 縦書きではイタリックの指定があってもフォントがグリフを持っていない場合はイタリックにはなりません。
- フォント埋め込み処理において、埋め込み対象となるフォントが動作環境に存在していなければなりません。
- フォント埋め込み処理において、フォントが埋め込み禁止である場合は埋め込み処理は行われません。
- フォント統合処理では重複するフォント情報をひとつに統合しますが、統合により必ずファイルサイズが小さくなるわけではありません。

10.2 原点・座標・単位

10.2.1 原点

- ・ デフォルトの座標の原点は、ページ表示上の「左下」です。
- ・ PtlOption クラスを利用して、原点を「左上」に切り替えて処理を行うことができます。

10.2.2 座標

- ・ 座標の扱い方について、デフォルトは表示上の座標です。PDF Tool API を利用する側においては、PDF ページの加工や編集処理のさいに元のページが回転されているかどうかを考慮する必要はありません。原点を「左下」としている場合、どのページにおいても表示上の「左下」が原点となります。
- ・ PtlOption クラスを利用して、PDF のユーザースペース座標に切り替えて扱うことができます。ユーザースペース座標では、ページが回転していると座標軸も回転します。たとえば、回転しているページに文字を挿入した場合、文字も回転します。

10.2.3 単位

- ・ デフォルトは、「mm」（ミリ）です。
- ・ PtlOption クラスを利用して、「PT」（ポイント）単位に切り替えて処理を行うことができます。

10.3 画像の扱い

10.3.1 対応形式

PDF Tool API が対応している画像形式は以下の通りです。

bitmap / jpeg / png / tiff / gif

10.3.2 画像のマスク処理

PDF Tool API でサポートしているマスクの種類は、

- ・ ステンシルマスク
- ・ カラーキーマスク
- ・ 明示マスク
- ・ ソフトマスク

の4種類です。

マスクの種類により、取り扱う画像に特長があります。

10.3.2.1 ステンシルマスク

入力画像には白黒画像を指定してください。白黒画像以外の場合はエラー（234:STENCIL_MASK_IS_NOT_SINGLE）となります。

参考サンプルコード：ImageToPdfStencilMask

10.3.2.2 カラーキーマスク

入力画像には、カラースペースが「RGB」であるビットマップを指定してください。ビットマップ以外の画像や、indexed color を持つビットマップの場合はエラー（235:UNSUPPORTED_IMAGE_FOR_COLORKEY_MASK）となります。

参考サンプルコード：ImageToPdfColorkeyMask

10.3.2.3 明示マスク

マスクする画像には、白黒画像を指定してください。白黒画像以外の場合はエラー（237:EXPLICIT_MASK_IS_NOT_SINGLE）となります。

入力画像にサポートされていない画像が指定された場合は、エラー（232:UNSUPPORTED_IMAGE）となります。

参考サンプルコード：ImageToPdfExplicitMask

10.3.2.4 ソフトマスク

マスクする画像には、グレースケール画像の指定を推奨します。指定された画像でマスクすることができなかった場合はエラー（232:UNSUPPORTED_IMAGE）となります。

入力画像にサポートされていない画像が指定された場合は、エラー（232:UNSUPPORTED_IMAGE）となります。

参考サンプルコード：ImageToPdfSMask

10.3.3 ダウンサンプリングと JPEG 圧縮

- ダウンサンプリングの対象とする画像の条件として、Filter の種類を指定することができます。対象 Filter を 1 種類だけ指定した場合は、指定したその Filter のみを持つ画像だけがダウンサンプリングの対象となります。
- 画像データの中には、Filter を配列で複数持つものがあります。その Filter 配列の中にダウンサンプリングの対象として指定した Filter が含まれていても、その画像データはダウンサンプリングの対象とはなりません。
- ダウンサンプリングを行ったとき、カラースペースが元のデータとは異なる場合があります。

10.3.4 画 像 抽 出

- 画像抽出処理は、オブジェクト 1 個をひとつの画像ファイルとして出力します。表示上は 1 個の画像であってもオブジェクトが細かく分かれている場合があります、このようなケースでは出力される画像ファイル数が多くなります。
- 画像抽出処理は、ページコンテンツ全体を対象としています。表示領域がページコンテンツの一部分である場合、表示されていない部分の画像も抽出対象となります。
- 画像抽出処理は、クリッピングされて表示されていない部分の画像も抽出対象となります。

10.4 テキスト透かし

テキスト透かしの挿入方法として、

横書き

縦書き

任意の角度付き

タイリング

があります。

PDF Tool API では改行コードをサポートしますが、任意の角度付きとタイリングの挿入方法では、改行コードはサポートされません。改行コードを無視されます。

PDF Tool API では、任意の角度付きでタイリングが可能です。

10.5 Acrobat の透かし機能との互換性について

PDF Tool API V7.0 では、透かしの作成処理オプションにより、PDF Tool API が作成する透かしの Adobe Acrobat の透かし機能で扱うことができるようにしました。

ただし、PDF Tool API 独自のオプションが付いている場合、その透かしは Adobe Acrobat では扱うことができません。

PDF Tool API で作成した透かしの Adobe Acrobat で扱う場合は、PDF Tool API 独自のオプションは付けずに透かし作成処理を行ってください。

【PDF Tool API 独自のオプション】

- テキスト透かし・画像透かし・PDF 透かし・色透かし共通
 - タイリング指定がある
 - 挿入ページ指定で連続しないページ番号の指定がある
 - 余白設定がある
 - 挿入位置の矩形指定がある
- テキスト透かし
 - 文字の輪郭色の指定がある
 - 対角線配置の指定がある

Adobe Acrobat の透かし対応オプションと上記のオプションを同時に指定している場合、透かし作成処理の実行時にエラーとなります。

10.6 パスワードによるセキュリティ

- PDF ファイルに対しパスワードによるセキュリティ設定を行う場合、暗号化する文書コンポーネントを設定することができます (PtlEncrypt::setEncryptComponent(int component))。コンポーネントの種類と内容については以下の通りです。

定数名 (int component)	内容	対応 PDF バージョン
ENCRYPT_ALL	文書のすべてのコンテンツを暗号化	全バージョン
ENCRYPT_EXCEPT_METADATA	文書のメタデータを除くすべてのコンテンツを暗号化	PDF1.5 以降
ENCRYPT_ONLY_ATTACHED_FILE	添付ファイルのみを暗号化	PDF1.6 以降

- 「添付ファイルのみを暗号化」は、添付ファイルを暗号化し、PDF の文書の暗号化は行われません。添付ファイルを開くためのパスワードとして、ユーザーパスワードの設定 (PtlEncryptStandard::setUserPassword(const PtlParamString & password)) が必須です。
- 添付ファイルのみの暗号化処理 (ENCRYPT_COMPONENT.ENCRYPT_ONLY_ATTACHED_FILE) では、セキュリティの権限設定は行われません。したがって、「SetOwnerPassword」「SetPermission」は呼び出さないようにしてください。

10.7 証明書セキュリティ

証明書ファイルを用いてセキュリティ設定を行う機能です。
設定されているセキュリティ情報の取得もできます。

10.7.1 設 定

セキュリティ設定に使用する証明書の形式は次の2つです。

- X.509 形式 (公開鍵)
- PKCS#12 形式 (秘密鍵)

ひとつの PDF に対して複数の証明書を設定することができます。
証明書ごとに異なるセキュリティ権限の設定ができます。

10.7.2 オ ー プ ン

セキュリティが付いた PDF を開くときに使用する証明書の形式は以下です。

- PKCS#12 形式 (秘密鍵)

10.7.3 制 限 事 項

- 一度設定したセキュリティ権限の内容を変更することはできません。
- すでにセキュリティが設定されているファイルに対し、証明書を追加して設定することはできません。
- 複数の証明書セキュリティが付与されている PDF ファイルに対しセキュリティ削除処理を行った場合、すべての証明書セキュリティが削除されます。
- 証明書によるセキュリティ設定を行う場合、Web 表示用に最適化 (=線形化) する保存は行われません。

- Adobe Acrobat©で作成した 1024bit RSA の pfx ファイルは使用できません。

10.8 データ削除処理

- 指定した矩形内のテキスト、画像、図形を削除します。
- 図形を削除する場合、図形全体を矩形で囲むようにしてください。部分削除は不可なため、図形全体が矩形内に含まれていない場合は削除されません。
- 画像のデータ削除を行った場合、カラスペース、圧縮形式が元のデータとは異なる場合があります。
- リガチャ文字の削除について、グリフ自体がリガチャの場合その一部分を削除することはできません。

10.9 閲覧制限設定

- 閲覧制限設定された PDF は、指定された期間あるいは指定された場所から開かれた場合に閲覧が可能となります。
- 閲覧の可・不可の制御は、Acrobat Java Scriptで行っています。したがって、Acrobat Java Script をサポートしていない PDF ビューアや、Acrobat Java Script の機能が無効になっている PDF ビューアでは設定された制限動作はせず、常に閲覧不可状態、もしくはページが真っ白な状態で表示されます。また、Acrobat Java Script をサポートしていても、PDF ビューアによっては期待する制限動作をしない場合があります。
- 閲覧制限を設定する場合、セキュリティの権限とオーナーパスワードも同時に設定することをお勧めします。
- 閲覧制限設定を行った PDF から閲覧制限設定を取り除くことはできません。
- すでに閲覧制限設定された PDF に対し再度閲覧制限設定を行った場合、設定された閲覧制限の制限動作は保証されません。
- 閲覧可能パスを設定する場合、パスの途中にワイルドカード「*」を使用することができます。
- 開始日・終了日の時刻設定には「協定世界時」(UTC)を使用しています。

※制限事項「閲覧制限」(p. 26) もご参考ください。

10.10 注釈の扱い

PDF Tool API では、ISO 32000-2 で規定されている注釈タイプのすべてを判別します。

既存の注釈に対しては、情報取得と削除ができます。

10.10.1 新規作成

PDF Tool API では、以下の注釈を新規に作成することができます。

- テキスト (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場合があります)
- ファイル添付
- リンク
- スタンプ (既定)
- カスタムスタンプ
- ハイライト

※ハイライト注釈を扱う `PdfAnnotTextMarkup` クラスのオブジェクトに対しては、基底クラス `PdfAnnot` の `setRect` 関数で矩形座標の設定は行わないでください。

10.10.2 編集

PDF Tool API では、以下の注釈を編集することができます。

- テキスト（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場合があります）
- ファイル添付
- リンク
- スタンプ（既定）
- ハイライト

10.10.3 境界線の幅

PDF Tool API において、境界線の幅は「enum PdfAnnot::BORDER_LINE_WITDH」で定義され、以下のように表されています。

- BORDER_WIDTH_NONE：非表示
- BORDER_WIDTH_THIN：細い（＝太さ「1」）
- BORDER_WIDTH_MIDDLE：中（＝太さ「2」）
- BORDER_WIDTH_THICK：太い（＝太さ「3」）

注釈の境界線幅の編集は、以下の条件で行われます。

設定する境界線幅の値	編集処理仕様
BORDER_WIDTH_THIN	元の境界線幅が 0.5 以上 1.5 未満の場合は BORDER_WIDTH_THIN とみなし編集は行われない。 上記以外の線幅であるとき、太さ「1」を設定する。
BORDER_WIDTH_MIDDLE	元の境界線幅が 1.5 以上 2.5 未満の場合は BORDER_WIDTH_MIDDLE とみなし編集は行われない。 上記以外の線幅であるとき、太さ「2」を設定する。
BORDER_WIDTH_THICK	元の境界線幅が 2.5 以上の場合は BORDER_WIDTH_THICK とみなし編集は行われない。 上記以外の線幅であるとき、太さ「3」を設定する。

10.11 テキスト抽出

- ページコンテンツ内のテキストを抽出します。注釈やPDF フォームデータなどのテキストは抽出されません。
- 矩形が指定された場合、矩形内のテキストを抽出します。矩形の指定がない場合はページ全体が抽出対象です。表示領域がページコンテンツの一部分である場合、表示されていない部分のテキストも抽出対象となります。また、クリッピングされて表示されていない部分のテキストも抽出対象となります。
- 「TEXT_SORT」（座標でソート）オプションの場合、次の加工を行い抽出します。
 - 抽出対象が欧文の場合、単語間にスペースを挿入する。
 - 行末の「ハイフン (-)」を、単語が2行にわたっている場合に表記されるものとみなして削除する。
- 抽出処理オプションには以下のものがあります。
 - 取得した順(TEXT_RAW)：取得したテキストをそのまま抽出します。このため、見た目とは異なる順序で抽出される場合があります。
 - 座標でソート(TXT_SORT)：取得したテキストを座標でソートして抽出します。
 - 指定されたユニコード文字をスペースに置換する：指定されたユニコードの文字をスペース(U+0020)に置換します。ユニコード文字の指定がない場合、置換は行われません。
 - 指定されたユニコード文字を削除する：指定されたユニコードの文字を削除します。したがって該当の文字は抽出されません。ユニコード文字の指定がない場合、削除は行われません。
 - 同じ行にある文字とみなすしきい値：文字と文字がどれだけ重なっていると同行とみなすか、その割合を設定します。この設定は、座標でソートする場合に有効です。
 - 同じ文字が重なっているとき抽出から除外するしきい値：同じ文字が重なっているとき、文字と文字がどれだけ重なっているときに取り除くか、その割合を設定します。この設定は、座標でソートする場合に有効です。
 - 違う文字が重なっているとき抽出から除外するしきい値：違う文字が重なっているとき、文字と文字がどれだけ重なっているときに取り除くか、その割合を設定します。重なっていると判定したとき、先に現れた文字、つまり、重なりの下になっている文字が取り除かれます。この設定は、座標でソートする場合に有効です。
 - 文字が矩形内にあるとみなすしきい値：文字が抽出範囲として指定された矩形とどれだけ重なっているときに抽出対象とするか、その割合を設定します。

10.11.1 制 限 事 項

- 英語など欧文テキストに対して、テキストを座標でソートして抽出するモードの場合に、単語のかたまりを判定して単語間に空白を挿入した状態で抽出します。単語であるのに「かたまり」と認識できなかった場合は、次の単語との間に空白が入らず続いた状態で抽出されます。
- 2段以上に段組みされたページにおいて、座標でソートして抽出する場合、段をまたいでソートを行うため、段ごとにテキストは抽出されません。1段ずつ抽出範囲として矩形指定することで、段ごとに抽出できるようになります。
- 縦書き、横書きが混在したページからは期待通りにテキスト抽出できません。抽出するには、縦書きのみ、横書きのみとなるよう抽出範囲を指定してください。
- 異なった文字サイズが混在している場合など1行と認識できないとき、期待通りに抽出されません。

10.12 PDF フォームの値の取得

PDF Tool API V7.0 では、PDF フォームの値を「XFDF」という形式のファイルにエクスポートする機能を追加しました。

XFDF 形式ファイルを介して、PDF フォームの値の取得ができます。

10.12.1 エクスポート対象となる PDF フォームの種類

- チェックボックス
- ラジオボタン
- テキスト
- リストボックス
- ドロップダウン（コンボボックス）

10.12.2 XFDF の仕様について

XFDF の形式は、「ISO 19444-1」に準拠しています。

<記述例>

```
<?xml version="1.0" encoding="UTF-8"?> --- ①
<xfdf xmlns="http://ns.adobe.com/xfdf/" xml:space="preserve"> --- ①
<f href="C:\test\test.pdf"/> --- ②
<ids original="7A0631678ED475F0898815F0A818CFA1" modified="BEF7724317B311718E8675B677EF9B4E"/> --- ③
<fields>
<field name="Text1"> --- ④
<value>アンテナハウス株式会社</value> --- ⑤
</field>
<field name="CheckBox1">
<value>Off</value>
</field>
</fields>
</xfdf>
```

※PDF Tool API が出力する XFDF では、上記のような 1 行空けやインデントなどのレイアウトは行われません。

①		先頭 2 行はこの記述内容で固定されています。
②	<f href="xxx">	"xxx"は XFDF の元である PDF ファイル名
③	<ids original="xxx" modified="xxx">	「ids original」は XFDF の元である PDF ファイルの FDF 辞書の ID エントリ。 「modified」は ID エントリの第 2 識別子。 「ID エントリ」は PDF ファイルが持つ固有の情報です。
④	<field name="xxx">	"xxx"はフォームフィールドのフィールド名
⑤	<value="xxx">	"xxx"はフォームフィールドの値

10.13 PDF/A 変換

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

変換に使用する icc プロファイルの指定ができます。

[Windows 版]

icc プロファイルの指定がない場合、sRGB2014.icc または JapanColor2001Coated.icc を次の順で検索します。

(1)PdfTk70.dll があるフォルダ内

(2)環境変数「PTL70_ICCPROFILE_PATH」で指定されたフォルダ内

[Linux / Amazon Linux2 X86 版]

icc プロファイルの指定がない場合、sRGB2014.icc または JapanColor2001Coated.icc を環境変数「PTL70_ICCPROFILE_PATH」で指定されたフォルダ内で検索します。

10.14 PDF/A の準拠確認

PDF ファイルが特定の規格に準拠しているか否かをチェックします。

V7.0 においてチェック可能な規格は、「PDF/A-1b」と「PDF/A-2b」の2種類です。

準拠エラー箇所がある場合、関数はエラーのコンテナを返します。

10.15 画像の入れ替え

ページコンテンツへの画像・PDF の追加と、指定エレメントの削除に対応することにより、画像の入れ替えを実現します。

Pattern の画像は、入れ替え対象とはなりません。

10.15.1 画像の追加

ページコンテンツ内の指定の場所に画像を追加します。

元の画像がクリッピングされているケースにも対応します。

10.15.2 PDF の追加

ページコンテンツ内の指定の場所に PDF のページを FormXObject として追加します。

10.15.3 エレメントの削除

ページコンテンツ内の指定の場所のエレメントを削除します。

10.16 PDF2.0 対応について

- PDF2.0 の PDF ファイルを入力ファイルとして指定できます。この場合に限り、出力 PDF の PDF バージョンが「2.0」となります。
- PDF2.0 のファイルに対するセキュリティ設定は PDF2.0 仕様となります。AES256bit、「R(リビジョン)6」の暗号化処理になります。これ以外のレベルの暗号化処理はエラーとなります。
- PDF2.0 のファイルとその他の PDF バージョンのファイルを結合することが可能です。この場合、出力結果のファイルの PDF バージョンは PDF2.0 となります。ただし、PDF2.0 以外の PDF バージョンファイルの部分は元の構成のままとなり、PDF2.0 仕様にそったものに変更することはありません。結合とともにその他の編集処理を行った場合についても、内容を PDF2.0 用に変更することはありません。
- PDF 透かしとして PDF2.0 のファイルを指定することができます。出力 PDF は、入力 PDF のバージョンとなります。

第 11 章 仕様変更について

11.1 V7.0 におけるセキュリティ設定の仕様変更

11.1.1 40bit RC4 設定処理は非対応

PDF Tool API V7.0 では、パスワードによるセキュリティ設定処理において、40bit RC4 による暗号化を非対応としました。このため、PDF Tool API V4.0 / V5.0 / V6.0 を使用しているプログラムで 40bit RC4 のセキュリティ設定処理を行っている場合、V7.0 ではエラーとなります。

40bit RC4 で暗号化された PDF の読み込み、セキュリティの削除の各処理は、V7.0 で行うことができます。

11.1.2 256bit AES 設定処理の仕様変更

PDF Tool API V7.0 では、256bit AES による暗号化を「R（リビジョン）6」で行うよう仕様変更しました。「ISO 32000-2」において、256bit AES の場合は「R 6」と規定されました。

PDF Tool API V4.0 / V5.0 / V6.0 では、「R 5」仕様です。

11.2 V7.0 改訂 1 におけるテキスト抽出の仕様変更

V7.0 改訂 1 において、英語など欧文テキストの抽出処理仕様を変更しました。

改訂 1 では、テキストを座標でソートして抽出するモードの場合に、単語のかたまりを判定し、単語間に空白を挿入した状態で抽出します。

11.3 文書情報の Metadata 作成について

[V7.0]

文書情報設定時、入力ファイルに Metadata が存在しない場合、Metadata が新たに作成されます。

PDF Tool API では、保存処理時に文書情報の「更新日時」を設定しますので、API 利用時では保存を行うと自動的に Metadata が作られることになります。

[V6.0 以前のバージョン]

PDF Tool API が新たに Metadata を作成することはありません。

文書情報設定時、入力ファイルに Metadata がない場合、Document Information 辞書のみ更新します。

第 12 章 評価版の仕様について

- Windows 版ではインストール後 30 日間、Linux / Amazon Linux2 X86 版ではインストール後 PDF Tool API を初めて実行した日から 30 日間、使用できます。
- 期間を過ぎると使用できません。エラーコード「11」(License file is expired.) が返ります。
- 出力される PDF ファイルの各ページに、透かし文字列「Antenna House PDF Tool」が挿入されます。

第 13 章 サンプルコードについて

サンプルコードは、製品版／評価版パッケージの「SampleCode」フォルダに格納されています。

C++、.NET(C#)、Java の各言語ごとにあります。

処理内容については、「SampleCode」フォルダ内の「AHPDFToolAPI7_サンプルコード一覧.pdf」をご参照ください。

第 14 章 エラーコード一覧

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

210 *1	Need password. パスワードが必要です。
211 *2	Need user password. ユーザーパスワードが必要です。
212 *3	Need owner password. オーナーパスワードが必要です。
213	Invalid encrypt key length. 暗号化キー長が間違っています。
214	Invalid encrypt permission. 権限が間違っています。
215	Invalid encrypt component. 暗号化する文書コンポーネントに誤りがあります。
216	Invalid encrypt method. 暗号化メソッドが間違っています。
217 ※V7.0 追加	Need PKCS12. PKCS12 が必要です。
220	Can not read attached file. 添付ファイルの読み込みが出来ません。
221	Can not write attached file. 添付ファイルの書き出しが出来ません。
222	No attached file. 添付ファイルがありません。
223	Attached file has no name. 添付ファイルに名前がありません。
230	Can not read image file. 画像ファイルの読み込みが出来ません。
231	Can not write image file. 画像ファイルの書き出しが出来ません。
232	Unsupported image. 未対応の画像です。
233	Unsupported image for stencil mask. ステンシルマスクとしてサポートしていない画像です。
234	Image is not single. ステンシルマスクがモノクロ画像ではありません。
235	Unsupported image for color key mask. カラーキーマスクとしてサポートしていない画像です。
236	Unsupported image for explicit mask. 明示マスクとしてサポートしていない画像です。
237	Image is not single. 明示マスクがモノクロ画像ではありません。
238	Image is not gray scale. ソフトマスクとしてサポートしていない画像です。
240	Image processing error. イメージ処理で問題が発生しました。
241 ※V7.0 追加	Can not read ICC Profile. ICC プロファイルの読み込みが出来ません。
245	Font processing error. フォント処理で問題が発生しました。
250	Can not insert page. ページの挿入が出来ません。
251	Can not delete page. ページの削除が出来ません。
252	Has no pages. ページが存在しません。
260 *4	Free docproperty error. DocProperty がドキュメントからフリーです。
261 *4	Free openmode error. OpenMode がドキュメントからフリーです。

262 *4	Free embeddedfiles error. EmbeddedFiles がドキュメントからフリーです。
263 *4	Free pages error. Pages がドキュメントからフリーです。
264 *4	Free page error. Page がドキュメントからフリーです。
270	Can not set to root outline. ルートアウトラインには設定出来ません。
271 *5	Can not set to free outline. フリーアウトラインには設定出来ません。
280	Invalid FDF file. FDF が異常です。
281	Cannot read FDF. FDF の読み込みができません。
282	Cannot write FDF. FDF の書き出しができません。
290 ※V7.0 追加	Cannot read PKCS12. PKCS12 の読み込みができません。
291 ※V7.0 追加	Cannot read X509. X509 の読み込みができません。
500 *6	Linearize processing error. 線形化処理で問題が発生しました。
700	null value. null 値です。
800	No Object. オブジェクトが存在しません。
900	Not enough memory. メモリが不足しています。
901	Internal error. 内部エラーです。
902	Other error. その他のエラーです。
999	Sorry, not implemented. 未実装です。

※備考

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

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

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

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

*5[271] しおりが作成できないことを示します。

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

第 15 章 商標・著作権情報

15.1 商 標 情 報

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

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

Adobe、Acrobat、および Reader は、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 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

This product includes software developed by Independent JPEG Group

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

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

* libtiff

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

Copyright (c) 1988-1997 Sam Leffler

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

* zlib

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

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

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

* libpng

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

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

Copyright (c) 2000-2002 Glenn Randers-Pehrson

Copyright (c) 1998, 1999 Glenn Randers-Pehrson

Copyright (c) 1996, 1997 Andreas Dilger

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

* j2k(jasper)

JasPer License Version 2.0

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

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

Copyright (c) 2001-2003 Michael David Adams

All rights reserved.

*** LittleCMS**

Little Color Management System

Copyright (c) 1998-2016 Marti Maria Saguer

*** ICU**

This product includes softwares developed by:

International Business Machines Corporation

International Components for Unicode (ICU) libraries

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

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

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

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

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

*** FreeType**

This product includes softwares developed by:

Portions of this software are copyright 2017 The FreeType Project (www.freetype.org).

All rights reserved.

*** OpenSSL**

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

履歴

日付	内容
2021.4.27	初版
2021.6.29	<ul style="list-style-type: none"> ・Linux と Java API についての記述を追加しました。 ・「制限事項」の「テキスト抽出」項において、英語など欧文テキストについての記述を修正しました。 ・「仕様変更について」の章において、「V7.0 改訂 1 1 におけるテキスト抽出の仕様変更」を追加しました。
2021.7.29	・「仕様について」の章において、「PDF フォームの値の取得」の項を追加しました。
2021.9.24	・「制限事項」-「テキスト抽出」の説明を修正しました。
2021.12.16	・「動作環境」を修正しました。
2022.1.20	・「8.1.全般」の制限事項 1 項を修正しました。保存される PDF のファイルサイズが 2GB を超える場合、エラーコードは「902」ではなく「103」となります。
2022.4.1	・「7.1.1 モジュールファイル名」の「依存関係にあるファイル」を修正しました。
2022.4.14	<ul style="list-style-type: none"> ・「第 2 章 特長」を「特長とおもな機能」に変更し、「V7.0 の新機能」と「その他のおもな機能」の項を追加しました。 ・「6.1 Windows 版」「6.2 Linux 版」それぞれに「インストール先の構成」の項を追加しました。 ・「第 12 章 仕様について」に「PDF/A 変換」「PDF/A の準拠確認」「画像の入れ替え」の 3 項を追加しました。
2022.5.9	・「8.5 閲覧制限」「12.7 閲覧制限設定」「12.13 テキスト抽出」の説明を修正しました。
2022.6.29	・「15.2 第三者ライブラリー著作権情報」を修正しました。
2022.7.12	・「13 サンプルコード一覧」に「59 ExtractTextElem (テキストエレメントの取得)」を追加しました。
2022.8.29	・「8 制限事項」に「8.2 保存」項を追加しました。「8.1 全般」と「8.6 閲覧制限」を修正しました。
2022.11.10	・「8 制限事項」の「8.6 閲覧制限」の説明を修正しました。
2022.11.11	・「8 制限事項」の「8.6 閲覧制限」の説明を修正しました。
2023.2.3	<ul style="list-style-type: none"> ・「3.2 プログラム言語」の「Java」の項に「Java8」を追加しました。 ・「6.1.7 インストールされる「PdfTkJava70.jar」について」「6.2.5 インストールされる「PdfTkJava70.jar」について」を追加しました。
2023.3.15	・「対応環境とプログラム言語」に「Amazon Linux2 X86」を追加しました。
2023.3.29	<ul style="list-style-type: none"> ・「制限事項」と「仕様について」を修正しました。 ・章構成を変更しました。
2023.5.29	・改訂 6 版にて .NET6 に対応しました。「2.1 特長」「2.2.1.V7.0 の新機能」「3.2 プログラム言語」「7 モジュールファイルについて」の各項に .NET6 用の記述を追加しました。
2024.1.25	・「3.1 対応プラットフォーム」-「3.1.1 Windows」を修正しました。
2024.3.27	・「第三者ライブラリー著作権情報」において一部の URL アドレスを修正しました。
2024.6.6	・「第 9 章 制限事項」に「PDF の表示あるいは描画」項を追加しました。
2024.11.8	<ul style="list-style-type: none"> ・「第 13 章」のタイトルを「サンプルコードについて」に変更し、説明文を修正しました。 ・「9.4 閲覧制限」の制限事項の説明を修正しました。
2024.12.13	・.NET6 用モジュールファイルは .NET8 で使用可能です。関係箇所「.NET8」の記述を追加しました。
2024.12.23	・「2-1 特長」の「処理対象ファイル」の説明を修正しました。
2025.2.3	・「3.1 対応プラットフォーム」-「3.1.1 Windows」に「Server 2025」を追加しました。
2025.2.18	・「10.5 Acrobat の透かし機能との互換性について」の説明文を修正しました。
2025.4.2	<ul style="list-style-type: none"> ・「2.2.1 V7.0 の新機能」の「テキストボックス機能」の説明を修正しました。 ・「3.2 プログラミング言語」の「.NET」と「Java」の説明を修正しました。 ・「10.1.6 フォントの統合」の説明を修正しました。 ・「11.3 文書情報の Metadata 作成について」の項を追加しました。 ・「12 評価版の仕様について」の説明を修正しました。
2025.10.21	・「3.1 対応プラットフォーム」-「3.1.1 Windows」を修正しました。

