

Antenna House PDF Tool API

V5.0/V6.0 から V8.0 へのバージョンアップについて

アンテナハウス株式会社

内容

1. はじめに.....	3
2. 仕様変更について	4
2.1. セキュリティ設定の仕様変更.....	4
2.1.1. 40bit RC4 の非対応.....	4
2.1.2. 256bit AES 設定の変更点.....	4
2.2. 文書情報の Metadata の作成.....	4
2.3. 文字列取得に関する戻り値の変更.....	5
2.4. 綴りの修正.....	5
3. 開発環境について	6
4. 動作環境について	7
5. ファイル名、環境変数名	8
6. ライセンスファイル.....	8
7. バージョンアップ方法 — Windows.....	9
7.1. 開発環境の準備.....	9
7.2. ライセンスファイルの準備	9
7.3. C++ API の場合	9
7.4. .NET Framework API の場合.....	9
7.5. Java API の場合	10
7.6. コマンドライン.....	10
7.7. ビルドした実行ファイルの配置について	10
7.8. 環境変数について	10
7.8.1. ライセンスファイル	10
7.8.2. フォント構築ファイル.....	11
8. バージョンアップ方法 — Linux	12
8.1. 開発環境の準備.....	12
8.2. ライセンスファイルの準備	12
8.3. C++ API の場合	12
8.4. Java API の場合	12
8.5. コマンドライン.....	13
8.6. ビルドした実行ファイルの配置について	13
8.7. 環境変数について	13
8.7.1. ライセンスファイル	13
8.7.2. フォント構築ファイル.....	13

1. はじめに

本書では、PDF Too API V5.0 / V6.0 で開発を行っていたプログラムを PDF Tool API V8.0 にバージョンアップするさいの手順について説明します。

仕様や開発／動作環境等についてのバージョン間の違いもあわせて説明します。

2. 仕様変更について

2.1. セキュリティ設定の仕様変更

2.1.1. 40bit RC4 の非対応

V8.0 では、40bit RC4 のセキュリティ設定は非対応です。処理はエラーになります。
40bit RC4 のセキュリティ設定された PDF ファイルの読み込みやセキュリティの削除の処理には対応しています。

2.1.2. 256bit AES 設定の変更点

V8.0 では、256bit AES のセキュリティ設定は「R（リビジョン）6」で行います。

2.2. 文書情報の Metadata の作成

V8.0 では、入力ファイルに Metadata が存在しない場合、PDF Tool API が新たに Metadata を作成します。

V5.0/V6.0 では、入力ファイルに Metadata が存在しない場合、PDF Tool API が新たに Metadata を作成することはありません。文書情報設定を行ったときは Document Information 辞書を更新します。

2.3. 文字列取得に関する戻り値の変更

過去バージョンにて「PtlParamString」クラスを戻り値として受け取っていた各種関数に関して、V8.0 では「const PtlParamString&」の形でポインタを戻り値として受け取るように変更しました。

【例】

V6.0/V5.0	V8.0
PtlParamString getText()	const PtlParamString& getText ()

2.4. 綴りの修正

V5.0 では綴りの誤りがありました。V6.0 以降で修正されています。

V5.0 での綴りの誤り	V8.0	
PtlEncryptPermissionType2::setAccessability	PtlEncryptPermissionType2::setAccessability	PtlEncryptPermissionType2 クラス
PtlEncryptPermissionType2::getAccessability	PtlEncryptPermissionType2::getAccessability	PtlEncryptPermissionType2 クラス
PERM_MODIFY_ASSEMBLEANDFORM	PERM_MODIFY_ASSEMBLEANDFORM	enum PtlEncryptPermissionType1::PERMISSION_MODIFY のメンバー
PERM_MODIFY_ASSEMBLEDOSC	PERM_MODIFY_ASSEMBLEDOSC	enum PtlEncryptPermissionType2::PERMISSION_MODIFY のメンバー
PtlParamWaterMark::setDisplayWaterMark	PtlParamWaterMark::setDisplayWaterMark	PtlParamWaterMark クラス

3. 開発環境について

PDF Tool API の各バージョンは、以下の開発ツールおよびバージョンで開発されています。

PDF Tool API の Ver.	V5.0	V6.0	V8.0
Windows	Microsoft Visual Studio 2010	Microsoft Visual Studio 2015	Microsoft Visual Studio 2022
Linux	GCC 4.1	GCC 4.8	GCC 11.4
.NET Framework	4.0	4.5.2	4.8
.NET Core	非対応	非対応	.NET 8
Java	Java 7	Java 8	Java 11 ※改訂 2a(8.0.2.1)以降

4. 動作環境について

PDF Tool API の Ver.		V5.0	V6.0	V8.0
Windows	11	-	○	○
	10	⊖ ※1	⊖ ※1	非対応
	Server 2025	非対応	非対応	○
	Server 2022	非対応	○	○
	64bit	○	○	○ ※2
	32bit	○	○	非対応
Linux (X64)		GCC 4.1 をサポートするバージョン (RedHat 系 ver.5)	GCC 4.8 をサポートするバージョン (RedHat 系 ver.7)	GCC11.4 をサポートするバージョン (RedHat 系 ver.9、Amazon Linux 2023)
	64bit	○	○	○
	32bit	○	非対応	非対応
.NET Framework		4.0 以上	4.5.2 以上	4.8 以上
.NET Core		非対応	非対応	.NET 8
Java		Java 7	Java 8	Java 11/17/21/25 ※3

※1 Windows 10：2025 年 10 月 14 日、Microsoft のサポートが終了したため、本製品において動作保証外となります。

※2 Windows 版：V8.0 では 64bit のみ対応しています。

※3 Java11 用モジュールファイルにて、Java 17 / 21 / 25 環境で動作確認済みです。

5. ファイル名、環境変数名

dll などのファイル名や環境変数名は、バージョン固有の名称です。
このため、V5.0/V6.0 と V8.0 では名称が変更となっています。

6. ライセンスファイル

PDF Tool API のライセンスファイルは、バージョンごとに固有の情報を持っています。
V8.0 の動作には、V8.0 用ライセンスファイルが必要です。

7. バージョンアップ方法 — Windows

PDF Tool API V5.0/V6.0 を使用したプログラムを V8.0 用にバージョンアップする方法について説明します。

7.1. 開発環境の準備

- ・ V8.0 で提供する API は「64bit」のみです。プログラムが「32bit」で作成されている場合、64bit 対応をしてください。
- ・ C++ API、.NET Framework API のプログラムの場合、Microsoft Visual Studio 2022 をご使用ください。
- ・ .NET Framework 利用の場合、対象のフレームワークバージョンは「.NET Framework 4.8」以上です。
- ・ Java API の場合、対象のバージョンは「Java 11/17/21/25」です。

7.2. ライセンスファイルの準備

V8.0 の動作には、V8.0 用のライセンスファイルをご使用ください。

V5.0/V6.0 用のライセンスファイルでは動作しません。

7.3. C++ API の場合

- (1) PDF Tool API V5.0/V6.0 のプログラムのプロジェクトを開きます。
- (2) リンクする lib ファイルを「PdfTk80.lib」に変更します。
- (3) 参照するヘッダファイルの場所を V8.0 用に変更します。
- (4) ビルドします。

7.4. .NET Framework API の場合

- (1) PDF Tool API V5.0/V6.0 のプログラムのプロジェクトを開きます。
- (2) 参照する DLL ファイルを「PdfTkNet80.dll」に変更します。
- (3) プロジェクトのプロパティ「対象のフレームワーク」において、「.NET Framework 4.8」以上を選択します。

- (4) ビルドします。

7.5. Java API の場合

- (1) PDF Tool API V5.0/V6.0 のプログラムのプロジェクトを開きます。(IDE ツール利用の場合)
- (2) リンクする jar ファイルを「PdfTkJava80.jar」に変更します。
- (3) 環境変数「PATH」に依存関係 DLL (PdfTk8JNI.dll など) が存在するパスを追加します。
インストール時にオプション「環境変数 PATH へ追加する」にチェックを入れて行っている場合は「PATH」は設定されています。
- (4) ビルドします。

7.6. コマンドライン

コマンドラインの exe ファイル名を「AHPDFToolCmd80.exe」に変更してください。

7.7. ビルドした実行ファイルの配置について

ビルドした実行ファイルの配置場所は、PDF Tool API V5.0/V6.0 用の実行ファイルの運用時と同様です。

PDF Tool API のモジュールファイルを V8.0 用のものにしてください。

7.8. 環境変数について

7.8.1. ライセンスファイル

PDF Tool API V5.0/V6.0 のプログラムの運用時、ライセンスファイルの配置に環境変数を使用している場合、V8.0 用の環境変数を作成し、指定したパスにライセンスファイルを配置してください。

環境変数名：PTL80_LIC_PATH

7.8.2. フォント構築ファイル

PDF Tool API V5.0/V6.0 のプログラムの運用時、フォント構築ファイル用の環境変数を設定している場合、V8.0 用の環境変数を作成し、フォント構築ファイルの場所を設定してください。

環境変数名：PTL80_FONT_CONFIGFILE

8. バージョンアップ方法 — Linux

PDF Tool API V5.0/V6.0 を使用したプログラムを V8.0 用にバージョンアップする方法について説明します。

8.1. 開発環境の準備

- V8.0 で提供する API は「64bit」のみです。V5.0 にてプログラムが「32bit」で作成されている場合、64bit 対応をしてください。
- C++ API の場合、GCC 11.4 をご使用ください。
- Java API の場合、対象のバージョンは「Java 11/17/21/25」です。

8.2. ライセンスファイルの準備

V8.0 の動作には、V8.0 用のライセンスファイルをご使用ください。
V5.0/V6.0 用のライセンスファイルでは動作しません。

8.3. C++ API の場合

- 「LD_LIBRARY_PATH」に V8.0 用の lib ディレクトリを指定します。
- g++ コマンドで「lib」ファイルを指定している場合、一部のファイル名については接頭語が「libPtk」に変更されていますので、該当ファイル名については変更してください。

8.4. Java API の場合

- (1) PDF Tool API V5.0/V6.0 のプログラムのプロジェクトを開きます。(IDE ツール利用の場合)
- (2) リンクする jar ファイルを「PdfTkJava80.jar」に変更します。
- (3) 「LD_LIBRARY_PATH」に V8.0 用の lib ディレクトリを指定します。
- (4) ビルドします。

8.5. コマンドライン

コマンドラインの実行ファイル名を「AHPDFToolCmd80」に変更してください。

8.6. ビルドした実行ファイルの配置について

ビルドした実行ファイルの配置場所は、PDF Tool API V5.0/V6.0 用の実行ファイルの運用時と同様です。

PDF Tool API のモジュールファイルを V8.0 用のものにしてください。

8.7. 環境変数について

8.7.1. ライセンスファイル

PDF Tool API V5.0/V6.0 のプログラムの運用時、ライセンスファイルの配置に環境変数を使用している場合、V8.0 用の環境変数を作成し、指定したパスにライセンスファイルを配置してください。

環境変数名：PTL80_LIC_PATH

8.7.2. フォント構築ファイル

PDF Tool API V5.0/V6.0 のプログラムの運用時、フォント構築ファイル用の環境変数を設定している場合、V8.0 用の環境変数を作成し、フォント構築ファイルの場所を設定してください。

環境変数名：PTL80_FONT_CONFIGFILE

以上