PDF Tool API V7.0 サンプルコードのビルド手順（Linux/Amazon Linux2）

アンテナハウス株式会社

目　　次

1 PDF Tool API の開発環境を整える（Linux / Amazon Linux2 X86）

1.1 ライブラリファイル、ヘッダーファイルを配置する

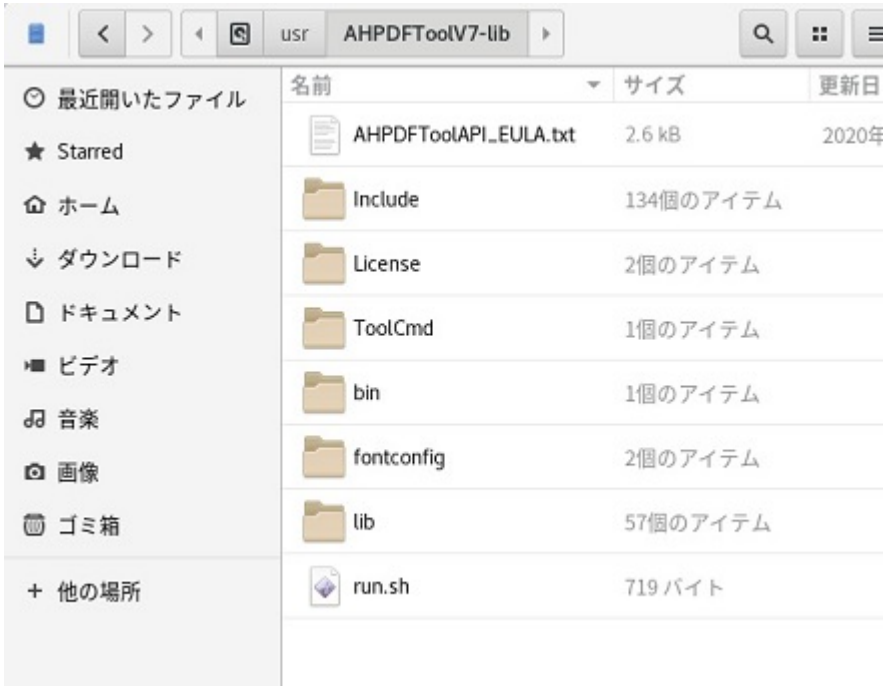
インストーラを利用してセットアップを行います。

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） --prefix を省略した場合、「/usr/AHPDFToolV7-lib」にインストールされます。



1.1.1 インストールディレクトリの内容

インストールされるディレクトリ、ファイルの内容は次の通りです。

ディレクトリ/ファイル名	内容
include	ヘッダファイル
lib	API ライブラリファイル
bin	Java インターフェース用（Java11 コンパイルファイル）※
ToolCmd	コマンドライン実行ファイル
fontconfig	フォント構築ファイル
License	ライセンスファイル

run.sh	コマンドライン用シェルスクリプト

※「Java8」をご使用になる場合は、「{製品データ}/Lib-Linux/Java8」ディレクトリにある「PdfTkJava70.jar」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。

1.1.2 API ライブラリファイルの参照方法

PDF Tool API を使用したプログラムを実行するには、環境変数「LD_LIBRARY_PATH」に「{インストールディレクトリ}/lib」へのパスを追加してください。

（設定例）

$ export LD_LIBRARY_PATH=[Install directory]/lib:${LD_LIBRARY_PATH}

1.2 ライセンスファイルを配置する

- インストーラにより、ライセンスファイルは「{インストールディレクトリ}/License」に配置されます。
- 環境変数「PTL70_LIC_PATH」にライセンスファイルが存在するディレクトリを設定してください。

  （設定例）

  $ export PTL70_LIC_PATH=[Install directory]/License

1.3 フォントの準備

テキスト透かしを挿入したりページ上に文字を描画するには、フォント情報が必要です。

PDF Tool API は、システムに存在するフォントを参照します。

そのため、あらかじめ、参照するフォントの場所を「フォント構築ファイル」に設定します。

Linux 環境においては、「フォント構築ファイル」の設定は必須です。この設定がない場合、テキスト透かしや文字描画の処理は行われません。

1.3.1 フォント構築ファイルの作成

1. フォント構築ファイルは、下記の場所にあります。
   {インストールディレクトリ}/fontconfig

2. fontconfig ディレクトリ内には以下の 2 つのファイルがあります。
   font-config.xml　：フォント構築ファイルのひな型
   font-config.dtd　：font-config.xml の定義ファイル

3. font-config.xml の「font-folder path」タグに、フォントファイルが存在するディレクトリを記述します。

（例）

```
<font-config>
  <font-folder path=”/usr/share/fonts/ipa-gothic”></font-folder>
  <font-folder path=”/usr/share/fonts/ipa-mincho”></font-folder>
</font-config>
```

4. font-config.xml と font-config.dtd を任意の場所に配置します。

5. 環境変数「PTL70_FONT_CONFIGFILE」を作成し、font-config.xml のパスを設定します。
（設定例）
$ export PTL70_FONT_CONFIGFILE=[Install directory]/fontconfig/font-config.xml

1.4 必要なランタイムライブラリ

[Linux]
glibc 2.28
libstdc++ 6.0.25

[Amazon Linux2 X86]
glibc 2.26.amzn2
libstdc++ 7.3.1-15.amzn2

1.5 環境変数のまとめと参考シェルスクリプト

1.5.1 環境変数のまとめ

Linux 環境において PDF Tool API を使用したプログラムを実行するためには、次の環境変数の設定が必須です。

環境変数名	設定値
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
PTL70_FONT_CONFIGFILE	フォント構築ファイル「font-config.xml」のフルパス
LD_LIBRARY_PATH	API ライブラリファイルが存在するディレクトリ
PTL70_CODEPAGE	PDF Tool API を特定の文字コードで使用する場合に文字コード名を指定します。 [設定例] PTL70_CODEPAGE=EUC-JP 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。
PTL70_ICCPROFILE_PATH	ICC プロファイルが存在するディレクトリ PDF/A-1b、PDF/A-2b 変換処理を行う場合は設定が必要です。

1.5.2 参考シェルスクリプト

PDF Tool API を使用したプログラムを実行するための環境構築については、シェルスクリプト
「{インストールディレクトリ}/run.sh」
をご参考ください。
「run.sh」は、コマンドライン「AHPDFToolCmd70」を実行するスクリプトです。

2 C++サンプルコードのコンパイルと実行について

2.1 環境変数の設定

環境変数を設定します。変数の内容については、「環境変数のまとめ（p. 3）」を参照してください。

（設定例）

```
$ export LD_LIBRARY_PATH=[Install directory]/lib:${LD_LIBRARY_PATH}
$ export PTL70_FONT_CONFIGFILE=[Install directory]/fontconfig/font-config.xml
$ export PTL70_ICCPROFILE_PATH=[Install directory]/icc
$ export PTL70_LIC_PATH=[Install directory]/License
```

2.2 サンプルコードのコンパイルと実行

PDF Tool API では、C++用のサンプルコードを用意しています。
ソースファイルは、「SHIFT-JIS」を使用しています。
以下は、サンプルプログラム「AppendPages.cpp」をコンパイルして実行する例です。

[注意事項]
Linux 版は GCC8.3.1 で、Amazon Linux2 X86 版は GCC 7.3.1-15.amzn2 でビルドされています。サンプルコードをコンパイルする場合は、それぞれと互換のあるコンパイラをご使用ください。
GCC のバージョンは、以下のコマンドで確認できます。

```
$ g++ --version
```

1.「AppendPages.cpp」に「#include <stdio.h>」を追加します。
2. カレントディレクトリを「AppendPages.cpp」がある場所に移動します。
3. 次のコマンドでコンパイルします。

```
$ g++ AppendPages.cpp -o AppendPages -I [install directory]/Include -L [install directory]/lib -lPtkAHCommon -lPtkAHDMC -lPtkAHGraphicService -lPtkAHFontService -lPdfTk -lPdfTkEx -lPtkAHPDFLib -lPtkAHPDFEditLib -lPtkPDFLinearizer -licuuc -licui18n -licudata
```

4. 次のようにして実行します。

```
$ ./AppendPages in.pdf out.pdf append.pdf
```

3 Javaサンプルコードのコンパイルと実行について

3.1 環境変数の設定

環境変数を設定します。変数の内容については、「環境変数のまとめ（p. 3）」を参照してください。

```
$ export LD_LIBRARY_PATH=[Install directory]/lib:${LD_LIBRARY_PATH}
$ export PTL70_FONT_CONFIGFILE=[Install directory]/fontconfig/font-config.xml
$ export PTL70_ICCPROFILE_PATH=[Install directory]/icc
$ export PTL70_LIC_PATH=[Install directory]/License
```

3.2 環境変数「CLASSPATH」の設定

「CLASSPATH」には、「PdfTkJava70.jar」のフルパス名を設定します。

（設定例）

```
$ export CLASSPATH=[Install directory]/bin/PdfTkJava70.jar:${CLASSPATH}
```

3.3 アプリケーションサーバにおける使用について

Tomcatなどのアプリケーションサーバにおいて本インターフェイスを使用する場合、PdfTkJava70.jarをWEBアプリケーションのWEB-INF/libに置かないようにしてください。

JavaVMの仕様によりJNIのネイティブライブラリは複数のクラスローダから読み込めないようになっているため、各WEBアプリケーションディレクトリにPdfTkJava70.jarを置くと複数のWEBアプリケーションから使用することができなくなります。

これを防ぐにはシステムクラスローダなどロードが一度だけ行われるクラスローダで読み込ませるように設定してください。

3.4 サンプルコードのコンパイルと実行

PDF Tool APIでは、Java用のサンプルコードを用意しています。

ソースファイルは「UTF-8」を使用しています。

これらのサンプルプログラムをコンパイルして実行する手順は以下の通りです。

＊＊ 「AppendPages.java」サンプルの場合 ＊＊

1. カレントディレクトリをサンプルコードが存在するフォルダに切り替えます。

```
$ cd {任意のディレクトリ}/SampleCode/Java
```

2. 「AppendPages.java」をコンパイルします。

```
$ javac AppendPages.java
```

3. サンプルコードでは「package SampleTryWithResources;」が定義されています。実行するには、コンパイルされた「AppendPages.class」を「SampleTryWithResources」フォルダに移動します。

```
$ mkdir SampleTryWithResources
$ mv AppendPages.class SampleTryWithResources
$ java SampleTryWithResources.AppendPages in.pdf out.pdf append.pdf
```

4 サンプルコード一覧

	ファイル名	処理内容
1	AllocatePages	ページ割り付け
2	AppendAnnotFileAttachment	ファイル添付注釈の作成
3	AppendAnnotLink	リンク注釈の作成
4	AppendAnnotStamp	スタンプ注釈の作成
5	AppendAnnotText	テキスト注釈の作成
6	AppendColorWatermark	色透かしの挿入
7	AppendEmbeddedFile	ファイルの添付
8	AppendImageWatermark	画像透かしの挿入
9	AppendOutline	しおりの追加
10	AppendPages	PDF ファイルの結合
11	AppendPdfWatermark	PDF 透かしの挿入
12	AppendTextWatermark	テキスト透かしの挿入
13	Decrypt	セキュリティの解除
14	DividePage	ページ抽出
15	DrawForm	フォームの描画
16	DrawImage	画像の描画
17	DrawLayer	レイヤーの描画
18	DrawShape	パス（矩形、線）の描画
19	DrawTextBox　※V7.0 新規	テキストボックス
20	Encrypt	セキュリティ設定
21	EncryptPubSec　※V7.0 新規	証明書セキュリティ設定
22	ExportFormDataXFDF　※V7.0 新規	PDF フォームデータの XFDF へのエクスポート
23	ExtractImage	画像抽出
24	ExtractPage	ページ抽出
25	ExtractText	テキスト抽出
26	FixUpPDFA　※V7.0 新規	PDF/A-1b 変換
27	GetAnnots	注釈情報の取得
28	GetDocInfo	文書情報の取得
29	GetEncryptInfo	セキュリティ情報の取得
30	GetOpenMode	開き方情報の取得

31	GetOutline	しおり情報の取得
32	ImageToPdf	画像ファイルの PDF 化
33	ImageToPdfColorkeyMask	カラーキーマスクによる画像の PDF 化
34	ImageToPdfExplicitMask	明示マスクによる画像の PDF 化
35	ImageToPdfSMask	ソフトマスクによる画像の PDF 化
36	ImageToPdfStencilMask	ステンシルマスクによる画像の PDF 化
37	ImportFormDataXFDF ※V7.0 新規	PDF フォームデータの XFDF からのインポート
38	InsertImagePage	画像ファイルの PDF 化と既存 PDF へのページ追加
39	Optimize	最適化
40	OutputEmbeddedFile	添付ファイルの書き出し
41	RemoveAnnots	注釈の削除
42	RemoveEmbeddedFile	添付ファイルの削除
43	RemoveOutline	しおりの削除
44	RemovePages	ページの削除
45	RemoveWaterMark	透かしの削除
46	ReplaceImage　※V7.0 新規	画像の入れ替え
47	RotatePage	ページの回転
48	SearchTextAndDrawPath	テキスト検索と矩形描画
49	SearchTextAndHighlight	テキスト検索とハイライト注釈追加 （注釈追加の位置指定処理）
50	SearchTextAndHighlightAuto	テキスト検索とハイライト注釈追加 （注釈追加は関数による自動処理）
51	SearchTextAndMaskAuto	テキスト検索とデータ削除
52	SeparationColor ※V7.0 新規	特色を使用した円の描画
53	SetDocInfo	文書情報の設定
54	SetMask	墨消し処理
55	SetOpenMode	開き方の設定
56	SetRestriction	閲覧制限設定
57	WriteString	テキストオブジェクトの挿入
58	ZoomPage	ページの拡大・縮小
59	ExtractTextElem	テキストエレメントの取得

5 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 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 ※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。

PDF Tool API V7.0 サンプルコードのビルド手順（Linux/Amazon Linux2）

著　者　アンテナハウス株式会社