Antenna House PDF Driver V8.0 付属 API 説明書

目次

Antenna House PDF Driver V8.0 付属 API 説明書 1
１．　付属 API の概要 3
１．１．出力先指定 API 3
１．２．設定編集 API 3
１．３．印刷開始・終了通知（設定編集 API） 3
１．４．32bit／64bit　ネイティブ対応 3
１．５．マルチプロセス／マルチスレッド対応 3
２．　動作環境 5
３．　開発環境 6
３．１．出力先指定 API 6
３．１．１．　インターフェース 6
３．１．２．　コンパイラ 6
３．２．設定編集 API 7
３．２．１．　インターフェース 7
３．２．２．　コンパイラ 7
３．３．32bit と 64bit の区別 7
４．　制限事項 8
４．１．付属 API の制限事項 8
４．２．PDF Driver の制限事項 10
５．　付属 API の使い方 13
５．１．PDF Driver のインストール 13
５．１．１．　PDF Driver のインストール方法 13
５．１．２．　PDF Driver のアンインストール方法 14
５．２．API の導入方法 15
５．２．１．　開発環境への導入手順 15
５．２．２．　実行環境での API 実行ファイルの配置 16
６．　ライセンスファイルについて 17
６．１．PDF Driver 製品のファイル名と配置場所 17
６．１．１．　ライセンスファイル名 17
６．１．２．　ライセンスファイルの場所 17
６．２．PDF Driver API 製品のファイル名と配置場所 17
６．２．１．　ライセンスファイル名 17
６．２．２．　ライセンスファイルの場所 17
６．３．正規利用方法 17
６．４．保守期限と付属 API の動作について 17
６．５．ライセンスファイルの入れ替えについて 17
６．６．ライセンスファイルの削除 18
７．　付属 API の詳細 19
７．１．プリンター選択処理 19
７．１．１．　プリンターの選択：SelectPrinter の仕様 19
７．１．２．　出力先指定 API の「PDFDriverSetting::SelectPrinter」と設定編集 API の「CEditSettingFile::SelectPrinter」の同時使用について 20

７.１.３.　SelectPrinter のエラーコード「4」について 20
７.１.４.　選択状態の解放：ReleasePrinter 20
７.２. サンプルコード 21
７.３. プログラミングの注意事項 21
７.４. 出力先指定 API 22
７.４.１.　Namespace 名と Class 名 22
７.４.２.　関数仕様 22
７.４.３.　API による出力先指定と「同名のファイルがある場合」の設定について ... 25
７.５. 設定編集 API 26
７.５.１.　Namespace 名と Class 名 26
７.５.２.　関数仕様 26
７.５.３.　印刷開始終了通知のしくみ 35
８.　旧バージョンとの違いについて 36
８.１. ファイル名について 36
８.１.１.　出力先指定 API 36
８.１.２.　設定編集 API 36
８.１.３.　SelectPrinter 処理ライブラリーファイル 36
８.２. 出力先指定 API について 36
８.２.１.　ヘッダーファイル名(C++) 36
８.２.２.　Name Space 名(C++) 36
８.２.３.　Name Space 名(.NET Framework) 36
８.２.４.　対応する.NET Framework バージョン 37
８.３. 設定編集 API について 37
８.３.１.　ヘッダーファイル名 37
８.３.２.　Name Space 名 37
８.３.３.　追加された定義 37
８.４. 印刷設定ファイルについて 38
８.５. 開発環境 38
９.　商標と著作権情報 39
９.１. 商標情報 39
９.２. 第三者製ライブラリー著作権情報 39
履歴 43

１．付属 API の概要

『付属 API』は、『Antenna House PDF Driver V8.0』（PDF Driver V8.0）を制御する API です。『出力先指定 API』、『設定編集 API』の 2 種類があります。どちらの API についても 32bit 用、64bit 用それぞれのモジュールを用意しています。
これらの付属 API では、付属 API 側で印刷処理は行いません。

１.１.　出力先指定 API

おもな機能は、PDF Driver V8.0 の出力先ファイルパスの指定です。出力先を指定することにより、プログラムにおいて PDF Driver の印刷処理を行うとき、PDF Driver の保存ダイアログが表示されることなく PDF 出力が行われます。

対応インターフェース：C++、.NET Framework 4.6.2 以降
※　.NET Framework 用 API は C++ API のラッパークラスとして作成されています。
※　.NET Core、.NET5 には非対応です。
※　.NET6 には対応予定です。

１.２.　設定編集 API

PDF Driver V8.0 の印刷設定の編集や PDF Driver への設定値の指示、PDF Driver の印刷通知処理を行います。API で編集した設定値を PDF Driver へ送りその後 PDF Driver で印刷処理を実行したとき、出力される PDF は API で編集した設定値となります。API で編集した設定値は、印刷が終了すると破棄されます。

対応インターフェース：C++

１.３.　印刷開始・終了通知（設定編集 API）

PDF Driver V8.0 は、印刷の開始と終了をメッセージブロードキャストで通知します。設定編集 API でメッセージの識別番号や文字列を指定すると、その番号や文字列で Windows メッセージを受信し、プログラム側で印刷の開始や終了を認識することができます。（印刷開始終了通知のしくみ）

１.４.　32bit／64bit　ネイティブ対応

出力先指定 API、設定編集 API は、32bit、64bit にネイティブ対応しています。モジュールファイルは、32bit 用、64bit 用それぞれに用意されています。

１.５.　マルチプロセス／マルチスレッド対応

・　付属 API は、マルチスレッドあるいはマルチプロセスで利用できます。これにより、複数の文書を同時に印刷するような処理が可能です。ただし、マルチスレッドでは、各スレッドで印刷処理を行う場合は、あらかじめ、モデル名が「Antenna House PDF Driver 8.0」であるプリ

ンターを複数個用意する必要があります。マルチプロセス利用においては、それぞれのプロセスの起動ユーザー名が同一になるケースでは、あらかじめ、プリンターを複数個用意する必要があります。

・ スレッド１つにプリンター１つを割り当てて印刷を行います。このため、付属 API により出力先や設定編集等の処理を行う場合、最初に、プリンター選択処理（SelectPrinter(()）が必要となります。

・ １つのスレッドから同時に２つの印刷処理が行われないよう、制御する必要があります。

・ Windows Server のターミナルサービスなどの Server - Client システムで、Server 上の付属 API を使用したプログラムを複数の Client ユーザーが使用する場合に、Server 上の 1 個のプリンターで処理を行うことができます。ただし、同一名の複数の Client ユーザーからの利用があるケースでは、1 個のプリンターでは処理を行うことはできません。あらかじめ、モデル名が「Antenna House PDF Driver 8.0」であるプリンターを複数個用意してください。

２．動作環境

オペレーティングシステム(OS) ※3	Microsoft® Windows 11/10※1　各日本語版 Microsoft® Windows Server 2025/2022 各日本語版 ※2
CPU	32bit：Intel Core2、Atom プロセッサ又は上位互換のプロセッサ。2.0GHz 以上推奨 64bit：AMD64 互換のプロセッサ。2.0GHz 以上推奨
RAM	上記 OS が必要とする最低メモリに加えて 1GB 以上のメモリ
必要なランタイムライブラリ	Microsoft Visual　C++ 2015-2022 再頒布パッケージ ※4
.NET Framework （出力先指定 API の.NET）	.NET Framework4.6.2 以上
必要なソフトウェア	Antenna House PDF Driver V8.0

※1　Windows 10 S モードには対応していません。
※2　Windows Server については GUI 環境で検証しています。
※3　ただし、印刷を行うソフトウェアの動作環境に準じます。
※4　PDF Driver の開発は、Microsoft Visual Studio 2015 で行っています。このため、Microsoft Visual C++ 2015-2022 再頒布パッケージのインストールが必要となります。最新版をご使用ください。動作環境にランタイムコンポーネントが存在しない場合、PDF Driver V8.0 のインストール処理開始時にランタイムコンポーネントのセットアップが起動されます。

３．開発環境

３.１.　出力先指定 API

３.１.１.　インターフェース

３.１.１.1.C++

ネームスペース：DrvSet
ヘッダーファイル：DrvSet.h
ライブラリーファイル：AH_DrvSet80.lib、drvselector80AH.lib
DLL ファイル：AH_DrvSet80.dll
lib ファイル、dll ファイルは 32bit 用、64bit 用それぞれにあります。

３.１.１.2..NET Framework

ネームスペース：AntennaHouse.PDFDriver.DS80
DLL ファイル：AH_DrvSetNet80.dll、AH_DrvSet80.dll
dll ファイルは 32bit 用、64bit 用それぞれにあります。

３.１.２.　コンパイラ

３.１.２.1.C++

Microsoft Visual Studio 2015[1]でビルドされています。
呼び出し側のプログラムは、互換性あるコンパイラ使用してください。

３.１.２.2..NET Framework

Microsoft Visual Studio 2015 でビルドされています。
呼び出し側のプログラムは、互換性あるコンパイラ使用してください。

[1] Microsoft Visual Studio については、最新の Service Pack を適用されることをお勧めします。

３.２.　設定編集 API

３.２.１.　インターフェース

３.２.１.1.C++

ネームスペース：PDFSetFileModify
ヘッダーファイル：CEditSettingFile.h
ライブラリーファイル：EditPS80.lib、drvselector80AH.lib
DLL ファイル：EditPS80.dll
dll ファイルは 32bit 用、64bit 用それぞれにあります。

３.２.２.　コンパイラ

３.２.２.1.C++

Microsoft Visual Studio 2015 でビルドされています。
呼び出し側のプログラムは、互換性あるコンパイラ使用してください。

３.３.　32bit と 64bit の区別

出力先指定 API、設定編集 API のモジュールファイルは、32bit API 用と 64bit API 用が区別されています。
32bit で動作するプログラムを開発・実行する場合は 32bit API 用のモジュールを、64bit で動作するプログラムを開発・実行する場合は 64bit API 用のモジュールをそれぞれご使用ください.

４．制限事項

４.１.　付属 API の制限事項

1. 付属 API は、PDF Driver の出力先の指定や印刷設定を編集る API です。印刷を行うアプリケーションの印刷処理は制御しません。印刷を行うアプリケーションの仕様によっては、付属 API で出力先を指定していても PDF Driver のファイル保存ダイアログが表示されるなど、期待される動作と異なる場合があります。

2. PDF Driver V8.0 は、PDF Driver V7.5 以前の付属 API では制御できません。

3. PDF Driver V8.0 の付属 API では、PDF Driver V7.5 以前の PDF Driver を制御することはできません。

4. 付属 API を利用して印刷を行うプログラムが動作している環境では、付属 API 以外から PDF Driver V8.0 が使用されないようにしてください。

5. プログラムで「通常使うプリンター」を変更し、Windows API の「ShellExecute」などを使用してファイルに関連付けられたアプリケーションで印刷させる場合、印刷ジョブが投入される前に他のスレッドから「通常使うプリンター」が変更されると、アプリケーションが変更されたプリンターを使ってしまうため、プログラムでは期待通りの印刷処理が行われません。アプリケーションが使用するプリンターを外部から指定できる場合を除き、「通常使うプリンター」の変更から印刷ジョブが投入されるまでの処理は、セッションレベルで CriticalSection に入れる必要があります。

6. **付属 API を利用して印刷を行うプログラムが、ひとつのユーザー環境で複数同時に動作する場合は、あらかじめ、モデル名が「Antenna House PDF Driver 8.0」であるプリンターを複数個用意してください。**プログラムでは、印刷処理実行前に「SelectPrinter」によるプリンター選択処理を行ってください。

7. **付属 API を利用した複数スレッド構成のプログラムにおいて各スレッドで印刷を行う場合、あらかじめ、モデル名が「Antenna House PDF Driver 8.0」であるプリンターを複数個用意してください。**プログラムでは、印刷処理実行前に「SelectPrinter」によるプリンター選択処理を行ってください。

8. 付属 API を利用した複数スレッド構成のプログラムにおいて各スレッドで印刷を行う場合、１つのスレッドに１つのプリンターを割り当てて印刷を行うようにしてください。

9. **１つのスレッドから同時に２つの印刷処理が行われないよう、制御する必要があります。**たとえば、

```
{
  ...省略...
  int job1 = StartDoc(printerDC1,&docinfo1);
  int job2 = StartDoc(printerDC2,&docinfo2);
  ...印刷...
```

```
  EndDoc(printerDC1);
  EndDoc(printerDC2);
}
```

のように、**１つのスレッド内で２つ以上同時に印刷を行うと、プリンタードライバーは正常に動作しません。**

４.２.　PDF Driver の制限事項

1. PDF Driver の出力結果は、オペレーティングシステム（OS）の種類やバージョンや更新、印刷を行うアプリケーションの更新やバージョンアップにより、出力 PDF の内容が更新／バージョンアップ前の結果と異なる、あるいは元文書のレイアウトやデザインと異なる、PDF 変換できない、Office アドイン変換のオプションが効かない、印刷速度が遅くなるなど印刷品質に影響が出る場合があります。
2. Adobe Acrobat などの PDF ビューアから PDF Driver を指定し印刷する動作、および印刷結果である PDF ファイルについては保証されません。
3. PDF Driver は印刷アプリケーションの動作に依存します。このため、停止しないで動作することは保証していません。停止が問題になるようなサービス等では、上位システムでエラーや停止時の対処また負荷分散などの機能を組み込まれる事を推奨いたします。
4. 印刷処理速度や消費メモリ、出力結果の PDF などについて、特定の性能や品質を保証するものではありません。Adobe Acrobat や Adobe Acrobat の「Adobe PDF」を含む他社製品の PDF 生成ソフトウェアと同等の品質や性能を保証するものではありません。
5. PDF Driver はネットワークプリンターとしてご利用いただくことはできません。
6. １つのスレッドから同時に２つ以上の印刷を行うようなしくみのアプリケーションにおいて２つ以上のファイルを同時に PDF Driver に対して印刷すると PDF Driver は正しく動作しません。
7. 保存ダイアログで「キャンセル」を行った場合、PDF Driver は印刷処理を続行します。最後に PDF ファイルを出力しないことで「キャンセル」とする動作仕様となっています。このため、印刷内容によりキャンセルされるまでに時間がかかる場合があります。
8. PDF Driver V8.0 で印刷中にエラーが発生し印刷が中止された場合、PDF Driver V8.0 用の一時フォルダ「C:¥Users¥ログインユーザー名¥AppData¥Local¥Temp¥AHPD80_temp」に拡張子「.tmp」のファイルが残る場合があります。
9. Windows のプリンター一覧に表示されるプリンター名を変更しないでください。一部の機能が正常に動作しなくなるおそれがあります。
10. PDF Driver を使用した場合に作成可能な PDF バージョンは、PDF1.3～PDF1.7 と PDF2.0 です。
11. 縦書きや文字が回転されている場合、フォントを埋め込む設定で印刷を行ってください。PDF にフォントが埋め込まれていないとき、文字が元文書とは異なる方向に回転したり、表示位置がずれる場合があります。
12. 縦書き文字の出力対応は、MS ゴシックと Arial Unicode MS を基準としています。
13. 対応フォントは、True Type、Open Type、Type1 フォントです。Windows で使用可能なフォント以外については動作保証されません。また、フォントの仕様に沿っていないフォントについては動作保証されません。
14. Open Type Font では、フォントによって縦書き記号が回転する場合があります。
15. Open Type Font をフォント作成ツール「FontForge」で True Type Font に変換したフォントは正しく変換されません。
16. 透過画像や図形が含まれているとき、出力された PDF では黒く塗りつぶされた表示になる場合があります。
17. 透過画像や図形が含まれているとき、PDF/A や PDF/X 設定時に PDF が出力されない、あるいは、ファイルサイズが 0KB である PDF ファイルが残る場合があります。
18. 画像や図形のグラデーションやパターンが元文書の通りに変換されない場合があります。

19. Excel ファイルにおいて、「ページ設定」の印刷品質（解像度）設定値の変更を行った場合、出力される PDF ファイルの結果が印刷プレビューと異なり、行やページが増減する場合があります。PDF Driver で出力する場合、あらかじめ、PDF Driver を選択した状態でシートのレイアウトを行ってください。
20. プリンターが出力するときのピクセル数は、「用紙サイズ（インチ換算）×解像度」で決まります。この値が非常に大きくなる設定のときに、印刷が行われず PDF ファイルが出力されない場合があります。そのような場合は、用紙サイズを小さくするか、あるいは、解像度を低い値に設定してください。設定値の目安は、「用紙サイズ（縦）×解像度」あるいは「用紙サイズ（横）×解像度」が 16bit の上限である「65535」より小さくなる値です。（参考：3600 dpi なら約 460mm、600 dpi なら約 2750mm が上限値となります）
21. PDF Driver は、Modern UI に対応していません。Modern UI を持つソフトウェアから印刷を行った場合、ユーザーインターフェースが期待される動作をせず、ファイル保存ダイアログはデスクトップ上に表示されます。
22. PDF Driver は、Windows のサービス（Windows サービス、Internet Information Service など）での利用において、外字出力には対応していません。
23. 透かしの配置を最背面に設定した場合、印刷を行うアプリケーションによっては、透かしが他のデータの背面に配置され見えない状態になることがあります。
24. 透かし文字列のフォントとして「Roman」など fon 形式フォントを選択した場合、PDF 出力に失敗し、PDF が出力されない、もしくは 0 バイトの PDF ファイルが残ります。
25. PDF Driver は印刷ジョブ名を PDF ファイル名としています。Windows においてファイル名に使用できない 9 種類の記号 ￥ / : * ? “ < > | が印刷ジョブ名に含まれている場合、PDF Driver はこれらの記号を「_」（0x5F）に置換し PDF ファイル名とします。
26. PDF Driver は印刷ジョブ名を PDF ファイル名としています。印刷ジョブ名に制御コード（0x01～0x1F）が含まれる場合、これを半角スペースに置換し PDF ファイル名とします。
27. グレースケールあるいはモノクロ変換設定でかつ PDF ファイルを透かしとして指定している場合、透かし部分はグレースケールおよびモノクロ変換は行われません。
28. Web フォントを使用した Web ページの PDF 出力には対応していません。
29. Windows のバリアブルフォントには対応していません。
30. 設定オプション「文中の URL にリンクをつける」について、URL 文字列内に改行が存在したり、印刷を行うアプリケーションの仕様により印刷データとして URL アドレスの終端までをひとまとまりの文字列として扱われていない場合に、正しくリンク設定が行われません。
31. PDF ファイルを「透かし」とする場合、PDF バージョンは「PDF のバージョン」で指定する PDF バージョンと同一、もしくはそれ以下の PDF バージョンのファイルを指定してください。
32. 「PDF のバージョン」に「PDF2.0」を指定したとき、「透かし」として PDF を指定する場合は PDF2.0 のファイルを使用してください。
33. 「PDF のバージョン」に「PDF/A-2b」を指定しカラープロファイルが「CMYK」、かつ「圧縮」タブで「JPEG2000」を指定している場合、JPEG2000 ではなく JPEG 圧縮が行われます。カラープロファイルが「RGB」の場合は JPEG2000 圧縮が行われます。
34. Word と Excel において、ファイルの用紙サイズ設定が「Letter」であるときに「A4」のページサイズで PDF 出力される場合があります。Word／Excel のオプション「詳細設定」において、「A4 または 8.5 x 11 インチの用紙サイズに合わせて内容を調整する」という設定が有効であるとき A4 サイズとなります。Letter サイズで PDF 出力する場合はこの設定を無効にしてください。
35. PowerPoint ファイルのスライドの印刷の向きが「横」の場合、印刷設定の「用紙設定」

で「サイズ指定」したとき、PDF ではページサイズの幅と高さが入れ替わった大きさになります。

36. ２つの画像を SRCAND のラスタオペレーションで印刷する処理に対し、PDF Driver では「Blend Mode」という描画方法で PDF に変換しています。「Blend Mode」は「PDF1.4」以降の仕様であるため、「PDF バージョン」が「PDF1.3」設定では、一方の上に他方がのった状態で出力されます。また、透過が禁止されている「PDF/A-1b」「PDF/X-1a」「PDF/X-3」設定の場合も、一方の上に他方がのった状態で出力されます。

５．付属 API の使い方

５.１.　PDF Driver のインストール

付属 API を利用するプログラムの開発環境あるいは実行環境には、PDF Driver のインストールが必要です。

５.１.１.　PDF Driver のインストール方法

５.１.１.1.インストールを実行する前の注意事項

- インストールは、管理者権限を持つユーザーがログインしている環境で行ってください。
- インストールを行う環境において、すべての印刷が終了していることを確認してください。
- インストールを行う環境において、すべてのアプリケーションが終了していることを確認してください。

５.１.１.2.インストール手順

(1) 製品パッケージ ¥PDFDriver_setup フォルダにある AHPDFDriver80_setup_xxx_MRxx.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) インストールの途中に、Microsoft Visual C++ 2015-2022 再頒布パッケージのインストーラが起動する場合があります。起動した場合は、指示にしたがって必ず再頒布パッケージをセットアップしてください。
(4) インストールが完了すると、「コントロールパネル」の「デバイスとプリンター」に「Antenna House PDF Driver 8.0」のプリンターアイコンが表示されます。
(5) 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¥PDF Driver 8.0

※１
インストール終了後、システムの再起動が促される場合があります。

※２
PDF Driver を複製している場合、改訂版にて上書きインストールを行うと複製したプリンターは削除されます。上書きインストール後、再度、プリンターを複製してください。上書きインストール前と同一のプリンター名で複製すると、プリンターが選択する設定ファイルは上書きインストール前と同じものになります。

５.１.２.　PDF Driver のアンインストール方法

５.１.２.1.アンインストールを実行する前の注意事項

- アンインストールは、管理者権限を持つユーザーがログインしている環境で行ってください。
- アンインストールを行う環境において、すべての印刷が終了していることを確認してください。
- アンインストールを行う環境において、すべてのアプリケーションが終了していることを確認してください。

５.１.２.2.アンインストール手順

(1) 「コントロールパネル」の「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 Driver V8.0」を選択します。
(3) 「アンインストール」をクリックし、指示にしたがってアンインストールを実行します。

※　アンインストール終了後、システムの再起動が必要となる場合があります。

５.２.　API の導入方法

付属 API のモジュールファイルやヘッダーファイルは「製品パッケージ¥付属 API」フォルダにあります。

５.２.１. 開発環境への導入手順

付属 API のモジュールファイルについてはインストーラはありません。
「付属 API」フォルダごと、任意の場所に配置してください。

５.２.１.1.「出力先指定 API」C++ API での開発

・　【実行ファイル】: AH_DrvSet80.dll
API の実行ファイルは、32bit 用と 64bit 用でファイルが異なります。
32bit プログラム実行 :「bin32」フォルダ内のファイル
64bit プログラム実行 :「bin64」フォルダ内のファイル

・　【インクルードファイル】: DrvSet.h、AHPD_ErrorCode.h
プロジェクトでは、インクルードディレクトリに下記の２つのフォルダパスを追加します。
　付属 API¥Include
　付属 API¥Include¥DrvSetCP
ソースコードにおいて、「DrvSet.h」と「AHPD_ErrorCode.h」を Include してください。

・　【ライブラリファイル】: AH_DrvSet80.lib、drvselector80AH.lib
プロジェクトでは、ライブラリディレクトリに「付属 API¥lib32」または「付属 API¥lib64」フォルダパスを追加します。
ライブラリファイルは 32bit 用と 64bit 用は異なるファイルです。開発を行う bit 数に適切なライブラリファイルをリンクしてください。
32bit プログラム開発 :「lib32」フォルダ内のファイル
64bit プログラム開発 :「lib64」フォルダ内のファイル

５.２.１.2.「出力先指定 API」.NET API での開発

・　【プロジェクトの「参照」に追加するファイル】: AH_DrvSetNet80.dll
プロジェクトでは、.NET API 用のモジュールファイルを参照に追加します。
32bit 用と 64bit 用は区別されています。開発を行う bit 数に適切なモジュールファイルを参照に追加してください。
32bit プログラム開発・実行 :「bin32」フォルダ内のファイル
64bit プログラム開発・実行 :「bin64」フォルダ内のファイル

・　【実行ファイル】: AH_DrvSetNet80.dll、AH_DrvSet80.dll
.NET API は C++ API のラッパーとして開発されています。API 動作のためには C++ API 実行ファイル「AH_DrvSet80.dll」が必要です。.NET API 用モジュールファイルと同じ場所、あるいは、環境変数「PATH」に設定されているフォルダパスに配置してください。

５.２.１.3.「設定編集 API」C++ API での開発

・　【実行ファイル】: EditPS80.dll

API の実行ファイルは、32bit 用と 64bit 用でファイルが異なります。
32bit プログラム実行 :「bin32」フォルダ内のファイル
64bit プログラム実行 :「bin64」フォルダ内のファイル

・　【インクルードファイル】: CEditSettingFile.h、AHPD_ErrorCode.h

プロジェクトでは、インクルードディレクトリに下記の２つのフォルダパスを追加します。
付属 API¥Include
付属 API¥Include¥EditSettings
ソースコードにおいて、「DrvSet.h」と「AHPD_ErrorCode.h」を Include してください。

・　【ライブラリファイル】: EditPS80.lib、drvselector80AH.lib

プロジェクトでは、ライブラリディレクトリに「付属 API¥lib32」または「付属 API¥lib64」フォルダパスを追加します。
ライブラリファイルは 32bit 用と 64bit 用は異なるファイルです。開発を行う bit 数に適切なライブラリファイルをリンクしてください。
32bit プログラム開発 :「lib32」フォルダ内のファイル
64bit プログラム開発 :「lib64」フォルダ内のファイル

５.２.２. 実行環境での API 実行ファイルの配置

付属 API の実行ファイル（DLL）のは、以下の a)、b)のどちらかに配置してください。

a) プログラム実行ファイルと同じ場所に配置
b) プログラム実行ファイルが参照できる場所に配置（＝環境変数「PATH」が通っている場所）

６．ライセンスファイルについて

６．１．　PDF Driver 製品のファイル名と配置場所

６．１．１．　ライセンスファイル名

ファイル名：ahpdfdriver.lic

６．１．２．　ライセンスファイルの場所

PDF Driver V8.0 のインストールフォルダ

６．２．　PDF Driver API 製品のファイル名と配置場所

６．２．１．　ライセンスファイル名

ファイル名：pdalic.dat

６．２．２．　ライセンスファイルの場所

PDF Driver V8.0 のインストールフォルダ¥lic

６．３．　正規利用方法

ライセンスファイルは、PDF Driver のインストーラによりインストールされます。
インストールされるライセンスファイルは、インストール後 30 日間有効な評価用ライセンスです。
製品ご購入後、弊社からライセンスファイルを送付します。このライセンスファイルに入れ替えることにより、正規利用できるようになります。

６．４．　保守期限と付属 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ついては、アップデートしてご使用になると付属 API は評価版として動作します。このため、出力される PDF のページには評価用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。

６．５．　ライセンスファイルの入れ替えについて

評価版から正規版に移行する場合や保守契約を更新した場合、弊社から新しい正規のライセンスファイルを送付します。
新しいライセンスファイルに入れ替え、ご利用ください。
ライセンスファイルの入れ替えは、付属 API および PDF Driver 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。
ファイル入れ替え後にコンピュータの再起動は必要ありません。

６.６.　ライセンスファイルの削除

PDF Driver をアンインストールしてもライセンスファイルは削除されません。

７．付属 API の詳細

７．１．　プリンター選択処理

７．１．１．　プリンターの選択：SelectPrinter の仕様

プリンター選択関数「SelectPrinter」は、出力先指定 API、設定編集 API のそれぞれに存在します。出力先指定や設定編集の処理をどのプリンターに対して行うのかを決める関数です。選択対象は、モデル名が「Antenna House PDF Driver 8.0」であるプリンターです。
複数のスレッドあるいは複数のプロセスから同時に PDF Driver V8.0 で印刷処理をさせる場合、プログラムのコードでは、印刷実行前に「SelectPrinter」をコールするようにしてください。

SelectPrinter の仕様	
SelectPrinter(NULL)	引数が NULL の場合は、プリンター名「Antenna House PDF Driver 8.0」を選択します。
SelectPrinter(L"")	引数が空文字列の場合は、SelectPrinter 側で処理が可能な状態にあるプリンターを選択します。プリンターが見つからない場合は、エラーになります。プリンターの選択に成功した場合は、「GetSelectedPrinter」によりプリンター名を取得することができます。
SelectPrinter("[プリンター名]")	引数に名称が指定されている場合、すでにその名称のプリンターが存在するときにはそれを選択状態にします。存在しないときには、SelectPrinter 側でプリンターを作成します。 **プログラム側でプリンターを作成する場合は、プログラム自体、あるいは、プログラムを実行するユーザーに、プリンターが作成可能な権限が必要となります。** SelectPrinter 側でプリンターを作成した場合、印刷終了後にそのプリンターは削除されます。
プログラム側で SelectPrinter の呼び出しが行われなかった場合	「SelectPrinter(L"")」と同じ仕様で動作します。

プリンターが「選択状態」※になると、他のプロセスやスレッドからはそのプリンターを付属 API では使用できなくなりますが、付属 API 以外からは使用できます。
付属 API からプリンターを利用するプログラムが動作している間は、付属 API 以外からそのプリンターが使用されないようにしてください。

※　ここで言う「選択状態」とは、システムのプリンターとして選択されるという意味ではありません。出力先指定 API あるいは設定編集 API が、API から利用するプリンターとしてみなす処理です。

7.1.2. 出力先指定 API の「PDFDriverSetting::SelectPrinter」と設定編集 API の「CEditSettingFile::SelectPrinter」の同時使用について

出力先指定 API の「PDFDriverSetting::SelectPrinter」と設定編集 API の「CEditSettingFile:::SelectPrinter」は同じプログラムの中で同時に使用しても問題はありません。
同じインスタンス内である場合、「PDFDriverSetting::SelectPrinter("")」（空文字列指定の場合）が成功してプリンター選択を行った後に「CEditSettingFile::SelectPrinter("")」を呼び出したときは、「PDFDriverSetting::SelectPrinter("")」で選択されたプリンターが選択状態であることに変わりはありません。

【注意点】
「PDFDriverSetting::SelectPrinter」と「CEditSettingFile::SelectPrinter」で異なるプリンターは選択しないでください。
たとえば、「PDFDriverSetting::SelectPrinter("")」でプリンター選択し出力先を指定したあとに「CEditSettingFile::SelectPrinter("AHDriver8(2)")」としてプリンター名を指定して呼び出し設定編集して印刷処理を実行した場合、そのインスタンスの印刷は「AHDriver8(2)」で行われることになります。設定編集は有効ですが、指定した出力先は無効となります。

7.1.3. SelectPrinter のエラーコード「4」について

- 同じプリンター名を SelectPrinter するスレッドが複数動作しているときに発生した場合、他のスレッドの処理が終了するのを待機するようにしてください。

- SelectPrinter に空文字列を指定するスレッドが複数動作しているときに発生した場合、モデル名「Antenna House PDF Driver 8.0」のプリンターの複製を追加してください。

7.1.4. 選択状態の解放：ReleasePrinter

プログラムで使用が終わったプリンターは、「ReleasePrinter」で選択状態を解放します。
「ReleasePrinter」は、デストラクタで呼ばれるよう設計されています。
「SelectPrinter」が作成したプリンターは、自動的に削除されます。

７.２.　サンプルコード

サンプルコードを用意しました。ご参考ください。
サンプルコードの場所：付属 API¥SampleCode

- EditSettingSample
  出力先指定 API と設定編集 API の C++サンプルプロジェクトです。
  exe の引数に指定した入力ファイルと同じ場所に Antenna House PDF Driver 8.0 を使って PDF を出力します。「c:¥透かし付き(社外秘).printSetting3」を読み込み、文書情報と透かしの設定編集を行っています。
  Microsoft Visual Studio 2015 C++　コンソールアプリケーションプロジェクトです。

- DrvSetNetSample
  出力先指定 API の C#サンプルプロジェクトです。
  PrintDocument()を利用して、Antenna House PDF Driver 8.0 で印刷を行います。
  Microsoft Visual Studio 2015 C#　コンソールアプリケーションプロジェクトです。

７.３.　プログラミングの注意事項

出力先指定 API や設定編集 API によって指定された PDF の出力先や印刷設定を印刷処理で正しく反映させるためには、印刷が終了するまでそれぞれのクラスのインスタンスが有効でなければなりません。
印刷終了後は、インスタンスを破棄し、印刷処理のプロセスを終了することができます。

７.４.　出力先指定 API

７.４.１.　Namespace 名と Class 名

インターフェース		名称
C++	Namespace	DrvSet
	Class	PDFDriverSetting
.NET	Namespace	AntennaHouse.PDFDriver.DS80
	Class	PDFDriverSetting

７.４.２.　関数仕様

以下は、出力先指定 API(C++)の関数仕様です。

７.４.２.1.setOutputPDFFilenameFullPath

定義	bool setOutputPDFFilenameFullPath(const char* path) bool setOutputPDFFilenameFullPath(const wchar_t* path)
引数	path：出力する PDF ファイルのフルパス名
戻り値	true=成功　／　false=失敗
説明	出力先となる PDF ファイルパスを指定します。UNC（Universal Naming Convention）パス名での指定も可能です。 PDF へ出力する際、指定されたファイル名で上書きするためプログラム側でファイルの存在チェックが必要です。 また、出力先フォルダは存在していなければなりません。

７.４.２.2.setOutputPDFFilenameClear

定義	bool setOutputPDFFilenameClear()
引数	なし
戻り値	true=成功　／　false=失敗
説明	指定した出力先 PDF ファイルパスをクリアします。

7.4.2.3.setSettingsFile

定義	bool setSettingsFile(const char* settingsFile) bool setSettingsFile(const wchar_t* settingsFile)
引数	settingsFile ：PDF 出力設定ファイル名（フルパス名またはパスなしのファイル名）
戻り値	true=成功 ／ false=失敗
説明	PDF 出力設定ファイル名を指定します。パスを含まないファイル名が指定された場合は、 1) PDF Driver V8.0 のインストールフォルダ¥Settings 2)「CSIDL_APPDATA」で取得されるフォルダ¥AntennaHouse¥PDF_Driver¥<パージョン>¥CustomSettings の順に検索します。

7.4.2.4.setSettingsFileClear

定義	bool setSettingsFileClear()
引数	なし
戻り値	true=成功 ／ false=失敗
説明	指定した PDF 出力設定ファイル名をクリアします。

7.4.2.5.setOpenPDF

定義	bool setOpenPDF(bool openFlag)
引数	openFlag ：rue=オープンする ／ false=オープンしない
戻り値	true=成功 ／ false=失敗
説明	PDF 出力後に PDF 表示するかどうかを指定します。 本関数による設定は、印刷設定の「作成後 PDF を表示」の設定値より優先されます。

7.4.2.6.SelectPrinter

定義	int SelectPrinter(const wchar_t* printerName)
引数	printerName： ・NULL：「Antenna House PDF Driver 8.0」プリンターを選択 ・空文字列：処理可能状態にあるプリンターを自動で選択 ・プリンター名（NULL 文字終端で 32 文字(= 64 バイト)まで）
戻り値	AHPD_ErrorCode.h を参照してください。
説明	PDF の出力先指定を行うプリンターを選択します。

7.4.2.7.ReleasePrinter

定義	int ReleasePrinter()

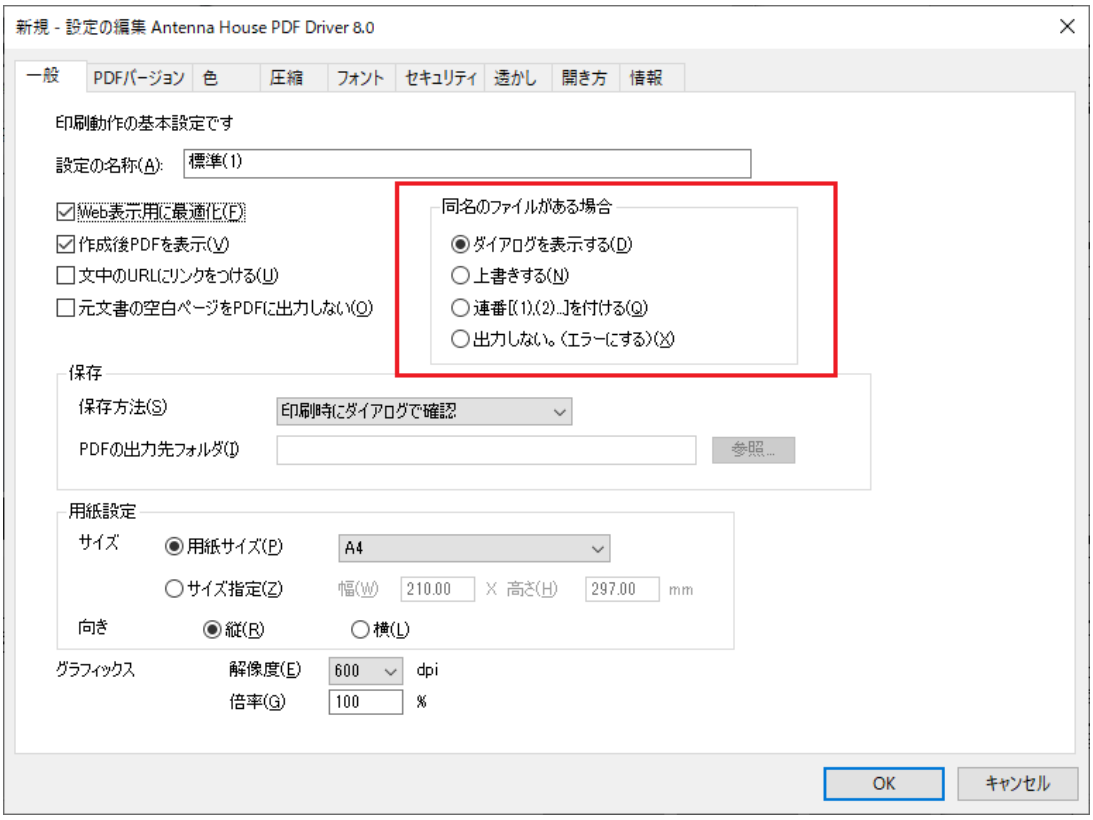
引数	なし
戻り値	AHPD_ErrorCode.h を参照してください。
説明	SelectPrinter で選択したプリンターを解放し、他のプロセスやスレッドでも使えるようにします。 この関数は、デストラクタで呼ばれます。

７.４.２.8.GetSelectedPrinter

定義	int GetSelectedPrinter(wchar_t* buf, size_t bufsizeInWORD)
引数	buf：選択されているプリンター名が格納されるバッファ。 bufsizeInWORD：buf のバッファ文字数。最低 32 文字（=64 バイト）確保してください。
戻り値	AHPD_ErrorCode.h を参照してください。
説明	SelectPrinter で選択したプリンター名をバッファ（buf）に返します。

７.４.３.　　API による出力先指定と「同名のファイルがある場合」の設定について

印刷設定の「一般」タブに「同名のファイルがある場合」の設定があります。
出力先にすでに同じ名前のファイルが存在する場合の動作オプションです。



出力先指定 API で出力先を指定し印刷処理を行うとき、「同名のファイルがある場合」の設定値に関係なく、**出力先に同名ファイルが存在する場合は上書き動作**をします。
上書きできない場合、エラーとなります。

７.５.　設定編集 API

７.５.１.　Namespace 名と Class 名

インターフェース		名称
C++	Namespace	PDFSetFileModify
	Class	CEditSettingFile

７.５.２.　関数仕様

以下は設定編集 API の関数仕様の説明です。API リファレンスも合わせてご参考ください。

７.５.２.1.load

定義	int load(const char* settingFile) int load(const wchar_t* settingFile)
引数	settingFile：読み込む出力設定ファイル名（フルパス名またはパスなしのファイル名）
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	指定された出力設定ファイルを読み込みます。パスを含まないファイル名が指定された場合は、 1) PDF Driver V8.0 のインストールフォルダ¥Settings 2)「CSIDL_APPDATA」で取得されるフォルダ¥AntennaHouse¥PDF_Driver¥<パージョン>¥CustomSettings の順に検索します。

７.５.２.2.save

定義	int save()
引数	なし
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	編集した設定内容をロードした出力設定ファイル名で上書き保存します。

７.５.２.3.saveAs

定義	int　saveAs(const char* fileName) int　saveAs(const wchar_t* fileName)
引数	fileName：保存先の設定ファイルのフルパス名
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	編集した設定内容に名前を付けて保存します。

７．５．２.4.getCommon

定義	int getCommon(PDFCommon * pdfCommonSet)
引数	pdfCommonSet：取得した値を格納するための「一般」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「一般」設定を取得します。 PDFCommon 構造体定義は「PDFTableSet.h」を参照してください。

７．５．２.5.setCommon

定義	int setCommon(const PDFCommon * pdfCommonSet)
引数	pdfCommonSet：格納する「一般」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「一般」設定を格納します。 PDFCommon 構造体定義は「PDFTableSet.h」を参照してください。

７．５．２.6.getPDFVersion

定義	int getPDFVersion(PDFVersion* value)
引数	value：取得した値を格納するための「PDF バージョン」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「PDF バージョン」設定を取得します。 PDFVersion 構造体定義は「PDFTableSet.h」を参照してください。

７．５．２.7.setPDFVersion

定義	int setPDFVersion(const PDFVersion* value)
引数	value：格納する「PDF バージョン」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「PDF バージョン」設定を格納します。 PDFVersion 構造体定義は「PDFTableSet.h」を参照してください。

７.５.２.8.getCompress

定義	int getCompress(PDFCompress * pdfCompressSet)
引数	pdfCompressSet: 取得した値を格納するための「圧縮」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「圧縮」設定を取得します。 PDFCompress 構造体定義は「PDFTableSet.h」を参照してください。

７.５.２.9.setCompress

定義	int setCompress(const PDFCompress * pdfCompressSet)
引数	pdfCompressSet：格納する「圧縮」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「圧縮」設定を格納します。 PDFCompress 構造体定義は「PDFTableSet.h」を参照してください。

７.５.２.10.getFont

定義	PDFFont*　getFont()
引数	なし
戻り値	「フォント」設定の構造体ポインタ
説明	「フォント」設定を取得します。 取得したポインタや「フォント」設定内のポインタを上位で開放する必要はありません。 PDFFont 構造体定義は「PDFTableSet.h」を参照してください。

７.５.２.11.setFont

定義	int setFont(const PDFFont * pdfFontSet)
引数	pdfFontSet：格納する「フォント」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「フォント」設定を格納します。 PDFFont 構造体定義は「PDFTableSet.h」を参照してください。

７.５.２.12.getSecurity

定義	int getSecurity(PDFSecurity * pdfSecuritySet)
引数	pdfSecuritySet:取得した値を格納するための「セキュリティ」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「セキュリティ」設定を取得します。 PDFSecurity 構造体定義は「PDFTableSet.h」を参照してください。

７.５.２.13.setSecurity

定義	int setSecurity(const PDFSecurity * pdfSecuritySet)
引数	pdfSecuritySet：格納する「セキュリティ」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「セキュリティ」設定を格納します。 PDFSecurity 構造体定義は「PDFTableSet.h」を参照してください。

７.５.２.14.getWaterMark

定義	int getWaterMark(PDFWaterMark * pdfWaterMarkSet)
引数	pdfWaterMarkSet：取得した値を格納するための「透かし」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「透かし」設定を取得します。 PDFWaterMark 構造体定義は「PDFTableSet.h」を参照してください。

７.５.２.15.setWaterMark

定義	int setWaterMark(const PDFWaterMark * pdfWaterMarkSet)
引数	pdfWaterMarkSet：格納する「透かし」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「透かし」設定を格納します。 PDFWaterMark 構造体定義は「PDFTableSet.h」を参照してください。

７．５．２.16.getOpenMode

定義	int getOpenMode(PDFOpenMode * pdfOpenModeSet)
引数	pdfOpenModeSet：取得した値を格納するための「開き方」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「開き方」設定を取得します。 PDFOpenMode 構造体定義は「PDFTableSet.h」を参照してください。

７．５．２.17.setOpenMode

定義	int setOpenMode(const PDFOpenMode * pdfOpenModeSet)
引数	pdfOpenModeSet：格納する「開き方」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「開き方」設定を格納します。 PDFOpenMode 構造体定義は「PDFTableSet.h」を参照してください。

７．５．２.18.getInformation

定義	int getInformation(PDFInformation * pdfInformationSet)
引数	pdfInformationSet：取得した値を格納するための「情報」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「情報」設定を取得します。 PDFInformation 構造体定義は「PDFTableSet.h」を参照してください。

７．５．２.19.setInformation

定義	int setInformation(const PDFInformation * pdfInformationSet)
引数	pdfInformationSet：格納する「情報」設定の構造体ポインタ
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	「情報」設定を格納します。 PDFInformation 構造体定義は「PDFTableSet.h」を参照してください。

７．５．２.20.getName

定義	int getName(wchar_t* buf, size_t bufsize)
引数	buf：取得した設定表示名を格納する文字列変数のポインタ bufsize ：buf のサイズ。WORD 単位で NULL 文字分を含めてください。
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	印刷設定ダイアログ上で表示される設定の表示名を取得します。設定ファイル名とは一致しません。表示名がない場合は空文字列が取得されます。

７．５．２.21.setName

定義	int setName(const wchar_t* value)
引数	value：設定する設定表示名の文字列変数のポインタ。終端は NULL 文字にしてください。 指定可能な文字数は、UNICODE で 100 文字(200 バイト)までです。以降は切り捨てられます。
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	印刷設定ダイアログ上で表示される設定の表示名を指定します。設定ファイル名とは一致しません。ファイル名に使用できない文字や記号の指定が可能です。 プログラムで setName を行わない場合、あるいは、setName で空文字列や NULL を指定した場合は、印刷設定ダイアログ上では拡張子を除く設定ファイル名で表示されます。

７．５．２.22.initSetting

定義	int initSetting()
引数	なし
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	設定内容を規定値に戻します。

７．５．２.23.sendToDriver

定義	int sendToDriver ()
引数	なし
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	設定内容を PDF Driver V6 へ渡します。 設定内容を印刷処理で正しく反映させるためには、印刷が終了するまで sendToDriver を行った CEditSettingFile のインスタンスが有効でなければなりません。 PDF Driver へ渡した設定内容は、PDF Driver での印刷後、本クラス消滅後、もしくは clearSettings メソッド呼出し後にクリアされます。

７．５．２.24.clearSettings

定義	int clearSettings ()
引数	なし
戻り値	エラーコード：Settingerror.h を参照してください。 0=正常
説明	PDFDriver V6 へ渡した設定内容をクリアします。

７．５．２.25.SelectPrinter

定義	int SelectPrinter(const wchar_t* printerName)
引数	printerName： ・NULL：「Antenna House PDF Driver 8.0」プリンターを選択 ・空文字列：処理可能状態にあるプリンターを自動で選択 ・プリンター名（NULL 文字終端で 32 文字(= 64 バイト)まで）
戻り値	AHPD_ErrorCode.h を参照してください。
説明	印刷設定編集を行うプリンターを選択します。

７．５．２.26.ReleasePrinter

定義	int ReleasePrinter()
引数	なし
戻り値	AHPD_ErrorCode.h を参照してください。
説明	SelectPrinter で選択したプリンターを解放し、他のプロセスやスレッドでも使えるようにします。 この関数は、デストラクタで呼ばれます。

７.５.２.27.GetSelectedPrinter

定義	int GetSelectedPrinter(wchar_t* buf, size_t bufsizeInWORD)
引数	buf：選択されているプリンター名。入力値は無視されます。 bufsizeInWORD：buf のバッファ文字数。最低 32 文字（=64 バイト）確保してください。
戻り値	AHPD_ErrorCode.h を参照してください。
説明	SelectPrinter で選択したプリンター名をバッファ（buf）に返します。

７.５.２.28.getNotificationID

定義	int getNotificationID(uintptr_t* id)
引数	id：識別番号を格納する変数へのポインタ
戻り値	EDIT_PARAM_ERR(0x130) ＝ 引数エラー その他の 0 以外の値 ＝ その他のエラー 0 ＝ 正常
説明	印刷の開始や終了を通知するための識別番号を取得します。

７.５.２.29.setNotificationID

定義	int setNotificationID(uintptr_t id)
引数	id：通知を受け取るさいの識別番号 0 ＝ 通知は行わない 0xFFFFFFFF 、 0xFFFFFFFFFFFFFFFF(64bit の み) ＝ GetCurrentThreadId()の値を識別番号とする
戻り値	EDIT_PARAM_ERR(0x130) ＝ 引数エラー その他の 0 以外の値 ＝ その他のエラー 0 ＝ 正常
説明	印刷の開始や終了を通知するための識別番号を設定します。 複数の印刷ジョブで同じ設定を使用すると id が重なります。印刷ジョブごとに id を変更、再設定してください。 **識別番号設定を 64bit プロセスで行い、Windows メッセージ受信を 32bit プロセスで行うと、識別番号の上位 32bit が欠落します。**

７.５.２.30.getNotificationMessageString

定義	int getNotificationMessageString (messageReason reason, wchar_t* buf, size_t bufsize)

引数	reason : 文字列を取得する Windows メッセージの種類。次の４種類からひとつを指定します。
	num messageReason{
	PrintStart = 0, // 印刷開始
	PageStart = 1, // ページ開始
	PageEnd = 2, // ページ終了
	PrintEnd = 3, // 印刷終了
	};
	buf : 文字列を格納する変数のポインタ
	bufsize : buf のサイズ。WORD 単位で NULL 文字分を含めてください。

戻り値	EDIT_PARAM_ERR(0x130) = 引数エラー EDIT_MEMORY_ERR(0x140) = 文字列処理エラー その他の 0 以外の値 = その他のエラー 0 = 正常
説明	setNotificationMessageString()で指定された文字列を取得します。

７.５.２.31.setNotificationMessageString

定義	int setNotificationMessageString (messageReason reason, const wchar_t* str)

引数	reason : 文字列を設定する Windows メッセージの種類。次の４種類からひとつを指定します。
	num messageReason{
	PrintStart = 0, // 印刷開始
	PageStart = 1, // ページ開始
	PageEnd = 2, // ページ終了
	PrintEnd = 3, // 印刷終了
	};
	str : Windows メッセージを受信するための文字列。NULL 終端で NULL を含み 31 文字以内。超過する場合はエラーになります。
	空文字列(L"")、NULL が指定された場合、指定がない場合は、メッセージは送信されません。

戻り値	EDIT_PARAM_ERR(0x130) = 引数エラー EDIT_MEMORY_ERR(0x140) = 文字列処理エラー その他の 0 以外の値 = その他のエラー 0 = 正常
説明	印刷の開始や終了の Windows メッセージを受け取るための文字列を指定します。この関数で指定した文字列で RegisterWindowMessageW()を行い、Windows メッセージを受信することで、印刷の開始や終了を認識することができます。

７.５.３.　　印刷開始終了通知のしくみ

API では、次の通知を受け取ることができるように設定することができます。

通知の種類	内容
印刷開始	ファイルの印刷開始
ページ開始	ページの印刷開始
ページ終了	ページの印刷終了
印刷終了	ファイルの印刷終了

設定編集 API の setNotificationMessageString()で指定した文字列で
RegisterWindowMessageW()を行い、Windows メッセージを受信します。
受信したメッセージの内容は次の通りです。

WPARAM	32 -63 bit ※64bit のみ	印刷ジョブ ID
	0 - 31 bit	ページ数（ページの開始/終了の場合） 最初のページは０
		印刷ジョブ ID（印刷開始/終了の場合）
LPARAM	設定編集 API「setNotificationID」で指定した識別番号	

＜注意事項＞

識別番号設定を 64bit プロセスで行い、Windows メッセージ受信を 32bit プロセスで行うと、識別番号の上位 32bit が欠落します。

８．旧バージョンとの違いについて

Antenna House PDF Driver V6.0、 V7.5、V8.0 の付属 API の相違点は以下の通りです。

８.１.　ファイル名について

８.１.１.　出力先指定 API

V6.0	C++	AH_DrvSetCP60.dll, AH_DrvSetCP60.lib
	.NET Framework	AH_DrvSetNet2_60.dll（.NET Framework 2.0/3.0/3.5 用） AH_DrvSetNet4_60.dll（.NET Framework 4.0/4.5 用）
V7.5	C++	AH_DrvSet75.dll, AH_DrvSet75.lib
	.NET Framework	AH_DrvSetNet75.dll
V8.0	C++	AH_DrvSet80.dll, AH_DrvSet80.lib
	.NET Framework	AH_DrvSetNet80.dll

８.１.２.　設定編集 API

V6.0	EditSettingFileCP60AH.dll, EditSettingFileCP60AH.lib
V7.5	EditPS.dll, EditPS.lib
V8.0	EditPS80.dll, EditPS80.lib

８.１.３.　SelectPrinter 処理ライブラリーファイル

V6.0	-
V7.5	drvselector75AH.lib
V8.0	drvselector80AH.lib

８.２.　出力先指定 API について

８.２.１.　ヘッダーファイル名(C++)

V6.0	DrvSetCP.h
V7.5	DrvSet.h
V8.0	DrvSet.h

８.２.２.　Name Space 名(C++)

V6.0	DrvSetCP
V7.5	DrvSet
V8.0	DrvSet

８.２.３.　Name Space 名(.NET Framework)

V6.0	DrvSetNET2_60（.NET Framework 2.0/3.0/3.5 用） DrvSetNET4_60（.NET Framework 4.0/4.5 用）
V7.5	AntennaHouse.PDFDriver.DS75
V8.0	AntennaHouse.PDFDriver.DS80

８.２.４.　対応する.NET Framework バージョン

V6.0	2.0,3.0,3.5,4.0,4.5
V7.5	4.5.2 以降
V8.0	4.6.2 以降

８.３.　設定編集 API について

８.３.１.　ヘッダーファイル名

V6.0	CEditSettingFile.h
V7.5	CEditSettingFile.h
V8.0	CEditSettingFile.h

８.３.２.　Name Space 名

V6.0	PDFSetFileModify
V7.5	PDFSetFileModify
V8.0	PDFSetFileModify

８.３.３.　追加された定義

PDF Driver V7.5/V8.0 で追加された設定に対応するため、定義テーブル（PDFTableSet.h）には以下が追加されています。

定義	内容
enum SAMENAME	印刷設定「一般」タブの「同名のファイルがある場合」ラジオボタンに対応
enum PDFVER のメンバ	印刷設定「PDF バージョン」タブの「PDF のバージョン」コンボボックスに追加された選択肢に対応 PDFVER_X_4(=「PDF/X-4:2008」) ※V7.5,V8.0 PDFVER_A_2B(=「PDF/A-2b」) ※V7.5,V8.0 PDFVER_20(=「PDF2.0」) ※V8.0 を追加
enum COLORMODE	印刷設定「色」タブの「カラー／グレースケール／モノクロ」に対応
enum GRAYDECREASINGMETHOD	印刷設定「色」タブの「グレースケール減色法」に対応
enum MONODECREASINGMETHOD	印刷設定「色」タブの「モノクロ減色法」に対応
enum IMAGEFILTER	印刷設定「色」タブの「モノクロ減色前のフィルタ」に対応
struct PDFColor	印刷設定「色」タブに対応

８.４.　印刷設定ファイルについて

- ファイル拡張子は、V6.0 と V7.5、V8.0 で変更があります。V7.5 と V8.0 のファイル拡張子は同一名です。
- V7.5 は、V6.0 で作成された印刷設定ファイルを読み込むことができます。
- V8.0 は、V6.0 および V7.5 で作成された印刷設定ファイルを読み込むことができます。

	ファイル拡張子	ファイル形式
V6.0	printSetting3	バイナリー
V7.5	ps4	JSON 仕様
V8.0	ps4	JSON 仕様

８.５.　開発環境

V6.0	Microsoft Visual Studio 2010
V7.5	Microsoft Visual Studio 2015
V8.0	Microsoft Visual Studio 2015

９．商標と著作権情報

９.１.　商標情報

Adobe、Acrobat、および Reader は、アドビ システムズ社の米国ならびに他の国における登録商標または商標です。
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。
TrueType は米国その他の国で登録された米国アップルコンピュータ社の商標です。
その他記載されている全ての会社名および製品名は、個々の所有者の登録商標または商標です。

９.２.　第三者製ライブラリー著作権情報

- libtiff

Silicon Graphics has seen fit to allow us to give this work away. It is free. There is no support or guarantee of any sort as to its operations, correctness, or whatever. If you do anything useful with all or parts of it you need to honor the copyright notices. I would also be interested in knowing about it and, hopefully, be acknowledged.

The legal way of saying that is:

Copyright (c) 1988-1997 Sam Leffler
Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE

OF THIS SOFTWARE.

- zlib

(C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

- BOOST

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

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, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

- picojson

Copyright 2009-2010 Cybozu Labs, Inc.
Copyright 2011-2014 Kazuho Oku
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- POCO C++ LIBRARIES

THE BOOST SOFTWARE LICENSE 1.0
Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software,

and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

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, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

※その他、PDF Driver V8.0 インストールフォルダ内のドキュメントをご参照ください。

履歴

日付	修正内容
2020.2.7	初版
2020.3.26	・「6.付属 API の相違点について」：比較対象に V6.0 の記述を追加しました。
2020.8.13	・「5.4.3.API による出力先指定と「同名のファイルがある場合」の設定について」を追加しました。
2021.9.13	・「2.動作環境」を修正しました。
2022.5.2	・「PDF Driver の制限事項」の 1 項めを追記しました。
2022.10.6	・「2.動作環境」を修正しました。 ・全体の構成を見直しました。
2023.3.28	・「4.2.PDF Driver の制限事項」に 1 項追記しました。
2023.6.9	・「9.2.第三者製ライブラリー著作権情報」に「POCO C++ LIBRARIES」の項を追加しました。
2023.11.15	・「2.動作環境」の表の欄外に注意書き「※3」を追加ました。 ・「4.1.付属 API の制限項」を修正ました。 ・「4.2.PDF Driver の制限事項」を修正しました。 ・「7.1.プリンター選択処理」の説明文体裁を修正しました。
2024.3.13	・「2.動作環境」を修正しました。
2024.6.10	・「4.2.PDF Driver の制限事項」の(1)の説明文を修正しました。
2024.10.9	・「4.2.PDF Driver の制限事項」に「36」を追加しました。 ・「5.1.1.PDF Driver のインストール方法」 - 「5.1.1.2.インストール手順」において「※２」を追加しました
2024.10.16	・「7.1.1.プリンターの選択：SelectPrinter の仕様」：「SelectPrinter("[プリンター名]")」を修正しました。
2025.1..20	・「2.動作環境」に「Server 2025」を追加しました。 ・「5.2.モジュールファイルの配置について」を廃止し、「5.2.API の導入方法」の項を追加しました。 ・「6.6.ライセンスファイルの削除」：説明を修正しました。PDF Driver 製品の改訂 20 版以降、ライセンスファイルはアンインストールしても削除されないようになりました。
2025.1.30	・「7.1.2.」の項を追加しました。

Antenna House PDF Driver V8.0

付属 API 説明書 2025.1.30

© Antenna House Inc. 2021-2025 All Rights Reserved.