

PDF Direct™

Version 2.0J



Results Through Focused Technology...

PDF Direct™

Version 2.0J



©1998-2003 Soft Solutions, Inc. All Rights Reserved.

©2003 K's Room. (japanese version)

Published World-Wide by Soft Solutions, Inc.

Soft Solutions, Inc.
2900 Chamblee Tucker Road
Building 12, Suite 200
Atlanta, GA 30341 USA



Results Through Focused Technology...

IMPORTANT-READ CAREFULLY BEFORE OPENING.

By installing this software program, you indicate your acceptance of the following Soft Solutions, Inc. License Agreement.

Software License and Limited Warranty Agreement:

This is a legal agreement between you, the end user, and Soft Solutions, Inc. By opening this package, you are agreeing to be bound to the terms of this agreement. If you do not agree to the terms of this agreement, promptly return the unopened disk package and the accompanying items (including written materials and binders or other containers) to the place you obtained them for a full refund.

Soft Solutions, Inc. Software License

1. GRANT OF LICENSE. Soft Solutions grants to you the right to use one development copy of the enclosed Soft Solutions software program (the "SOFTWARE") on a single terminal connected to a single computer (i.e. with a single CPU), or on a LICENSED COMPUTER NETWORK. Each concurrent user of the SOFTWARE must have exclusive access to a Soft Solutions SOFTWARE manual during his/her use. Soft Solutions, Inc. as Licensor, grants to you, the LICENSEE, a non-exclusive, non-transferable right to use this Software subject to the terms of the license as described in the following sections:

A. You may make backup copies of the Software for your use provided they bear the Soft Solutions, Inc. copyright notice.

B. You may use this software in an unlimited number of custom or 4D- compiled commercial database applications created by the original licensee. No additional product license or royalty is required.

2. COPYRIGHT. The SOFTWARE is owned by Soft

Solutions or its suppliers and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material (e.g., a book or musical recording) except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purposes, or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the written materials accompanying the SOFTWARE.

3. OTHER RESTRICTIONS. You may not rent or lease the SOFTWARE, but you may transfer the SOFTWARE and accompanying written materials on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. You may not reverse engineer, decompile, or decompile the SOFTWARE. If SOFTWARE is an update, any transfer must include the update and all prior versions.

LIMITED WARRANTY. Soft Solutions, Inc. warrants that (a) the SOFTWARE will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt. Any implied warranties on the SOFTWARE are limited to ninety (90) days and one (1) year, respectively. Some states do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

CUSTOMER REMEDIES. Soft Solutions, Inc. entire liability and your exclusive remedy shall be, at Soft Solutions option, either (a) return of the price paid or (b) repair or replacement of the SOFTWARE that does not meet Soft Solutions Limited Warranty and which is returned to Soft Solutions with a copy of your receipt. This Limited Warranty is void if failure of the SOFTWARE has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period of thirty (30) days, whichever is longer.

NO OTHER WARRANTIES. Soft Solutions disclaims all other warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the SOFTWARE, the accompanying written materials, and any accompanying hardware. This limited warranty gives you specific legal rights. You may have others, which vary from state to state.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. In no event shall Soft Solutions or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use this Soft Solutions product, even if Soft Solutions has been advised of the possibility of such damages. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

Governing Law. This entire agreement shall be governed by the laws of the State of Georgia.

PDF Direct™ is a trademark of Soft Solutions, Inc.

Apple is a registered trademark of Apple Computer, Inc.

4th Dimension, 4D Compiler are registered trademarks of ACI and ACI US, Inc.

PDFWriter, Acrobat Exchange and Adobe Acrobat are registered trademarks of Adobe Corporation.

目次

はじめに	6
本ドキュメントにおけるコマンド構文	7
インストール	7
どのプラグインは必要ですか?	7
動作条件	8
概要	8
プラットフォーム別印刷の違い	10
コマンド解説	12
PD_Register	12
PD_IsAvailable	12
PD_GetPrinter	13
PD_SetPrinter	14
PD_GetPath	15
PD_Define	16
PD_GetStatus	17
PD_Cleanup	18
サンプル・アプリケーション	19
準備	19
4D Print Selection Printing	28
4D Print Form Printing	30

はじめに

Portable Document Format (PDF) は、世界で最も一般的に用いられるドキュメントフォーマットであり、オンライン上のドキュメントを見る場合、最も広く用いられているフォーマットである。PDF Direct™ は、4th Dimension のデータベース内から、直接この PDF ファイルを出力できるようにするプラグインである。PDF Direct を用いることで、4D デベロッパーは、適切な仕様にのっとり、プログラムを使って 4D Print ジョブに名前をつけて、Adobe Acrobat “PDF” フォーマットファイルに書き出すことができるのだ。

PDF フォーマットは、読者が、紙媒体、すなわち “ハードコピー” を必要としないケースにおいて、多大な支持を集めている。e-mail と、フリーソフトである Acrobat Reader の組み合わせは抜群で、PDF ドキュメントを e-mail に添付し全世界へ送信することができる。スピーディーに、かつコストを節約しながらに、である。同様に、Fax 送信されたドキュメントと比較し、PDF は、送信によって解像度が低下することはない。作成されたときと全く同じ解像度を、送信先でも再現できるのだ。

PDF は、カラーやグレイスケールをサポートし、一般には、作成者のドキュメントに含まれる情報を損なうことなく、読者の利用するコンピュータのプラットフォームに関わらず利用できるよう保存されたものである。Adobe 社、すなわち、PDF フォーマットの提唱組織では、“Acrobat PDFWriter” という、PDF 用プリンタドライバを、Windows 及び Mac OS 用に提供している。一般のプリンタドライバ同様、PDFWriter では、手動で OS から利用選択することができる（一端 Windows のプリンタディレクトリや、Mac OS のセレクトを経由する）。選択後は、どのユーザーでも、直接、PDF 出力フォーマットへ直接「プリント」できる。PDF Direct を通じて、プログラマやエンドユーザに、4D のデータベース内から、シームレスに一貫して、この優れた PDF 作成機能を提供するものである。

PDF Direct は 4D 内部から直接、PDF ドキュメントへの出力を可能にするものであるが、PDF Direct の出力機能を PDF 言語やコマンドを使って直接 PDF ドキュメントを作成することと混在してはならない。そのような、ばらばらに細分化されたものを組み立てて PDF ドキュメントに作り上げていく機能は、現在の PDF Direct にはないからである。

本ドキュメントにおけるコマンド構文

Command(変数(フィールド型))->\$X(I)

ここで、\$Xは、コマンドセットドキュメンテーションで特定されり、ステータスコードを受け取るものである。

フィールドタイプ
B = Boolean (ブーリアン)
S(n) = スtring。nはString長を指定する
T = テキスト
I = Integer (整数型)
L = Long integer (長整数型)

インストール

PDF Direct は、Windows および Mac OS に固有の圧縮フォーマットで供給されるので、これをハードディスクに展開するだけでよい。各プラットフォーム用に含まれる PDF Direct のバージョン 2 は以下の通りである：

- 1) 4th Dimension version 6 対応の製品ソースコードデモ
- 2) PDF Direct Plug-in
- 3) PDF Direct プログラムのドキュメント

Acrobat™ PDFWriter が必ず必要である。PDF Direct では、PDFWriter を供給していないので注意すること。Acrobat™ PDFWriter は、Adobe 社、あるいは Adobe 社の製品を取り扱う公認組織や、販売店から有料で購入できる。PDF Direct のリリース 2.0 では、Acrobat™ PDFWriter を手に入れるために、最低でも、Adobe Acrobat™ あるいは、Acrobat™ PDFWrite を含む、その他の類似した Adobe 社の製品を購入しなければならない。ただし、Mac OSX では Acrobat の商品は必要ありません。

どのプラグインは必要ですか？

	Mac OS 9.2 以前	Mac OS X 以上	Windows 全
4D 6.8 以上	Mac4DX or Mac4CX	Mac4CX	Win4DX
4D 6.7 以前	Mac4DX	---	Win4DX

動作条件

Mac OS X : Mac OS X の 10.2 以上が必要です。

それ以前の Mac OS の場合は Acrobat PDFWriter が必要です。

Windows の全バージョンの場合は acrobat PDFWriter が必要です。

概要

PDF Direct に用意された個々のコマンドを詳細に解説する前に、PDF Direct で出力プロセスをコーディングする場合の、基本ステップの概要を説明しておこう。

PDF Direct を用いるにあたり、ディベロッパー／ユーザーが、Acrobat™ PDFWriter の Version3.0 以降を準備していなければならない。PDFWriter が正しくインストールされており、プリンタドライバ類に対し、適切な場所に置かれているものとする（どの場所にあるべきかは、プラットフォームによって異なる）。コマンド **PD_IsAvailable** で、4D ディベロッパーは、PDF ドキュメントを出力する指定する前に、PDFWriter が利用可能かを知ることができる。

4D アプリケーションを起動するに先立って、通常は、物理的にデフォルトの出力装置を指定しておく。というのも、コンピュータのユーザーは、一般的に、標準もしくはデフォルトのプリンタドライバを、それぞれのマシンに対して指定しておくもので、読者（あるいは読者の抱えるユーザー）は、PDF ドキュメントを 4D 内部で作成した後、デフォルトの出力装置を再度指定し直すからだ。そして、PDF Direct は、こういった出力装置の切替をプログラムで行えるような機能を提供する。読者は、出力装置切替が発生したとき、必要なツールを提供されている訳である。

4D 内部から、Acrobat™ PDFWriter へ出力するには、Acrobat™ PDFWriter（Windows ではプリンタドライバ、MacOS では、機能拡張ファイル）への切替が必要である。これは、4D アプリケーションに、PDFWriter の名前と場所を記憶させるためである。この情報が一端マシンに確保されれば、Acrobat PDF ファイルへ出力し、プリンタドライバを直ちに元々設定されていたデフォルトのプリンタへ再設定する。こういったタスクは、**PD_GetPrinter** に含まれており、現在選択されているプリンタについて、名前と型、ゾーンやドライバについての情報を、このコマンドで知ることができる。

PDFWriter への切り替えは、**PD_SetPrinter** を発行すればよい。**PD_SetPrinter** を **PD_GetPrinter** と組み合わせて使うと、プリンタ名を、プリンタ種類やゾーン、ドライバと一緒に指定できる。プリンタをリセットするには、**PD_SetPrinter** を発行すればよい。**PD_SetPrinter** が発行されると、ディ

ベロッパが **PD_GetPrinter** を使って指定したデフォルトのプリンタに設定が戻される。

PDF Direct によって、次に挙げる事が可能となる：

- 1) PDF ドキュメントにプログラムを使って名前をつけることができる
- 2) 4D が作成した PDF ドキュメントにユーザが名前をつけることができる
- 3) ユーザの必要に応じて、変更可能なプログラム化できる名前を規定することができる。

PD_GetPath を使えば、ディベロッパは、PDF ドキュメントを新たに作成し、そのドキュメントに名前を与えるといたファイルコントロールを、完全に行うことができる。このコントロールは、カスタマイズ可能な“別名で保存...”メッセージダイアログの形で行われる。このダイアログの中で、ユーザは特定のファイル名（ファイル名を変更することも可能）を指定したり、ファイル名を空白として与え、ユーザがファイル名だけでなくファイルの種類も含めてすべて指定できる。たとえば、ユーザは標準のファイルメニューに現れる“別名で保存...”ダイアログを操作して、パスを完全に変更することができる。

ファイル名が決定されたら、そのファイル名を PDFWriter のプリンタドライバへ伝える必要がある。この作業を行うのが、**PD_Define** である。**PD_Define** は、PDFWriter に作成されたドキュメントのフル・パス名を連絡する。さらに、**PD_Define** によって、4D ディベロッパは、PDF ドキュメントが作成される時に、Acrobat Reader プログラムを起動させるかどうかを指定することもできる。これは、PDF ドキュメントを素早く準備し、テストする際に、大変便利である。

PD_Define でフル・パス名を空のままにすると、PDFWriter のプリンタダイアログが表示され、ディベロッパにフル・パス名を入力するよう求める。これは、ディベロッパあるいはエンドユーザが、特殊な属性を持つような PDF ドキュメントを作成しようという場合に役立つだろう。

MacOS X の印刷方法はちょっと違います。プリンティングシステムが全仕事を行っています。Mac OS X が印刷ダイアログを表示し、PDF Data として出力し、プリントジョブを作ります。それからプリンタードライバーは PDF Data を受け取ります。

その為にも新しいコマンドが増えました。**PD_GetStatus** は PDF ファイルの印刷ステータスを取得します。その為に **PD_Define** に新しい引数が増え、その結果が **PD_GetStatus** に使用できます。

4D から PDF ドキュメントを作成した後に役立つコマンドが、**PD_Cleanup** である。このコマンドは、

PDFWriter で作成するよう設定されたものを、使用後に消去するものである。というのも、4D 環境下では、PDFWriter のプリンタダイアログを呼び出すことができ、前回の印刷ジョブについての印刷設定が、以降の印刷ジョブに対しては、ふさわしくない場合がありえる。PD_Cleanup は、現在の設定(以前の印刷ジョブの設定を持ち越している)をリセット(イニシャライズ)し、PDFWriter のデフォルト設定に戻すコマンドである。

プラットフォーム別印刷の違い

Mac OS X

Mac OS X 上の 4D 6.8 は起動時にデフォルト・プリンターを設定します。そのため、4D の「Print Settings」コマンドで PDF Direct が新しいデフォルトプリンターであることを指定する必要があります。4D の管理方法のため、開発者が「Print Settings」ダイアログを表示しなければなりません。ユーザが PDF を作成する前に PDF Direct プリンターを指定する必要があります。PDF Direct の「PD_SetPrinter」コマンドの後に 4D 「Print Settings」コマンドを呼ぶといいのです。

「Print Settings」コマンドはページ・セットアップダイアログ表示後にプリントセッティングダイアログを表示します。「Print Settings」ダイアログ内、ユーザは PDF Direct プリンターを選択しないとけません。それから 4D は新しいデフォルトプリンターを認識します。

PDF Direct の「PD_SetPrinter」コマンドは正確にプリンタドライバを切り替え、OS X のプリンターセンターも正しく切り替わりますが、4D は認識しません。その為にユーザが手動で PDF Direct プリンタドライバに切り替える必要があります。4D 2003 では印刷が改良され、上記の問題は新しいコマンドで発生しません。

4D Windows 版

4D の「Print Form」コマンドを使用すると 4D が新しいプリンタへの切り替えに失敗することがあります。(最初のプリントジョブ) そのため、4D PDF Direct Win 版のデモでは 4D のダミープリントジョブをコールしています。その最初のダミー実行のために。次の本当の PDF 印刷が正しく動作します。あなたのアプリケーションにも同じ様なメソッドを作成するとよいでしょう！

4D Write 6.5 (以上) Windows 版

4D Write はスタンドアロン・アプリケーションに似ています。その為、4D 6.5/6.8 で 4D Write エリアから印刷する場合、開発者は PDF Direct デモ版内のサンプルをよく参照すべきです。

レター出力フォームメソッドにはコードサンプルが含まれます。4D Write エリアがロードされる前に切り替わります。入力フォームに切り替える時に必ず実行されるので、4D は PDF Direct がデフォルトプリンタと認識します。Win 版の 4D 6.5 以上には必要なコーディングです。

コマンド解説

PD_Register

構文：

PD_Register(LicenseeName(S);LicenseCode Macintosh(S);LicenseCode Windows(S)): ->\$X(L)

解説：

この関数は、4D デベロッパが、PDF Direct プログラムの登録をするための関数である。PD_Register は、どんな 4D アプリケーションにおいても、スタートアップ・プロシージャで呼び出されると、PDF Direct を使用可能にする。正しい登録情報であれば、long integer で値 1 が返され、プラグインの全機能を使うことができる。もし、誤った登録情報が渡された場合、すなわち、空もしくは誤った登録番号が、PD_Register を発行したマシンから送られた場合には、long integer で値 0 が返され、プラグインの機能は 30 分間しか使うことができない。

変数：

Licensee Name.....PDF Direct ライセンス契約で規定されたストリング。

LicenseCode Macintosh.....Mac OS 用の PDF Direct のライセンス契約を受けたデベロッパに適合するよう規定されたストリング。

LicenseCode Windows.....Windows 用の PDF Direct のライセンス契約を受けたデベロッパに適合するよう規定されたストリング。

例：

\$ReturnCode:=PD_Register("Name";"Mac Code";"Windows Code") `Soft Solutions, Inc. が発行した登録コード

PD_IsAvailable

構文：

PD_IsAvailable->\$X(I)

解説：

PD_IsAvailable は、現在 Acrobat PDFWriter が利用可能かを、返り値によって調べる関数である。PDF ドキュメントを印刷しようとする前に、このコマンドを発行しておかねばならない。このコマンドは、4D アプリケーションからリクエストしたプロセスに対して、PDFWriter が、現在他で使用途中で、利用できない状況かどうかを調べるときに役に立つ。

返り値：

ResultCode.....PDFWriter が利用可能なら、integer で 0 が返される。エラーが発生した場合は、ゼロ以外の値が返される。

例：

```
$X:=PD_IsAvailable
```

PD_GetPrinter

構文：

```
PD_GetPrinter(Name(S);Type(S);Zone(S);Driver(S))->$X(I)
```

解説：

PD_GetPrinter は、現在のデフォルトプリンタの、アドレス情報を取るために使うコマンドである。Windows 版、Mac OS 版とも、同じパラメータをとる。Alpha 80 で渡す変数は、4 つである。返り値 0 は、コマンドが正しく実行されたことを示す。コマンドが正しく実行された場合、変数それぞれに、プリンタ情報が返されている。

変数：

Name現在のプリンタ名が入る文字列。選択されたプリンタがローカル（コマンドを発行したマシンに直接接続されている）プリンタの場合この変数の値には、“Modem” あるいは “Printer” が入る。

Typeプリンタの種類を入れる文字列。たとえば、Apple 社の全 LaserWriter の場合、この変数には “LaserWriter” と入る。

Driver現在のプリンタドライバ名を入れる文字列。たとえば、“AppleTalk ImageWriter” など。

Zoneプリンタの属するゾーンを入れる文字列。もし、現在選択されたプリンタが、このコマンドを発行したマシンと同じゾーンに属している場合、この変数には “*” が入る。また、違うゾーンに属している場合には、そのゾーン名がこの変数に返される。

返り値：

ResultCode.....操作が正しく実行された場合、integer で 0 が返される。もし、エラーが発生した場合、0 以外の値が返される。

例：

```
$x:=PD_GetPrinter(Name;Type;Zone;Driver)PD_SetPrinter
```

PD_SetPrinter

構文：

PD_SetPrinter(Name(S);Type(S);Zone(S);Driver(S))->\$X(I)

解説：

PD_SetPrinter は現在のデフォルトプリンタのアドレス情報をリセットするために使う。Windows 版も Mac OS 版も、同じ変数を取る。それぞれ、4つの変数からなる、Alpha80 のプロセス変数を使うこと。このコマンドが正しく実行されたとき、戻り値として0が戻され、現在のプリンタドライバは、4つの変数で指定されたものに変更される。

変数：

Name現在のプリンタ名が入るstring。選択されたプリンタがローカル（コマンドを発行したマシンに直接接続されている）プリンタの場合この変数の値には、“Modem”あるいは“Printer”が入る。

Typeプリンタの種類を入れるstring。たとえば、Apple社の全LaserWriterの場合、この変数には“LaserWriter”と入る。

Zoneプリンタの属するゾーンを入れるstring。もし、現在選択されたプリンタが、このコマンドを発行したマシンと同じゾーンに属している場合、この変数には“*”が入る。また、違うゾーンに属している場合には、そのゾーン名がこの変数に返される。

Driverユーザあるいはディベロッパが使いたいプリンタドライバ名を入れるstring：“Acrobat™ PDF-Writer”など。

戻り値：

ResultCode.....正しく実行されると、integerで0が返される。もし、-43という値が返された場合、Name変数で指定された名前のプリンタが見つからなかったことを示す。

例：

\$x:=PD_SetPrinter(Name;Type;Zone;Driver)

PD_GetPath

構文：

PD_GetPath(Pathname(S);Prompt(S);DefaultFileName(S))->\$X(I)

解説：

PD_GetPath は、ファイルメニュー中の“別名で保存 ...”ダイアログを、ディベロッパが定義したデフォルト情報と共に発行するために使う。Windows 版も Mac OS 版も、同じ変数を取る。最初の 2 つの変数は、Alpha80 のプロセス変数を使い、3 番目の変数には、Alpha31 のプロセス変数を使うこと。返り値 0 は、コマンドが正しく実行されたことを示す。コマンドが正しく実行された場合、FilePath 変数の値は、SFDefault 変数で指定された本来のファイル名か、あるいはユーザによって指定されたファイルのパスおよびファイル名となる。

変数：

FilePath作成したいファイルの名前に対する、フルパスを指定するストリング変数。ユーザに新規ファイルの作成場所指定を許す場合、このコマンドをコールする前は、この変数をヌルストリングに割り当てる；この場合、コマンドは、標準のファイルパッケージである“別名で保存 ...”ダイアログをユーザに示し、ファイル名と、PDF ファイルを保存するフォルダ先を入力するように求める。コマンドが正しく実行されると、ユーザによって指定されたファイルのフルパスが、この変数に割り当てられる。

SFPrompt.....ユーザに標準のファイルパッケージダイアログを表示するよう指定するストリング。コマンドがコールされる以前に、パス値が FILEPATH に割り当てられている場合、この変数は無視される。

SFDefault標準のファイルパッケージダイアログが表示された場合、ユーザに対して、デフォルトのフォルダおよび（あるいは）ファイル名を指定するためのストリング。コマンドがコールされる以前に、パス値が FILEPATH に割り当てられている場合、この変数は無視される。ここで、実際に存在するフォルダへのフルパスを渡すと、標準ファイルダイアログは、そのフォルダをあらかじめセレクトして表示し、ユーザにそのフォルダの中に含まれるファイルカタログリストを示す。よって、この“示唆された”フォルダが、ファイルダイアログ中のファイル名で指定した新規ファイルの保存先となる。もし、実際に存在するフォルダへのフルパスと共に、ファイル名も渡した場合、ファイルダイアログは、保存先のフォルダおよびファイル名両方を、デフォルトの名前としてあらかじめセレクトする。

返り値：

ResultCode.....コマンドが正しく実行された場合、integer で 0 が返される。エラーが発生した場合、ゼロ以外の値が返される。

例：

\$x:=PD_GetPath(Pathname;Prompt;DefaultFileName)

PD Define

構文：

PD_Define(Pathname(S);LaunchReader(I);Capture Reference(S))->\$X(I)

解説：

PD_Define は、現在の印刷ジョブで作成される PDF ドキュメントに名前を与え、Adobe Acrobat Reader を起動するかどうかを指定するために使われる。返り値 0 は、正しく実行されたことを示す。コマンドが正しく実行された場合、このコマンド以降の 4D 印刷ジョブは、Pathname 変数で指定されたファイル名として印刷する。

重要事項:4D 印刷ジョブに関して、マルチページ・フォーム、あるいはマルチページ・印刷セレクションを行う場合、すべての“印刷される”4D フォームを、PDF への出力ファイルとするためには、4D Customizer の中で、“Print One Job”を指定しておかねばならない。4D PRINT FORM コマンドは、印刷するそれぞれのページをメモリに割り当てる。メモリ上のページがいっぱいになったり、PAGE BREAK がコールされた場合に、それぞれのページは印刷される。さらに、PRINT FORM を使っている場合は、4D の“PAGE BREAK”コマンドは、オプションとして、変数 (>) を取る。このオプション変数は、同じ印刷ジョブの残りの部分について、継続して同じフォームページの使用を許すものである。この変数を加えることで、印刷ジョブを 1 つのファイルとしてスプールできる。

戻り値のキャプチャ・リファレンス番号は「PD_GetStatus」コマンドで使用します。

PRINT FORM を使った場合、最後のページまで印刷した後は、必ず PAGE BREAK で終了しなければならない。そうしないと、最後のページがメモリ上でフルになっていない場合には、その情報がずっとメモリ上に残ってしまい、印刷されない (4th Dimension v6 manual からの抜粋)。

変数：

Pathname.....これから保存しようとするドキュメントのフルパス名を指定するストリング。これをマルチストリングとすると、PDFWriter のダイアログが呼び出される。PDFWriter のダイアログとは、ファイルの“別名で保存...”ダイアログのことで、これから作成する PDF ドキュメントのファイル名と、保存先のディレクトリへのパスを指定するよう、ユーザに求める。さらに、PDFWriter のダイアログでは、解像度や、圧縮率、ページサイズ、ページ方向、拡大縮小率などといった、さまざまな印刷オプションも提供される。

LaunchReader.....作成された PDF ドキュメントを見るために、Adobe Acrobat Reader を起動するかどうかを指定する、integer 型のストリング。この変数に 0 を渡すと、Acrobat Reader は起動しない。この変数に 1 を渡すと、Acrobat Reader を直ちに起動し、PDF ドキュメントを閲覧可能にする。

CaptureReference.....PD_GetStatus コマンドに必要な文字列

返り値：

ResultCode.....コマンドが正しく実行された場合、integer で 0 が返される。
エラーが発生した場合、ゼロ以外の値が返される。

例：

```
$X:=PD_Define(Pathname;LaunchReader;Reference)
```

PD_GetStatus

構文：

```
PD_GetStatus(Capture Reference(S);Status(I))->$X)
```

解説：

PD_GetStatus コマンドは出力中の PDF ファイルのステータスを得る為に使用します。

注意:Mac OS X上の4D Writeを使った場合、4D Writeでの印刷ダイアログ表示を非表示できません。
それから、4D Writeはプリンター情報もドキュメント内に保存することで、デフォルト・プリンターを無視します。印刷ダイアログが表示されている時には"PDF Direct"を選択してください。

変数：

Capture Refer.....PD_Define で受け取ったキャプチャ・リファレンス文字列
Statusinteger

1 = 印刷準備 OK , 2 = 印刷中 , 3 = PDF 作成完了

返り値：

ResultCode.....コマンドが正しく実行された場合、integer で 0 が返される。
エラーが発生した場合、ゼロ以外の値が返される。

例：

```
$X:=PD_GetStatus(vi_Reference;vs_Status)
```

PD_Cleanup

構文：

PD_Cleanup->\$X(I)

解説：

PD_Cleanup は、コマンドが正しく実行されたら、integer で 0 を返し、正しく実行されなかった場合には、ゼロ以外のコードを返す。プリンタドライバの選択を管理・変更するにあたっては様々なメモリを使うため、PDF Direct で提供されるコマンド群を必要に応じて使った後は、PD_Cleanup を発行しておくことを推奨する。

例：

\$X:=PD_Cleanup

サンプル・アプリケーション

PDF のデモ・アプリケーションを以下に示す。PDF Direct で提供されるプラグインコマンドをすべて用いた、ソースコード例である。

準備

4D ドキュメントと、Adobe Acrobat の PDF フォーマットで書き出す前に、ディベロッパあるいはユーザは、まず、Adobe PDFWriter に、プリンタドライバ情報を渡しておかねばならない。現在のデフォルトのプリンタについて、ユーザが最初にデモ・アプリケーションのメニューアイテムをセレクトしたときに、以下のダイアログが表示される。



メモ	最初に、ダイアログは現在のデフォルトプリンタとして、物理的に接続されているプリンタを表示する。Adobe AcrobatWriter 用に、物理的に存在するプリンタのプリンタドライバを保存しようとしなさい。もし、そのように保存した場合には、4D の印刷ジョブは、デフォルトのプリンタに送られることになる。
----	--

PDF Writer をデフォルトのプリンタに設定することが、データベース中で、PDF Writer のプリンタドライバをキャプチャするための最初のステップである。一旦 Adobe PDFWriter に名前をつけ、その位置情報をキャプチャしたら、—将来行うであろう Acrobat への印刷に備え、プリンタ情報を適当な場所に保存しておけばいいのだ。

デフォルトプリンタの確保は、使っている開発プラットフォーム上に用意されている適切なツールを使って、簡単に実行できる。PDF Direct 4D デモ・アプリケーションでは、シングルユーザ・アプリケーションとして設定されているが、複数のクライアント・マシンを持つ 4D Server 用アプリケーションに簡単に変更できる。

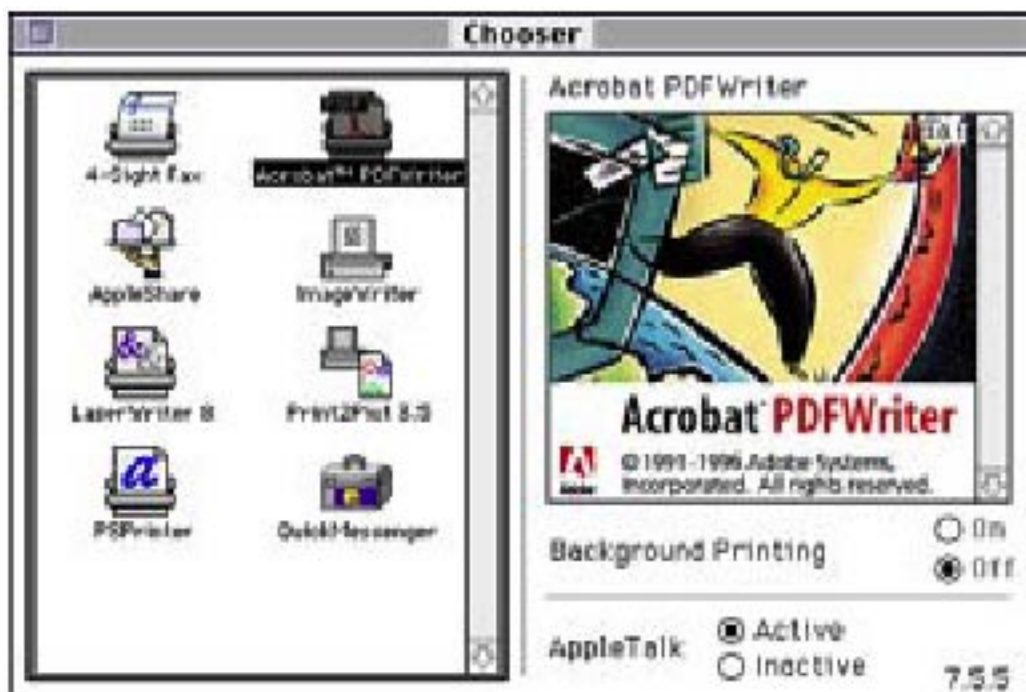
データベース・アプリケーションによっては、Acobat 用プリンタドライバの確保までユーザに行

うべきかどうか、迷うかもしれない。ユーザは、しばしばマシンを変更するため、デフォルトとした Adobe PDFWriter の位置情報が最新の状態と異なってきたり、全く間違っただけのものになるからだ。理想的には、ローカル接続された Adobe PDFWriter の位置情報が、ローカル接続されたハードディスクに、プリファレンスファイルとして書き込まれるようにしておくことが望ましい。Adobe PDFWriter をプリンタドライバとして使う可能性のあるマシンそれぞれについて、そのような設定をしておくのは、とてもいい方法だ。PDFWriter の名前が変更されたり、位置情報が変更されたとしても、この方法は、最も安全で、長持ちする方法だろう。

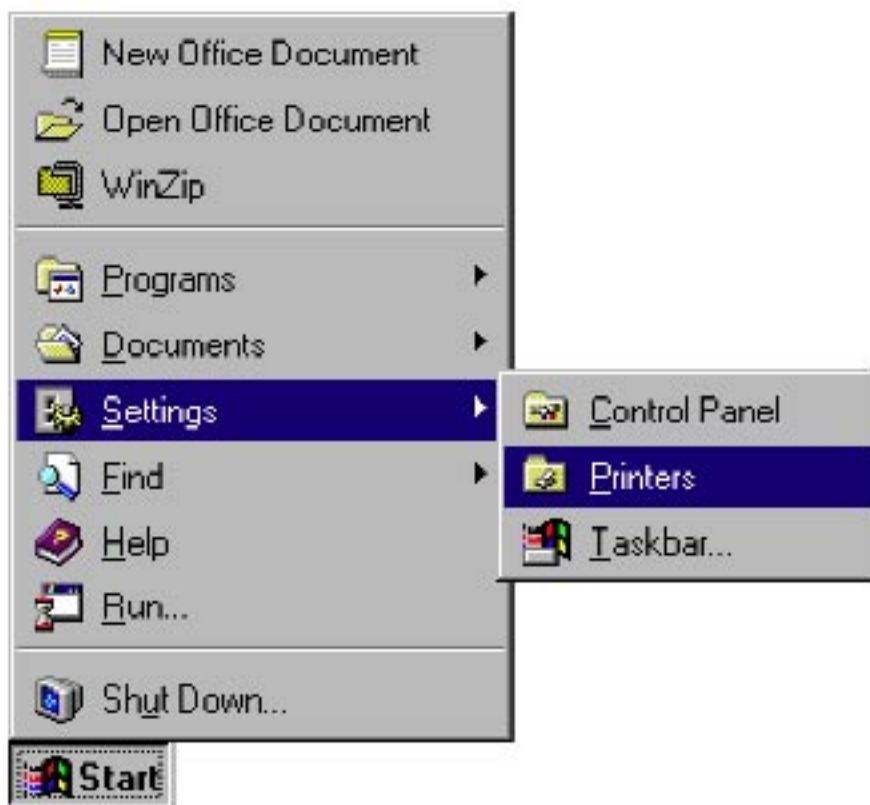
逆に、Adobe PDFWriter のプリンタドライバ設定を、“ハード・コーディング”しておくのは、あまり推奨できない。プラットフォームが違い、OS のバージョン違いや、AdobePDFWriter のバージョン違いなど、度重なる変更が加わったり、ユーザによって変更が加えられたりすることから、問題やエラーを起こしがちだからだ。

以下のページに、Mac OS および Windows 両方のユーザ向けに、Acrobat PDFWriter をデフォルトのプリンタとするためのステップを紹介する。

Mac OS では、ユーザは、“セレクトラ”を選び、直接 Adobe PDFWriter を選択しなければならない。



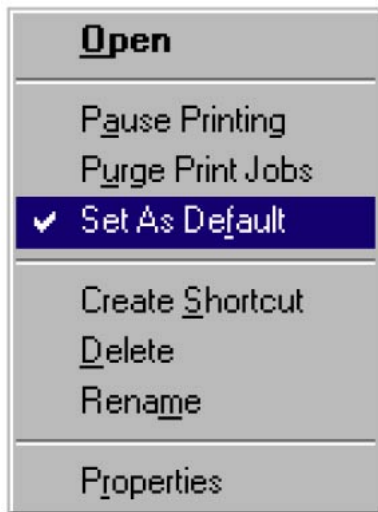
Windows では、まず、設定メニューからアクセスできる、プリンタのディレクトリへ移動しなければならぬ。



Windows の“プリンタ”ディレクトリに移動したら、Acrobat PDFWriter のプリンタドライバを、右クリックし、Acrobat PDFWriter のプリンタオプションを表示させる。



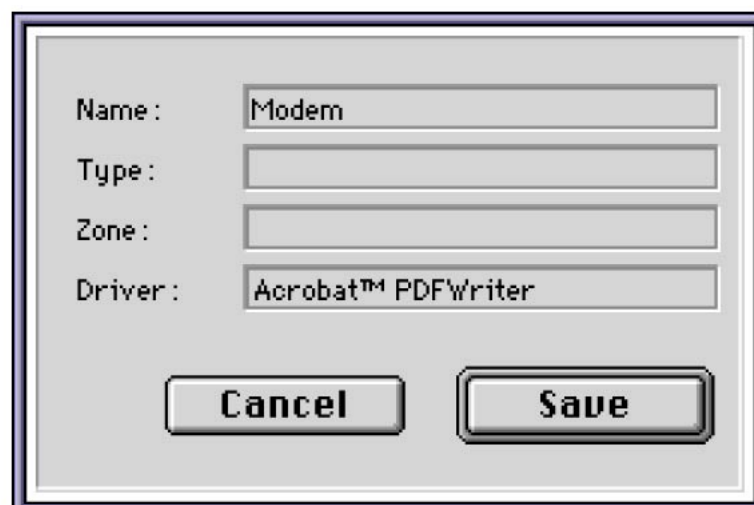
“通常使うプリンタに設定”を選択すれば、Acrobat PDFWriter を現在のデフォルトプリンタとして指定できる。



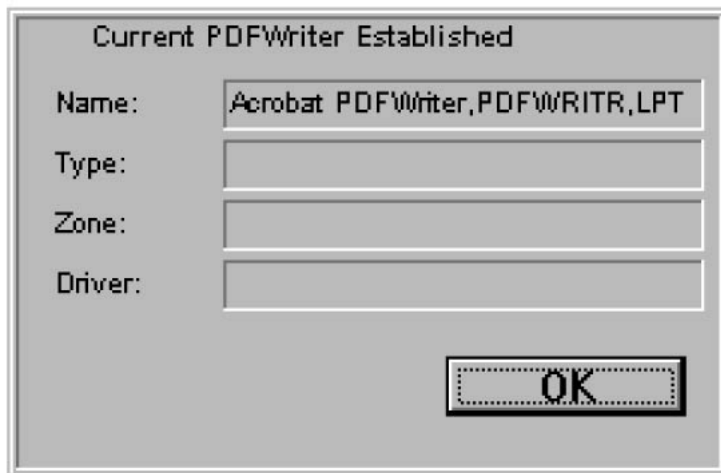
一旦、Acrobat PDFWriterをデフォルトのプリンタ(ドライバ)として確保したら、ディベロッパ/ユーザは、4th Dimension のアプリケーションに、その情報を返し、メニュー・オプションを使って、それらの情報をデータベースの設定として保存しておかねばならない（以下に示す PDFWriter の割り当てを参照）。



Mac OS では、Driver が Acrobat? PDFWriter に割り当てられていても、普通、Name は “モデム” あるいは “プリンタ” に割り当てられる。



Windows では、普通、以下に示すポート情報とともに、Name が “Acrobat PDFWriter” に割り当てられる。Driver 情報は空欄（ヌル）となる。



Acrobat PDFWriter の設定が（Acrobat PDF ドキュメントを印刷する可能性のある）それぞれのマシンに保存されたら、ユーザが、デフォルトのプリンタを PDFWriter から本来のプリンタへ、OS のツールを使ってもどしておけばよい。

以下に、いくつかの 4D 印刷タスクで用いられる 4D コーディング例を挙げておく。

4D Write PDF Printing

4D Write で “Mail Merge” するという方法は、PDF ドキュメントを作成し、名前をつけるという作業をプログラムで実現する際に用いることができる。しかしながら、それにはいくつかの大切なルールがあり、そのルールに従わなければ、4D アプリケーションがユーザにとって問題の種になってしまう。4th Dimension は、異なる 4D の印刷ジョブの扱い方（および一般のプリンタ・ドライバの扱い方）が、大変特殊である。したがって、これらの大切なルールについて、読んでおく価値が絶対にあるはずだ。

Mac OS:

4D の “For ループ” の内部で、WR PRINT を発行し、プリンタ・ドライバを自動的に、連続する数字に加える。たとえば、要求された名前が “John Smith.PDF”、“Bill Jones.PDF” および “Jack Smith.PDF” で、最初のドキュメントは正しく “John Simith.PDF” と名前がつけられたとしよう。しかし、2 番目、3 番目のドキュメントは “Bill Jones.PDF2” や “Jack Simith.PDF3” という名前になってしまう。したがって、4D Write では、“WRDO” というコマンドを、4D の “For ループ” 内でのメール・マージで使うことを推奨している。さらに、4D Write のデータベース・メニューで、“Show

References”メニューではなく、“Show Values”を選択することが、エンドユーザにとっては必須事項である。リファレンスを表示しても、変数を正しくマージできないが、“Show Values”を使えば、それができるからだ。

```
$Err:=PD_GetPrinter (vname:vptype:vzone:vdriver)
$Err:=PD_SetPrinter (<>PDF_Name;<>PDF_Type;<>PDF_Zone;<>PDF_Driver)
```

or `gp_OnMenuPDF は、PDF Direct Mac/Win version 1.0 で使う

```
C_LONGINT($1;$2;$3;$Err;$R;$i)
C_TEXT($Name;$Path;$DocName)
Case of
```

```
¥ ($2=1 12) `Print Merge Menu Item
    USE NAMED SELECTION("Merge")
```

```
vPrompt_Option:=False ` メール・マージについて、PDF Print Dialog を無効とする
vViewer_Option:=False ` Reader に、メール・マージのロードを禁止する
bLaunchViewer:=0
```

```
gpCenterWindow (228;265;1;")
vMerge_Message:=String(Records in selection([Contacts]))+" Contact records will be merged with
the letter named "+[Letters]LetterName+"."
vTitle:="PDF Merge Setup"
DIALOG([Constants];"PDF_Options")
CLOSE WINDOW
```

(上記の) コードは、以下のダイアログを表示する



If (bOk=1) ユーザが PDF ボタンを選択した

ディベロッパは、エンドユーザに提示するファイル名として、デフォルトの名前をつけておく

MESSAGES OFF `メッセージを OFF にしておく

FIRST RECORD([Contacts])

\$Name:=[Contacts]First_Name+" "+[Contacts]Last_Name

ディベロッパは、現在のデフォルトのプリンタ情報を確保しており、Acrobat PDF の印刷ジョブを完了した後に使うために、その情報をデータベース変数にストアする

vname="" `ポート名 ("name" は、旧世代の Mac の名残)、Windows および Mac で用いる

vptype="" `プリンタ種類; シリアル接続時 = "/ プリンタは "LaserWriter"/Mac のみ使用

vzone="" `デフォルトゾーンは "*" / シングルゾーンかローカルゾーンを示す /Mac のみ使用

vdriver="" `Mac OS 機能拡張 (で与えられるプリンタドライバ) の名称

\$Err:=PD_GetPrinter (vname;vptype;vzone;vdriver)

ディベロッパは、出力先を Acrobat PDF プリンタドライバに設定する

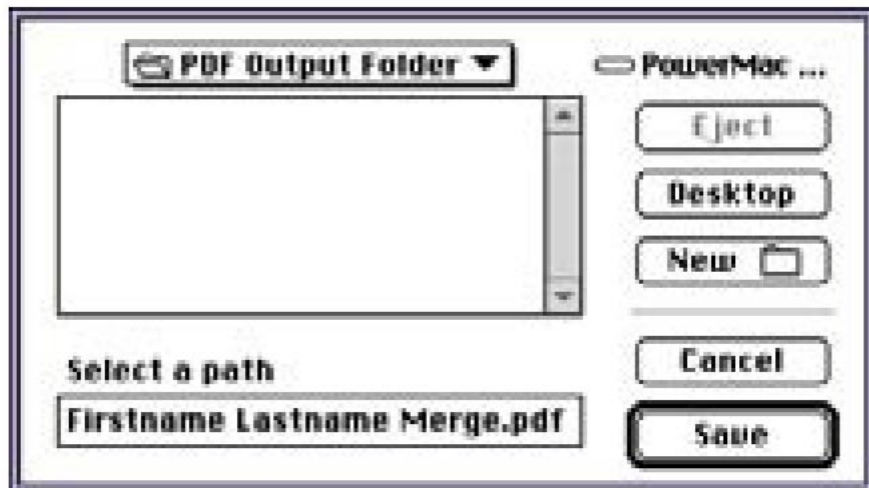
\$Err:=PD_SetPrinter (<>PDF_Name;<>PDF_Type;<>PDF_Zone;<>PDF_Driver)

If (bPromptDefault=1)

\$Err:=PD_GetPath (\$Path;"Select a path";\$Name+" Merge.pdf")

End if

上記で用いたコマンド "PD_GetPath" は、以下に示すダイアログをユーザに提示する：



\$WhereDelete:=Position(\$Name+" Merge.pdf";\$Path) `ドキュメント名を明らかにする

\$ShortPath:=Delete string(\$Path;\$WhereDelete;250) `ここでパス名を確保することが可能

```

$R:=Records in selection([Contacts])
For ($i;1;$R)
    Case of
        ¥ (bAutoName=1)
            $Name:=[Contacts]First_Name+" "+[Contacts]Last_Name
            $DocName:=$Name+" Merge.pdf"
        ¥ (bPromptDefault=1)
            $Name:=[Contacts]First_Name+" "+[Contacts]Last_Name
            $DocName:=$ShortPath+$Name+" Merge.pdf"
    End case

    $Err:=PD_Define ($DocName;bLaunchViewer) `Mailmerge 中での Acrobat を起動防止
    If (<>z_n_Platform=Windows)
        WR PRINT (Letter;0) ` マージするドキュメントエリアを指定 / 値提示 /
        ' リファレンスを示すのではない
    Else ` ユーザにマージできるよう値を選択させることに注意!!!
        WR DO COMMAND (Letter;1 1 1) `4D Write WR Do は排除された
        ' ファイル名を使用
    End if

    If (bCreateLog=1)
        gpBuild_PDF_Log ($Name+" Merge.pdf")
    End if

    NEXT RECORD([Contacts])
End for ` ($i;1;$R)

$Err:=PD_SetPrinter (vname;vptype;vzone;vdriver)
` プリンタドライバをデフォルトに戻す
$Err:=PD_CleanUp ` Acrobat をデフォルト状態にリセット、メモリをクリーンアップ

MESSAGES ON ` サーチ&ソートメッセージをオンにする
Else ` ユーザがキャンセルボタンを選択した

End if ` If(bOK=1) ` ユーザが、次へボタンを選択した

Else ` ユーザが Print Merge 以外のメニューアイテムを選択した
    WR DO COMMAND ($1;$2;$3) `4D Write を通じてコマンドを実行する
End case

```

4D Print Selection Printing

4D Write 内で“Print Selection”を実行するという方法は、プログラムでPDFドキュメントを作成し、その名前を与えるときによく用いられる。

```
`gpPrintSelection  
C_BOOLEAN($Continue)  
$Continue:=False
```

If (<>z_n_Platform=Windows) ` この例は、シングルユーザ用のみの設定である

```
  If (<>PDF_Name # "") `Windows 用 PDF Driver がすでに確保されているのを確かめる  
    $Continue:=True  
  Else  
    gpCenterWindow (352;131;1;")  
    vMessage:="The PDFWriter printer must first be made the default printer and then saved"  
    vMessage:=vMessage+" into the database by selecting menu item Assign PDFWriter."  
    DIALOG([Constants];"Show Status")  
  End if  
Else  
  If ((<>PDF_Name # "") & (<>PDF_Driver # ""))  
    $Continue:=True  
  Else  
    gpCenterWindow (352;131;1;")  
    vMessage:="The PDFWriter printer Driver must first be made the Chooser default and then"  
    vMessage:=vMessage+"saved into the database by selecting menu item Assign PDF Writer."  
    DIALOG([Constants];"Show Status")  
  End if  
End if
```

If (\$Continue)

```
  gpCenterWindow (260;130;4;"Output Options")  
  DIALOG([Constants];"Print_or_PDF")  
  CLOSE WINDOW
```

If (bOK=1) ` ユーザが、次へボタンを選択した

If (bPrint=1) ` 印刷先を現在のプリンタドライバとする

```
    PRINT SELECTION(vFilePtr->)
```

Else `PDFWriter を印刷先とする

```
  $Err:=PD_GetPrinter (vname;vptype;vzone;vdriver)
```

```
  $Err:=PD_SetPrinter (<>PDF_Name;<>PDF_Type;<>PDF_Zone;<>PDF_Driver)
```

```
  PRINT SELECTION(vFilePtr->)
```

```
$Err:=PD_SetPrinter (vname;vptype;vzone;vdriver) ` ドライバをデフォルトに戻す  
$Err:=PD_CleanUp ` Acrobat をデフォルトの状態にリセットする  
End if
```

```
End if
```

```
End if
```

4D Print Form Printing

4D 内部から “Print Form” を実行する場合、Windows については、特別にコーディングが必要である。Windows 版の 4th Dimension では、プリンタドライバ変更後の初回印刷に問題が起こるからだ。このことは、4D で “Print Form” コマンドを扱うときに、非常に重要なポイントである。以下に示すルールは、Windows において重大である：Windows マシンの場合、プリンタドライバを変更した直後に、4D の “Print Selection” を発行し、レコードセットを空（ヌル）にしておかねばならない（PDF Direct の “PD_SetPrinter” コマンドを用いる）。

```
If (<>z_n_Platform=Windows) ` この例はシングルユーザ用のみの設定である
```

```
    Query([PDF_Log];[PDF_Log]Unused1="***?***") ' レコードはなく、印刷されないことを保証
```

```
    Print Selection([PDF_Log];*) ' アスタリスクで、プリントダイアログ非表示を指定する
```

```
End If
```

```
`gpOrder ; ユーザが PDF Direct を使うのを手助けする
```

```
C_BOOLEAN($Continue)
```

```
$Continue:=False
```

```
If (<>z_n_Platform=Windows) ` この例はシングルユーザ用のみの設定である
```

```
    If (<>PDF_Name # "") `Windows で PDF Driver が確保されていることを保証する
```

```
        $Continue:=True
```

```
    Else
```

```
        gpCenterWindow (352;131;1;")
```

```
        vMessage:="The PDF Writer printer must first be made the default printer and then saved"
```

```
        vMessage:=vMessage+" into the database by selecting menu item Assign PDF Writer."
```

```
        DIALOG([PDF_Log];"Show Status")
```

```
    End if
```

```
Else
```

```
    If ((<>PDF_Name # "") & (<>PDF_Driver # ""))
```

```
        $Continue:=True
```

```
    Else
```

```
        gpCenterWindow (352;131;1;")
```

```
        vMessage:="The PDFWriter printer Driver must be made the Chooser default and then"
```

```
        vMessage:=vMessage+"saved into the database by selecting menu item Assign PDF Writer."
```

```
        DIALOG([PDF_Log];"Show Status")
```

```
    End if
```

```
End if
```

```
If ($Continue)
```

```
    gpCenterWindow (540;290;4;"PDF Direct Order Form") `Set up for 13" Screen
```

```
    DIALOG([Contacts];"OrderEntry")
```

```
    CLOSE WINDOW
```

```
If (bOk=1) ` ユーザが PDF ボタンをクリックした
```

```
    vPrompt_Option:=True
```

```

vViewer_Option:=True ` Acrobat Reader 起動オプション選択可能—ユーザが目で確認
vMerge_Message:="Print Current Order Form"
gpCenterWindow (228;265;1;")
vTitle:="PDF Print Setup"
DIALOG([Constants];"PDF_Options")
CLOSE WINDOW

C_TEXT($Path)

$Name:=vCreator

vname:="" ` ポート名 (旧式 Mac)、Windows および Mac OS 両方で使用される
vptype:="" ` プリンタの種類; シリアル = "" ; "LaserWriter" として印刷; Mac のみ
vzone:="" ` デフォルトゾーンは "*" で、シングルゾーンあるいはローカルゾーン
vdriver:="" ` Mac OS 機能拡張で与えられるプリンタドライバ名
$Err:=PD_GetPrinter (vname;vptype;vzone;vdriver) ` 現在のデフォルトプリンタ情報入手

$Err:=PD_SetPrinter (<>PDF_Name;<>PDF_Type;<>PDF_Zone;<>PDF_Driver)
` ここで Adobe PDFWriter へ設定

```

Case of

¥ (bAutoName=1)

```

$Name:=vCreator+" Order.pdf"
$Path:=$Name

$DocName:=$Path

```

¥ (bPromptName=1)

```
$Path:=""
```

¥ (bPromptDefault=1)

```

$Name:=vCreator+" Order.pdf" ` 指定フォームに入力された名前を入手
$Err:=PD_GetPath ($Path;"Select a path";$Name) ` ユーザに編集選択するファイル名を提示

$WhereDelete:=Position($Name;$Path)
$ShortPath:=Delete string($Path;$WhereDelete;250)

```

End case

If (bPromptName=0)

```
$Err:=PD_Define ($Path;0) ` 1 の場合 True、0 の場合 False と処理する
```

Else

```
` ユーザが Acrobat ダイアログを要求
```

End if

```
If (<z_n_Platform=Windows) ` この例はシングルユーザ用のみの設定である
    Query([PDF_Log];[PDF_Log]Unused1="**??**") ' レコードはなく印刷されないことを保証
    Print Selection([PDF_Log];*)
End If

For ($y;1;vCopies) ` 4D Customizer を使い、“Print One Job” を ON に設定する
    PRINT FORM([Contacts];"OrderForm")
End for

PAGE BREAK

If (bPromptName=0) ` Acrobat の外で名前をつけたドキュメントにのみアクセスする
    ARRAY STRING(31;aDocList;0)
    DOCUMENT LIST($ShortPath;aDocList) ` カレントディレクトリのリストを入手
    $x:=Find in array(aDocList;$Name)
    If ($x=-1) ` ドキュメントが見つからない
        ALERT("Doc name may be altered (appended with numeric suffix) or a memory problem may e")
    End if
End if ` If (bPromptName=0)

$Err:=PD_SetPrinter (vname;vptype;vzone;vdriver) ` プリンタドライバをデフォルトに戻す

End if ` If (bOk=1) ` ユーザが PDF 印刷ボタンをクリックした場合
End if
```


索引

アルファベット

P

PD_Cleanup 18

PD_Define 16

PD_GetPath 15

PD_GetPrinter 13

PD_GetStatus 17

PD_IsAvailable 12

PD_Register 12

PD_SetPrinter 14