

4D Office Library

By Silvio Belini, Technical Services Team Member, 4D Inc.

Technical Note 09-08

要約

このテクニカルノートでは、4D データベースと Microsoft Office 2007 間でのデータ交換を可能にするコンポーネント、4D Office Library for 4D v11 SQL を紹介します。4D Office Library の実装を説明し、機能のデモを行います。

はじめに

4D Office Library は利用価値の高いコンポーネントであり、ユーザは Windows 上で Microsoft Office 2007 の Word, Excel, および Outlook に、Mac OS 上でアドレスブックや MS Office 2008 の Word、Excel にデータを書き出すことができます。さらに Microsoft Office 2007 Outlook やアドレスブックからデータを読み込むことができます。

4D Office Library for 4D v11 SQL は 4D Office Object Library for 4D 2004 のアップデートバージョンです。4D Office Object Library for 4D 2004 は Word や Excel でひとつの XML ファイルを生成していましたが、4D Office Library for 4D v11 SQL は、最新バージョンの Microsoft Office (Office 2007 for Windows / Office 2008 for Mac) でサポートされた新しい Office Open XML フォーマットを使用して Microsoft Office ドキュメントを作成します。4D Office Object Library for 4D 2004 と 4D Office Library for 4D v11 SQL は両方ともコンポーネントですが、4D v11 SQL データベースへのコンポーネントのインストールは、4D 2004 データベースへのそれよりも簡単です。インストールに関する詳細は" 4D Office Library のインストール"の節で説明します。

概要

4D Office Library for 4D v11 SQL は利用価値の高いコンポーネントであり、4D v11 SQL データベースが Microsoft Office アプリケーションと相互に協働することを可能にします。出力フォーム上でレコードを選択するだけで Microsoft Word や Excel ドキュメントを自動で作成したり、指定したフィールドをドキュメントにバインドしたりできる柔軟性をユーザに提供します。また 4D Office Library を使用して Microsoft Outlook から 4D データベースに連絡先を読み込んだり、逆に 4D v11 SQL データベースから MS Outlook に書き出したりできます。Mac 上ではアドレスブックアプリケーションに対し読み書きが可能です。

Microsoft Office 2007 と 2008 の Word と Excel は Office Open XML ファイルフォーマット (別名 Office XML フォーマット) を使用します。Office Open XML ファイルフォーマットは XML と ZIP のテクノロジーを使用しているため、あまねくアクセスが可能であり、またファイル損傷の危険性を減少させています。新しい Office ファイルフォーマットは ZIP ファイルコンテナに圧縮された複数ファイルで構成されます。これらのファイルはほとんどがアプリケーションデータ、メタデータ、顧客データ等を記述する XML ファイルで、コンテナファイルの内部に格納されます。他に画像など XML でないファイルもコンテナパッケージに含まれることがあります。これらの XML および非 XML ファイルはドキュメントパーツとも呼ばれます。ドキュメントパーツは業界標準の ZIP フォーマットでコンテナファイルやパッケージに格納されます。2007 Microsoft Office システムファイルの個々のパーツを ZIP コンテナにラップすることで、ドキュメントは一つのファイルとして扱うことができます。つまりユーザはいままでと同じ利用方法で Office 2007 ドキュメントを扱うことができます。Office XML フォーマットパッケージには ZIP が選択されました。これは ZIP がよく理解された業界標準だからです。

Office Open XML ファイルフォーマットに関する詳細は以下の Web サイトを参照してください: <http://msdn.microsoft.com/ja-jp/library/aa338205.aspx>

すべてのドキュメントパーツを ZIP コンテナに圧縮するために、4D Office Library はそれぞれのプラットフォームごとにコマンドラインのファイルアーカイバが必要です。Mac 上で 4D Office Library は、Mac OS X に含まれる zip コマンドを使用します。zip コマンドは圧縮およびファイルパッケージ化ユーティリティであり、ZIP アーカイブの中にひとつまたは複数のファイルやディレクトリ構造全体を置くことが可能です。zip コマンドは以下のシンタックスを持ちます:

```
zip [-options] zipfile file1 file2 ...
```

zip コマンドに関する完全な説明と利用可能なオプションについては以下を参照してください:

<http://developer.apple.com/documentation/Darwin/Reference/ManPages/man1/zip.1.html>

Windows にはコマンドプロンプト上に Mac の zip と同様の動作をおこなう組み込みのコマンドを持っていないので、4D Office Library は 7-zip を使用します。7-zip は 7z と呼ばれる独自の圧縮フォーマットを持ち、また ZIP, GZIP, BZIP2 そして TAR (パッケージ化及びパッケージ化解除)、および ARJ, CAB, CHM, CPIO, DEB, DMG, HFS, ISO, LZH, LZMA, MSI, NSIS, RAR, RPM, UDF, WIM, XAR そして Z (パッケージ化解除のみ) などの他のフォーマットもサポートするファイルアーカイバです。Windows では、4D Office Library はすべての必要なファイルを ZIP コンテナに圧縮するために、コマンドラインバージョンの 7-zip を使用します。7-zip は以下のシンタックスを持ちます:

```
7z <command> [<switch>...] <base_archive_name> [<arguments>...]
```

7-zip コマンドラインのシンタックスとダウンロードについては <http://www.7-zip.org/> を参照してください。7-Zip はオープンソースのファイルアーカイバであり、GNU LGPL でライセンスされています。

動作環境

4D Office Library はコンポーネントであり、すべての 4D v11 SQL データベースで使用できます。

Word, Excel, および Outlook (Windows のみ) の機能を使用するためには、コンピュータに Microsoft Office 2007 (Windows) または Microsoft Office 2008 (Mac) がインストールされていなければなりません。

Microsoft Office XP と 2003 のユーザも、

<http://www.microsoft.com/downloads/details.aspx?FamilyID=941b3470-3ae9-4aee-8f43->

[c6bb74cd1466&displaylang=ja](#) で入手できる Microsoft Office Compatibility Pack をインストールすることで 4D Office Library を使用できます。

Microsoft Windows 上で 4D Office Library を使用する場合、7-zip が必要です。7-zip はコンポーネントに含まれています。また<http://www.7-zip.org/>からダウンロードが可能です。

4D Office Library の内部

4D Office Library は Open XML ファイルフォーマットを使用して Word や Excel ドキュメントを生成します。このタスクを行うために、4D XML DOM コマンドを使用して必要な XML ファイルを生成し、Mac 上では zip コマンド、Windows 上では 7-zip アーカイバを使用してこれらの XML ファイルを ZIP フォーマットに圧縮します。

Word や Excel ファイルを作成する際、4D Office Library は以下の動作を行います：

- 1) 複数の XML ファイルを生成します。データが Word ドキュメントに書き出される場合、4D Office Library は document.xml, document.xml.rels, そして header1.xml ファイルを作成します。データが Excel ドキュメントに書き出される場合、作成されるファイルは sharedStrings.xml, styles.xml, sheet1.xml, そして sheet1.xml.rels です。これらのファイルを作成するために XML DOM コマンドが使用されます。Word や Excel で Office コンテナ内部に他の XML ファイルが複数存在しますが、4D Office Library で作成されるドキュメントは先に示したドキュメントの実の作成を必要とします。このテクニカルノートでは前述の XML ファイルのみを説明します。Office Open XML を構成する他のファイルに関する詳細な情報は以下のサイトを参照してください: <http://msdn.microsoft.com/ja-jp/library/aa338205.aspx>

Word ドキュメント中で、document.xml には 背景とロゴ画像への参照 (ドキュメントに含まれる場合)、データ自身、そしてヘッダやフッタファイルへの参照が含まれます。document.xml.rels ファイルはコンテナ内のすべてのファイルの関連を記述します。関連は以下のように記述されます：

```
<Relationship Id="someID" Type="relationshipType"
Target="targetPart"/>
```

Id はユニークな識別子です。Type は関連のタイプであり、関連を他の関連と区別し、また関連の目的のヒントを提供します。Type は Office XML フォーマットのタイプを定義するスキーマを指します。最後に、Target は関連のターゲットを含むフォルダやファイルのパスです。以下は document.xml.rels に書かれる関連の一例です:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Relationships xmlns = "http://schemas.openxmlformats.org/package/2006/relationships">
<Relationship Id = "rId1" Target = "styles.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles"/>
<Relationship Id = "rId3" Target = "settings.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/settings"/>
<Relationship Id = "rId4" Target = "webSettings.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/webSettings"/><Relat
ionship Id = "rId5" Target = "footnotes.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/footnotes"/>
<Relationship Id = "rId6" Target = "endnotes.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/endnotes"/>
<Relationship Id = "rId8" Target = "header1.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/header"/>
<Relationship Id = "rId9" Target = "footer1.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer"/>
<Relationship Id = "rId10" Target = "fontTable.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/fontTable"/>
<Relationship Id = "rId11" Target = "theme/theme1.xml" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme"/>
<Relationship Id = "rId2" Target = "media/image1.jpeg" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"/>
<Relationship Id = "rId7" Target = "media/image2.jpeg" Type =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/image"/>
</Relationships>
```

header1.xml ファイルは、どのページにヘッダがあるかやスタイル情報など、ドキュメントのヘッダ情報を含んでいます。またドキュメント内の透かしに関する情報も含んでいます。この場合、4D Office Library は透かし情報を扱います。

Excel ドキュメントでは、sharedStrings.xml ファイルは実際のデータを含みます。しかしスプレッドシート中にデータがどのように置かれるかについては sheet1.xml ファイルで定義されます。このファイルにはまたセルごとのスタイル (フォントサイズ、タイプや背景色など) も記述されます。スプレッドシート上で使用されるスタイルは styles.xml file で定義されます。sheet1.xml.rels ファイルは、sheet1.xml 上で記述されたスプレッドシートの関連を記述します。特に sheet1.xml.rels はスプレッドシートに背景イメージがあるかどうか

を定義します。

4D Office Library には上記のファイルを作成するメソッドが実装されています。これらのメソッドは以下のパターンを使用して命名されています:

Create_<name-of-file-being-created>

以下のコードは Create_document.xml メソッドから取り出したもので、以下のように動作します: 選択されたレコードのフィールドをそれぞれ検索し、重要なデータを XML ファイルに追加します。例えば Word や Excel ドキュメントにバインドされたフィールドのデータなどです。このタスクを実行するため、4D Office Library は XML DOM コマンドを使用して XML タグを作成し、データを割り当てます。選択されたデータの検索が終了して XML ツリーが完成すると、4D Office Library はファイルを Resources フォルダ内の作業フォルダ ("word" または "xl") に保存します。

```
.
.
.
For ($i;2;numOfRecords+1)
    XPath:="/w:document/w:body/w:tbl/w:tr["+String($i)+"]/"
    xElement:=DOM Create XML element (document.xml_Ref;XPath;"w:rsidR";"008F27FD";
"w:rsidRPr";"008F27FD";"w:rsidTr";"008F27FD")

    For ($j;1;numOfFields)

        fieldName:="[<>tableName+]">OFOBJL_Bound{$j}
        EXECUTE FORMULA("fieldType:=Type("+fieldName+")")

        XPath:="/w:document/w:body/w:tbl/w:tr["+String($i)+"]/w:tc["+String($j)+"]/"
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:tcPr/w:tcW/";"w:w";"5000";
"w:type";"dxa")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/";"w:rsidR";"008F27FD";
"w:rsidRPr";"008F27FD";"w:rsidRDefault";"008F27FD";"w:rsidP";"008F27FD")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:pPr/w:
contextualSpacing/")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:pPr/w:jc/";"w:val";
"center")
        vxElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:pPr/w:rPr/w:rFonts/";
"w:ascii";"Verdana";"w:hAnsi";"Verdana")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:pPr/w:rPr/w:sz/";
"w:val";"24")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:pPr/w:rPr/w:szCs/";
"w:val";"24")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:r/";"w:rsidRPr";
"008F27FD")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:r/w:rPr/w:rFonts/";
"w:ascii";"Verdana";"w:hAnsi";"Verdana")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:r/w:rPr/w:sz/";
"w:val";"24")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:r/w:rPr/w:szCs/";
"w:val";"24")
        xElement:=DOM Create XML element(document.xml_Ref;XPath+"w:p/w:r/w:t/")
```

```

If ((fieldType#3) & (fieldType#30))

    EXECUTE FORMULA("fieldValue:=String("+fieldName+")")
    DOM SET XML ELEMENT VALUE(xElement;fieldValue)

End if

End for

NEXT RECORD

End for
.
.
.

```

- 2) 二番目のステップはすべての XML ファイル、および背景やロゴとして使用される画像を圧縮し、ZIP コンテナに格納することです。この処理は CreateWordDocument や CreateExcelDocument から LAUNCH EXTERNAL PROCESS コマンドを呼び出して行います。基本的に作業フォルダの内容が既存の word.zip や excel.zip ファイルに圧縮されます。ZIP ファイルは既に存在することに留意してください。この中には完全な Office ドキュメントとして必要な他のすべてのファイルが格納されています。4D Office Library は作成したファイルのみを追加します。ZIP ファイルは Resources フォルダにあります。

CreateWordDocument メソッドから取り出した以下のコードの最初の 4 行が、今説明したことを行います。

```

.
.
.
.
$cmd:=appPath+"7za.exe a word.zip word\\*"
SET ENVIRONMENT VARIABLE( "_4D_OPTION_CURRENT_DIRECTORY";appPath)
SET ENVIRONMENT VARIABLE( "_4D_OPTION_HIDE_CONSOLE";"true")
LAUNCH EXTERNAL PROCESS($cmd)
$cmd:="cmd.exe /c copy /y word.zip word.docx"
SET ENVIRONMENT VARIABLE( "_4D_OPTION_CURRENT_DIRECTORY";appPath)
SET ENVIRONMENT VARIABLE( "_4D_OPTION_HIDE_CONSOLE";"true")
LAUNCH EXTERNAL PROCESS($cmd)
.
.
.

```

- 3) 最後のステップは、ZIP ファイルの名称を Word や Excel の拡張子、docx や .xlsx に変更することです。4D Office Library は ZIP ファイルを再利用するので、ZIP ファイルを同じフォルダにコピーして word.docx や excel.xlsx といった名称を付けます。そして元の ZIP ファイルはとっておきます。この手順は先のコードの後半 4 行で行われています。この処理も先と同様、LAUNCH EXTERNAL PROCESS コマンドを使用して行われています。

Outlook やアドレスブックとデータ交換を行うために、4D Office Library はスクリプトを生成、使用します。スクリプトはコンピュータプログラムコードであり、相対的に易しいプログラム言語で記述します。これらは一連のテキストコマンドであり、スクリプトの処理器で解釈される特別なフォーマットで書かれます。4D Office Library は Windows 上で Visual Basic Scripts を、Mac 上で Applescript を使用します。

4D Office Library の実装中で、Create_ExportContacts.vbs, Create_ImportContacts.vbs, そして Create_ImportAppointments.vbs は Windows 用の Visual Basic scripts を生成するメソッドで、Create_ExportContacts.scpt と Create_ImportContacts.scpt は Mac 用の Applescripts を生成します。

```
.
.
.
$script:=$script+"\t\tNext"+$CrLf
$script:=$script+"\t Else"+$CrLf
$script:=$script+"\t\tErr.Clear"+$CrLf
$script:=$script+"\t End If"+$CrLf
$script:=$script+$CrLf
$script:=$script+"\tWScript.StdOut.Write strOutput 'Writes all contact info back to
calling application"+$CrLf
$script:=$script+"\tSet objContacts = Nothing"+$CrLf
$script:=$script+"End Sub"+$CrLf
$script:=$script+$CrLf
$script:=$script+$CrLf
$script:=$script+" "

TEXT TO BLOB($script;$tblob;UTF8 text without length )
$docref:=Create document("\\Scripts\\ImportContacts.vbs")

If (OK=1)

    CLOSE DOCUMENT($docref)
    BLOB TO DOCUMENT(DOCUMENT;$tblob)

End if
```

このコードは Create_ImportContacts.vbs メソッドの一部で、スクリプトをどのように生成しているかを示しています。メソッドではテキスト変数を作成し、スクリプトコマンドを追加して、ImportContacts.vbs ドキュメントに保存しています。スクリプトはコンポーネントストラクチャと同階層の Scripts フォルダに置かれます。

スクリプトの実行は以下のシンタックスを使用して動作します:

```
cscript myScript.vbs param1 param2 ...
```

ここで **cscript** は **myScript.vbs** を実行するために使用されるコマンドで、**param1, param2, ...** はスクリプトで使用される引数です。スクリプトが引数をとるかどうかは実装に基づきます。結果は標準出力に送信されまた、ファイルや変数にリダイレクトされる場合もあります。

連絡先やイベントを読み込むために、4D Office Library は (Outlook やアドレスブックなどの) アプリケーションにデータ (連絡先やイベント) を要求するスクリプトを使用します。結果は以下のパターンを使用して標準出力に返されます: 連絡先やイベントのフィールドはタブで区切られ、レコードはラインブレークで区切られます。連絡先を書き出すために、スクリプトは選択されたレコードごとに一回呼び出され、フィールドの値は引数としてスペースで区切られて渡されます。

先に示した通り、スクリプトはスクリプト処理器によって解釈されます。スクリプトを実行するためにはコマンドを使用しなければなりません。Visual Basic scripts は **cscript** コマンドを使用して呼び出され、Applescript は **osascript** コマンドを使用して呼び出されます。これは後述の ImportContacts メソッドでお見せします。 **cscript** コマンドに関する詳細は

<http://technet.microsoft.com/ja-jp/library/bb490816.aspx> を、**osascript** に関する詳細は <http://developer.apple.com/DOCUMENTATION/DARWIN/Reference/ManPages/man1/osascript.1.html> を参照してください。

4D Office Library は **LAUNCH EXTERNAL PROCESS** コマンドを使用してスクリプトを実行します。スクリプトの結果は OS の標準出力に送られ、テキスト変数 **\$output** でキャプチャされます。以下の ImportContacts メソッドでこのコードをお見せします。その後 **\$output** の内容が解析され、連絡先やイベントがレコードとして保存されます。

```
.
.
.
If (<>pathSpacer="\\" ) `Windows
    ` get script path (where to create/save/delete the script file)
    $scriptPath:=$scriptPath+"ImportContacts.vbs"

    If (Test path name($scriptPath)#Is a document )
        Create_ImportContacts.vbs `Writes the entire script contents to the variable vScript.
    End if

    $cmd:="cscript //Nologo //B "+$DQ+$scriptPath+$DQ+" "+$DQ+"Get Contacts"+$DQ+"
    "+$DQ+vCategory+$DQ

    Else

        ` get script path
        $scriptPath:=$scriptPath+"ImportContacts.scpt"

    If (Test path name($scriptPath)#Is a document )
```

```

Create_ImportContacts.scpt
End if

$scriptPath:=Replace string(Substring($scriptPath;Position(":";$scriptPath)+1);";":"/")
$scriptPath:="/"+Replace string($scriptPath;Char(Space );"\\"+Char(Space ))

$cmd:="osascript "+$DQ+$scriptPath+$DQ

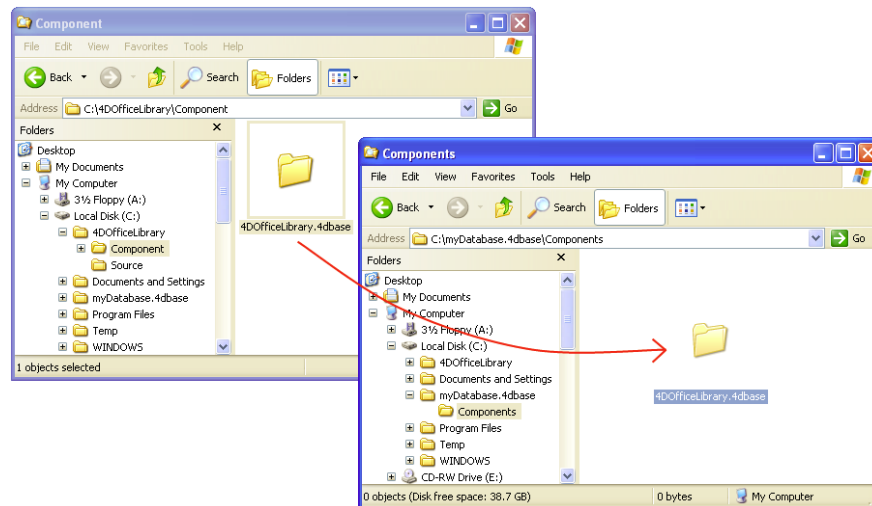
End if

SET ENVIRONMENT VARIABLE( "_4D_OPTION_HIDE_CONSOLE";"true" )
LAUNCH EXTERNAL PROCESS( $cmd;$input;$output )
.
.
.

```

4D Office Library のインストール

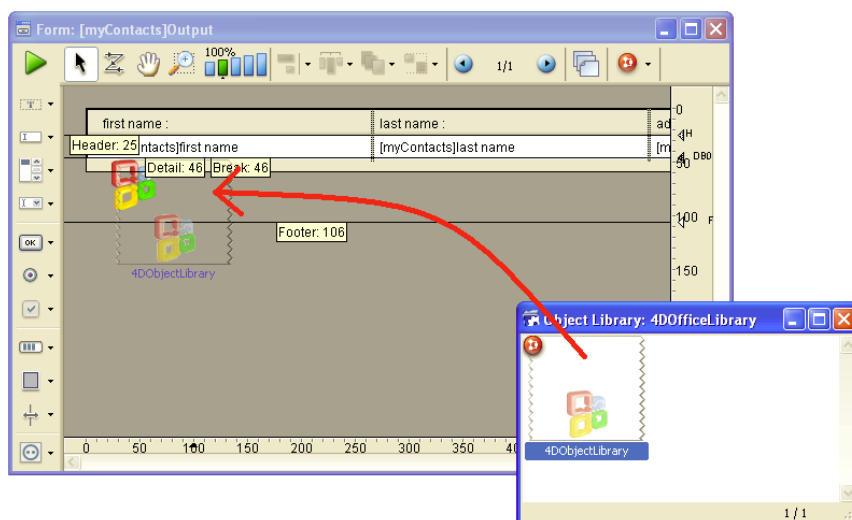
4D Office Library をインストールするには、4DOfficeLibrary.4dbase コンポーネントをホストデータベースの Components フォルダにコピーします。エイリアス (Mac) やショートカット (Windows) を使用することもできます。



Components フォルダはホストデータベースのストラクチャファイルと同階層になければなりません。コンパイルされ 4D Volume Desktop と統合されたデータベースでは、Components フォルダを実行可能ファイルと同階層 (Windows) またはパッケージ内の Contents フォルダ内に配置します。

4D Office Library を使用する

4D Office Library はデータベースの出力フォームで動作するようにデザインされています。4D Office Library を使用するには、オブジェクトライブラリから 4DObjectLibrary をドラッグして出力フォームに配置します。ブレークとフッタラインの間に、オブジェクトを配置するための間隔が空いていることを確認してください。

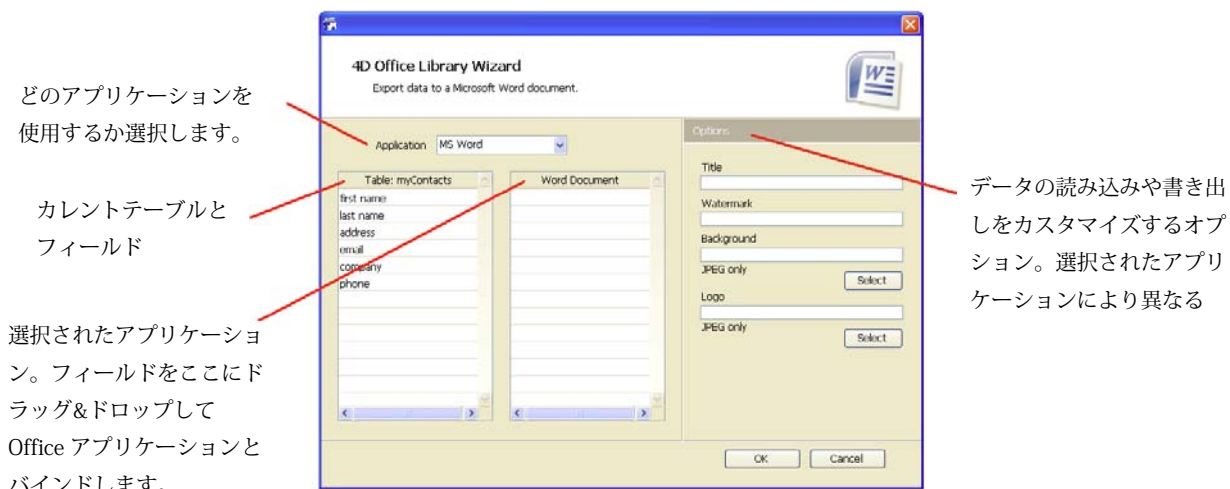


4D Office Library を使用するために、出力フォームで 1 つ以上のレコードを選択し、4DOfficeLibrary オブジェクトをクリックしてください。少なくとも 1 つ以上のレコードが選択されていなければなりません。連続したレコードを選択するには先頭のレコードをクリックし、Shift キーを押しながら最後のレコードをクリックします。複数の非連続レコードを選択するには、CTRL キー (Windows) またはコマンドキー (Mac) を押しながらレコードをクリックします。選択されたレコードはハイライトされます。

The screenshot shows the 'testingOOL - myContacts: 5 of 5' window. It displays a table with 5 rows of contact data. The first row is highlighted.

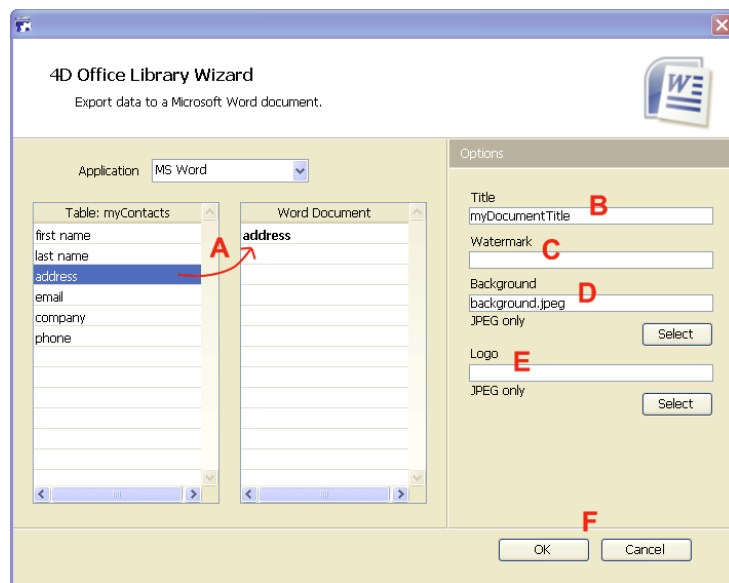
first name :	last name :	address :	email :	company :
James	Brown	123 A st	jbrown@email.com	4D Inc
Richard	Clark	2525 Winchester R	rclark@email.com	4D Inc.
Mark	Smith	845 Monroe St	msmith@email.com	4D Inc.
Robert	Mills	2121 Stevens Cree	rmills@email.com	4D Inc
Peter	Griffin	56 Old St	pgriffin@email.com	4D Inc.

4DOfficeLibrary がクリックされると、4D Office Library GUI が表示されます:



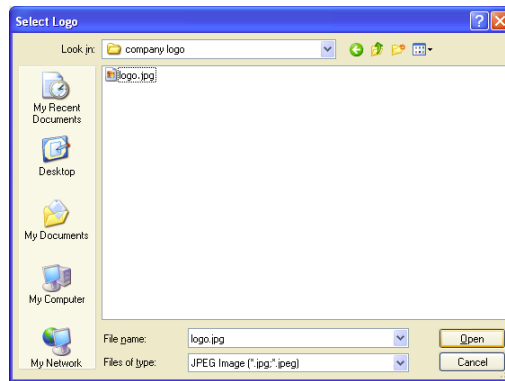
Word

このオプションでは選択したレコードの指定したフィールドを Word ドキュメントに書き出すことができます。以下は Word を選択した際の 4D Office Library GUI です:

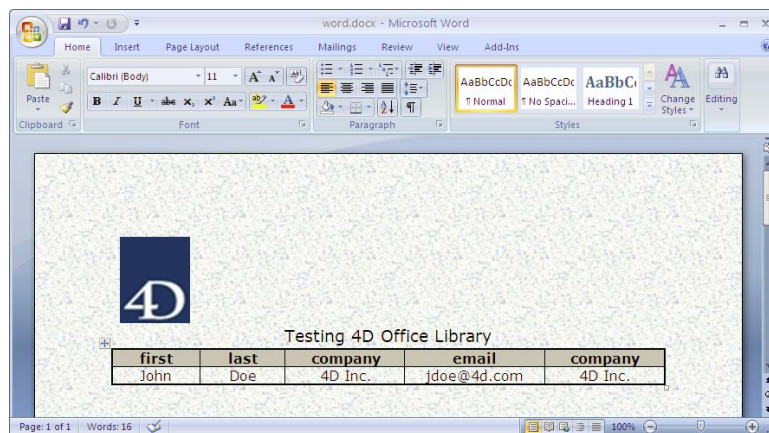


- A) テーブルエリア (左側のリストボックス) からフィールドをドラッグして、右側のリストボックスエリアにドロップします。並び替えを行うこともできます。
- B) ドキュメントにタイトルを追加するには、Title フィールドに入力します。

- C) Watermark フィールドに透かしを指定できます。
- D) オプションで JPEG フォーマットの画像を背景として指定できます。
- E) ロゴも追加できます。ロゴはドキュメントの左上部に配置されます。

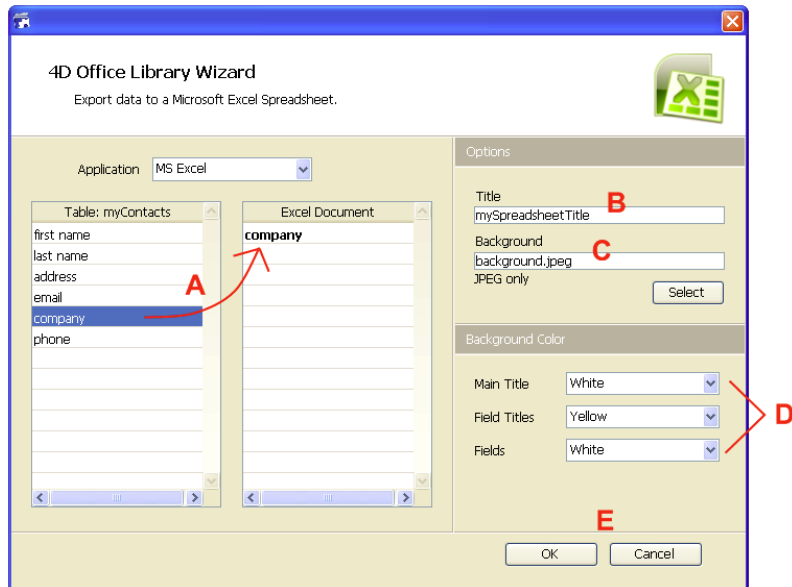


- F) OK をクリックすると、選択されたレコードのフィールドデータが記述された Word ドキュメントが自動で開きます。キャンセルをクリックすると処理がキャンセルされます。以下は作成される Word ドキュメントの一例です。

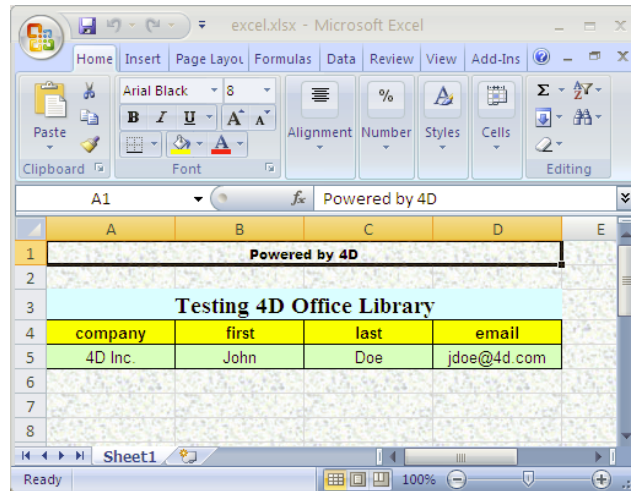


Excel

Excel アプリケーションが選択されると、ユーザは選択したレコードのフィールドを Excel スプレッドシートに書き出せます:

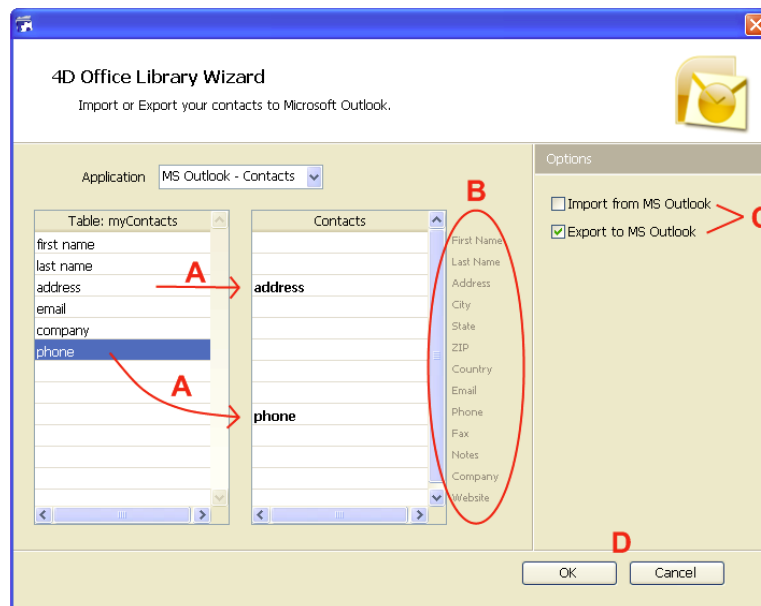


- A) テーブルエリア (左側のリストボックス) からフィールドをドラッグして、右側のリストボックスエリアにドロップします。並び替えを行うこともできます。
- B) ドキュメントにタイトルを追加するには、Title フィールドに入力します。
- C) オプションで JPEG フォーマットの画像を背景として指定できます。
- D) メインタイトル、フィールドタイトル、そしてフィールドのカラーを指定します。これらはデフォルトで白です。
- E) OK をクリックすると、選択されたレコードのフィールドデータが記述された Excel ドキュメントが自動で開きます。キャンセルをクリックすると処理がキャンセルされます。以下は作成される Excel ドキュメントの一例です。



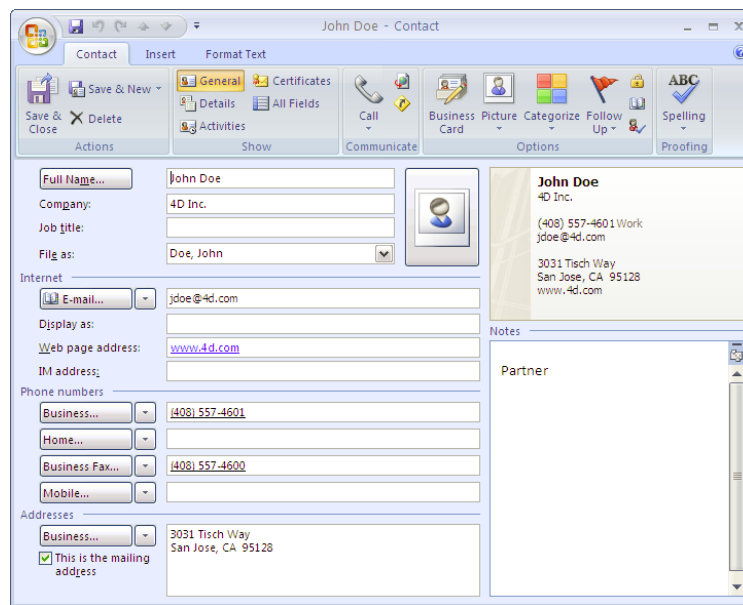
Outlook の連絡先とアドレスブック

このオプションは MS Outlook (Windows) やアドレスブック (Mac) との間で連絡先を読み込み/書き込みできるようにします。

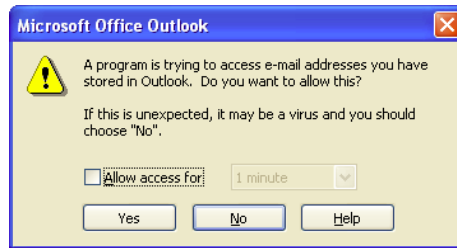


- A) テーブルエリア (左側のリストボックス) からフィールドをドラッグして、右側のリストボックスエリアの対応するフィールドにドロップします。対応する Outlook やアドレスブックのフィールドは B に表示されます。

- B) このエリアに表示されたフィールドは Outlook やアドレスブックの新規連絡先フィールドの一部であり、カレントのデータベーステーブルのフィールドとバインドできます。Outlook において、住所を構成するフィールド (address, city, state, zip, そして country) や phone および fax number は仕事用のアドレスに対応する点に留意してください。(Outlook はホーム、ビジネスなど複数の住所や電話番号を定義可能です。)
- C) 選択したレコードを書き出すには"Export to MS Outlook"または"Export to Address Book"を選択します。読み込みを行うには"Import from MS Outlook"または"Import from Address Book"を選択します。
- D) OK をクリックすると新しい連絡先が読み込み、あるいは選択した連絡先が書き出しされます。キャンセルをクリックすると処理がキャンセルされます。以下は選択されたレコードから書き出されたデータが入力された Outlook 連絡先の例です。

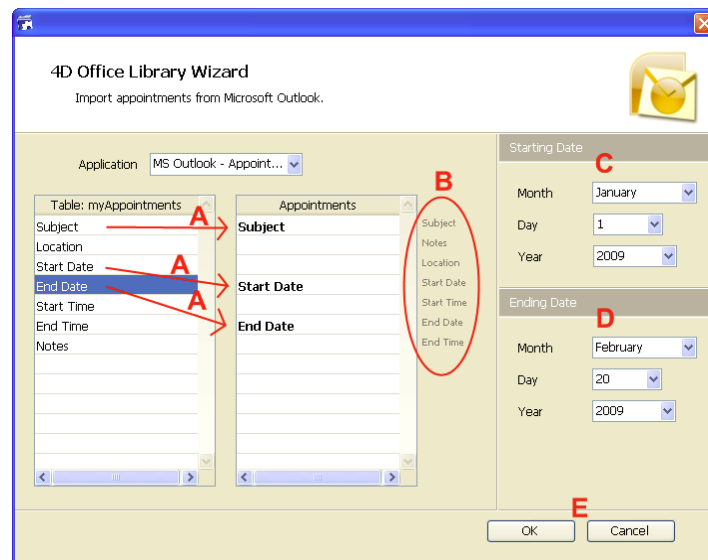


4D Office Library が MS Outlook から連絡先を読み込む際は、警告メッセージが表示され、Outlook へのアクセスを許可するかしないかを尋ねられます。読み込みを行うためには、はいをクリックします。



Outlook カレンダー (Windows のみ)

このオプションを使用して MS Outlook の予定表を読み込みできます。



- A) テーブルエリア (左側のリストボックス) からフィールドをドラッグして、右側のリストボックスエリアの対応するフィールドにドロップします。対応する Outlook の予定表フィールドは B に表示されます。
- B) このエリアに表示されたフィールドは Outlook の新規予定フィールドの一部であり、カレンダーのデータベーステーブルのフィールドとバインドできます。
- C) 読み込む予定の開始日を選択します。
- D) 読み込む予定の終了日を選択します。
- E) OK をクリックすると指定された日付の予定が選択されたテーブルに読み込まれます。

連絡の読み込みと同様に警告メッセージが表示され、Outlook へのアクセスを許可するかしないかを尋ねられます。読み込みを行うためには、はいをクリックします。

まとめ

このテクニカルノートでは 4D Office Library for 4D v11 SQL の機能について記述しました。コンポーネントのインストールからライブラリの使用法まで、すべてのステップを説明しました。また動作を実行する際のステップの実装の説明も行いました。

参照

4D 2004 Office Object Library:

http://www.4d.com/products/office_ol.html

Office 2007 open XML file formats:

<http://msdn.microsoft.com/ja-jp/library/aa338205.aspx>

7-Zip:

<http://www.7-zip.org/>