



Standard Edition



Professional Edition



Classic Edition



Reporting Edition

前編

後編

5. VS-VIEW 7.0J Classic Edition	7
▶ VS-VIEWの特徴	7
▶ VSPrinterコントロール	7
▶ 基本的な出力	7
▶ VSPrinterコントロールの機能	8
▶ 高度な表の作成	8
▶ エクスポート機能	9
▶ テキスト検索とスクロール機能	10
▶ VSDrawコントロール	10
▶ VSPDFコントロール	11
▶ VSViewPortコントロール	11
6. VS-VIEW 7.0J Classic Editionのまとめ	11
7. VS-VIEW 7.0J Reporting Edition	11
▶ VS-VIEW Reporting Editionの構成	11
▶ VSReportデザイナー	11
▶ Accessファイルからのインポート	13
▶ VSReportコントロール	13
8. VS-VIEW 7.0J Reporting Editionのまとめ	13
9. 比較評価	14

5. VS-VIEW 7.0J Classic Edition

VS-VIEWの特徴

VS-VIEWは、VSPrinter、VSDraw、VSViewPort、VSPDFの4つのコントロールから構成されます。多彩なプロパティやメソッドの装備により細かな制御が可能で、帳票系の出力にとどまらず、あらゆる形式の出力結果を得ることができます。

VSPrinterコントロール

VSPrinterコントロールはVBのPrinterオブジェクトを高機能に拡張したコントロールで、VS-VIEWの中心となるコントロールです。

基本的な出力

VSPrinterコントロールへの出力は、完全にコード記述で行います。ここでは簡単な例で基本的な出力を行ってみます。

1. VBの新規プロジェクトを作成し、コンポーネントメニューからVSPrinterコントロールを参照し、(図16)のようにコマンドボタンとともにフォームに貼ります。
2. フォームへ下記のコードを記述します。

Option Explicit

```
Private Sub SetTitle(vp As VSPrinter, strDat As String)
```

```
    With vp
```

```
        .Font.Bold = True
```

```
        .Font.Italic = True
```

```
        .Font.Underline = True
```

```
        .Font.Size = 18
```

```
        .TextColor = vbGreen
```

```
        .Text = strDat
```

```
        .Font.Bold = False
```

```
        .Font.Italic = False
```

```
        .Font.Underline = False
```

```
        .Font.Size = 12
```

```
        .TextColor = 0
```

```
    End With
```

```
End Sub
```

```
Private Sub cmdPreview_Click()
```

```
    Dim strFormat As String
```

```
    Dim strHeader As String
```

```
    Dim strData As String
```

```
    With VSPrinter1
```

```
        .Preview = True
```

```
        .StartDoc
```

```
        Call SetTitle(VSPrinter1, "文字の出力")
```

```
        .Paragraph = ""
```

```
        .Paragraph = ""
```

```
        .Text = "VS-VIEWは、VSPrinter、VSDraw、VSViewPort、VSPDFの"  
            & vbCrLf & "4つのコントロールから構成されます。"
```

```
        .Paragraph = ""
```

```
        .Paragraph = ""
```

```
        Call SetTitle(VSPrinter1, "表の出力")
```

```
        .Paragraph = ""
```

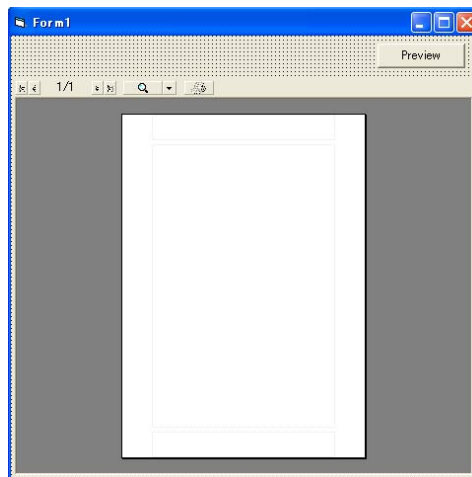
```
        .Paragraph = ""
```

```
        strFormat = ">+2000|<+2000|>+1500|^+1000|^+2000;"
```

```
        strHeader = "EmployeeID|Name|BirthDate|Country|HomePhone;"
```

```
        strData = "1|Davolio Nancy|08-Dec-48|USA|(206) 555-9857;"
```

(図16) VSPrinterコントロール



```

& "2|Fuller Andrew|19-Feb-52|USA|(206) 555-9482;" _
& "3|Leverling Janet|30-Aug-63|USA|(206) 555-3412;" _
& "4|Peacock Margaret|19-Sep-37|USA|(206) 555-8122;" _
& "5|Buchanan Steven|04-Mar-55|UK|(71) 555-4848;"

```

```

.TableBorder = tbAll
.Table = strFormat & strHeader & strData
.EndDoc
End With
End Sub

```

3. 実行して結果を確認したのが(図17)です。

(図17)実行結果



この例では、まずVSPrinterコントロールの.PreviewプロパティをTureに設定し、プレビューモードにしてから、.StartDocメソッドで印刷ジョブを開始しています。次に、SetTitleというプロシージャでタイトルを表示させてから.Textプロパティへ文字列をセットして出力、また、列幅と列内の文字位置を指定したフォーマット文字列を含めたデータを.Tableプロパティへセットして、表を出力します。最後に.EndDocメソッドで印刷ジョブを終了しています。間に入れている.Paragraph = "" は、段落の変更をあらわします。

VSPrinterコントロールの機能

VSPrinterコントロールの基本的な出力方法がわかったところで、このコントロールの優れた機能について見てみることにします。

高度な表の作成

7ページ「基本的な出力」で表の出力を行いました。VS-VIEWでは各セルのフォーマットの変更、行や列のマージといった、さらに高度な表を作成することができます。7~8ページ「基本的な出力」のコードへ次のようなコードを追記して実行したものが(図18)です。

```

` 2つ目の表
.Paragraph = ""
.Paragraph = ""
.StartTable
.AddTable strFormat, strHeader, strData, &H1, &HC0FFFF, False
.TableCell(tcForeColor, 0, 0, 0, 5) = &HFFFFFF
.TableCell(tcFontBold, 0, 0, 0, 5) = True
.EndTable

` 3つ目の表
.Paragraph = ""
.Paragraph = ""
.StartTable
.AddTable strFormat, strHeader, strData, &H808000, &HC0C0C0, False
.TableCell(tcForeColor, 0, 0, 0, 5) = &HFFFFFF
.TableCell(tcFontBold, 0, 0, 0, 5) = True

`セルのマージとスタイルの設定
.TableCell(tcFontBold, 2, 4) = True
.TableCell(tcBackColor, 2, 4) = vbYellow
.TableCell(tcForeColor, 2, 4) = vbRed
.TableCell(tcColSpan, 2, 4) = 2

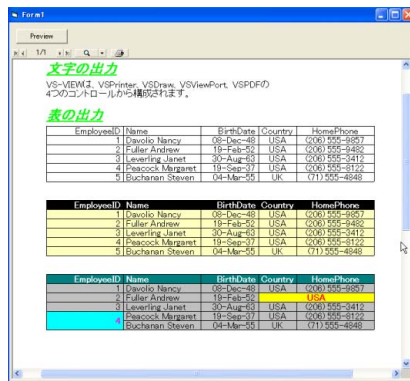
.TableCell(tcFontBold, 4, 1) = True
.TableCell(tcBackColor, 4, 1) = vbCyan
.TableCell(tcForeColor, 4, 1) = vbMagenta
.TableCell(tcRowSpan, 4, 1) = 2
.EndTable

```

2つ目の表はAddTableメソッドで作成し、その時点で表のヘッダや本体の色を指定し、.TableCellプロパティでヘッダ行のスタイルを変更しています。3つ目の表は、2行4列目のセルの行をマージ、4行1列目は列方向にマージを行っています。

また、これ以外にもNewPage()イベントを利用して、複数行のヘッダを持つ表を作成するといったこともできます。

(図18)いろいろな表出力



エクスポート機能

VS-VIEWにもエクスポートの機能があります。エクスポートできるファイルの種類は、HTMLとRTFです(PDFについては、後述するPDFコントロールでエクスポート可能です)。

エクスポートは、ドキュメントを作成する前に.ExportFormatプロパティに出力するファイルの形式を、ExportFileプロパティにエクスポートファイル名をそれぞれ設定しておくだけです。前項の「高度な表の作成」で作成したレポートの場合、

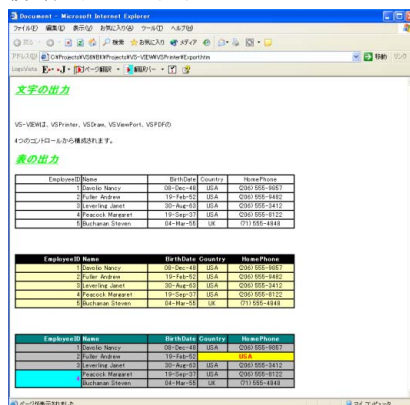
With VSPrinter1

```
.ExportFormat = vpxPlainHTML
.ExportFile = App.Path & "¥Export.htm"
.Preview = True
...
```

のようにしてエクスポートします(図19)。

また、.ExportRowプロパティでHTML内へスクリプトを挿入することも可能です。

(図19)データのエクスポート

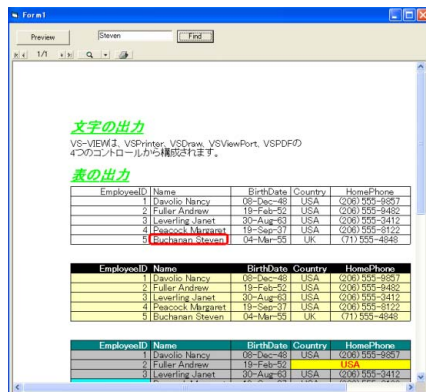


テキスト検索とスクロール機能

指定した文字列をドキュメント内から検索することができます。検索は、FindTextメソッドで行い、検索する文字列、大文字と小文字を区別するか否か、検索を開始するページ、検索を終了するページ、検索を開始する最初のページ上の位置を引数として渡します。このメソッドは、文字列が見つかった場合にはページ番号が返され、またVSPrinterのX1、Y1、X2、Y2の各プロパティに文字列の座標がセットされます。文字列が見つからなかった場合には-1が返されます。7ページ「基本的な出力」のフォームへコマンドボタンとテキストボックスを追加し、次のようなコードを追記して実行したものが(図20)です。

```
Private Sub cmdFind_Click()
Dim intPage As Integer
With VSPrinter1
.PenWidth = 60
.PenColor = vbRed
.PenStyle = psSolid
.BrushStyle = bsTransparent
intPage = .FindText(txtFind.Text, , intPage)
If intPage <= 0 Then Exit Sub
.StartOverlay intPage
.DrawRectangle .X1 - 100, .Y1, .X2 + 100, .Y2
.EndOverlay
.PreviewPage = intPage
.ScrollIntoView .X1, .Y1, .X2, .Y2
End With
End Sub
```

(図20)検索とスクロール



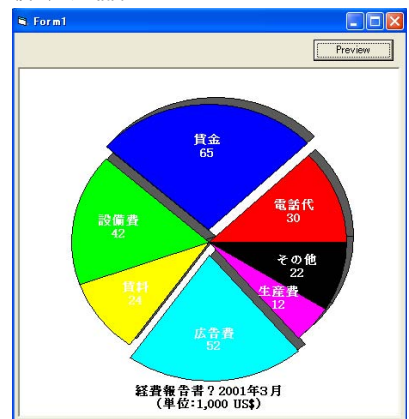
この例では、FindTextで返されたページのX1、Y1、X2、Y2の座標にDrawRectangleメソッドで角丸四角形を描画し、ScrollIntoViewメソッドでその座標が表示領域内に現れるようにスクロールしています。

このほかにも、ナビゲーションバーのカスタマイズ、VS-FlexGrid Proとの連携、目次の挿入とそのクリックによるページの表示、スタイルの一括定義、多数のプロパティやメソッドでの精密なカスタマイズ、Visual C++やInternet Explorerでの使用といった多くの機能が内包され、プロフェッショナルユーザのコントロールになっています。

VSDrawコントロール

VSDrawコントロールは図形を描画、出力するコントロールで、単純な図形から地図、グラフなど、必要に応じたさまざまなグラフィックを出力することが可能です(図21)「VS-VIEW 7.0Jオンラインヘルプ内サンプルより」。

(図21)グラフ描画

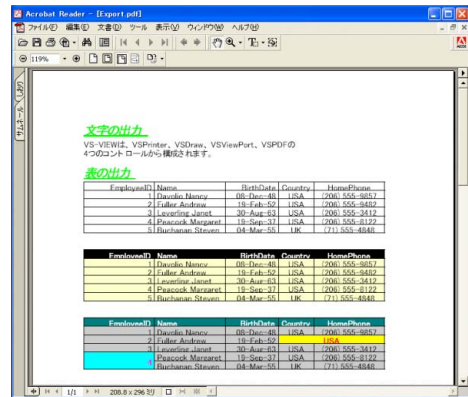


VSPDFコントロール

VSPDFコントロールは、VSPrinterコントロールで作成したプリントドキュメントをPDFファイルへ変換することができるコントロールです。使用法はいたって簡単で、VSPrinterコントロールへのプリントドキュメントの出力の後、VSPDFコントロールの.ConvertDocumentメソッドの引数にVSPrinterコントロールオブジェクトとPDFファイルのファイル名を指定するだけです。

7 ページ「基本的な出力」のレポートをPDF出力したものが(図22)です。

(図22)PDFファイルへの出力



VSViewportコントロール

VSViewportコントロールは、スクロール可能なコンテナコントロールです。VirtualHeight、VirtualWidthという仮想領域をあらゆるプロパティに値を設定することによりスクロールを行えるようにするもので、PictureBoxとScrollBarを同期させるといった、よく行われるコード記述が不要になります。表示領域よりも大きな入力用フォームへのUIをこのコントロールで作成し、VSPrinterコントロールでそのフォームの内容をプリントするなどの使用方法があります。

このコントロールには、アクティブなコントロールの位置までビューを自動的にスクロールするFocusTrackプロパティがあり、入力フォーカスを持つコントロールを自動的にユーザーが見える位置にスクロールすることなどができます。

6. VS-VIEW 7.0J Classic Editionのまとめ

このようにVS-VIEW 7.0J Classic Editionは、ActiveReportsのようにビジュアルに帳票デザインするといったことは不可能ですが、細かな制御が可能でどのような印刷物にも対応することができるコントロールです。600ページ近いマニュアルのすべてのプロパティ、メソッドを知り尽くし使いこなすには多くの習熟期間が必要ですが、このコントロールだけでプリント処理のプロフェッショナルになれることは確実です。また、このコントロールのすべてを知ることが無理でも、自分に必要な機能だけをうまく使うことができれば、あらゆるニーズに柔軟に対応する、使い勝手の良いコントロールといえます。

7. VS-VIEW 7.0J Reporting Edition

VS-VIEW 7.0J Reporting Editionの構成

VS-VIEWの上位エディションであるReporting Editionは、Classic Editionのすべての機能に加え、元来VS-VIEWが苦手としたレポートのビジュアルなデザインを可能にするVSReportデザイナと、そのデザイナで作成したXML形式のレポート定義からVSPrinterコントロールを介してレポートを生成できるVSReportコントロールが付属しています。

VSReportデザイナ

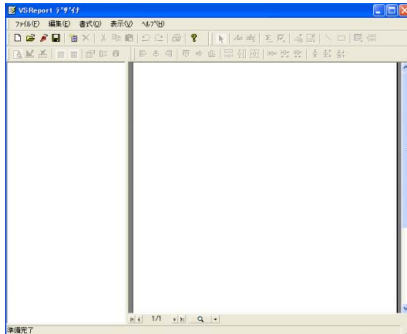
VSReportデザイナは、VSReportコントロールで読み込み可能なレポート定義ファイルの作成、および編集に使用するVSRpt7.exeという実行形式の単一アプリケーションです。このアプリケーションはレポートに基づくデータソースから、簡単にレポートの定義を作成できることはもとより、Microsoft Accessファイル(MDB)からレポートをインポートすることも可能です。

これもNWIND.MDBのデータをもとにレポート定義ファイルを作成してみます。

1. スタートメニューから〈PowerTools VS-VIEW Ver.7.0J Reporting Edition〉→〈VSReportデザイナ〉、または、インストールフォルダ¥VSVIEW7RE¥Controls¥VSRpt7.exeを起動します(図23)。
2. 〈ファイル(F)〉→〈レポートの新規作成(R)〉メニューでレポート作成ウィザードが起動します。
3. 接続文字列とテーブル、またはSQL文字列を指定します。この例では(図24)のように指定しました。
4. 利用可能なフィールドリストからグループ化するフィールドと詳細セクションに表示するフィールドを選択し、右のボックスへドラッグ&ドロップします(図25)。
5. レポートのフォントや形式を選択するダイアログで各項目を選択し、最後にレポート名を入力します。これまでの操作で(図26)のようなレポートデザインが完成しました。
6. データベースのPhotoフィールドの写真も表示させるため、デザイナをデザインモードとし、ツールバーのバウンディングボックスボタンで「Photo」フィールドを選択して詳細セクションに貼り、プレビューしてみます(図27)。
7. 〈ファイル(F)〉→〈名前をつけて保存(A..)〉メニューでレポート定義として保存します。

このようにデータベースを基にしたレポート定義が簡単に作成できます。

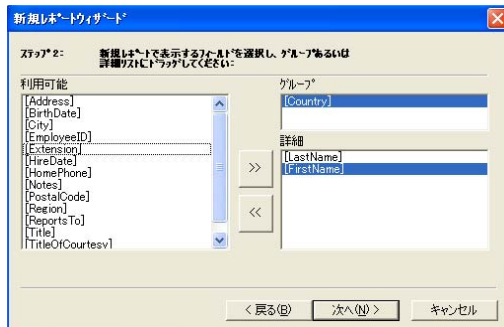
(図23) VSReportデザイナ



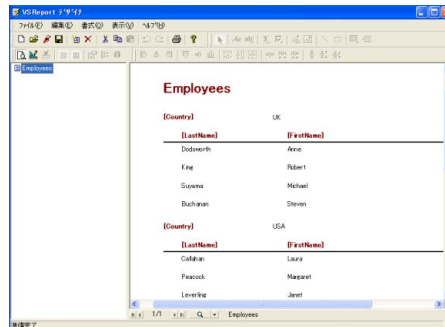
(図24) 接続文字列の指定



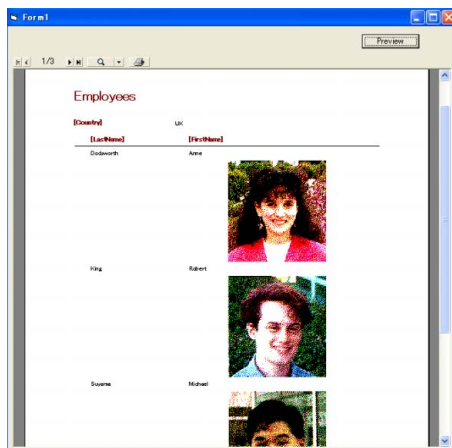
(図25) フィールドの選択



(図26) レポートデザイン



(図27) 完成したレポート



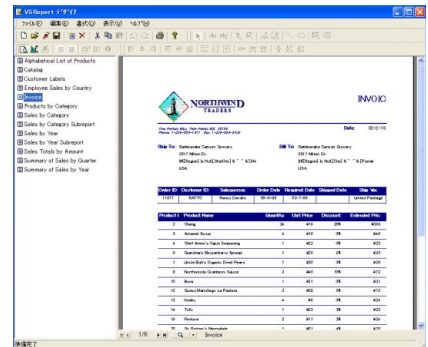
Accessファイルからのインポート

次にAccessのファイルからレポートをインポートしてみます。

これは<ファイル(F)>→<インポート(I)>メニューで表示されるファイル選択ダイアログで、Accessファイルを選択するだけの簡単な操作です。これだけでファイル内のすべてのレポートがインポートされ、プレビューできます(図28)。

このことは何を意味するかというと、今まで使い慣れたAccessのレポートデザイナーでデザインしたレポートをVS-VIEWですぐに使用することができ、過去にAccessで作成したレポートも無駄にはならないということです。すばらしく画期的なものだと思いますが、皆さんはどう思われますか？

(図28) Accessからインポート



VSReportコントロール

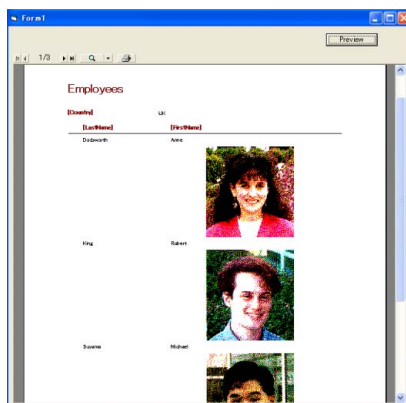
VSReportデザイナーで作成したレポート定義は、VSReportコントロールを使用してVSPrinterコントロールへ送ることが出来ます。先ほど作成した定義ファイルからドキュメントを生成するプロジェクトを作成してみます。

フォームへコマンドボタンとVSPrinterコントロール、それにVSReportコントロールを配し、コマンドボタンのクリックイベントプロシージャへ次のコードを記述します。

```
Private Sub cmdPreview_Click()  
    With VSReport1  
        .Load App.Path _  
            & "¥Employees.xml", "Employees"  
        .Render VSPrinter1  
    End With  
End Sub
```

これだけで(図29)のようなレポートを出力することができます。

(図29) VSReportコントロール



8. VS-VIEW 7.0J Reporting Editionのまとめ

今までVS-VIEWで不得意であった、レポートレイアウトの視覚的な作成を可能にしたReporting Edition。レポート作成という面で、できないことがないと言っても過言ではないVS-VIEWにこのビジュアル環境が加わって、レポート定義の作成時間短縮という面でもクリアできるものになっています。

9. 比較評価

「ビジュアルな環境でのレポート作成」のActiveReports、「あらゆる用途に対応できる細かな制御」が売りのVS-VIEWという双方の異なる思想。この思想が時を経て、ActiveReportsは多彩なイベントを得て細かな制御を、VS-VIEWはVSReportデザイナを手にしてビジュアル性を補完し完成度を高めるうちに、機能差というものほとんどなくなり、非常に似通った製品になっているようです。

当初は、ごく単純なデータを簡単な表形式で出力するだけのレポートを短い期間で作成したいのならActiveReports、複雑なデータで各ページ間のフォーマットも共通ではないようなレポートはVS-VIEWでと考えていたのですが、それぞれに弱点を強化して完成度の高い製品になっています。

レポート出力のための製品をこれから購入しようとしている方にとって、今回ご紹介した内容で判断が付くのか否かは判りかねますが、グレープシティ株式会社のホームページからトライアル版がダウンロードできますので、まずは使ってみてご判断されるのが賢明かと思えます。

最後に要望となりますが、定型用紙への出力という、かなり高頻度でニーズのあるシチュエーションにおいて、標準の伝票(GB〇〇請求書、PCF〇〇納品書 等)への出力を想定したスケルトンが用意されていれば便利なのではないでしょうか。

評価項目に対する各製品の対応を一覧して、この比較を終わります。

	ActiveReports		VS-VIEW	
	Standard	Professional	Classic	Reporting
文字列や数値などの配置	各コントロールのAlignmentプロパティで指定可		フォーマット文字列等で指定	デザイナのAlignプロパティで指定可
フォントのスタイルや色の指定	各コントロールのAlignmentプロパティで指定可		VSPrinterコントロールのプロパティをコードで操作	デザイナのプロパティで指定可
罫線の線種、太さ、色の指定	コントロールのプロパティで指定可		VSPrinterコントロールのプロパティをコードで操作	デザイナのプロパティで指定可
図形の種類、塗りつぶしの指定	図形の形は3種、塗りつぶし可		VSDrawコントロールでどのような図形も可	
画像の対応フォーマット	BMP, DIB, JPG, GIF, WMF, EMF, ICO, CUR			
バーコード印刷	BOC BarCode Control 1.0 付属			
出力対応形式	PDF, RTF, Text, Excel, TIFF, HTMLの各形式のExport Filterを使用		HTML・RTFはエクスポート可、PDFはVSPDFコントロール使用	
データとのリンク機能	専用のデータコントロール		VBのデータコントロールとのバウンド可	
各ページでの異フォーマット対応	イベントを駆使してコードで		コードで作成	
プリンタ選択・ページ設定機能	独自のダイアログで指定可		Windows標準のダイアログで指定可	
プレビューの方法	コントロールのビューで行なう			
定型帳票への対応	記述なし			
エンドユーザーのカスタマイズ	不可	可	不可	別途ライセンス契約により可
Web Cache	不可	可	不可	
他言語対応	不可		VC++対応	
マニュアル	445ページ共通		590ページ	Classic プラス276ページ
ヘルプ	VBから使用可のオンラインヘルプ			
サンプル	充実しており、サンプルのチュートリアルも付属			
価格(定価)	98,000	258,000	58,000	98,000

プロフィール

小森 大輔 氏
 愛知県在住。某大手自動車会社勤務のかたわら、Visual Basic Users Group 名古屋支部に所属するフリーのVBプログラマーで、執筆活動やカンファレンスのスピーカーとしても活躍。
 著作：
 「Office 2000 Developer プログラミング入門」(アスキー刊)
 Visual Basic Magazine (翔泳社) に連載の「コンポーネント工房」等