

# 1 Delphi 2.0 から引き継がれている特長

## 目次

2	はじめに
3	製品仕様
4	トライアル版について
5	トライアル版の使用方法
6	Delphi 3 の概要
8	サンプルプログラムのオープン
10	簡単なアプリケーションの作成
12	コード入力支援機能
14	コンポーネントテンプレート
15	デバッガの新機能
16	新しいコンポーネント
19	その他の新機能
20	ActiveX コントロールの作成 (*)
24	Decision Cube による多次元解析 (*)
26	マルチデータベースエンジンのサポート
27	リモートデータ制御 (*)
28	Web アプリケーションの開発 (*)
30	Delphi 2.0 から引き継がれている特長
32	Q&A

□□□^,^,□ffff",,,—,……,□

□^ %□-□,□Š",Delphi 3,,,,□,,,,□^,•Ž,,,,□□□□•^,,□□,,,,□,,,,□,—□,,,,□

Copyright (C) 1997 Borland International, All rights reserved.

All Borland product names are trademarks or registered trademarks of Borland International, Inc.

„□□•-,“,ŠŽ,□•,,“~□•,,□

□•,Ž—,□%—,,—□,•□,,,,,,□

## Delphi 2.0 から引き継がれている特長

Delphi8, Š

1995” □□, Delphi, “□, ^ □ Delphi, CE □ f f f f f , f f f f f f / f □ Š” Ž, □, • %  
 , Ž, □ • □, Ž Ž, “ , , , , , □ □ ‘ , f f f f □ f f f Š” , ” CE, f f f □ f f f , — , , □ < “ , □ Ž □ , , , , , , Delphi, □ ‘ □ Š, 60 — , , □  
 % □ Ž □, Ž, □ Ž” f f f , . f f f f f f / f □ f □ % f f f f □ f f f □ , , , f f f f f f % □ f f f □ f f f , Š” , • □, — , , , , , □  
 , , , □ NASA, f f f f f f f , f f f , f □ f  
 % □ f f □ f f f , □ ‘ < , Ž • Ž □, f f f f □ , , , , □ Delphi, Ž , , , □ , □ NASA, % Š < □ □ • f f f f , □ Delphi, Š” , , , , □ Info  
 World Ž, CE □ , , □ , , , , f f f f f f / f □ f □ f f f f □ f f f , , , , , ~ , , , , □  
 American Airlines, □ , , , , Ž, ‘ , , , , ” CE — “ , f f f f □ ” Ž ‘ , Ž ‘ %  
 , , , □ DB2, Ž, , SABRE f f f f , f f f f , f f f f □ f f f , Delphi, Š” , , , □  
 “ — , □ □ □ CE □ , “ Ž f □ f f f f CABINET, Š” , Delphi, □ — , , , , □ □ % □ □ f f f %  
 , Ž CE , , , □ CE □ CABINET, 4,500 ‘ , f f f f , CE, Š Š < — , f f f f , , Š — , , , , , □  
 , , □ □ “ Š , ‘ , Š < , Delphi, □ ‘ □ , CE — □ , ‘ , □ □ — , , , , , □ Delphi “ “ Ž — , , , , , http://www.borland.co.jp/  
 , — , , , , , □ □  
 , , □ f □ f f f , □ □ ‘ , < □ , ‘ < , , , , , □ , , , Ž , , , ‘ □ □ , ‘ < , , , , , , □ □ Ž , , , □ Delphi, Ž , , , , ‘ < , , < ” , — , □ Š □ , Ž —  
 , — “ , f f f f f f , , , • , □ , Ž, “ , , , , , □  
 , , □ , , , □ , , “ CE , , , , ” , , , , Delphi — , Ž □ □ □ , □ “ — CE , , , , , □ 10 □ , , , , □ “ — □ , □ “ , < □ , □ %  
 , , , , □ • □ , • — , , , , , , Š < , □ ” , , , , , □  
 “ □ CE • □ , f f f f □ f , □ Š, □ Delphi 3, □ □ ‘ , < □ , Ž , , , , CE , ‘ < , , □ f f f f □ f f f , • %  
 , □ , □ , , Active X f f f f □ f , Active f f □ f , □ □ □ CE — “ , f □ f □ — , Ž CE , , , , f f □ f f □ f f f f , Decision  
 Cube , , , , f □ f , Ž CE % □ □ ‘ □ , • CE, Ž, f f □ f f f f □ f f f □ f f f □ f f f f f f f f f f , ‘ %  
 , Web f □ f □ f f f f □ f f f , Š” , , □ Delphi, — , , , ” ^ , , , , □ , , , □  
 Š” , , , , Š ‘ — , , , , % □ , , , □ f □ f f f f f , f f f f f , Š □ ^ □ , • Ž □ ‘ □ “ — , Ž CE , “ —  
 , Ž Š, f f , CE , ‘ < ” Ž, □ , , , , , □ , □ f f f □ f f f , f □ f f □ , , , □ , f f f □ f f f , ‘ % □ f f f □ f , , , Ž □ f f f f f f f , □ □  
 % □ f f f f f , f f f □ f f f , □ CE, Ž CE , , f f f □ f f f f f f f □ f , , ‘ < ” , , □ , , , , Ž , , , , , , □  
 , , , □ f f f f f Š” , □ Ž □ □ □ — , , , f f f f f □ f f f f f f f □ f f f f f f f , □ f □ f f □ f ‘ % □ , □ □ — , , , , ^ CE , , , , □  
 Delphi 3, < □ , Š” Ž Š, — CE, — , □ , , , , f f f f □ f f f , Š” , □ □ , Š” f □ f , □ □ ” □ , < % □ , Delphi 3, , Ž , , , □  
 Ž □ , — , , , < ” , f f f □ f f f , , □ • CE ‘ , , , , ^ , □ □ , , , Ž f □ f , □ • Ž — , — , , , , □

### Delphi 2.0 から引き継がれている特長

Delphi 3,,□Standard”□STD□□Professional”□PRO□□Client/Server Suite”□CSS□,3,,□•CE’,,,,,□ffff”□Š”’,Standard”’,,,□□,’,’,’,□□

【動作環境】

CPU Intel 80486^□,CPU  
 fff 16MB^□,fff  
 f□fffff 60MB^□,f□fffff  
 OS “-CE”Windows95,,Windows NT 4.0/3.51(SP5)  
 ,,’ CD-ROMffff□fff,,fffffffffffff

【機能一覧】

↳”	STD	PRO	CSS
NEW! fff□f□fff□ffffffffff,DLL%□	✓	✓	✓
NEW! f□f“-Ž%□”	✓	✓	✓
NEW! f□ffffŽ•%□,DLLffff	✓	✓	✓
NEW! fff□ffffffffff□f	✓	✓	✓
NEW! fff□fff□f□fff□,Š,100^□,fff□fff	✓	✓	✓
NEW! ffff□ff□fffff,ff□f	✓	✓	✓
NEW! COMff□f,’□,	✓	✓	✓
NEW! fffff,←-,f□fff□•,-,□,	✓	✓	✓
NEW! Š’,,,QuickReports 2.0	✓	✓	✓
NEW! Š’,,,fffffffff	✓	✓	✓
NEW! ActiveXffff□f□Activeff□f,□□		✓	✓
NEW! TeeChart,,,’□,fff•Ž		✓	✓
NEW! HTML□□,Š,fff□ffffffffff□fff			✓
NEW! Decision Cube,,,’ŽCE%□			✓
NEW! ff□ff□ffff			✓
NEW! Webf□f□ffff□fffŠ”			✓
NEW! NSAPI/ISAPI,ff□f,,WebBridge			✓

【マニュアル】

Delphi f□f□ffff	✓	✓	✓
Delphi Š”Žfff	✓	✓	✓
Object Pascal(CE)fff	□	✓	✓
Delphi Visual Component Libraryffffffffff□2•□□	□	✓	✓
InterBase Server for Windows95&Windows NT f□f□ffff		✓	✓
InterBase Server f□f’□fff			✓
InterBase Server (CE)fffff			✓

□,’,’,’,□fffff,ffffffffffffffffff,Š’,’,’,□

## Delphi 2.0 から引き継がれている特長

Delphi 3 の Delphi 3 から引き継がれている特長

### Delphi 3 (トライアル版) 利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

### トライアル版の制限事項

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定

Delphi 3 の利用規定





## 7 Delphi 2.0 から引き継がれている特長

Delphi 3 を起動しただけでは、表示されないウィンドウもあります。こうしたウィンドウの中で主なものを解説します。

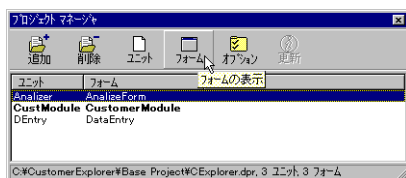
### 画面

### 名称と用途



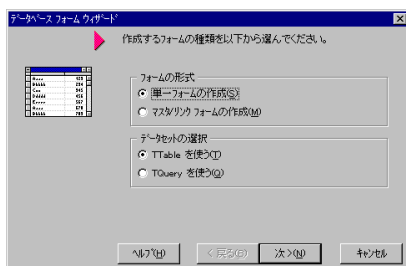
#### 新規作成ダイアログ

[ファイル(F)|新規作成(N)]で表示されます。フォーム、コンポーネント、プロジェクト、テキストファイルなど新しい要素を作成するために使うウィンドウです。空の要素を作成するほか、登録済みのフォームやプロジェクトを利用したり、別のフォームを継承する場合もオブジェクトリポジトリを使います。



#### プロジェクトマネージャ

[表示(V)|プロジェクトマネージャ(P)]で表示されます。アプリケーションを構成するファイルを管理するウィンドウです。ファイルの一覧、登録、削除、プロジェクトオプションの設定ができます。プロジェクトは、Object Pascal のソースプログラムで管理されており、[表示(V)|プロジェクトソース(J)]で表示できます。



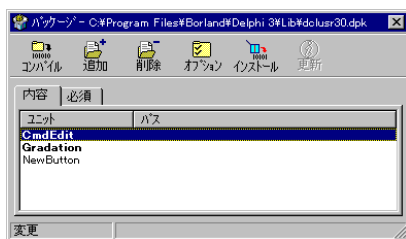
#### データベースフォームウィザード

[データベース(D)|フォームウィザード(F)]で表示されます。表示されるダイアログボックスのメッセージにしたがって、使いたいテーブルを選んだり、表示スタイルやマスター/詳細の設定を指定することで、希望するデータベース対応フォームを自動的に作成できます。



#### データベースエクスプローラ

[データベース(D)|エクスプローラ(E)]で表示されます。エリアスやテーブルの一覧を表示できる他、テーブルの内容を表示・編集、新しいエリアスの作成、SQL文を使った問い合わせなどができます。Professional 版ではデータディクショナリを管理できます。さらに、Client/Server Suite 版に付属する SQL エクスプローラでは、ストアドプロシージャやトリガーなど SQL データベースのメタデータも表示できます。



#### パッケージエディタ

新しいコンポーネントを登録するときに表示されます。任意のコンポーネントユニットの追加、削除やパッケージを構築するときのコンパイルオプションなどを設定できます。

## Delphi 2.0 から引き継がれている特長

実際に Delphi 3 で作成したアプリケーションの例を見るために、Demos ディレクトリに含まれているサンプルプログラムを利用できます。

### サンプルプログラム Fishfact のオープン

\Demos\Db ディレクトリには、データベースを使ったサンプルプログラムが含まれています。この中の Fishfact サブディレクトリには、Fishfact.dpr というサンプルプロジェクトが含まれています。Delphi 3 のメニューで [ファイル(F) | 開く(J)] を選び、このディレクトリに移動して Fishfact.dpr を選んでください。ここで、図 2 のような画面になります。

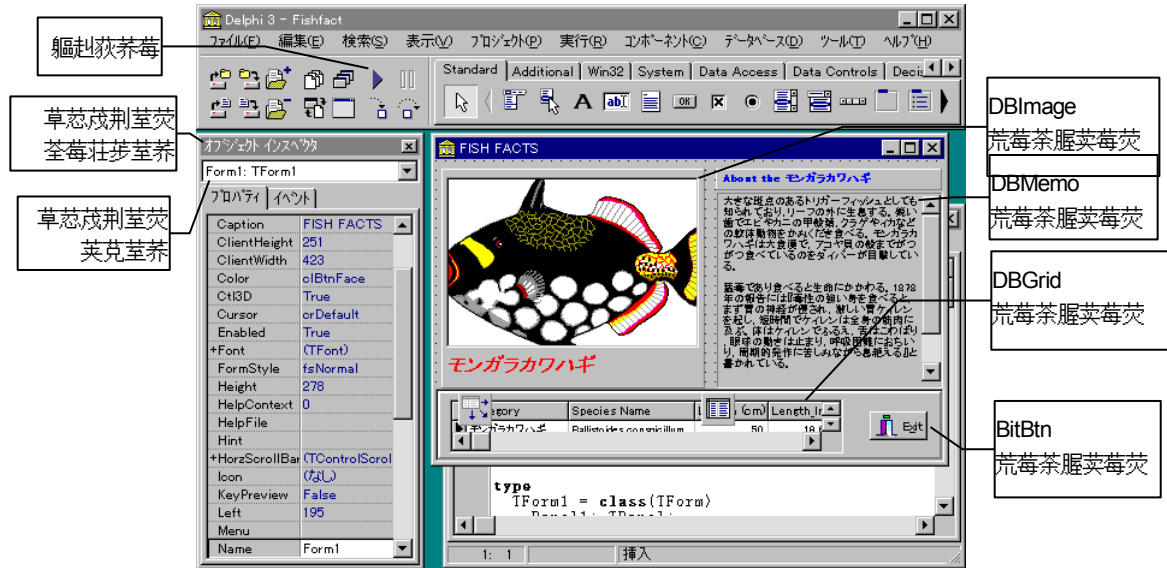



図 2 FishFact プロジェクトをオープンしたところ

このプロジェクトは、BIOLIFE.DB という Paradox 形式のテーブルを使っています。単にプロジェクトをオープンしただけですが、フォーム上には文字列やイメージなどのデータベースの内容が表示されています。Delphi ではアプリケーションを設計する段階で、実際のデータベースを参照できるため、データがどのように表示されるかを意識しながらユーザーインターフェースを設計できます。

フォーム上のそれぞれのコンポーネントをマウスでクリックすると、オブジェクトインスペクタに表示される内容が変わります。オブジェクトセレクタには、現在フォーム上で選択されているコンポーネントの名前と型名が表示されています (型名の先頭には T がついています)。ここで、フォーム上のコンポーネントを選ぶこともできます。

Delphi のデータベース対応のコンポーネントを使えば、プログラミングをしなくても、オブジェクトインスペクタで必要な設定をするだけでデータベースを処理するアプリケーションを開発できます。このプロジェクトでも、プログラムコードをまったく使っていません。各コンポーネントの詳しい説明が必要ときは、フォーム上のコンポーネントをマウスでクリックして選択し、[F1] キーを押します。

実行ボタン  を押すと、このプロジェクトを構築して実行できます。



## Delphi 2.0 から引き継がれている特長

Delphi は、プロジェクトを単独で実行できる実行形式のファイルにコンパイルします。ただし、データベースを使うアプリケーションでは、データベースエンジン（BDE＝Borland Database Engine）を別途インストールする必要があります。BDE は、Delphi をインストールすると自動的にシステムに組み込まれます。

Fishfact プロジェクトを実行して、左下の DBGrid の垂直スクロールバーのボタンを押すと、データベースのレコードが移動して表示される内容が切り替えられます。



図.3 FishFact の実行画面

Demos ディレクトリには、この他にも多くのプログラミング例が入っていますので、適当なプロジェクトファイルを開いて試してみてください。

## Delphi 2.0 から引き継がれている特長

Delphi で簡単なアプリケーションを開発してみましょう。

[ファイル(F)|アプリケーションの新規作成(T)]メニューを選択して、新しいプロジェクトを作成します。オープンしているプロジェクトに変更を加えていると、ダイアログボックスが表示されて、変更した内容を保存するかどうか問い合わせられます。

新しいプロジェクトをオープンすると、図.1 (p.6) のような画面になります。Delphi では、コンポーネントパレットからコンポーネントを選んで、フォームという設計領域に配置し、オブジェクトインスペクタを使ってイベントやプロパティを設定してアプリケーションを開発していきます。

まず、コンポーネントパレットの [Standard] というタブをクリックしてください。ここには、Windows が提供する標準的なコントロールや機能に対応するコンポーネント (部品) が登録されています。

左から 6 番目にボタン (Button) コンポーネントがあります。これをクリックすると、このボタンが押された状態になります。次に、フォーム上の適当な場所をクリックします。すると、ボタンがフォーム上に配置されます。コンポーネントが配置されると、そのコンポーネントには後ろに数値が付加された名前が自動的につけられます。



図.4 コンポーネントパレットの [Standard] ページ

このようにコンポーネントパレットでコンポーネントをクリックし、フォーム上でもう一度クリックすれば、そのコンポーネントがフォームに配置されます。このときコンポーネントはあらかじめ決められた大きさと配置されます。フォーム上で、マウスをドラッグ (マウスの左ボタンを押しながら位置をずらす) すると、指定した大きさとコンポーネントが配置されます。コンポーネントパレットで適当なコンポーネントをダブルクリックすれば、フォーム上の中央に決められた大きさとコンポーネントが配置されます。

同じコンポーネントを連続的に配置したい場合は、キーボードの [Shift] キーを押しながら、コンポーネントをクリックします。ボタンが青い枠つきで押された状態になり、左端のポインタアイコンをクリックするまで、連続的にコンポーネントを配置できます。

次に、コンポーネントパレットの [Dialogs] ページを選びます。ここには、コモンダイアログをあらわすコンポーネントがあります。ここで、左から 6 番目にある ColorDialog コンポーネントをフォーム上に配置します。



図.5 コンポーネントパレットの [Dialogs] ページ

このコンポーネントは「色の設定」ダイアログを呼び出す機能は持っていますが、自分自身は実行時にはフォーム上に表示されません。コンポーネントには、ボタンのように実行時に目に見えるもの (ビジュアル) と、コモンダイアログのように目に見えないもの (非ビジュアル) があります。非ビジュアルコンポーネントは、大きさを変更できません。

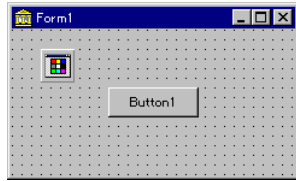


図.6 Button と ColorDialog を配置したフォーム

次に、ボタンの表現や動作を設定します。つまり、オブジェクトインスペクタを使ってプロパティとイベントハンドラを定義します。プロパティとは、コンポーネントの表示する文字列や保持する値などを文字列や数値などのデータとして表現するものです。イベントとは、コンポーネントに対して何が起きたかをあらわすもので、イベントハンドラはコンポーネントに対して発生した要因に対応する処理部分のことです。プロパティやイベントハンドラを定義するためにオブジェクトインスペクタを使います。

フォーム上でボタンをクリックして選択します。次に、オブジェクトインスペクタで [プロパティ] というタブをクリックし Caption という項目を探します。ここには Button1 と表示されているはずですから、これを Color と変更します。次に、[イベント] ページで OnClick という項目をダブルクリックします。

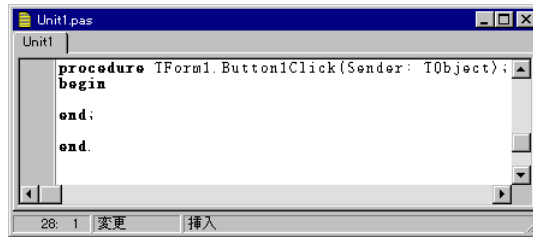



図.7 Button1 をダブルクリックしたときのイベントハンドラ

ここで、次のようなプログラムリストを入力します。

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  if ColorDialog1.Execute then           { 追加する行 }
    Color := ColorDialog1.Color;       { 追加する行 }
end
```

これは「色の設定ダイアログを呼び出して OK ボタンが押されたら、フォームの色をダイアログボックスで選んだ色に変更する」という意味のプログラムです。

実行ボタン  を押して、プログラムを実行してください。フォーム上の Color ボタンを押すと、「色の設定」ダイアログが表示され、適当な色を選んで OK ボタンを押すと、フォームの色が指定した色に変更されます。

ここで作成したアプリケーションは非常に単純なものです。より複雑なアプリケーションでも基本的な手法は変わりません。

## Delphi 2.0 から引き継がれている特長

Delphi 3 は、広範なアプリケーション開発に対応する数多くの部品や関数を用意しています。しかし、慣れないうちは完全な文法を覚えたり、関数引数の型を見つけ出すのは面倒かもしれません。Delphi 3 には、このような場合に役立つ便利な機能があります。

### コードテンプレート

コードテンプレートとは、IF、FOR、WHITE、CASE といった頻繁につかう構文を手作業で入力する代わりに、自動的に入力させる機能です(図.8)。たとえば、if と入力した直後に[CTRL]+[J]を押すと、単純な if 文や begin～end 付きのもの、else 節のあるものなどのリストが表示されます。ここから適当なものを選べば、対応するプログラムコードが埋め込まれ、if の直後にカーソルが移動します。

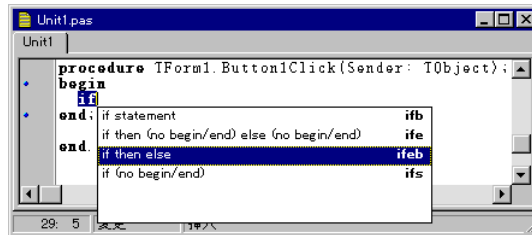


図.8 コードテンプレートの例

また、入力したキーワードに対応するプログラムコードがひとつしかなければ、即座にそのコードが埋め込まれます。たとえば、begin/end 付きの for 文をあらわす forb を入力してから[CTRL]+[J]を押すと、次のプログラムコードが埋め込まれ、for の直後( | の位置)にカーソルが移動します。

```
for | := to do
begin
```

```
end;
```

コードテンプレートで入力できるプログラムコードや短縮形式は、[ツール(T)|環境オプション(O)]の[支援機能]ページでカスタマイズできます。ここで設定すれば、プログラム構文だけでなく、関数やメソッド名を短縮入力するためにも使えます。



図.9 環境オプションの設定

### コード補完

コード補完は、フォーム上のコンポーネントなどオブジェクト名の直後にピリオド(.)を入力すると、プロパティやメソッドの名前をリストアップする機能です(図.10)。

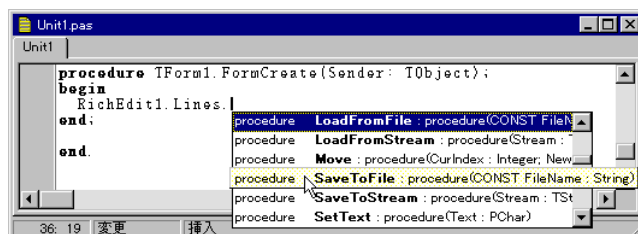


図.10 コード補完の例

通常これらの名前は、継承されているクラス別・種類別に並べられていますが、リスト表示されているときにマウスの右ボタンでスピードメニューを呼び出せば、アルファベット順でも表示できます。リスト中から目的のものを選ぶためには、項目をスクロールさせるカインクメンタルサーチが使えます。

入力している文が、代入文のときは左辺の型に対応する関数やプロパティだけが表示されます。たとえば、文字列型の変数への代入であれば、文字列型を返す関数や文字列型のプロパティが表示されます。また、フォント(Font)のようにさらにメソッドやプロパティを持っているクラスオブジェクトも表示されます。

コード補完は、編集中のプログラムコードに即座に反応します。プログラム中にレコード(record)やクラス(class)の定義があれば、これらに対する変数名の後にピリオドを入力した時に、定義されているデータフィールドやメソッドが表示されます。

コード補完の反応速度は、前述の環境オプションで0.5~1.5秒の範囲で変更できます。また、[CTRL]+[スペース]を押せば即座に呼び出せます。

### 宣言表示

宣言表示は、関数名やメソッド名の直後に左カッコ( )を入力すると、自動的に入力すべき値の型や仮引数の名前を示してくれる機能です。これは、標準的な関数やコンポーネントのメソッドだけでなく、Windows APIや自作の関数を含むすべてに対応します。

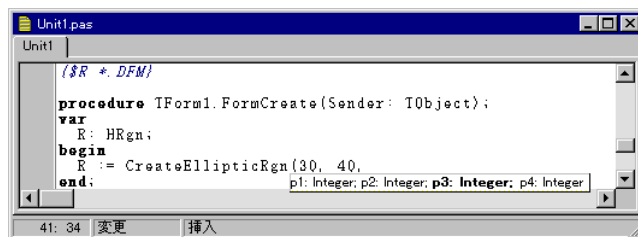


図.11 宣言表示の例

また、次に入力すべき引数の型は強調表示(ボールド)され、複数の引数を持つ関数の場合は、値を入力するごとに次の引数が強調されていきます。入力すべき型や順序が不安な場合でも、わざわざヘルプを呼び出す必要がありません。

## Delphi 2.0 から引き継がれている特長

Delphi 3 では既存のコンポーネントを利用して、ビジュアルに新しいコンポーネントを作成できるコンポーネントテンプレートという機能を提供しています。

コンポーネントテンプレートを使うには、フォーム上にコンポーネントを配置し、適当にプロパティやイベントハンドラを設定した後で、[コンポーネント(C)|コンポーネントテンプレートの作成(T)]を呼び出します(図.12)。

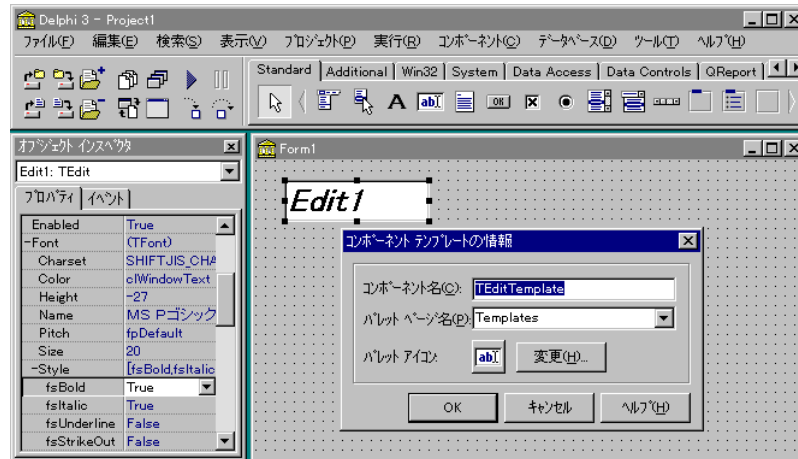


図.12 コンポーネントテンプレートの例

ここで、新しいコンポーネント名や登録するコンポーネントパレットのページ名を指定すると、その情報をもとに新しいコンポーネントが追加されます(図.13)。

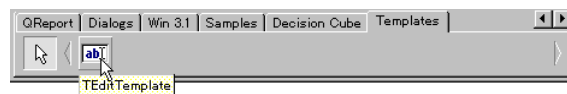


図.13 [Templates]ページに組み込まれた、新しいコンポーネント

Panel や ToolBar/CoolBar のようなコンテナコンポーネントを使えば、複数のコンポーネントを組み合わせて、ひとつのコンポーネントとして登録することもできます。図.14 は、Panel 上に3つの SpeedButton と1つの ComboBox を配置したものです。

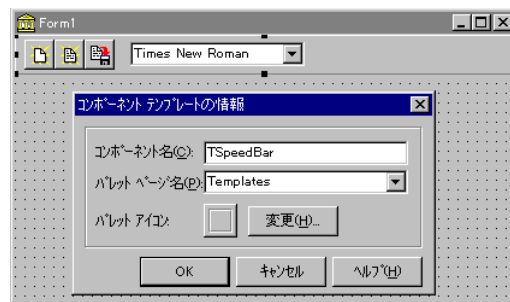


図.14 スピードバーをテンプレートとして登録

テンプレートとして登録されたコンポーネントは、VCLには組み込まれません。テンプレートコンポーネントを配置すると、登録したときに使われた VCL コンポーネントが配置され、設定されていたプロパティとイベントハンドラの両方が反映されます。

### ツールチップ式評価

Delphi 3では、プログラムをデバッグのために停止させているとき、マウスカールソルを変数やプロパティの上に移動させると、自動的にその値がツールチップとして表示されます。ここでは、オブジェクトや変数、定数などさまざまな式を評価できます。

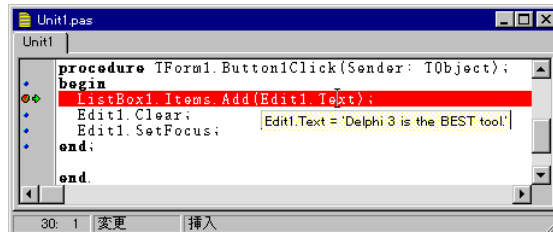


図.15 ツールチップ式評価の例

### DLL デバッグ

DLLを開発するプロジェクトでは、[実行(R)|実行時引数(P)]でホストアプリケーションを設定できます。ホストアプリケーションを設定して、DLLプロジェクトを実行させると、まずホストアプリケーションが実行され、DLLプロジェクトにブレイクポイントを設定したり、監視式を設定できます。

### モジュールウィンドウ

[表示(V)|モジュール(M)]で表示されるモジュールウィンドウは、プロジェクトの実行中メモリに読み込まれているモジュールの一覧を表示します。モジュールウィンドウは、それぞれのモジュールの名前、実行時のイメージベースアドレス、モジュールが読み込まれた場所(パス)を表示します。

名前	アドレス	パス
Project1.exe	\$00400000	C:\Program Files\borland\Delphi 3\
COMCTL32.DLL	\$BFC00000	C:\WINDOWS\SYSTEM\
RPCRT4.DLL	\$7FD00000	C:\WINDOWS\SYSTEM\
MSVCRT20.DLL	\$7FD00000	C:\WINDOWS\SYSTEM\
OLE32.DLL	\$7FF60000	C:\WINDOWS\SYSTEM\
OLEAUT32.DLL	\$65340000	C:\WINDOWS\SYSTEM\
WINNLS32.DLL	\$BFE30000	C:\WINDOWS\SYSTEM\
ADVAPI32.DLL	\$BFFD0000	C:\WINDOWS\SYSTEM\
GDI32.DLL	\$BFF30000	C:\WINDOWS\SYSTEM\
USER32.DLL	\$BFF60000	C:\WINDOWS\SYSTEM\
KERNEL32.DLL	\$BFF70000	C:\WINDOWS\SYSTEM\
IMM32.DLL	\$BFE80000	C:\WINDOWS\SYSTEM\
INDICDLL.DLL	\$7EA00000	C:\WINDOWS\SYSTEM\
MSIME95K.DLL	\$7FC40000	C:\WINDOWS\SYSTEM\
MSIME95.IME	\$7FC00000	C:\WINDOWS\SYSTEM\

図.16 モジュールウィンドウ

## Delphi 2.0 から引き継がれている特長

### CheckListBox

Delphi には数多くのコンポーネントが登録されています。Delphi 3 では、さらに便利なコンポーネントが追加されました。ここでは、新しいコンポーネントの機能や使い方について紹介します（トライアル版で利用できない機能については後述します）。

#### [Additional] ページ

Delphi の [Additional] ページには Windows の標準コントロールにない機能を持つコンポーネントが登録されています。

CheckListBox は、複数のチェックボックスをまとめて管理するものです。RadioGroup に似ていますが、CheckListBox は ListBox を拡張して実現されています。

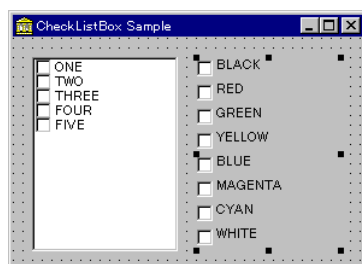


図.17 CheckListBox の使用例

図.17 で、左側は単純に項目を設定したもの、右側はさらに BorderStyle を bsNone に、Color を clBtnFace に、Style を lbOwnerDrawFixed に、ItemHeight を 20 に設定したものです。チェックボックスの状態は Checked プロパティや State プロパティで確認します。たとえば、2 番目の項目がチェックされているかどうかは CheckListBox1.Checked[2] で調べます。

Splitter は、領域を左右または上下に分割するバーを作ります。Align プロパティの設定によって実行時に隣のコンポーネントの幅や高さを変更します。デフォルトでは alLeft が設定されており、自分の左側のコンポーネントの幅を変更します。Align を alTop にすれば自分の上のコンポーネントの高さを変更することになります。

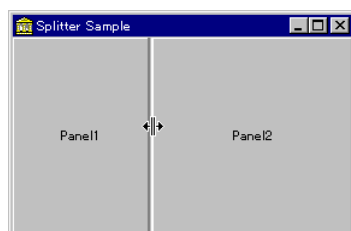


図.18 Splitter の使用例

Splitter とパネルなどのコンポーネントと組み合わせることで、フォームを複数の領域に分割し、簡単にサイズを変更できます。

StaticText はテキスト表示用のコンポーネントです。Label に似ていますが、ウィンドウハンドルを持っているため、Memo などの他のコントロールの上に重ねて表示できます。

また、Professional 版以上では ActiveX コントロールに変換する対象として指定したり、ActiveX コントロールのプロパティページを作成する際に、入力フォーカスを受け取るテキスト表示コントロールとして利用できます。



## Delphi 2.0 から引き継がれている特長

DateTimePicker

### [Win32] ページ

Delphi 3 の [Win32] ページは、Delphi 2.0 の [Win95] ページに相当します。Delphi 3 では、新しいコンポーネントが登録されています。

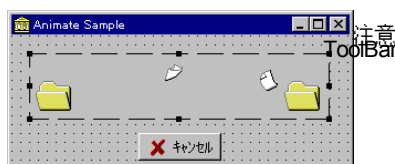


図.19 Animate の使用例

Animate は、音のないアニメーション (AVI) を表示するコンポーネントです。

ファイルをコピーしたりごみ箱に捨てるといった汎用的な定義済みのアニメーションは、CommonAVI プロパティで簡単に利用できます。Active プロパティを使えば、設計時にアニメーションの表示を確認することもできます (図.19)。

DateTimePicker は、日付や時間を設定するためのコンポーネントです。カレンダーを使



図.20 DateTimePicker の例

って日付を設定したり、上下ボタンを使って時刻を設定できます。

ToolBar は、ツールボタン (ToolButton) や他のコントロールを配置して、ツールバー (スピードバー) を作成するコンポーネントです。

従来、ツールバーを作成するには Panel や SpeedButtonなどを組み合わせて設計し、大きさや位置をプログラマが調整する必要がありました。ToolBar では、ImageList と組み合わせて簡単にボタンを配置でき、位置や大きさも自動的に調整されます。Flat プロパティを True にすれば、マウスカーソルに対応してボタンが浮き出る形式も使えます。

CoolBar は、ToolBar や他のコンポーネントを配置し、左端のつまみを使って場所を移し

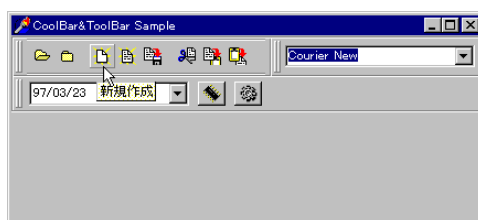


図.21 ToolBar と CoolBar を組み合わせた例

たり、サイズを変更できるコンポーネントです。

DateTimePicker と CoolBar を使うためには、バージョン 4.70 以降の COMCTL32.DLL が必要です。特にアプリケーションを配布する際には注意してください。これは、Internet Explorer 3.0 や MS-Office97 をインストールすると自動的に組み込まれます。

## Delphi 2.0 から引き継がれている特長

### OpenPictureDialog

#### [Dialogs] ページ

コモンダイアログが登録されている [Dialogs] ページでは、イメージファイルを開いたり保存するときに、プレビュー機能を持つコモンダイアログが追加されています。

OpenPictureDialog は、プレビュー機能付きの「ファイルを開く」ダイアログを呼び出します。イメージファイル (.bmp, .ico, .emf, .wmf) を選択するとダイアログボックスの右側にイメージが表示されます。

SavePictureDialog は、プレビュー機能付きの「ファイル名を付けて保存」ダイアログを呼び出します。



図.22 OpenPictureDialog の実行例

#### [Data Controls] ページ

書式付きで項目を表示する DBRichEdit が追加されました。

DBRichEdit は、書式付きでデータベース項目を表示・編集できます。DBRichEdit は、RichEdit と同じように、SetAttributes や Paragraph プロパティを使って文字単位、段落単位でフォントや位置揃えなどを指定できます。

#### [QReport] ページ

レポート出力を設計するための QuickReports コンポーネントは、大幅に改良されました。

新しい QuickReports 2.0 では、従来の QuickReport に代わり、QuickRep コンポーネントを使ってレポートを設計します。QuickRep はフォーム上にグリッドを表示するビジュアルコンポーネントで、レポートの設計がよりビジュアルにわかりやすくなりました。

また、高度なレポートを設計するための新しいコンポーネントに加え、レポート出力を印刷するだけでなく、カンマ区切りのテキストや HTML 形式でエクスポートすることもできるようになりました。さらに、テーブルや問い合わせに対するレポートを簡単に設計するための QuickReport ウィザードも組み込まれています。

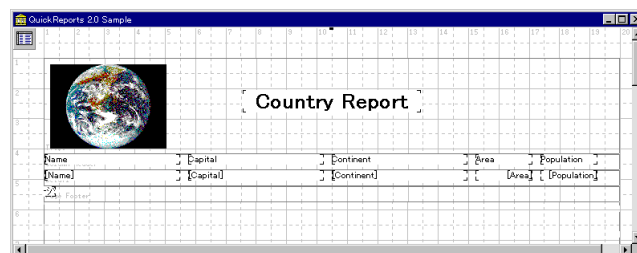


図.23 QuickReports 2.0 を使った設計画面

## Delphi 2.0 から引き継がれている特長

### パッケージ

コンポーネントライブラリの機能を DLL として提供する機能です。この DLL は、.DPL (Delphi Package Library) という拡張子がつけられています。

パッケージを利用することで、実行ファイルにコンポーネントライブラリのコードが含まれなくなるため、実行ファイルの大きさは小さいものでは 10KB 程度になります。これは、ActiveX コントロールのような部品をいくつも作成したり、Active フォームのようなインターネット経由で参照するものを作成する場合、あるいは多層環境で小さなクライアントアプリケーションを配布する場合などに効果的です。

### COM サポートの組み込み

多層モデル、OLE オートメーションサーバーや、ActiveX などのオブジェクトにシームレスなアクセスを実現するため COM (Component Object Model) インターフェースをプログラム中に直接記述できるようになりました。Delphi 3 の COM サポートは ActiveX、Active ドキュメント、Active フォームのような Microsoft 規格をサポートしています。

### コンパイラの拡張

システムの標準手続きとして Assert が追加されました。Assert は指定された条件が満たされない場合、例外が発生します。

Visual Basic との互換性のため、論理データ型の ByteBool、WordBool、LongBool 型に対する True は、-1 で表現されます (Boolean における True は、1 のままです)。

### 開発環境の拡張

オブジェクトリポジトリを共有できるようになりました。[ツール(T)|環境設定(O)]の[設定]ページでリポジトリを共有するためのディレクトリを設定します。

[検索(S)|ファイルからの検索(D)]を使うと、プロジェクト中のファイルやオープンしているファイルや指定したディレクトリ中のすべてのファイルから文字列を検索できます。

[プロジェクト(P)|オプション(O)]の[アプリケーション]ページでは、出力ファイルの拡張子を指定できます。たとえば、スクリーンセーバーを作成する場合は「SCR」を指定します。また、[バージョン情報]ページでは、アプリケーションに埋め込むバージョン情報を設定できます。

### データベースデスクトップ

テーブルや問い合わせを新規に作成するためのユーティリティとして、データベースデスクトップがサポートされます。

### Chart コンポーネント

Delphi 3 Professional 以上では、棒グラフや円グラフをはじめ 11 種類の表示形式を持つ Chart(TeeChart)コンポーネントが提供されます。TeeChart は多彩なスタイルや重ね合わせ、2次元・3次元の切り替え、立体化の比率や枠のスタイル、円グラフの開始角度など、きめこまかなスタイルを設定できます。

チャートには、プログラム中のデータを表示する Chart、データベースに対応した DBChart、レポート出力用の QRChart、後述の Decision Cube に対応する DecisionGraph があります。また、チャートを簡単に設計するために TeeChart ウィザードが用意されています。

## 注意

Delphi 3 Professional 以上では、ActiveX コントロール (OCX) を簡単に作成できます。  
 ActiveX コントロールを作成するためには、Delphi のコンポーネントを変換する方法と、  
 フォームと同じように ActiveX コントロールを設計する方法があります。  
 ActiveX コントロールの作成機能は、トライアル版にはありません。

## 簡単な Delphi コンポーネント

Delphi のコンポーネントを変換して ActiveX コントロールを作成する場合、新しい  
 Delphi のコンポーネントを作成してから ActiveX コントロールに変換してもかまいません  
 し、変換した後でコントロールに機能を追加することもできます。

以下のプログラムは、テキストを立体的に表示するコンポーネントの例です (抜粋)。

```

unit Label3D;
.....
type
  TLabel3D = class (TCustomControl)
  protected
    procedure Paint; override;
    procedure CMTextChanged(var Msg: TMessage); message CM_TEXTCHANGED;
  public
    constructor Create(AOwner: TComponent); override;
  published
    property Caption;
    property Color;
    property Font;
  end;
.....
constructor TLabel3D.Create(AOwner: TComponent);
begin
  inherited Create(AOwner);
  Width := 96;
  Height := 24;
  Font.Color := clWhite;
  Font.Height := -(Height-2);
end;
procedure TLabel3D.Paint;
var
  R: TRect;
begin
  R := ClientRect;
  InflateRect(R, -2, -2);
  OffsetRect(R, 2, 2);
  Canvas.Brush.Style := bsClear;
  Canvas.Font := Font;
  Canvas.Font.Color := clBlack;
  DrawText(Canvas.Handle, PChar(Caption), Length(Caption),
           R, DT_LEFT or DT_WORDBREAK or DT_EXPANDTABS);
  Canvas.Font := Font;
  OffsetRect(R, -2, -2);
  DrawText(Canvas.Handle, PChar(Caption), Length(Caption),
           R, DT_LEFT or DT_WORDBREAK or DT_EXPANDTABS);
end;
procedure TLabel3D.CMTextChanged(var Msg: TMessage);
begin
  Invalidate;
end;

```

## Delphi 2.0 から引き継がれている特長

### 注意

このコンポーネントを[コンポーネント(C)|コンポーネントのインストール(I)]で Delphi

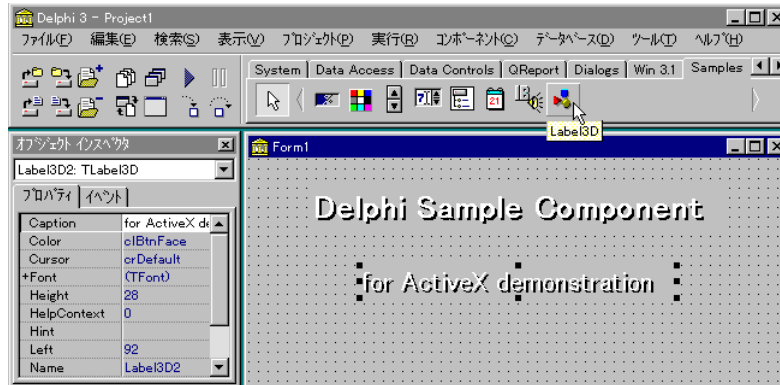


図.24 立体化テキストの使用例

に登録すると、図.24のようにコンポーネントパレットで選んで利用できます。このように自分自身で利用するコンポーネントを簡単に作成できるのも、Delphiの特長です。

### ActiveX コントロールへの変換

DelphiのコンポーネントからActiveXコントロールを作成するのは非常に簡単です。

まず、プロジェクトが空の状態では、[ファイル(F)|新規作成(N)]で表示される[新規作成]ダイアログの[ActiveX]ページから[ActiveX コントロール]を選びます。ActiveXコントロールウィザードが表示されるので、VCLクラス名には作成したTLabel3Dを選び、ファイル名やチェックボックスをチェックして[OK]を押します(図.25)。

作成したActiveXコントロールをVisual Basic 4.0でも利用したい場合は、[バージョン情報を追加(V)]をチェックします。



図.25 ActiveX コントロールウィザード

これで、Label3DのActiveXコントロール版であるLabel3DXを作成するためのプロジェクトやソースコードなどが自動生成されます。[プロジェクト(P)|再構築(B)]を選ぶと、プロジェクトが構築され、OCXファイルが作成されます。作成したActiveXコントロールをシステムに登録するのは[実行(R)|ActiveXサーバーの登録(A)]を選びます。

## Delphi 2.0 から引き継がれている特長

Delphi で作成する ActiveX コントロールは、制約のない完全なネイティブコードコンパイラで構築されるため、あらゆる目的に対応するコントロールを作成できます。

作成した ActiveX コントロールは、Visual Basic のような他のツールでも使えます。たとえば、Visual Basic4.0 では[ツール(T)|カスタムコントロール(C)]を選び[L3DCtrl Library]という項目をチェックします(名前は設定したプロジェクト名により異なります)。ツールボックスに Label3DX というアイコンが追加されますので、他のコントロールと同じようにフォームに配置して利用できます。

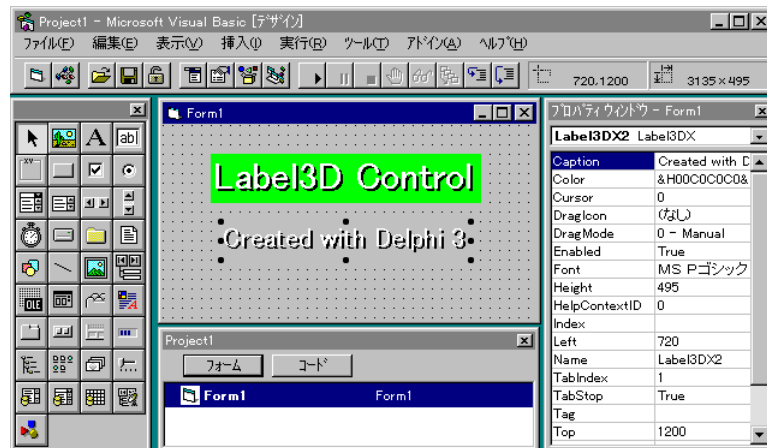


図.26 Label3DX を Visual Basic 4.0 に組み込んだ例  
タイプライブラリエディタ

ActiveX コントロールや COM オブジェクトのタイプライブラリは、[表示(V)|タイプライブラリ(T)]で表示されるタイプライブラリエディタで編集・追加できます(図.27)。



図.27 タイプライブラリエディタ

また、ActiveX コントロールのプロパティを設定するためのプロパティページを作成するときは、[ファイル(F)|新規作成(N)]で表示される[新規作成]ダイアログの[ActiveX]ページから[プロパティページ]を選びます。

### Active フォームの作成

Active フォームは、フォームスタイルの ActiveX コントロールです。Active フォームは、通常のフォームと同じように作成でき、特に Web ブラウザなどで表示する HTML ドキュメントに埋め込んで利用する場合に役立ちます。

Active フォームを作成するには、ActiveX コントロールと同様、[ファイル(F) | 新規作成(N)]の[ActiveX]ページで[Active フォーム]を選びます。ActiveX コントロールウィザードに似た Active フォームウィザードが表示されますが、VCL クラス名は TActiveForm に固定されています。ここでは、適切な ActiveX 名を入力します。

これで、空の Active フォームができあがります。通常のフォームと同じようにコンポーネントを配置したりイベントハンドラを定義できます。たとえば、図 28 のようなフォームを設計し、適切なイベントハンドラを割り当てます。

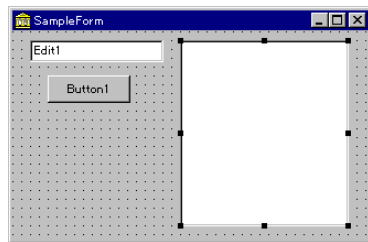


図 28 Active フォームの設計例

プロジェクトを保存し、[プロジェクト(P) | 再構築(B)]を選べば Active フォームができあがります。また、Web で配布するためには、[プロジェクト(P) | Web 配布オプション(W)]を使えます。このダイアログボックスで必要な項目を指定し、[プロジェクト(P) | Web 配布(D)]を選ぶと、Active フォームを使うための簡単な HTML ファイルが生成されます。

生成された HTML ファイルを Internet Explorer 3.0 のようなブラウザで呼び出せば、通常のフォームと同じように使うことがわかります (図 29)。

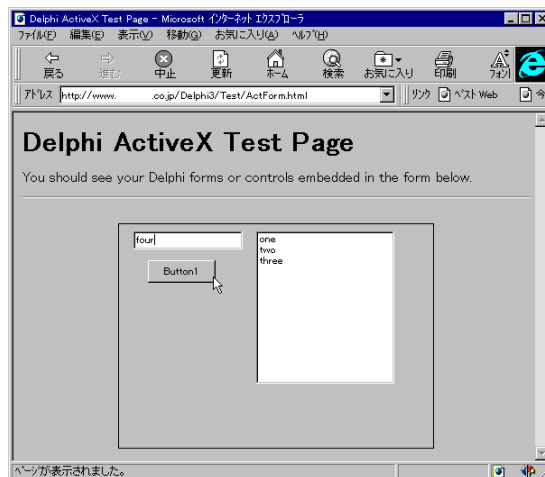


図 29 Internet Explorer 3.0 での表示

## 注意

Delphi 3 Client/Server Suite では、多次元解析を実現するための Decision Cube コンポーネントが提供されます。Decision Cube は、特定の目的に対応した調査に利用したり、重要な意思決定を求められるユーザーに高度な情報を提供します。アプリケーションに応じてカスタマイズできるため、誰もがダイナミックに変化する情報をアクセスできるようになります。

Decision Cube のドリルダウン、ピボット（軸）、総計といった機能は、データのビジュアルな解析を実現し、よりよい決定をサポートします。折りたたみのできる要約やドラッグアンドドロップをサポートしたインターフェース、さらに解析結果のグラフ表示などがビジネスの効率化に大きく寄与します。

Decision Cube は、トライアル版にはありません。

## Decision Cube の使い方

Delphi 3 Client/Server Suite の [Decision Cube] ページには、多次元解析を処理する 6 つのコンポーネントが登録されています（図.30）。

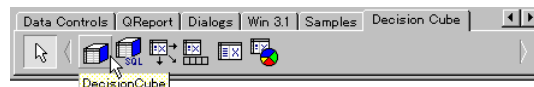


図.30 [Decision Cube] ページのコンポーネント

DecisionCube を使って多次元解析を実行するには、まず DecisionQuery を配置して問い合わせを設計します。DecisionQuery の機能は、Query と同等ですが、多次元解析のための専用のプロパティエディタを持っています。

配置した DecisionQuery をダブルクリックすると「ディジジョン問い合わせの設定」ダイアログが呼び出されます（図.31）。このエディタでは、対象となるテーブルやデータ項目を設定します。



図.31 ディジジョン問い合わせの設定

DecisionQuery の設定が終わったら、次に DecisionCube コンポーネントを配置します。このコンポーネントは多次元解析の中心的役割を果たします。DecisionCube をダブルクリックすると、「ディジジョンキューブの設定」ダイアログが表示され、どのように解析するか、項目ごとに設定したり、解析のために使用するメモリの大きさなどを設定します。



## Delphi 2.0 から引き継がれている特長

次に DecisionSource を配置します。DecisionSource は、データセットとデータ表示コントロールを仲介する DataSource に似ています。DecisionSource は、DecisionCube と表示用のコンポーネントの仲介役となります。

以上を配置したら、後は実際にフォーム上に解析結果を表示したり制御するためのコンポーネントを配置します。[Decision Cube] ページの右側の 3 つのコンポーネント DecisionPivot、DecisionGrid、DecisionGraph は、いずれも DecisionSource プロパティに配置した DecisionSource コンポーネントを設定します。

DecisionPivot はユーザーが多次元解析のピボットを設定できるようにするためのコンポーネント、DecisionGrid は実際に解析されたデータがグリッド形式で表示するコンポーネント、DecisionGraph は対応するグラフを表示するコンポーネントです。

設定が終わったら、DecisionQuery の Active プロパティを true にすると、設計画面で解析が実行されます。DecisionPivot にはボタンが表示され、DecisionGrid と DecisionGraph には解析されたデータとチャートが表示されます (図.32)。

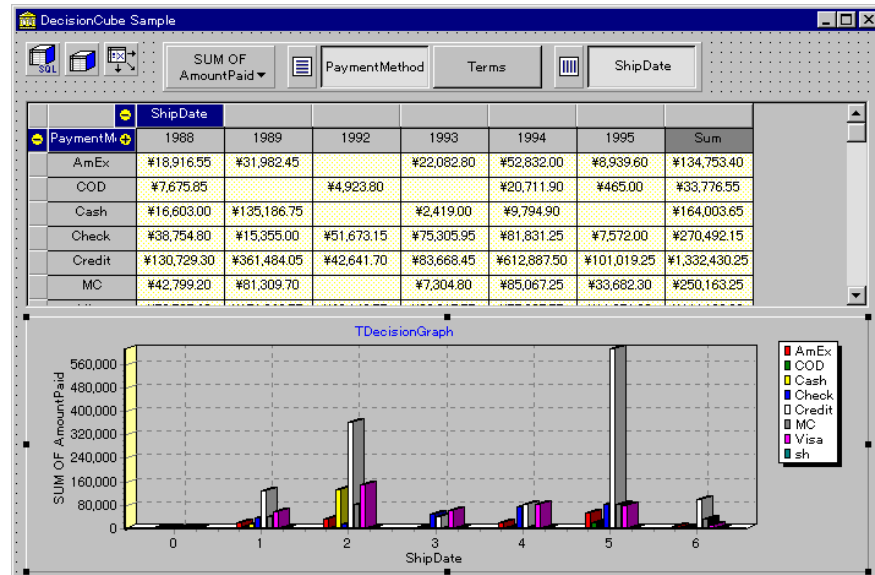


図.32 Decision Cube の使用例

設計画面で DecisionPivot1 に表示されているボタンの文字列の部分をクリックすると解析データの種類やピボットを変更できます。また、直接グリッド上に表示されている項目を互いにドラッグアンドドロップすることでも座標軸を交換できます。項目上に黄色で表示されている (+) や (-) の印をクリックすると、その項目の詳細を表示するかどうかを切り替えられます。

このように設計画面で実際の動作を確認できるのも、Delphi の特長です。実際に解析したデータを確認しながら、アプリケーションの要求に応じて設定をカスタマイズできます。

Decision Cube を使えばプロパティを変更したり、グリッド上で表示形式を変更することで、幅広いビジネスニーズに対応する多次元解析を実現できます。そして、この機能はコンポーネントとして提供されているため、プログラマが作成するアプリケーションに強力な解析能力を追加できます。

## Delphi 2.0 から引き継がれている特長

Delphi 3 データセットアーキテクチャは、複数のデータベースエンジンに対応できるように拡張されました。Table や Query のように Borland Database Engine を使うもの以外にも、開発者がデータベースの基本的な処理（キャッシュ、取り出し、言語照会、異種間アクセス、一般的なカーソルサポート）を実装することで、アプリケーション固有のデータセットを利用できます。これによって、他のデータベースエンジンを扱う場合にも、Delphi のデータベースに関する強力なビジュアルコントロールをシームレスに利用できます。Delphi 3 のデータベース対応コンポーネントの関連性を図 33 に示します。

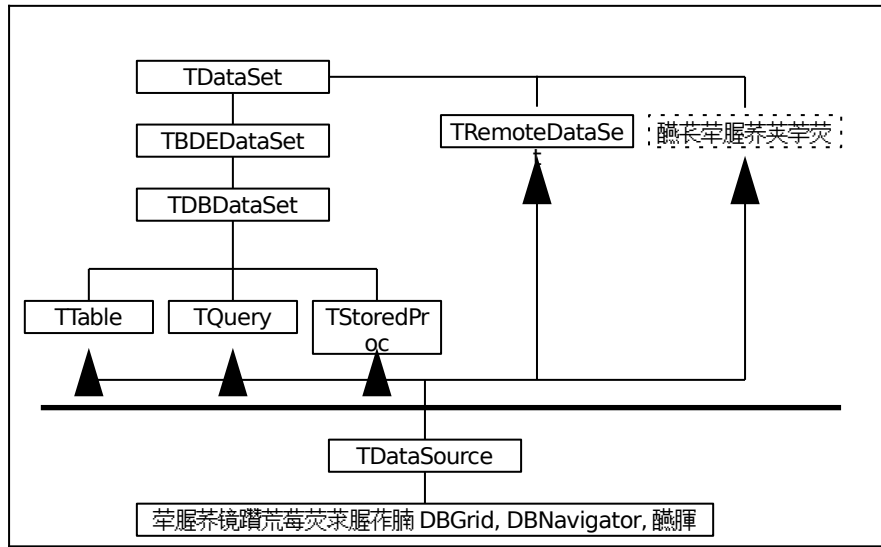
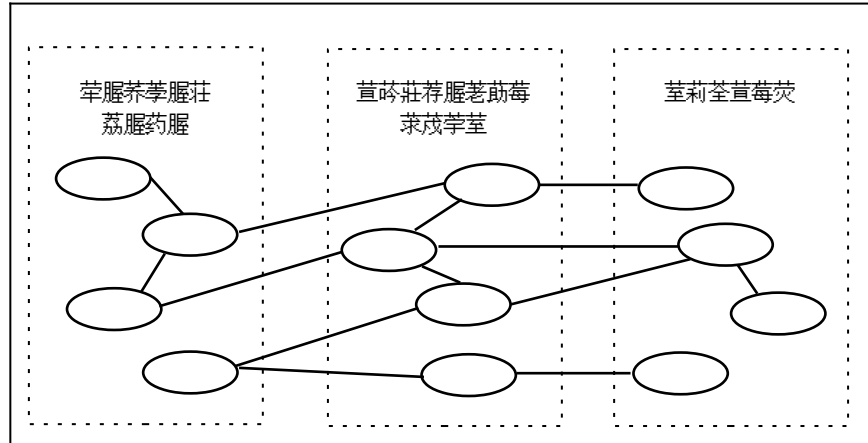


図 33 Delphi 3 のデータベース対応コンポーネントの関連性を示す図

Delphi 3 のデータベース対応コンポーネントの関連性を示す図。TDataSet は TBDEDataSet と TRemoteDataSe に継承される。TBDEDataSet は TDBDataSet、TTable、TQuery、TStoredProc に継承される。TRemoteDataSe は TDataSource に継承される。TDataSource は DBGrid、DBNavigator、TTable、TQuery、TStoredProc に継承される。



“fff,□Ž,f□ff□f□’,ffff□ffffff,^,,fff,Š—,□‘←,fff,Œ—%o,,,,,,’,,,,,Ž—,□  
 □.34,□“Œ“3’fff,—,□Delphi 3,□“fff,%o,,ffff□fff,□“f□ff□f□□Ž,Š”,,,,□  
 □.34□“Œ“3Š”

Delphi 3 における多層開発

Delphi 3,□,,,,,ffffff,“fff,%o,,,,,□ff□ff□f□Œ,%o,‘<’,fff□fff,’<,,,,,□  
 ff□ff□f□Œ’,□ffffff”,,,,,,□



□.35□Delphi 3 Client/Server Suite.[Data Access]f□f  
 ,,□Delphi 3,.[ファイル(F) | 新規作成(N)]  
 ,ff□ff□fff□f,□□,□Table,Query,”,,Š’,f□ff□ff□f□□□□,,,,□  
 TRemoveServer,□“ffff□fff,,,,ffffff”,□ff□ff□f□□□‘Š—  
 ,,□RemoteServer,ServerNameffff,□□‘,,f□f□ffffff,—<,,,,,□““‘,,,,,□Š—,,□  
 TClientDataSet,□TDataSet,,Œ□,,ffff□f,Œ—,f□ffff,,□ Borland Database  
 Engine,,f□f,Ž,Ž,TQuery,TTable,^,,□TClientDataSet,f□f,’□“,,,,,□TClientDataSet,Dataffff,,□  
 ,,f□f□DataPacket; f□ffff□□,,□□f□f□Delta; □•□,“,,,,,□BDEf□f,f□ffff□TTable,  
 TQuery, TStoredProc□,□TClientDataSet,Ž,f□ffff,’<,Providerffff,Ž,,,,,□  
 ,,fff□fff,’□,,,,,□’□,,,,,□,,,,ffffff,•—  
 ,,f□f□‘ffffff’.ffff□fff,□□,,,,,□,,,,,fff□fff,□,,‘<,fff□fff,Š”,,□□,‘%o  
 ,,□•,‘Ž,,,,,□















## Delphi 2.0 から引き継がれている特長

Web“%o,ffff□fff,Š”,,,,,□Delphi 3 Client/Server Suite,,Internet Solutions Pack,%  
 ,,□□,,fff□fff,fff□f,%o,,,,,,□  
 Webffff□fff,Š””,□ffff”,,,,,,□

### Internet 対応コンポーネント



□.36□fff□ffffff,Internetf□f

	ClientSocket	ffff□fff,TCP/IPffffff,,,,,□ff□f,f□ffff□f,”,,□
	ServerSocket	ffff□fff,TCP/IPf□f□,,,,,□ff□f,f□ffff□f,”,,□
	WebDispatcher	f□ffff□f,Webfff□f,•Š,,□HTTP,—(fff□f,%o ,,Webf□f□ffff□fff,,fff□fff,,□
	PageProducer	Webfff,,fffffffff□fff,□HTMLfff□f,,HTMLfff,•Ž— .Š,,,,Ž,fff□fff,,□
	QueryTableProducer	TQueryffffff,,ff□f,Ž,,□f□fCEŽ,HTMLfff—,□□,,,,fff□fff,,□
	DataSetTableProducer	TDataSetffffff,,ff□f,Ž,,□f□fCEŽ,HTMLfff—,□□,,,,fff□fff,,□
	FTP(*)	FTP□ffff“ffffff□,Š’,fff,,,,ActiveXfff□f,,□ff□fff,f□ffffŠ,f□ f,“,,□
	HTML(*)	HTMLffffff,Ž,□,□%□,□fff□f%”,—^•Ž,,ActiveXfff□f,,□
	HTTP(*)	HTTP□fff□fff“fffff) .ffffff,Ž,□’□HTTPffffff,Ž,□,,,,,□ActiveXfff□f,,□
	NNTP(*)	NNTP□fff□fff□f“ffffff□,,ff□ff□f□,ffff,— ^,,,,ActiveXfff□f,,□
	POP(*)	POP3ffff,f□f,,UNIX,,f□f□,f□f,Ž,□,,,,ActiveXfff□f,,□
	SMTP(*)	SMTPf□ff□f□,fff,,□f□f,fff,,,,ActiveXfff□f,,□
	TCP(*)	fffff,f□f□,— •,,,,□WinSock,TCP□fffffffff□CEffffff□,Ž’,ActiveXfff□f,,□
	UDP(*)	fffff,f□f□,— •,,,,□WinSock,UDP□f□f□fffffffff□,Ž’,ActiveXfff□f,,□

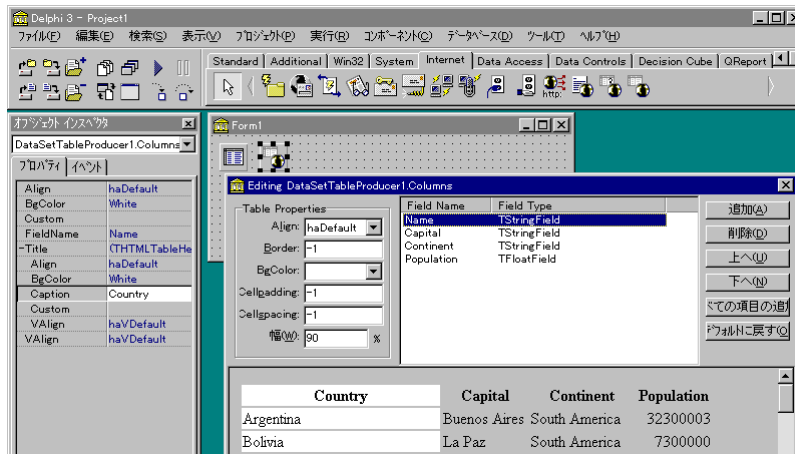
(\*),,,,□Internet Solutions Pack,,Delphi 3 Professional,,Š,,,,,□

## DataSetTableProducer の使い方

```

Web'%,fffff,DataSetTableProducer,Z,fffff,--,%,',
DataSetTableProducer,TDataset,CE,ffff,%,f,f,“—
,HTML,•f,f,f,CEŽ,Š,,TDataset,CE,ffff,f,f,,TableQueryStoredProc,%
,ClientDataSet,,ClientDataSet,Z,ff,f,f,f,f,f,f,Z,HTML,Š,,
-,Ž,,—,QueryTableProducer,Z,,QueryTableProducer,←,%,Ž“—
,,Ž,,
,fffff,,f,f,f,Table,”,,Table1,DatabaseNamefffff,TableNamefffff,“f,f,f,
,,Activefffff,True,,Ž,DataSetTableProducer,”DataSetfffff,Table1,’,,,Dat
aSetTableProducer1,fffff,,ffffffffff,CE,,.37ffffffffff,,f,f,f,f,,HT
ML,“—,Ž,%,Š”,,CE,,

```



```

.37ffffffffff,CE%—
ŽŽ,Contentffff,CE,,HTMLCEŽ,Ž—,Ž“,,ffff,Š,,Ž,fffff,,
var
  s: string;
  F: TFileStream;
begin
  s := DataSetTableProducer1.Content;
  F := TFileStream.Create('filename.html', fmCreate);
  F.Write(PChar(s)^, Length(s));
  F.Free;
end;
,,,fffff,,Webf,f,“,,fffff,,[ファイル(F)|新規作成(N)].[新規
作成]f,f,[Web サーバーアプリケーション],’,

```

## 30 Delphi 2.0 から引き継がれている特長

32 ビットコンパイラ	強力な最適化機能を持つ 32 ビットネイティブコードコンパイラにより、高速に実行されるアプリケーションを開発できます。また、実行形式ファイル(.EXE)だけでなく、他のアプリケーションから利用できる.DLLも作成できます。Windows API を呼び出したり、メッセージを処理することができ、マルチスレッドを含む Windows 95/NT のすべての機能を利用できます。
2Way-Tool	ビジュアル開発とソースコード開発を密接に連係させる機能です。ビジュアルに設計した内容は、同時に対応するソースコードに反映されます。また、ソースコードに入力した内容も自動的に評価されビジュアル開発側で利用できます。ビジュアルとソースコード間でのカットアンドペースト、プロジェクトのソースコードでの管理、フォームのテキスト表示切り替えなど、さまざまな側面で 2Way の機能が実現されています。
豊富なコンポーネント	Delphi 3 では 100 種類以上の豊富なコンポーネントによって、広範なアプリケーション開発に対応します。また、スクロールバーやトラックバーの初期位置の指定、ツリービューやリストビューでの項目の指定、セルでの編集機能を持つ文字列グリッドなど、ビジュアル性が高いユニークな機能は、迅速なアプリケーション開発に役立ちます。
ビジュアルなフォームの継承	Delphi の土台を築いているオブジェクト指向プログラミングを活かし、フォームやデータモジュールを継承できます。フォームのスタイルを統一したい場合や、既存のフォームを変更せずに新しいコンポーネントや機能を追加したい場合に役立ちます。
フォームリンク	[ファイル(F)] ユニットを使う(U)]でユニット名を指定すると、別のフォームやデータモジュールやそこに配置されているコンポーネントを利用できます。テーブルや問い合わせを別のフォームのデータ表示コントロールで利用できます。
データモジュール	テーブルや問い合わせ、コマンドダイアログなど、実行時に目に見えない非ビジュアルコンポーネントを配置し、アプリケーション内で共通化したいビジネスロジックを定義できます。
オブジェクトリポジトリ	汎用的なフォームやデータモジュール、プロジェクトを登録しておける場所です。アプリケーション間で共通に使用したいフォームのスタイルやデータ構造を登録しておくことで、複数のプログラムから構成されるアプリケーションを効率的に開発できます。Delphi 3 ではグループでリポジトリを共有できます。
コードエディタ	キーボードマクロやカラムブロック、構文強調表示など、高度な機能をサポートし、ソースコードプログラミングにも役立つ機能を備えています。キー操作も 4 種類の中から選べます。
InstallShield Express	アンインストールにも対応したインストーラ作成ツールです。
Internet Solutions Pack(*1)	インターネット・イントラネットプログラミングを容易にする ActiveX コントロールです。
統計・財務関数(*1)	幅広いビジネス処理に対応するために統計・財務関数がサポートされています。
VCL32 ソースコード(*1)	Delphi に登録されているビジュアルコンポーネントライブラリ(VCL)のソースコードです。コンポーネントの構造や内部動作を調べたり、新しいコンポーネントを作成する際に参考にすることができます。
Open Tools API(*1)	ビジュアル開発環境にツールを組み込むためのインターフェースです。さまざまなウィザードツールをシームレスに統合できます。
WinSight32(*1)	システムで使われているすべてのウィンドウや Windows メッセージを追跡するためのユーティリティです。

## 31 Delphi 2.0 から引き継がれている特長

データベースコンポーネント	テーブル、問い合わせ、ストアドプロシージャに対して、専用のコンポーネントが用意されています。また、データベースやセッションに対応するコンポーネントを使って複数のデータセッションを生成でき、問い合わせの並列処理などが実現できます。データ表示用のコントロールには、グリッドやテキスト、メモ、イメージを使った表示などがサポートされ、さらに Delphi 3 では書式付き表示やグラフ表示がサポートされています。
設計画面でのデータ表示	Delphi では設計画面で実際のデータを表示できるため、アプリケーションを実行させることなく、どのようにデータが表示されるかを設計時点で確認できます。
データベースグリッド	グリッド形式でデータを表示する DBGrid は、コンボボックスを使ってデータを入力するための選択リストを指定できます。また、データ入力のためにプログラムを実行したり、他のフォームを呼び出すこともできます。項目ごとに色やフォントを指定でき、同一の項目を複数表示するといった設定もできます。
マルチオブジェクトグリッド(*1)	複数レコードを表示するため、データ表示用コントロールを任意のスタイルで配置できるコンポーネントです。
キャッシュアップデート	データベースに対する編集をいったんデータベースエンジンでキャッシュし、まとめて適用したり取り消すことができます。特にクライアント/サーバー環境では、問い合わせの実行を局所化することで、パフォーマンスを向上できます。
データベースエクスプローラ	テーブルやエリアスの表示・編集をサポートし、テーブル単位または項目ごとにフォームにドラッグアンドドロップしてデータベース対応フォームを設計できます。また、Client/Server Suite 版に付属の SQL エクスプローラではストアドプロシージャやトリガーのような SQL データベースのメタデータも表示できます。
データディクショナリ(*1)	アプリケーション間で共通に使いたいデータベース固有の属性を設定しておけます。
Local InterBase Server(*1)	スタンドアロン環境で、クライアント/サーバースタイルのアプリケーションを開発するための SQL データベースです。ストアドプロシージャやトリガーなどを使ったデータベースアプリケーションを開発できます。
Borland Database Engine	dBASE/Paradox/ASCII テキストのようなローカルデータの操作から、ODBC 接続、SQL Link による SQL データベースへの直接接続をサポートするスケールビリティの高いデータベースエンジンです。Delphi のデータベース対応コンポーネントを使う場合は、Borland Database Engine(BDE)の詳細について知る必要はありませんが、高度なデータ処理が必要な場合は、BDE API を直接呼び出すこともできます。
SQL モニタ(*2)	SQL のパフォーマンスを調査するためのツールです。
SQL Link および再配布権(*2)	SQL データベースに対する高速なデータ処理を実現するため、InterBase、Oracle、MS-SQL Server、Sybase、Informix、DB2 に直接接続できるミドルウェアです。開発したアプリケーションとともに再配布できます。
Visual Query Builder(*2)	SQL 文をビジュアルに設計するツールです。Query コンポーネントのスピードメニューから呼び出せます。
InterBase Server(*2)	開発とテストの目的で使える、4 ユーザー版の InterBase Server for Windows 95/NT です。
PVCS(*2)	チーム開発やバージョン管理を支援するバージョンニングエンジンです。

\*1 Delphi 3 Professional、Delphi 3 Client/Server Suite でサポートされます。

\*2 Delphi 3 Client/Server Suite でサポートされます。

- .. Delphi, ffffff□fffff,,□,, “” ,,,,,□
- .. Delphi,,□←,□“%<”Ž,32ffffffffff,‘,□,,,,,□□‘,ffff□fff,Š”,,□□,,Windows  
API,CE,□,,□fff□f,□—,□Š”,,ffff□fff,□—  
,,,,,□ffff□fff,“Ž□ffff□.EXE□,DLL,,ffff,,,,,□ffffDLL,●—  
,,,,,□□,□DLL,Ž,Ž□ffff,‘,,□,,,,,□□,□fff□f,,<” ,—,□,□Delphi,,,,,Š”—  
,□DelphiŽ□,Delphi,Š” ,,,,,,——,□
- .. Object Pascal,,,,,CE□,□□
- .. Delphi,‘,□,, Object  
Pascal,□^<Š,Š” ,,Pascal(CE,fffffŽCEffffffffff,‘)“□,,,,,□,,Š□,□“ ,,,,,ffff,□□,,□  
□ffff%□,CE□,,,,fffffŽCEffffffffff,ff□,Š—,□f□fff,Pascalfffff,15”<—  
Ž,Ž,,,,,□,,,’%□—,CE,●,,,,fffff,—  
,,,,,□□□Ž,●,ffffffffff,ffff□fff,Š” ,,ffffffffff,●,□^□,,——,□
- .. ^‘,f□fff,,CEŠ□,●,,,,,□
- .. Delphi 3,,□Delphi 2.0,,16fff”Delphi  
1.0,□□,fffff,,,,,●□,,,,,□□’,,,,,□□,□QuickReports,‘●,%□—  
,,,,,□fffff,‘,□,Ž“ ,□,CEŽ,●Š,,,,,□
- .. CE□,C++Builder,□Delphi 3,□□,ff□f,fff□fff,——,□
- .. Delphi 3,□□,‘Ž,,,,,——,□,□C++Builder,Delphi 3,Š’ ,%□,,,,,□□,fff□fff,‘Ž,,,,,□□, C+  
+Builder,□——,□
- .. Delphi 3,□C++Builder,□□,ff□f,fff□fff,——,□
- .. ,,,,,□C++Builder,,Delphi,ff□f,fff□fff,——,□,□,□,□Object Pascalfffff,‘,□,,,,,□Delphi  
3,,C++fffff,‘,□,,,,,□
- .. 16fffff□fff,Š” ,ff□f,,,,,□
- .. Delphi 3,□32fff□—,Š”f□f,,□Delphi 3 Standard,,16fff”Delphi for Windows,□Delphi  
Professional 3,,Delphi Client/Server Suite 3,,□16fff”Delphi and Database Tools,fff,,□
- .. Delphi 3 Standard,□ODBC,SQL Link,Ž,,,,,□
- .. Delphi 3 Standard,□ODBC,SQL Link,,%□,,,,,□ODBC,Delphi 3 Professional^□□SQL  
Link,Delphi 3 Client/Server Suite,,——,□
- .. MS-Access,f□f,^,,,,,□
- .. □,,Borland Database Engine 4.0,,□Access95,,□MDBf□ff□f,%□,Accessffff,‘,□,,,,,□