

《软件开发指南》丛书



Delphi 5

开发指南

谢玉凤 姜进磊 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



(京)新登字 158 号

内 容 简 介

本书比较全面地介绍了 Delphi 5——当今非常强大, 富有潜力的开发工具之一。

本书首先深入剖析 Delphi 5 的初中级特性, 让读者对 Delphi 5 有个整体的掌握; 之后以专题的形式, 详细地介绍了 Delphi 5 的新特性和功能, 包括 Delphi 5 的可视化程序设计方法, Delphi 5 可视部件类库 (VCL) 中各种部件的使用方法和它们在各类应用程序中的应用, 有关 OLE、DDE、多线程技术、图形图像、异常与调试、DLL、因特网编程 Windows 高级特性的开发方法, 以及数据库高级应用开发方法等全套开发技术。

本书适合有一定编程和开发经验的中高级程序设计人员。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: Delphi 5.0 开发指南

作 者: 谢玉凤 姜进磊 编著

出 版 者: 清华大学出版社 (北京清华大学学研大厦, 邮编: 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 刘 彤

印 刷 者: 国防工业出版社印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 30.75 字数: 744 千字

版 次: 2001 年 3 月第 1 版 2001 年 3 月第 1 次印刷

书 号: ISBN 7-302-01346-2/TP·522

印 数: 0001~5000

定 价: 40.00 元

Delphi 以其优秀的语言、丰富的环境、友善的界面、高速的编译器、强大的数据库支持、与最新编程技术的紧密结合以及独特的组件技术吸引着各个层次的开发和使用人员。无论是初学者或是经验丰富的程序员，他们各自的应用要求都可以在 Delphi 的神殿之中得到满足。Delphi 的巨大成功已经不容置疑地证明它已是当今最强大、最具潜力的开发工具之一。每一代新版本 Delphi 的出现，都给人们带来一次更大的惊喜。为了帮助读者更好地掌握 Delphi 的特性和技术，我们编写了本书。

本书的特点

- **新颖全面** 书中结合 Delphi 5 强大的编程环境，较全面地介绍了当前最热门的高级开发技术。

- **深入详尽** 在剖析 Delphi 5 新特性及应用的同时，对相应的理论或实际背景知识也作详细的介绍，务求令读者对高级开发技术有一个更深层次的理解。

- **注重实践** 本书使用逐步教导的方法，引入大量的范例，帮助读者更好地领略 Delphi 5 的精髓，并且迅速、简单地掌握 Windows 高级特性开发方法。大部分范例都是一步步地建立，通过中间步骤，逐部分地讲解每个特性，让读者能够亲手做出范例，享受编程带来的快乐。

- **点滴积累** 在书中不断穿插笔者许多开发过程的小技巧，让读者能够在学的同时更快速和直接地积累有益的开发经验。

全书共分为 15 章。其内容可分为基础入门和高级专题介绍两大部分。

基础入门部分包括第 1 章到第 5 章，主要向初学者介绍 Delphi 的编程知识，同时向有经验的用户提供 Delphi 5 的许多使用和编程诀窍。第 1 章主要讨论 Delphi 5 的历史及其开发环境的使用。第 2 章重点是 Object Pascal 的语言特性及其与面向对象编程技术结合的编程方法。第 3~5 章则讨论 Delphi 5 给用户提供的常用标准组件和编程资源。

高级专题介绍部分包括第 6 章到第 15 章。第 6 章说明如何创建派生组件和复合组件。在第 7~9 章中分别介绍数据交换的四个主要方面，即剪贴板、动态数据交换（DDE）、动态链接库（DLL）和对象链接与嵌入（OLE），论述数据交换的理论和在 Delphi 5 中的实现方法。第 10 章讲

解如何在 Delphi 5 中实现多线程与线程间通信。第 11 章主要介绍 ActiveX 控件的建立与使用。第 12 章中结合 Delphi 5 带有的多个针对 Internet 开发的组件，对 Internet 编程做一个介绍。第 13 章介绍本地数据库和网络数据库的概念和实现方法。分布式计算机的通信方式，COM/DCOM 接口和最新的 CORBA 协议以及它的实现方法则在第 14 章里介绍。最后，在第 15 章中介绍在 Delphi 独有的 MIDAS (Multi-tier Distributed Application Services) 下如何实现多层数据库。

本书的问世，得到了很多人的帮助，特别是网络上不知名的人的帮助，借此机会向他们致以衷心的感谢和最良好的祝愿。

由于作者水平有限，书中错误或不妥之处在所难免，欢迎读者批评指正，以便今后改正和完善。

编 者

2000 年 9 月于北京

第一章

Delphi 5 的开发结构与集成开发环境

这一章详细介绍 Delphi 5 的开发结构,告诉读者在 Delphi 中一个应用程序的设计过程所包含的文件和各个文件所包含的信息;还介绍整个 Delphi 5 的集成开发环境,给用户提供很多开发的诀窍,让用户可以更快地掌握 Delphi 系统的开发工具。

本章重点

- 📁 Delphi 的开发结构
- 📁 Delphi 5 的集成开发环境

Delphi 5 开发指南

1.1 Delphi 的开发结构

Delphi 应用程序的开发过程都是以一个项目 (Project) 为基本单位, 项目内最基本的程序单位就是窗体 (Form) 和单元 (Unit)。Form 就是窗体, 而 Unit 就是 Pascal 的原始程序代码。每一个 Form 都会对应一个 Unit, 但不是每一个 Unit 都有对应的 Form 存在。因为我们所定义的 Form 是以继承 TForm 的类而来的, 所以一定要有一个 Unit 来记录它相应的程序代码。在 Unit 里面, 可以直接撰写一些函数或是过程, 不一定是窗体定义的, 所以可以单独撰写 Unit 而不必有相应的 Form。

1.1.1 项目 (Project)

项目管理器 (Project Manager) 用来管理一个项目的所有内容。与项目相关的文件包括 Project Source 文件和 Project Option 文件。

如图 1-1 所示, 每一个项目里面都有 Project Source 文件, 用来控制整个项目的内容, 它描述了这个项目到底包含几个 Unit 和 Form, 以及整个应用程序的执行进入点 Application.Run。

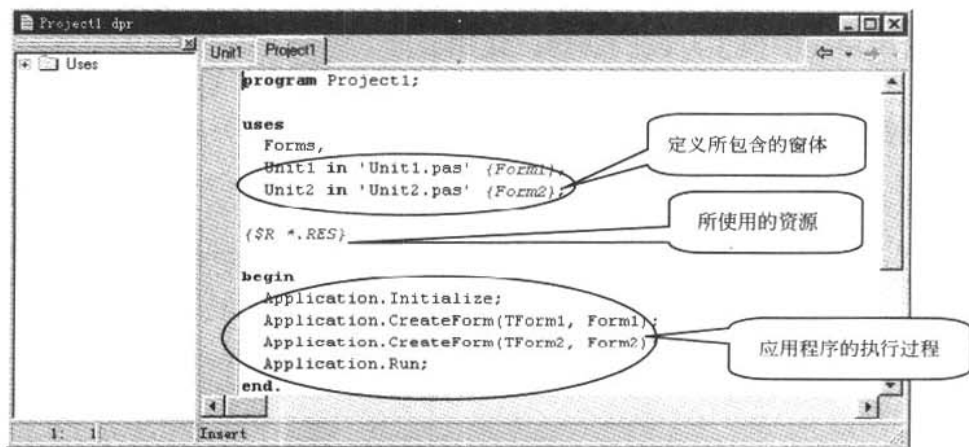


图 1-1 项目源文件, 其中包含项目中所有的窗体的单元文件存放的位置、使用的资源和应用程序的执行过程

Project Option 文件是存放每个项目本身特有的设定选项。用户可以在 Delphi 主菜单 Project 下的 Options 选项激活设定项目选项的窗体。在下一小节介绍 Delphi 的开发环境中会做更加详细的介绍。

Delphi 可使用和生成 DLL 文件、Delphi 编译后产生的 DCU 目的文件, 这种文件格式是 Object Pascal 专有的文件格式, 和一般的 OBJ 文件不同; 还可以使用和生成 ActiveX 组件; 也支持 COM/DCOM 标准, 它本身提供一组 Type Library 及一个用来定义 COM Interface 的工具。关于 ActiveX 和 COM 的技术在本书的其他章节有详细的介绍。

1.1.2 窗体 (Form) 和单元 (Unit)

Form 是 Delphi 应用程序存放可视化组件的容器 (Container)，当用户在设计阶段添加一个新的 Form 的时候，Delphi 会自动产生一个文本文件来描述 Form 的外观，这个文件的后缀名为 .DFM。如图 1-2 所示，在设计阶段用户只需选中窗体的弹出菜单的 View As Text 选项，就可以看到关于该窗体的一些描述。同样，我们在该窗体的文件描述中选中其弹出菜单的 View As Form 选项，就又可以以窗体的形式看窗体的设计结果。

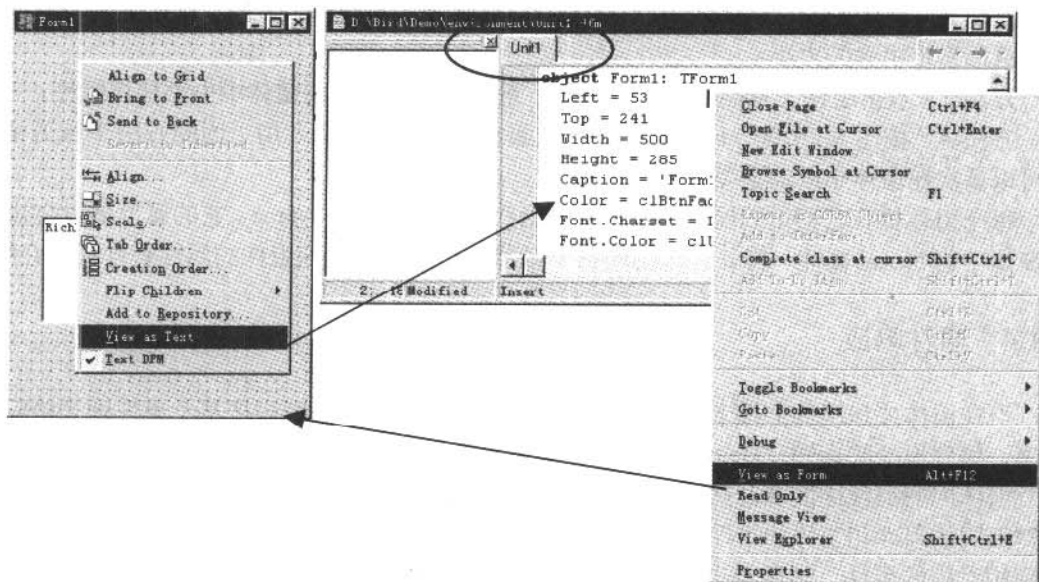


图 1-2 描述窗体的 .DFM 文件，可以在设计阶段打开，也可以直接在文本格式下面修改窗体的属性

窗体对应的单元 (Unit) 就是 Pascal 的程序代码文件，用户编写的关于该窗体的程序代码都保存在这里，包括窗体使用到的组件的事件响应程序、自行定义的函数。在 Delphi 环境下开发的应用程序基本上都是围绕着组件产生的事件信息撰写，因为 Windows 系统就是采用信息传递和信息驱动的方法，来联系组件和组件或是进行窗体和窗体之间数据的传送。

窗体有窗体特定的属性，用来设定窗体的一些外貌属性或是执行过程中的属性，所有的属性可以在 Delphi 的 Object Inspector 中查看。其中 Name 属性必须遵循 Pascal 中标识符的命名规则：

- ◆ 标识符是任意多个字母、数字和下划线等字符的排列。但是，只有前面 63 个字符是有效的。
- ◆ 标识符的第一个字符不能是数字，只能是字母或是下划线。
- ◆ 标识符不区分大小写，例如：Button1, button1, BUTTON1 都是指同一个标识符。

窗体中的属性用来帮助用户更方便地使用和管理窗体，用户可以在 Delphi 的帮助文档中找到各个属性的功能。在本书的范例中，我们也对用到的属性做详细的介绍。有些属性在修改之后，可以让别的属性调用，例如：当用户修改窗体的 Name 属性值之后，Caption 属性值就会调用该值，修改本身的数值。

1.1.3 使用组件

窗体是组件的容器，每个窗体都是一群组件或是控件的宿主。在 Delphi 主窗体中，用户可以在窗体右上的组件工具栏（Component Palette）中选择你所要的组件。向窗体中添加组件的方法有四种。

方法 1：单击组件工具栏中的组件图标，然后将光标移动到窗体内，在你所要的位置上单击一下，就可放置好高和宽均为默认值的组件。

方法 2：单击组件工具栏中的组件图标，然后将光标移动到窗体内，在你所要的位置上拖动鼠标设定组件的大小。

方法 3：双击组件工具栏中的组件图标，系统就会在当前的窗体中央添加该组件。

方法 4：按住 Shift 键，同时单击组件工具栏中的组件图标，用上面的方法 1 或方法 2 就可以多次在窗体中添加该类型的组件。

每个组件都有它特定的属性，惟一每个组件都有的属性是 Name，它的属性值设定规则和窗体中的 Name 属性一样。在本书的每个章节同样对所使用到组件的属性有详细的介绍。

1.1.4 事件响应

当在窗体或是组件上按下鼠标键时，Windows 会通过发送消息把某个事件通知给应用程序。接到事件的通知，Delphi 将调用相应的事件处理过程做出响应。程序员可以向窗体以及窗体内的组件提供多种这样的处理过程。Delphi 为每一个组件都设定了一系列的事件，当然不同的组件和窗体的事件表是不同的。例如图 1-3 所示，首先，在该组件的事件列表中找到相对应的事件，然后双击，就可以在窗体对应的单元中编写相应的代码。在这个例子中，当用户单击该组件之后，就会产生一个 OnClick 事件给 Windows，然后，Windows 就会调用 OnClick 相应的程序。

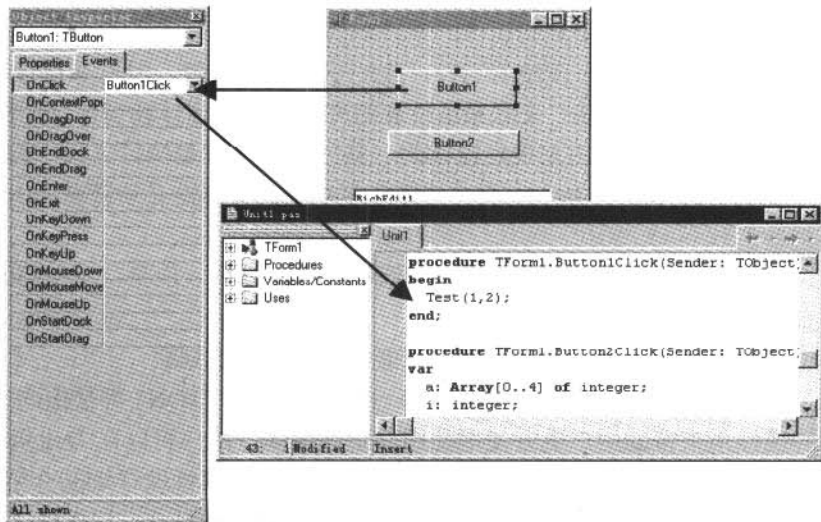


图 1-3 组件的事件响应程序的编写

下面给出 Form 对应的 Pascal 原始程序代码, 其中包含 Delphi 自动产生的组件声明、响应事件声明和函数声明的方法。

```
unit Unit1;
```

```
interface
```

```
uses Windows, Messages, SysUtils, Classes, Graphics,  
Controls, Forms, Dialogs, ComCtrls, StdCtrls;
```

```
type
```

```
  TForm1 = class(TForm)
```

```
    Button1: TButton;
```

```
    RichEdit1: TRichEdit;
```

```
    procedure Button1Click(Sender: TObject);
```

```
  private
```

```
    { Private declarations }
```

```
  public
```

```
    procedure add(x: integer);
```

```
end;
```

```
procedure Test(X: integer; Y: integer);
```

```
var
```

```
  Form1: TForm1;
```

```
implementation
```

```
{ $R *.DFM }
```

```
procedure Test(X: integer; Y: integer);
```

```
begin
```

```
  showmessage(inttostr(X)+' : '+ inttostr(Y));
```

```
end;
```

```
procedure TForm1.add(x: integer);
```

```
begin
```

```
  showmessage(inttostr(X)+' : '+ inttostr(Y));
```

```
end;
```

Delphi 自动产生的组件
声明

Delphi 自动产生的响应
事件声明

用户自定义的成员函
数声明

用户自定义的函数声
明

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    //程序开发者自行编写的程序部分  
    Test(1,2);  
end;  
  
end.
```

Delphi 自动产生的响应
事件过程样式

1.2 Delphi 5 的集成开发环境

作为一个可视化的开发工具，Delphi 给用户提供了一个灵活且容易使用的界面。作为一个专业的程序员，应该熟练掌握开发环境提供的功能，这样才可以开发一个 bug 较少的应用程序。

下面将介绍 Delphi5 的集成开发环境，本书并不是一个个地介绍菜单指令，而是通过设计一个应用程序来介绍如何使用集成环境中的工具。

1.2.1 项目和文件的使用

可以使用 Delphi 提供的主菜单 File 上的 Open 菜单来打开已有的项目文件，或使用 New 菜单创建一个新的项目（如图 1-4 所示）或使用 Reopen 菜单来打开曾经打开过的应用程序。Delphi 会为每一个应用程序生成一个项目文件 .DPR，为窗体文件生成 .DFM 和 .PAS 文件，不同的项目文件可以包含相同的窗体文件。当编译该项目文件的时候，将编译它所包含的所有单元定义。

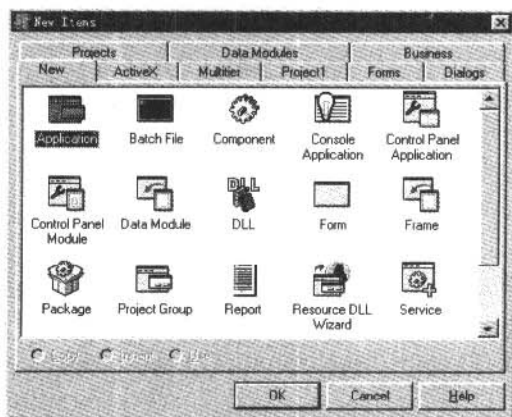


图 1-4 New 菜单弹出一个 New Items 对话框，让用户选择要创建的文件类型

对于应用程序项目来说，我们可以添加或删除一个单元（Unit），可使用主菜单 Project 上的 Add to project 或是 Remove from project 来实现。

每一个窗体文件对应一个 .DFM 文件，用来描述窗体中所使用到组件的属性值，可以

使用窗体的组件编辑器 (View as Text/ View as form) 来转换使用该文件或是设计窗体的显示, 通常情况下可以看到设计窗体。它还对应一个 .PAS 的文件, 它保存窗体使用到的组件和实现代码。我们可以使用 F12 键 (主菜单 View 上的 Toggle Form/Unit) 来实现设计窗体和文件之间转换。也可以使用单元查看器和窗体查看器来选择所要显示的单元或是窗体, 使用主菜单 View 下的 Unit 或 Form 选项来激活选择的窗体, 如图 1-5 所示。

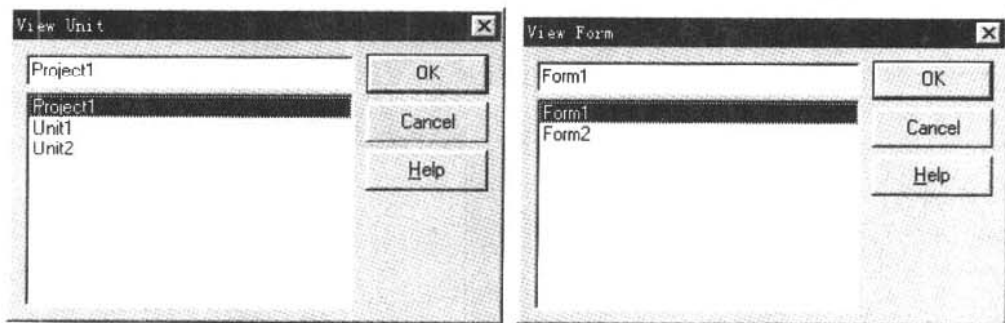


图 1-5 可以使用快捷键 Ctrl-F12 和 Shift-F12 来分别激活左右两个窗体

Delphi 允许用户依据不同的项目, 设定不同的桌面状态。例如: 一开启项目, 就打开窗体和它的工作环境。用户可以单击 Delphi 的主菜单 Tools 上的 Environment 选项, 在弹出的对话框中的 Preferences 中选定 Project Desktop 选项, 如图 1-6 所示。这样, 用户就会把目前项目的工作环境存成后缀名为 .DSK 的文件, 其实是用文本格式保存, 用户可以使用文本编辑器打开查看。

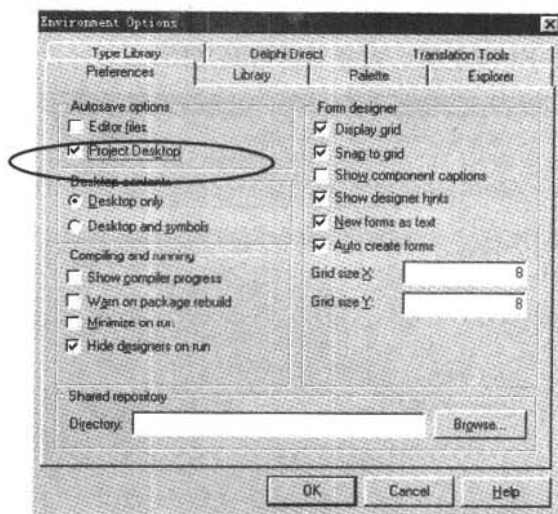


图 1-6 保存项目设计时候的桌面状态

在主菜单 Project 上的 Options 选项还可以设定关于该项目的属性, 包括主窗体、应用程序的图标、版本信息等。项目文件的属性值存在 .DOF 文件中, 该文件是一个文本文件, 用户可以直接使用文本编辑器 (Word) 打开查看, 但是不要随便更改设置。项目属性设定对话框总共包含 Forms、Application、Compiler、Linker、Directories/Conditionals、VersionInfo、

Packages 七个页面。下面对各个页面的属性设定做一下说明：

(1) Forms：在如图 1-7 所示页面中，Main Form 设定哪个 Form 是程序的主窗体。Auto-create Forms 设定当应用程序执行之后，需要自动建立哪些 Form 窗体。Available Forms 显示哪些 Form 是包含在这个项目里，但又不是在程序一开始就建立的窗体。

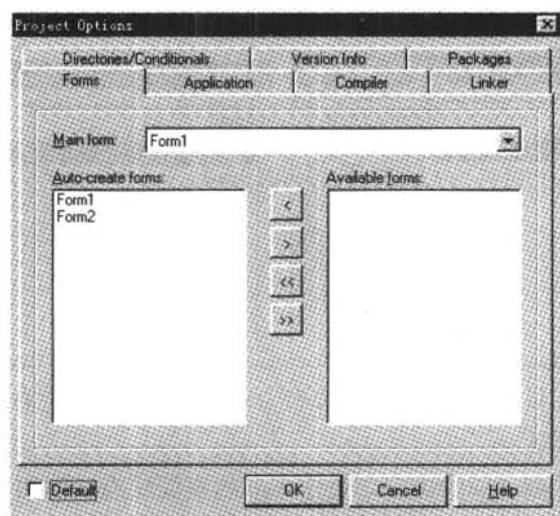


图 1-7 设定项目文件参数的 Form 页面

(2) Application：页面如图 1-8 所示，用来设定应用程序的标题 (Title)、图标 (Icon)、帮助文档 (Help file)。通常标题长度不可超过 255 个字符。若是用户没有设定图标的图案，Delphi 就使用默认的图标作为应用程序的图标。当用户设定了帮助文档之后，在应用程序的执行过程中，用户按下 F1 键之后，就会显示帮助文档。

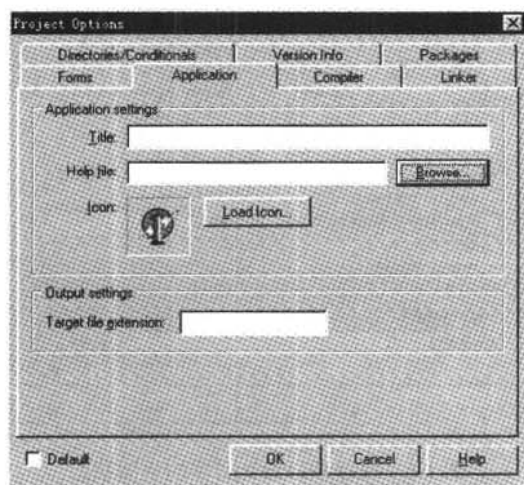


图 1-8 设定项目文件参数的 Application 页面

(3) Compiler：用来设定 Delphi 编译器的选项，这些选项分成 Code generation、Runtime errors、Syntax options、Debugging 和 Messages 五个选项组，见图 1-9。下面给出各个选项组内选项的说明：

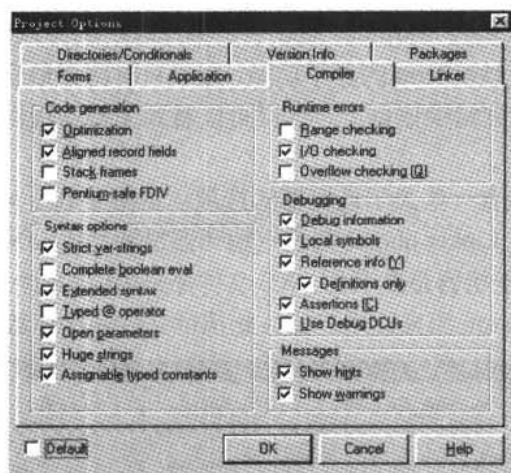


图 1-9 设定项目文件参数的 Compiler 页面

- ◆ **Code generation**: 这个选项组有四个复选框, Optimization 是设定用户在编译程序代码时, 是否要做最佳化的处理。最佳化的处理就是将应用程序执行文件变得更小、速度更快。Aligned record fields 是把记录类型内的各项数据在内存中计算的基本单位设定成 32 bit, 因为 Delphi 只能开发 32 位的应用程序。Stack frames 是把程序和函数的调用指定成 Windows 系统的使用方式。Pentium-safe FDIV 也是设定最佳化, 它只适用于 CPU 是 Pentium 的电脑。如果你的电脑不符合该条件, 绝对不要使用这个选项。在项目开发的过程中, 最好不要选择最佳化的选项, 因为这样会让编译速度变得很慢。最好是在项目开发完成之后选用最佳化处理, 再重新编译一遍即可。
- ◆ **Runtime errors**: 这个选项组用来设定是否将一些除错判断代码加到执行文件中, 当执行的过程中有发生指定类型的错误的时候, 应用程序就会中断执行, 并显示错误类型。Range checking 设定是否检查矩阵和字串的下标是否超出正常范围。I/O checking 是检查存取 I/O 设备时是否发生任何错误。Overflow checking 是检查程序中整数运算的结果是否超过最大允许值。
- ◆ **Syntax options**: 设定编译器的一些基本参数选项。Strict Var-string 设定字串参数具有错误检查功能。Complete boolean eval 设定布尔型变量具有完整的计算功能。Extended syntax 允许用户定义一个和程序参数回传方式一样的函数。Typed @ operator 使用 @ 运算符来控制各种类型的指标。Open parameters 允许用户在程序和函数的声明中开启字串参数。Huge strings 设定程序中声明的所有字串变量都会被指定为长字串, 最大可存放 2GB 的字串长度, 但是如果没有选中该复选框, 程序中声明的所有字串变量都会被设定成短字串, 最大可存放 256 个字节。Assignable typed constants 可以让编译器转成 Delphi 1.0 的编译标准, 只要不符合 Delphi 1.0 的语法都会被视成错误。
- ◆ **Debugging**: 设定包含哪些除错信息。Debug information 将除错信息加入编译过后的 .DCU 文件内。Local symbols 的功能是将区域符号加入编译过后的 .DCU 文件内。

- ◆ Messages: 设定 Show warnings 后, 会在编译时产生警告信息。而 Show hints 是设定错误信息窗体具有 Hint 的显示功能, 因为有时错误信息太长, 信息窗体放不下, 就可以将鼠标指到错误信息上, 这样可以看到完整的错误信息以 Hint 的方式显示出来。

(4) Linker: 该页面包含一些关于连接器的选项, 见图 1-10。Map files 选项组设定是否让连接器产生 Map 文件。Linker output 则设定编译器是要产生 .DCU 文件还是 .OBJ 文件。EXE & DLL options 告诉连接器是要产生 EXE 文件还是 DLL 文件。Memory sizes 设定栈的最大和最小空间。Description 用来让用户输入此执行文件的文字数据。

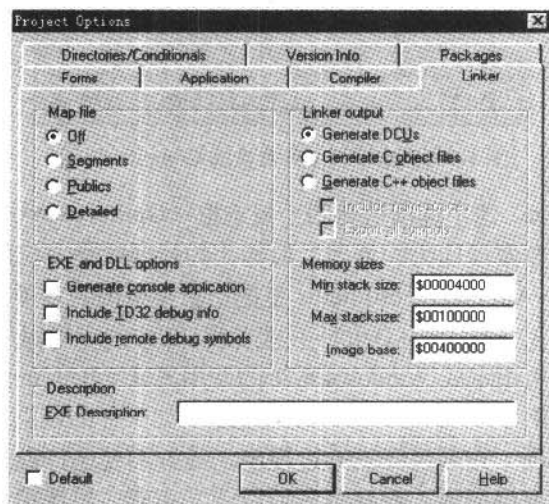


图 1-10 设定项目文件参数的 Linker 页面

(5) Directories/Conditionals: 该页面主要是设定项目产生文件的路径以及编译过程中寻找文件的路径, 如果没有设定, 就是保存在项目的相同目录下, 见图 1-11。Output directory 设定编译器产生 DCU 文件和 EXE 文件的存放目录。Unit output directory 另外指定一个目录来存放产生的 DCU 文件。Search path 设定所用到的原始程序的存放路径。Debug source path 这个路径是为除错设定的。如果用户在除错时会用到某些程序, 必须要将该程序的存放目录放在这个参数中。BPL output directory 和 DCP output directory 设定这两个文件输出的目录。Conditional defines 可以设定好几个目录, 每个目录必须以分号隔开。Unit aliases 可以将很多个 Unit 程序文件指定成同一个别名, 在程序中只要添加“uses 别名”即可, 其设定格式为 <oldunit> = <aliases>, 可以同时设定好几组, 并且以分号隔开。

(6) VersionInfo: 可以把版本信息加到项目的执行文件中, 见图 1-12。这个页面设定的信息保存在项目的资源文件中, 在本书后面有详细的介绍。

(7) Packages: 用来设定 Package 的功能。用户只要选中了 Build with runtime packages 选项, 则编译器会建立出 Package 类型的执行文档。

项目中所包含的窗体也就是所生成的应用程序中所包含的窗体定义。用户也可以动态地从项目中加入或删除一个窗体, 从而实现代码的重用。在主菜单 Project 上有 Add to Project 和 Remove from Project 两个选项, 可以让用户添加或是删除一个已设计好的窗体。如图 1-13 所示, 左图为向项目添加窗体的对话框, 用户可以选择一个已经设计好的窗体然后加

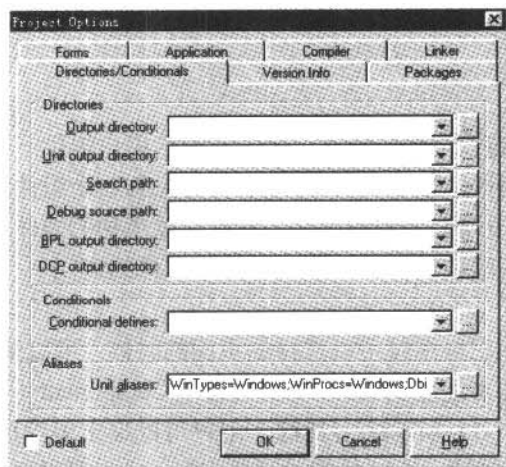


图 1-11 设定项目文件参数的 Directories/Conditionals 页面

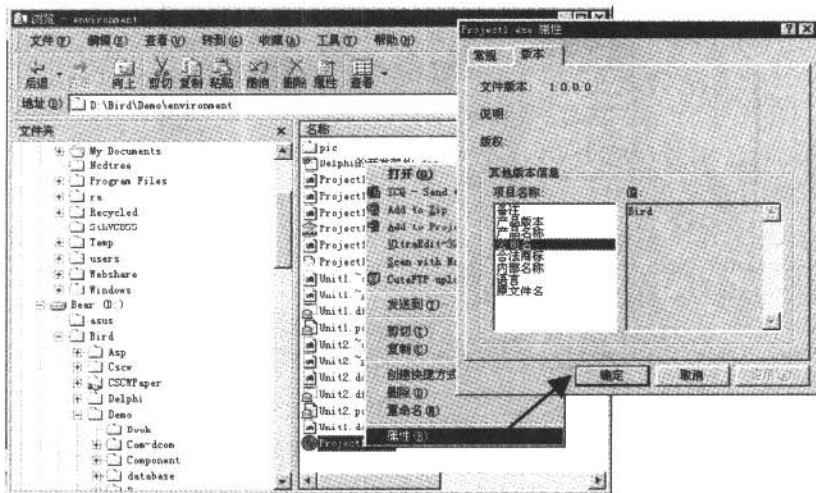


图 1-12 当应用程序的项目文件包含了版本信息之后，从编译出来的可执行文件中的属性可以查看到该应用程序的版本信息

入到该项目中，我们知道一个窗体的设计是保存在 .pas 文件中的。右图则是将项目中的窗体显示出来，用户可以选择想要删除的窗体名称。

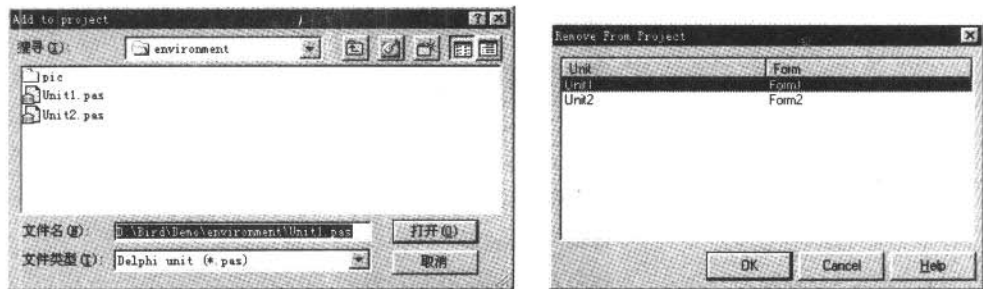


图 1-13 左图为向项目添加窗体的对话框；右图为删除项目中窗体的对话框

1.2.2 组件的使用

组件是 Delphi 编程的一个重要特性，所以必须给用户提供一个容易存取的方法以便存取在系统中已经注册的组件。在 Delphi 的程序界面上有一个组件面板列出了所有已经注册的组件。用户只需要单击所要使用的组件，然后在欲添加的窗体上单击，就会在光标单击的位置上加入该组件。下面我们介绍一些关于使用组件的窍门。

1. 组件的查找

当用户所注册的组件已经多过可以选择的页面的时候，可以使用下面两种方法来解决：

- ◆ 将常使用的组件页面放到前面。使用主菜单 Component 上的 Configure palette 选项，就会弹出如图 1-14 所示的对话框。用户可以设定组件页面的位置 (Move down/Move up)，甚至创建一个新的组件页面 (New)，也可以更改页面的名称 (Rename)。

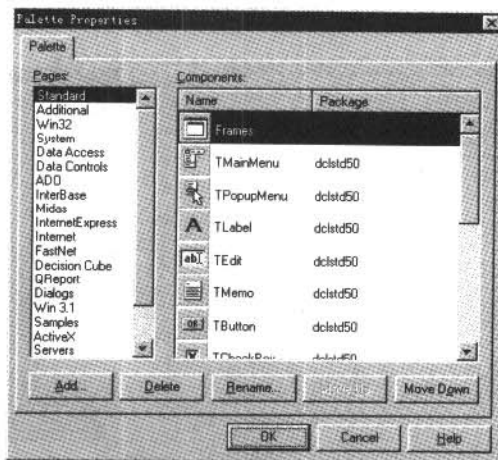


图 1-14 组件页面的设定，灵活的使用它所提供的服务可以减少对组件的查找时间

- ◆ 使用组件查找器查找。在主菜单 View 上的 Component List 选项，会激活如图 1-15 所示的对话框。下面的组件列表列出了所有已经注册的组件并按照名字排序，用户可以在上面的文本框中敲入要查找的组件名称，会依序显示匹配最接近的组件。这是一种比较科学的查找不常用组件的方法。



图 1-15 组件查找器，可以自动添加到设计窗体上

2. Object Inspector

每一个组件（包括窗体）都有它所包含的属性和事件，用户可以在 Object Inspector 中查看该组件的属性设定和各个事件是否激活相应的代码。如图 1-16 所示，组件的某些属性还包含子属性，用户可以单击属性旁边的加号来显示它的子属性。该属性扩展后，加号就会变成减号。如果属性值可以多重选择，则使用下拉式菜单供用户选择，双击该属性值会选中下一个选择。

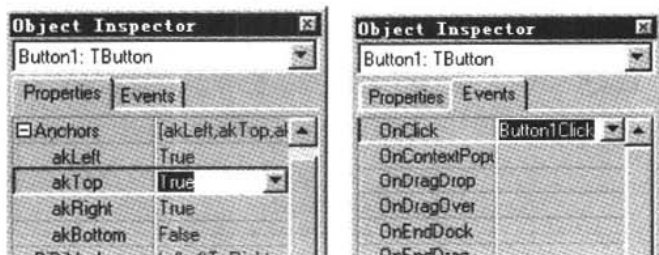


图 1-16 使用 Object Inspector 查看组件的属性

3. 组件的剪贴

在 Delphi 中可以实现将一个窗体中的组件粘贴到另一个窗体中。前面已经介绍了窗体的定义就是一系列窗体属性和它的成员组成的文本定义。所以，组件的粘贴就是将组件的文本描述复制到剪贴板中，然后，添加到我们所要加入的窗体定义中。如图 1-17 所示，当我们拷贝一个组件之后，在剪贴板中的数据为文本格式，并且是该组件的属性描述。

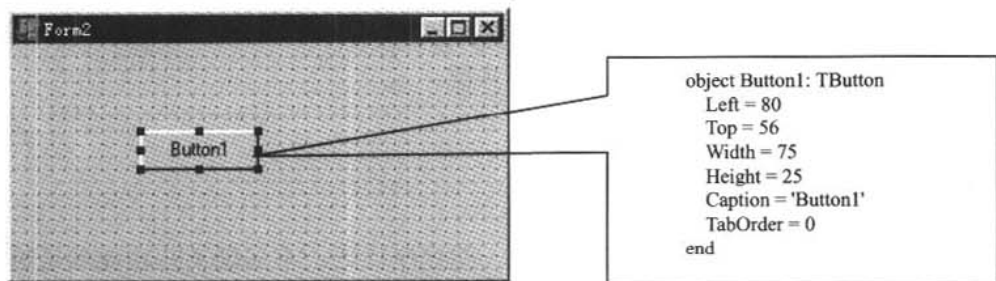


图 1-17 组件所对应的属性文本描述



如何让 Object Inspector 永远保持在窗体设计程序和代码编辑器的上面？

Object Inspector 有一个弹出菜单，其中有一个 Stay on Top 的指令，用户将它设为 True 之后，Object Inspector 窗体就会永远置于上方。使用过 ICQ 的用户一定知道这样设定的好处。

用户可以在文本编辑器中修改复制的组件属性描述，可以往一个组件属性描述中加入或是删除一个属性值，然后再复制到窗体中。如果其中有一个属性值是该组件没有的，就会出现错误信息。

```
object Button2: TButton
  Left = 32
```