

第一章 Delphi 2.0 安装及其新特性

Borland 公司在微机编译软件和微机数据库软件方面具有很强的技术背景。90 年代初, DOS 环境下的 Borland 系列编译软件曾风靡全球, 如今, 集图形用户环境和数据库技术为一体的 Borland Delphi 2.0 又被评为最优秀的小型数据库软件。

Borland Delphi 2.0 目前有以下三个版本:

- ①台式机版本(Delphi Desktop 2.0);
- ②开发版(Delphi Developer 2.0);
- ③客户机/服务器套件版本(Delphi Client/Server Suite 2.0)。

本章以 Windows 95 中文版上的开发版为例, 详细讲述 Delphi 的安装过程及最新功能, 以使初学者能很快掌握其精华。

Delphi 2.0 是 Borland 公司 1996 年推出的最新产品, 它在 Delphi 1.0 的基础上增加了许多新的特性, 本章后几节就 Delphi 2.0 中新增的特性作一介绍。关于 Delphi 2.0 的完整介绍请参阅本书的后续章节。

在安装 Delphi 2.0 之前, 请认真阅读本章内容, 在此我们假定读者已经比较熟悉中文 Windows 95 的环境。

1.1 Delphi 2.0 对软硬件环境的要求

安装 Delphi 2.0 的计算机要求满足以下条件:

- ①Windows 95、Windows NT 3.51 或者与之 100% 兼容的其它操作系统。
- ②至少用 80486 处理器, 8MB 内存; 最好用 P5100 以上的处理器, 16MB 内存。
- ③如果选择完全安装 Delphi 2.0, 大约需要 111MB 的硬盘空间, 其中 10MB 用于安装过程中存储一些临时文件, 安装结束后将被释放; 如果选择紧凑安装, 大约需要 34MB 空间, 再加上 10MB 的临时空间; 用户也可以选择定制安装, 仅仅把需要的内容安装到硬盘上。
- ④需要一个 CD-ROM 驱动器。
- ⑤一个与 Microsoft 兼容的鼠标器。

1.2 Borland Delphi 2.0 的安装

(1) 启动 Windows 95, 单击开始按钮, 运行 Windows 资源管理器, 在 Delphi 2.0 的光盘上运行它的安装程序 Setup.exe, 或打开运行对话框键入 Setup.exe 的路径名。注意, Delphi 2.0 的光盘中也附带了 16 位的 Delphi 1.0, 如果用户的机器是 Windows 3.X 环境, 就只能运行 Delphi 光盘上的“\DELPHI16\INSTALL\SETUP.EXE”程序。

(2) Setup 首先显示一个窗口,提示 Setup 正在安装一个安装外壳和向导,以帮助完成全部安装工作。下部的百分比显示安装向导的进程。这种风格是 Windows 95 鼓励的,一个好的 Windows 95 应用程序应当对一个新手来说也是友好的。

(3) 安装好向导以后,屏幕上出现一个对话框。对话框上显示欢迎您使用 Delphi 2.0 安装程序,并建议用户在运行 Setup 之前把所有 Windows 程序退出。如果需要这么做,请按下对话框的 Cancel 按钮,然后关闭所有的 Windows 程序;如果按下 Next 按钮,安装将进行下一步。另外,对话框还显示了此程序的版权信息。再按下 Next 按钮,屏幕上显示一个对话框,提醒用户仔细阅读 Install.txt。这是个英文的文档资料,用户可以不看,因为本章包括了它的主要内容。

(4) 如果按下了 Next 按钮,Setup 将做一些初始化工作,并把光标置成沙漏状,提醒用户稍候片刻;然后屏幕上出现一组单选按钮,让用户选择安装选项。共有三个选项可供选择,如图 1.1 所示,其中:

1) Full(完全); Setup 将把全部内容安装到用户的硬盘上。建议大多数用户用此选项。

2) Compact(紧凑); Setup 将只安装必须的内容。如果用户的计算机硬盘剩余空间有限,可以选择这个选项,不过许多有价值的内容就用不上了。

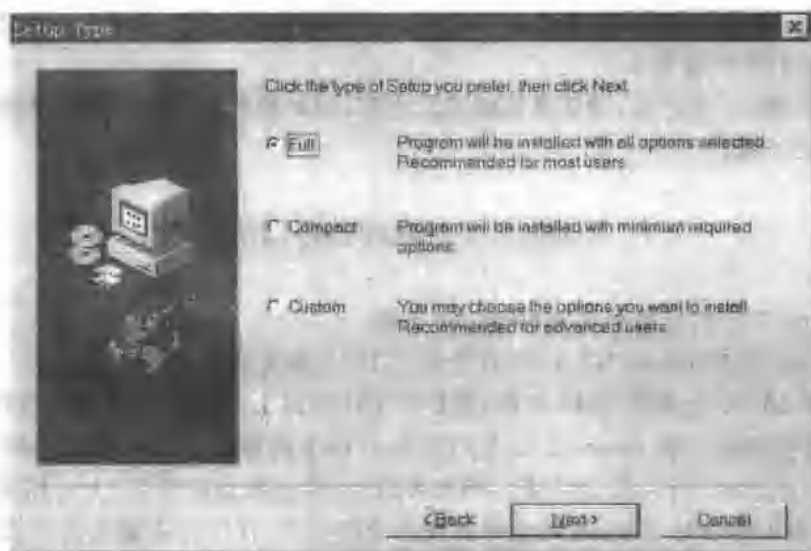


图 1.1 Delphi 2.0 的安装是可定制的

3) Custom(定制): 这个选项允许用户进一步控制安装的细节。高级用户可以选择这个选项。

(5) 如果用户选择的是定制选项并按下了 Next 按钮,屏幕上会出现一个对话框,显示 Delphi 2.0 的全部内容列表。一共有五项,每项左边是一个消除了边框的复选框,带对勾者表示选择了该项,空白表示没有选择该项;右边显示选择该选项需要的硬盘空间。稍下是每一选项的简短描述。有的选项还可以通过按下 Options 按钮进一步细化。如 Options 按钮变灰,则表示此选项无细化内容,如图 1.2 所示,其中:

1) Delphi 2.0: 这是 Delphi 2.0 的核心部分,一般情况下用户应当选择这个选项。按下

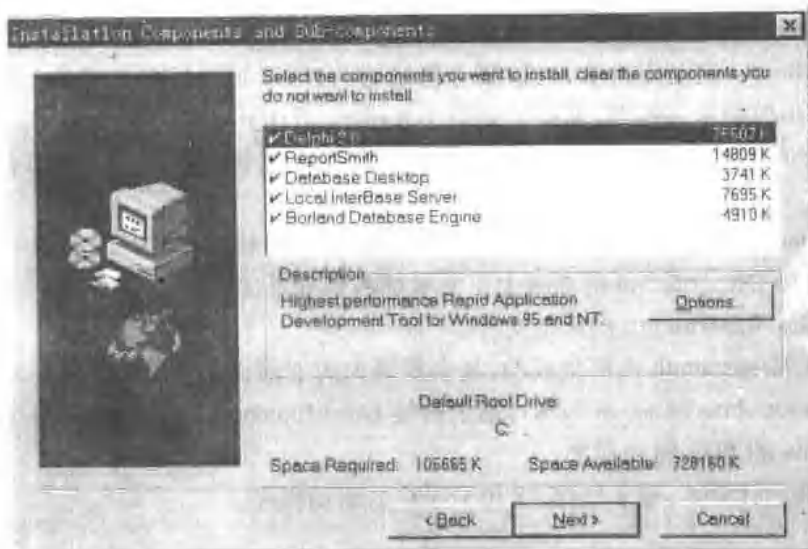


图 1.2 Delphi Developer 2.0 的安装选项

Options 按钮,可进一步控制安装细节,其细节如图 1.3 所示,其中:

- ① Program Files: 包括 Delphi 2.0 的 IDE、编译器及其相关文件。
- ② Image Editor: 生成和修改图像资源的编辑器。
- ③ WinSight32: 32 位的高级调试工具,可以显示 Windows 95 和 NT 的所有消息。
- ④ Sample Programs: Delphi 2.0 的示范程序,示范了 Delphi 2.0 大部分出色的特点。
- ⑤ Visual Component Library Source: Delphi 2.0 的 VCL 源程序代码,让用户学习 VCL 的内部机制以及如何生成自己的元件(Component)。
- ⑥ Team Development Support: 支持 PVCS(版本控制系统)。

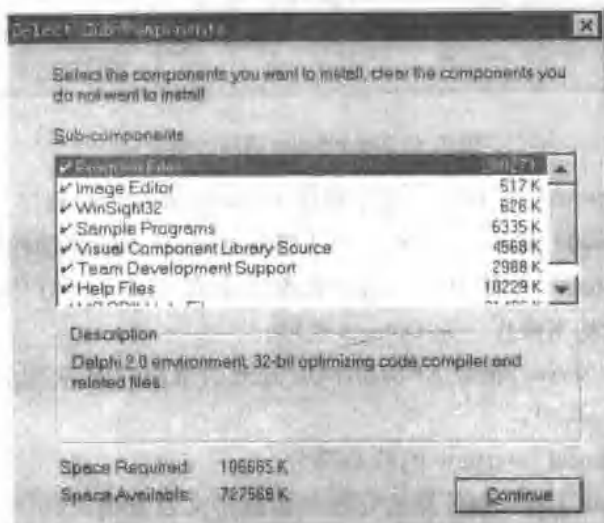


图 1.3 Delphi 2.0 选项的细化

⑦ Help Files; Delphi 2.0 及其实用工具的在线帮助。

⑧ MS SDK Help Files; 关于 Microsoft SDK 的在线帮助。

2) ReportSmith; 这是支持客户机/服务器结构的, 用于制作报表的工具。按下 Options 按钮, 可进一步控制安装细节, 其细节如图 1.4 所示, 其中:

① ReportSmith; ReportSmith 的核心部分, 可以创建、编辑和打印所有类型的数据库报表。

② Run-time Report Viewer; 运行期的 ReportSmith 报表观察器。

③ Help Files; ReportSmith 的在线上下文敏感的帮助系统。

④ Examples; ReportSmith 的示范文件。

⑤ Macros; ReportSmith 内部集成的、复杂而强大的宏语言示范文件。

⑥ Connection; InterBase (via BDE); 用于链接 Local InterBase 和 InterBase Server for NT、Netware 和 Unix 的 BDE 驱动程序。

⑦ Connection; ODBC; 用于链接 32 位 ODBC 的驱动程序。

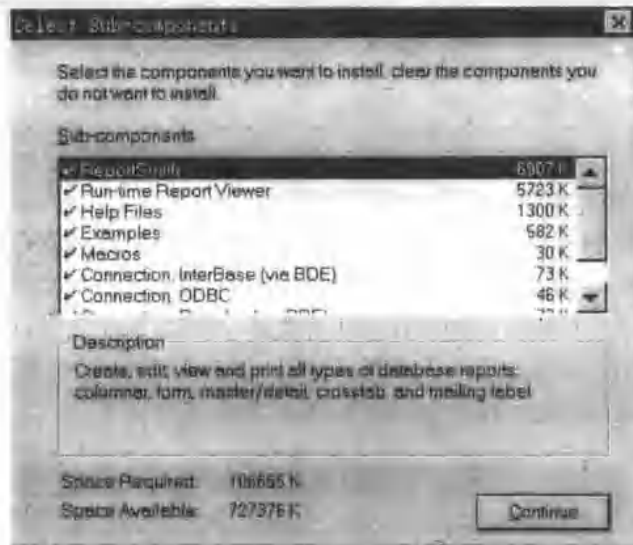


图 1.4 ReportSmith 选项的细化

⑧ Connection; Paradox (via BDE); 用于链接 Paradox 表的驱动程序。

⑨ Connection; Visuald Base (via BDE); 用于链接 dBASE 表的驱动程序。

3) Database Desktop; 一个简单而实用的数据库桌面工具, 让用户无需 dBASE, Paradox 就可以进行数据库的基本操作。无 Options 选项。

4) Local InterBase Server; 符合 ANSI-92SQL 规范的单用户服务器。按下 Options 按钮, 其细节如图 1.5 所示, 其中:

① Program Files; Local InterBase 的核心部分。

② InterBase Console Tools; 服务器的管理和数据定义及查询工具。

③ InterBase Windows Tools; InterBase 的 Windows 查询工具。

④ InterBase SQL Tutorial; 关于 Windows ISQL 和 SQL 的教程。

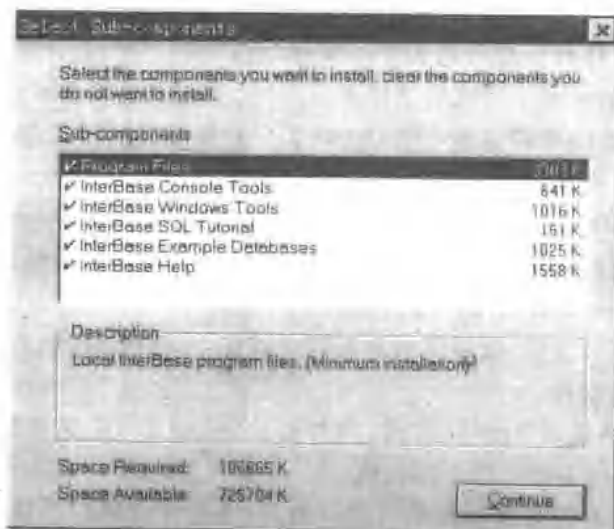


图 1.5 Local InterBase Server 选项的细化

⑤ InterBase Example Databases: 用于 InterBase 的示范程序和 SQL 教程的例子数据库, 包括一个职员管理数据库和一个销售管理数据库。

⑥ InterBase Help: InterBase 的帮助文件。

5) Borland Database Engine: Borland 的数据库引擎, 是 32 位高性能的数据库技术, 此项是必选项目。无 Options 选项。

(6) 上述选项选择好以后, 按下 Next 按钮, 屏幕上将显示 Delphi 2.0 的安装路径。缺省情况如图 1.6 所示。

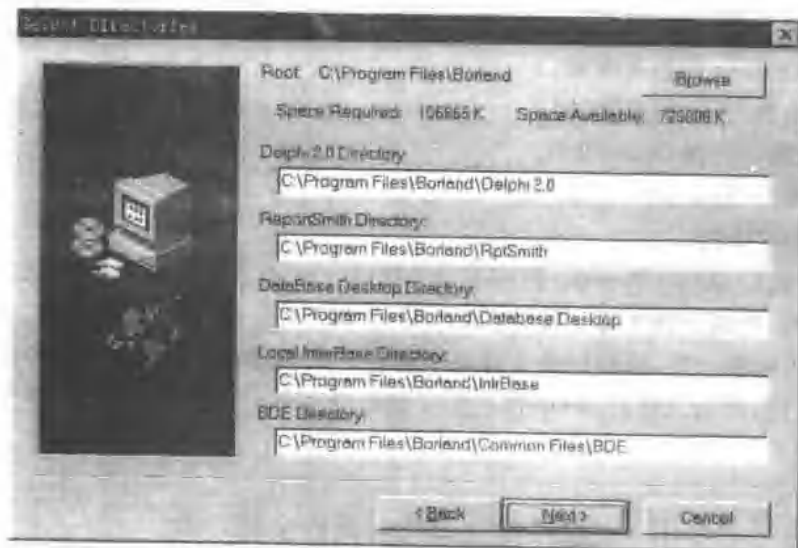


图 1.6 Delphi 2.0 文件的缺省路径

上述路径均可修改, 不过我们建议用户尽量不要修改。

(7) 按下 Next 按钮后, Setup 开始检查硬盘上是否有足够的空间。如果 Setup 发现没有

足够的空间,将提问用户是强行安装(Yes),还是返回到前面重新设置选项(No)。如果选择了强行安装,Setup 将开始执行拷贝工作至硬盘空间耗尽时,提示用户删除一些硬盘上的文件,以腾出足够的硬盘空间或者选择退出安装。如果有足够的硬盘空间,Setup 就开始拷贝工作,先前安装的向导将显示一些 Delphi 2.0 的特点及求助信息。在此过程中,用户随时可以按下 Cancel 按钮中止安装。

(8)拷贝结束后,Setup 将建立一个程序组,缺省名为 Borland Delphi 2.0。建立好程序组后,Setup 将进行联机注册。如果用户的计算机没有装调制解调器,可以按下 Cancel 按钮,中止联机注册。Setup 提醒用户用信函的方式注册。

建议用户一旦安装完毕,就用调制解调器或其它方式进行注册,以尽快得到购买正版软件用户所应得到的服务。

(9)最后,Setup 将显示 Readme 文件,内有关于 Delphi 2.0 的最新信息。

至此 Delphi 2.0 的安装工作全部完成。

尽管 Delphi 2.0 的安装路径都是可设置的,但还是建议用户尽量不要修改缺省的路径名,以免引起混乱。另外,即使用户并没有选择安装 Local InterBase Server (LIBS),Setup 程序总是将 InterBase Client Library DLL 拷贝到自己的 Windows 95 文件夹下的 System 文件夹,如果用户没有安装 LIBS,可以在安装结束后,从 System 文件夹中删掉 GDS32.DLL 文件。

1.3 Delphi 2.0 的卸载

如果用户的计算机上已安装了 Delphi 2.0,也可以随时卸载(Uninstall)它。注意,如果 Local InterBase Server 正在运行,用户必须用鼠标右键单击任务条上的 InterBase Server 图标,再在弹出菜单中单击 Shutdown,以关闭 InterBase Server。卸载的步骤如下:

- ①进入 Windows 的控制面板。
- ②双击“添加/删除程序”图标。
- ③在列表框中选择 Delphi 2.0。
- ④按下“添加/删除(r)...”按钮。
- ⑤按下“Yes”按钮。

如果用户运行 Delphi 的 Setup 程序不只一次,Uninstall 程序仅仅删除最后一次安装的内容。这是因为,所谓卸载实际上就是在安装过程中把文件拷贝及修改配置文件的操作记录下来,用户每安装一次,操作记录文件就被刷新一次。举例说,如果用户第一次安装时没有安装帮助文件,以后又运行 Setup 仅选择安装帮助文件,那么 Uninstall 程序仅仅把帮助文件删除。

1.4 编译器及运行时间库的新特性

1.4.1 数据类型

Delphi 2.0 定义了几种新的数据类型以适应 32 位的环境,分别是 Character 类型、String 类型、Variant 类型以及 Currency 类型(Currency 类型用于处理数值很大、精度要求

很高的场合)。

在 Delphi 1.0 中, Integer 和 Cardinal 是 16 位的,而在 Delphi 2.0 中改成 32 位的。如果要明确地声明一个 16 位的整型数据,请用 SmallInt 和 Word 类型。

Delphi 2.0 除了支持 255 字符长度的字符串类型以外,还支持几乎无限制长度的长字符串类型。新的编译器选项 \$H,用于控制保留字 String 声明的是短字符串类型,还是新的长字符串类型。\$H 选项缺省是 \$H+,表示缺省是长字符串类型。不过,即使设置了 \$H+,您还可以定义一个带指定长度的短字符串类型,例如:

```
{ $H+ } {设置长字符串类型}
Vars1:String; {长字符串类型}
s2:String[80]; {短字符串类型,最大长度为 80}
```

所有的 Delphi 2.0 元件都采用长字符串类型。如果您的 Delphi 1.0 应用程序用到的元件中含有字符串类型特性,请用 \$H+ 选项重新编译程序。

Delphi 2.0 还引进了 Variant 类型,主要用于当一个变量的类型在编译期是未知的,或者在运行期有可能要改变的情况下,让用户可以动态地改变一个变量的类型。这在实现 OLE 自动化或者在执行某种数据库操作时,对用户的客户端程序来说服务器的参数类型是未知的时候,这种变量类型是很有用的。

Variant 类型是一个 16 个字节的结构,可以表达 String、Integer 或者 Floating Point 值,编译器通过 Variant 类型的声明来识别它。在大多数情况下,用户可以把 Variant 类型当作其它数据类型用。当进行 OLE Automation 操作时,Variant 可以响应 OLE 服务器端的方法(Method)调用。

1.4.2 改进后的编译器

1. 编译器的新选项

Delphi 2.0 引进了两个新的编译器选项:

\$H:用于控制是否使用长字符串类型。

\$J:用于控制类型常量。

另外,对大多数单字母的编译器选项,都有一个长的编译器选项与之对应。例如 \$LONGSTRINGS 就对应着 \$H 选项。

2. 编译的优化

Delphi 的 32 位编译器在生成目标代码时提供了几种新的优化手段,使用优化可以提高代码执行速度的 3~4 倍。例如,寄存器优化能够把经常使用的变量和参数放到 CPU 寄存器,以减少访问这些变量和参数的时间。

通过使用 Project|Options... 命令,可以打开 Project Options 对话框,然后在对话框 Compiler 页的优化选项上设置优化选项。

3. 调用约定

Delphi 2.0 提供了几种选项,用于过程或函数的参数传递约定。除了在 Delphi 1.0 中的 Pascal、Cdecl 和 StdCall 以外,Delphi 2.0 增加了 Register 约定,这是 Pascal 和 C 调用的混合。在缺省情况下,Delphi 总是使用 Register 约定。

Register 约定在传递参数时尽可能地使用寄存器,头三个适合 32 位寄存器的参数被

依次传递到 EAX、EDX 和 ECX 寄存器中,其它参数遵循 Pascal 约定。另外,Register 约定还能保证 EBX、ESI 和 EDI 这三个寄存器是保留的。

4. 单元结束段

在一个单元中可以包括一个可选的结束段代码(前提是单元中已有 inalization 部分),您可以把它当作单元的“退出码”,相当于在 Delphi 1.0 中调用 ExitProc 和 AddExitProc 函数。结束段代码以保留字 Finalization 开始,它只能在 Initialization 段之后和在 End 之前。

1.5 改进的元件库

1.5.1 新元件

Delphi 2.0 中增加了几个新元件(Component),用于 Windows 95 的公共控制和新的数据库栅格。新的 Windows 95 元件的源代码在 SOURCE\VCL\COMCTRLS.PAS 中,VCL 的帮助文件包含有这些元件的特性、方法和事件。下面是新增加的元件:

| | | | |
|----------------|--------------|-------------|-----------|
| TTabControl | TPageControl | TTreeView | TTrackBar |
| THeaderControl | TProgressBar | TRichEdit | TUpDown |
| TListView | TStatusBar | TDBCtrlDrid | |

1.5.2 OLE 控制

Delphi 2.0 允许用户使用 OLE 控制(OCX),并把它们当作 Delphi 的元件使用,就好像使用 Delphi 1.0 中的 VBX 控制一样。

按以下步骤可引入 OLE 控制:

①用菜单上的 Component | Install 命令打开 Install Components 对话框,如图 1.7 所示。

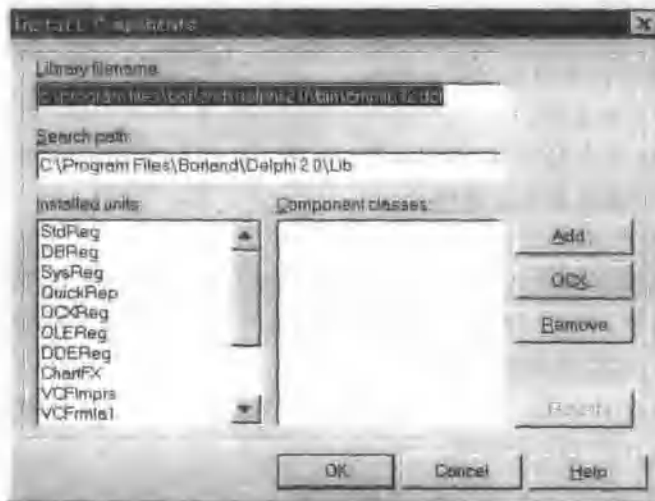


图 1.7 Install Components 对话框

②按下 OCX 按钮打开 Import OLE Control 对话框,如图 1.8 所示。

③OLE 控制在使用之前必须先注册。如果您想使用的 OLE 控制已出现在 Registered

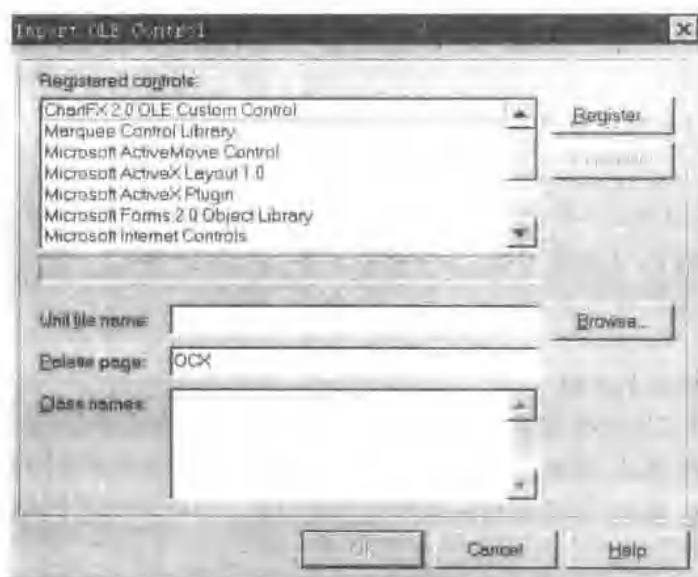


图 1.8 “Import OLE Control 对话框

Controls 列表框内,在该项上单击,然后进行步骤(6);否则单击 Register...按钮,打开 Register OLE Control 对话框,如图 1.9 所示。

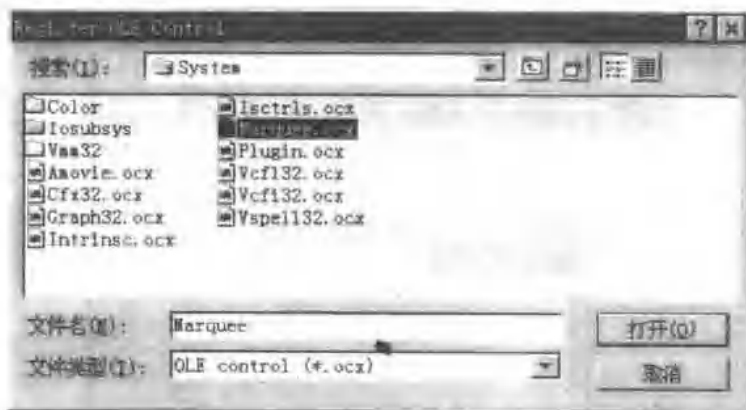


图 1.9 Register OLE Control 对话框

④在 Register OLE Control 对话框内定位,并打开一个 OLE 控制,该控制将出现在 Registered Controls 列表框内。

⑤在 Registered Controls 列表框内选择该项控制。当用户引入一个控制时,Delphi 必须为该控制生成一个元件外表,单元文件名和类名。编辑框允许用户为该控制指定一个文件名和类名,当然 Delphi 已在编辑框内自动生成了缺省的名字。

⑥按下 OK 按钮,Delphi 关闭 Import OLE Control 对话框,并返回到 Install Components 对话框,一个新的 OCX 控制就出现在 Installed Units 列表框内。

⑦按下 OK 按钮,Delphi 关闭对话框,然后重构元件库,并把该 OCX 加入到元件选项板上。

1.5.3 OLE 自动化

OLE 自动化(OLE Automation)用于 Windows 应用程序之间的相互操纵,其中被自动化操纵的应用程序称之为自动化对象或自动化服务器,而操纵其它程序的应用程序被称为自动化控制器或自动化客户。OLE 自动化就是一个应用程序控制另一个程序行为的协议。

Delphi 2.0 支持 OLE 2.0 自动化操纵,用 Delphi 设计的应用程序可以自动操纵其它程序,或充当其它程序的 OLE 服务器。关于 OLE 自动化的示范程序在 DEMOS\OLEAUTO 目录中(如果安装了示范程序的话)。

1.5.4 TDBGrid 的改进

用 Delphi 1.0 开发数据库应用程序,TDBGrid 是最常用的元件之一。现在 Delphi 2.0 又对它作了重大改进,允许用户修改栅格的显示特性,而不依赖于字段的特性,TDBGrid 支持数据单元调色、宽度、字段名和标题等特性的设置。TDBGrid 以下列两种方式之一工作:

①无列特性定义。这时候 TDBGrid 相当于 Delphi 1.0 中的 16 位 TDBGrid;有关列的信息(例如标题)直接来自数据库的字段;要改变列的宽度和列的顺序,就要分别修改相应字段的 DisplayWidth 和 Index 特性;在一个栅格中改变字段的宽度或顺序,将影响使用同一数据源的其它栅格。

②有列特性定义。这样,TDBGrid 就可以彻底改变栅格的外表,每一列都可以按任意顺序和任意组合指定字段名,而不依赖于数据集,如图 1.10 所示。

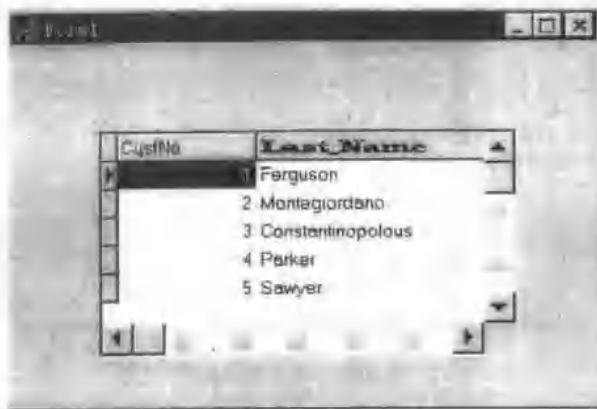


图 1.10 每个字段的标题字体可以不同

1.6 数据库的改进

1.6.1 表格的本地过滤技术

Delphi 2.0 允许用户显示一个表格时使用本地过滤,而不是像过去那样,使用查询技术从数据库服务器中过滤记录到本地的数据集中。使用过滤技术,用户可以从各种表格或某些查询中隐藏或显示本地化的记录。

TTable 元件和 TQuery 元件都增加了两个特性：一是 Filter 特性，用于设置过滤条件；另一个是 Filtered 特性，用于作为过滤开关。

Delphi 的数据集过滤是用 Object Pascal 语言编写的，其语法跟 SQL 的 Select 语句的 Where 部分相似。也就是说，用户可以通过写代码的方法计算或操纵过滤的行为。

1.6.2 TField 支持 Lookup 字段

Delphi 2.0 除了可以使用一般的数据字段和计算字段外，还支持一个新的字段类型 Lookup。Lookup 字段有点像 Calculated 字段，不过其“计算”用于其它表格的查找，并且 Lookup 字段总是只读的。

一个字段的 Calculated 特性和 Lookup 特性是互斥的：设置其中一个为 True，另一个将自动变为 False。如果您用了 Lookup 字段，则必须指定四个特性值，以确定去哪儿和怎么去查找。这四个特性如下所述，它们的名字反应出 Lookup 字段与 Lookup 表格的关系很像 Master Detail 的关系：

- ① LookupDataSet：指定要查找的数据集。
- ② LookupKeyFields：指定一个数据集中用于查找的关键字段。
- ③ LookupDetailFields：指定一个 Lookup 表格中对应于关键字段的字段。如果这个表格是本地表格 (Paradox 或 dBASE)，它必须具有一个索引以映射这些字段。
- ④ LookupResultField：Lookup 表格返回的字段名、返回的字段类型必须与 Lookup 字段类型相同。如果 Lookup 字段是字符串类型，必须确保这个字段的 Size 特性与返回字段的 Size 特性匹配，否则您将只看到空白。

后三个特性都是字符串，其中 Master 字段和 Detail 字段可以用分号隔开的字段列表。字段 Lookup 发生在字段 Calculation 之前，这样您可以把 Calculated 字段建立在 Lookup 字段的基础上，反之则不行。

1.6.3 菜单方面的改进

为了更好地符合开发工具菜单的标准，Delphi 2.0 的菜单与 Delphi 1.0 有所不同。主要区别是 Delphi 2.0 的 File 菜单作了重新设计，并且删掉了 Compile 和 Options 菜单，增加了 Project、Component 和 Database 菜单。以下是 Delphi 2.0 中新的菜单命令：

- ① File|New...：在当前工程中加入一个新的元件。
- ② File|Open...：打开文件。
- ③ File|Save As...：换名保存。
- ④ File|Use Unit：在单元的 Uses 部分加入一个要引用的单元。
- ⑤ Project|Add To Project：向当前工程中加入一个文件。
- ⑥ Project|Remove From Project：从当前工程中删除一个文件。
- ⑦ Component|Configure Palette：定制元件选项板。
- ⑧ Database|SQL Monitor：打开 SQL 监视器。
- ⑨ Database|Explore：运行 Database Explorer。
- ⑩ View|Threads：调试时显示所有的线程信息。

另外，File 菜单上显示先前打开的文件列表（最多 10 个）。其中 0~5 是最近打开的工

程文件,6~10 是最近打开的其它类型的文件。

Delphi 2.0 不再支持在 Delphi 1.0 中出现的下列菜单命令:

Tools|ReportSmith

Tools|BDE Config

Help|Interactive Tutors

如果要在代码编辑器里打开工程文件的文本(不是打开工程),则使用 File|Open 命令,在对话框中选择扩展名为 .DPR 的文件。

1.6.4 SQL 监视器

SQL 监视器允许用户看到通过 SQL Links 链接远程服务器,或通过 ODBC 链接 ODBC 数据源时的 SQL 语句调用。要打开 SQL 监视器,可以使用菜单上的 Database|SQL Monitor 命令。SQL 监视器如图 1.11 所示。

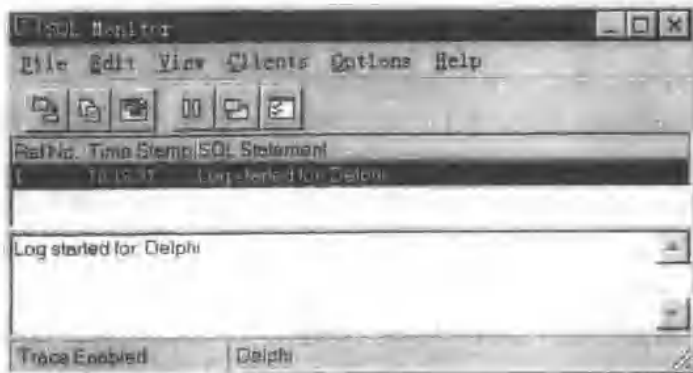


图 1.11 SQL 监视器

1.7 开发环境的新特性

1.7.1 可视化的窗体继承

我们知道, Borland 在面向对象领域一直处于世界领先的地位。在 Delphi 1.0 中,所有窗体(Forms)设计器设计的窗体都是从 TForm 类继承下来的。Delphi 2.0 作了重大改进,它允许您从其它窗体类继承,包括继承窗体上的元件、特性和代码,以作为自己窗体的起点。被继承的窗体可以是工程中或对象库中所有已存在的窗体。窗体的可继承性让您在一个应用程序内部或一组应用程序之间,建立一个标准的窗体平台库。当被继承的窗体发生变化时,继承的窗体马上也相应变化。您可以定制每一个从库中继承下来的窗体类型,同时保留修改标准窗体的能力,并且把变化马上反应到继承的窗体中来。

1.7.2 可视化的窗体链接

一个元件的特性可以被不在同一个窗体的元件使用,这种链接最常见的例子就是数据访问元件。例如可以在一个应用程序的窗体中放置一个 TTable 元件,然后在其它窗体

中,用不同的方法查看这个 TTable 元件所链接的数据库表的数据,例如用 TDBGrid 元件(栅格)或 TDBEdit(编辑框)等。Delphi 通过链接相应的单元链接窗体,例如有两个窗体分别是 Form1、Form2,相应的单元是 Unit1 和 Unit2,Form1 上的元件可以用到 Form2 的元件,只要在 Unit1 的 Uses 部分包含 Unit2。Delphi 2.0 允许用 File|Use Unit 这个菜单命令,在一个单元的 Uses 部分加入另一个单元。

1.7.3 新型编译信息

Delphi 2.0 的编译器在代码编辑器窗口的状态条上显示错误和警告信息,以及提示信息。您可以通过 Project|Options... 来改变有关选项,编译器也可以产生多行错误信息而不是停留在第一个错误的地方。当从编译器收到错误信息的时候,您可以在状态条的某项错误或警告信息上按下 F1 键,以得到该信息的解释,包括该错误是怎样引起的和怎样解决。

1.7.4 数据库浏览器

数据库浏览器(Database Explorer)允许您在程序开发阶段,保持同远程数据库服务器持续的链接,并且同 BDE 的别名和源数据对象协调工作。利用数据库浏览器,可以生成、查看和修改 BDE 别名和源数据对象,诸如表及预存储过程等。数据库浏览器能够显示所有的已合法定义的别名,选择其中一个可以看到该别名的定义。

要链接由一个别名指定的数据库,必须先选择一个别名,然后在菜单中选择 Object|Open,或从工具条上选择 Open 按钮。当链接好一个数据库以后,该别名左边的图标将变绿。

要扩展一个数据库,可以单击某个别名左边的“+”号,就可以列出所有的表。一旦链接好数据库,就可以对该数据库进行 SQL 操作。首先选择 Enter SQL 页,在语句区键入 SQL 语句,然后按下执行按钮,SQL 语句就被执行,并显示执行结果。数据库浏览器的界面和用法类似于 Windows 95 的资源管理器,如图 1.12 所示。我们估计它本身就是用 Delphi

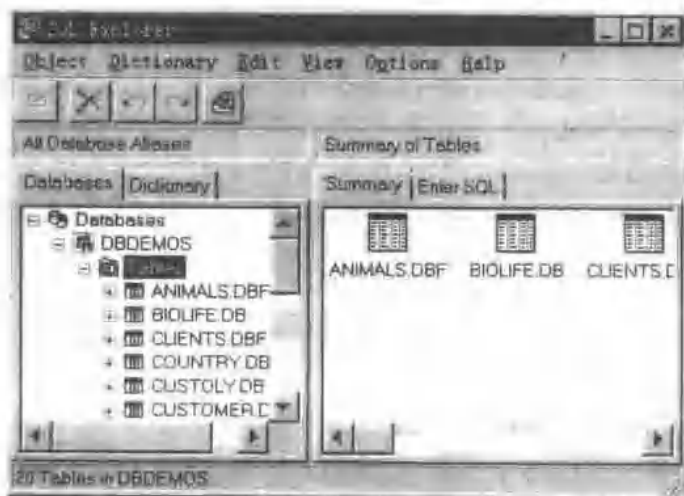


图 1.12 数据库浏览器

开发的,建议您把它作为开发应用程序的范例。

在 Delphi 2.0 的不同版本中,数据库浏览器有不同的名称。在台式机版本中不包含数据库浏览器,在客户机/服务器套件版本中,数据库浏览器称为 SQL Explorer。

1.7.5 字段编辑器的拖放操作

您可以对在一个窗体上的数据集(如 TTable 或 TQuery)使用字段编辑器。一旦某个数据集定义了字段,您就可以拖放一个字段名到包含该数据集的窗体上。如果您还没有把一个数据源元件加到窗体上,则 Delphi 自动将其加入到窗体上(当第一次拖放时)。Delphi 还能根据字段的类型自动生成相应的控制。例如,如果拖放的是 TStringField 类型,Delphi 就生成一个 TDBEdit 元件;如果拖放的是 TGraphField 类型,Delphi 就自动生成 TDBImage 元件。

1.7.6 从 Delphi 1.0 升级到 Delphi 2.0

Delphi 2.0 是 Delphi 1.0 版的升级产品。要把 16 位的程序变成 32 位的程序,只要把原来的代码在 Delphi 2.0 中重新编译一遍。不过某些代码需要适应 Delphi 2.0 的新数据类型和新的 Windows 特征,另外,Delphi 1.0 中使用的 VBX 控制在 Delphi 2.0 中已不再使用。如果 Delphi 1.0 程序中包括这些控制,在 Delphi 2.0 中编译将出错,这些控制是 TBiSwitch、TBiGauge、TBiPict、TChart。除了这些控制,原来的单元和窗体都可以继续在 32 位的 Delphi 2.0 中使用。

对同一个单元,不可以在 Delphi 1.0 和 Delphi 2.0 中都使用条件编译。例如不可以在 Delphi 1.0 中使用 TTabbedNotebook 元件;不可以在 Delphi 2.0 中使用 TPageControl 元件,而在单元中使用编译选项,如 \$IFDEF;也不可以在 Delphi 1.0 和 Delphi 2.0 间共享工程文件,因为相应的资源文件不同,一个是 Windows 3.1 的 16 位格式,一个是 Windows 95 的 32 位格式。当然可以采用这个办法,即在 Delphi 2.0 里生成一个新的工程文件,然后把老的单元和窗体加入到工程中去。

Delphi 2.0 还把 WinTypes 和 WinProcs 这两个单元,合并成一个称为 Windows 的单元。为了保持最大程度的兼容,Delphi 2.0 仍然允许使用这两个单元名。

虽然可以在 Delphi 2.0 里打开 Delphi 1.0 的工程文件,但是当重新编译或保存变化时,原有的资源文件将不能使用。也就是说,资源文件不能做到向后兼容。

为了处理 32 位的返回值,BlockRead 和 BlockWrite 这两个过程的最后一个可选参数应为 Integer 型,而 Delphi 1.0 中这个参数是 Word 类型。另外,Count 参数现在也是 Integer 类型。

1.8 对象库

Delphi 2.0 提供了一个对象库,以作为共享和重用窗体和工程的手段,对象库本身只是一个含有查找窗体、工程和专家线索的文本文件。Delphi 2.0 中的对象库替代了 Delphi 1.0 中的 Gallery。

对象库可用于工程间共享。只要把窗体、对话框和数据模块加入到对象库中,其它工

程就可以使用了。例如,所有的应用程序都要用到 About,只要把其中一个拷贝到对象库,其它工程就都可以使用这个 About 了。对象库的更高级的用法是,设计一个空的标准对话框,对话框内有公司和产品标志,以及标准的按钮放置,以后所有的工程中就可以继承这个对话框。

对象库也可以在一个工程内共享,Delphi 2.0 允许继承工程中已有窗体,当通过 File | New... 打开 New Items 对话框时,您将看到多页对话框上有一页是您的工程名。翻到这一页,将看到工程中所有窗体、对话框和数据模块,您可以从其中选择一项继承成新的项,并根据需要定制它。例如一个数据库应用程序可能有几个窗体需要显示相同的数据,但命令按钮不同。为了避免生成几个几乎差不多的窗体,您可以生成一个通用的窗体,包括用于显示数据的控制,然后在每一个窗体中继承下来并生成各自的命令按钮。通过这种方法可以大大缩短开发进程。

整个工程也可以共享,例如把整个工程加入到对象库中作为以后工程的平台。如果几个应用程序差不多,就可以把程序建立在一个标准程序的基础上。

还可以通过“应用专家”来帮助开发程序,对象库里包含了应用专家的内容。应用专家实际上就是一些小程序,引导您通过回答几个对话框,就可以快速生成应用程序。Delphi 2.0 提供了几个应用专家,用户也可以加入自己的应用专家。

第二章 Delphi 编程环境及高级用户接口

在这一章里,读者将了解如何使用构成 Delphi 环境的菜单与窗口。如果读者对于 Microsoft Windows 应用软件的形式还没有完全适应,那么这一章会提供一些帮助。总之,Delphi 本身就是一个设计完美的 Windows 应用软件,其元件的响应方式是典型的 Windows 应用软件的方式。有经验的 Windows 用户可以粗略地浏览这部分。但是请记住,在没有熟悉 Windows 应用软件的形式与使用之前,是不可能充分发挥 Delphi 的优势的。在读者阅读完这一章之后,就应该对联机帮助、编辑工具,以及 Delphi 内部的文件处理工具比较适应了。

如果读者是用 Delphi 为自己开发一个应用软件,那么倒是不必非要遵守 Windows 的标准。但是,如果其他的人要使用这一应用软件,那么遵守 Windows 的标准必然会减少学习使用这一软件过程中的波折。例如,Windows 的用户都希望用鼠标的单击来选择一个项,用双击来激活该项,那么设计者就应当保证自己设计的软件符合这一标准(关于 Windows 的应用软件最为完整的准则,可以参阅 Microsoft 的出版物:《The Windows Interface: An Application Design Guide (Windows 界面:应用软件设计指导)》(部分号 288921 或 ISBN 1-55615-439-9))。

读者在前面所看到的窗口缺省为一定的形状和大小,但是这也可以根据用户自己的需要来改变。类似地,菜单也会按用户定制的方式去响应。虽然 Delphi 并不要求设计者必须用这些缺省值,但是随意地改变并不一定更好。

在 Help 菜单中可以看到 Interactive Tutors(交互式教程),它是对本章提供的一些材料的很好补充。

Delphi 的强大功能,包括其丰富的工具和详细的菜单,都很容易使初学者不知所措。为了帮助初学者减少混乱的感觉,本章将详细描述 Delphi 编程环境所提供的內容。

2.1 主屏幕的综览

在启动 Delphi 时,读者首先将看到一个指明 Delphi 版本号的画面。在短暂地停顿之后,就自动进入了 Delphi 环境中,这在图 2.1 中可以看出。读者可以看到标准 Delphi 环境的四个部分:菜单条、对象观察器、窗体窗口以及代码编辑器窗口(代码编辑器窗口被部分掩盖了)。其它提供更专门功能的窗口(诸如用来调试的窗口)在需要使用它们时也会出现。

Delphi 会记住设计者在上次是如何布置屏幕的,并在以后重复使用这一布置。因此,读者所看到的屏幕可能与图 2.1 中的不一样。

下面对使用最普遍的主屏中的部分进行说明。

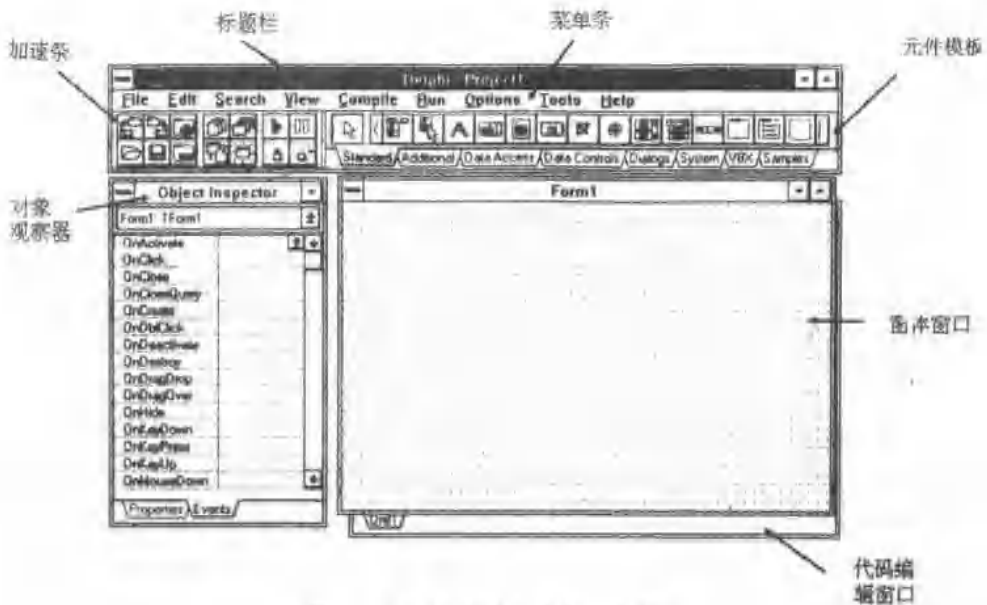


图 2.1 Delphi 初始状态的屏幕显示

2.1.1 标题栏

标题栏是水平方向的一个横条，在屏幕的上方，它告诉操作者应用软件的名称，这在所有的 Microsoft Windows 中都是一样的。标题栏与用户之间的交互式操作是由 Windows 而不是由应用软件来处理的。在 Windows 应用软件中，标题和菜单条下的第一项都称为客户区。读者设计的应用软件决定着放在客户区中的对象的外形、感观以及响应方式。

在 Delphi 中，当设计者使用默认的名称 Project1 来命名自己的设计时，标题栏即显示 Delphi-Project1，这是 Microsoft Windows 应用软件的典型情况。如在高级的程序（如 Delphi）中有多个状态，则标题栏不断改变以指示不同的状态。例如，当在 Delphi 环境中运行这个工程时，标题栏就会切换为 Delphi-Project1[Running]；而当设计者调试并暂时中止工程设计时（即更正工程设计代码中的错误时），标题栏就显示为 Delphi-Project1[Stopped]。

2.1.2 菜单条

从菜单中列出的下拉菜单里选项选项，是使用 Windows 应用软件功能的最普遍的一种方法。对 Delphi 也是这样。在 Delphi 中，菜单为设计者提供了开发、测试以及保存应用软件所需的工具。File(文件)菜单含有操作将编译成应用软件的文件的命令。Edit(编辑)菜单则包含有使代码编写更为容易的编辑工具，它们可以激活设计者的工程所用的界面。这里包括调整和布置工具。Search(搜索)菜单中有搜索-替换编辑工具以及代码定位工具。

View(查看)菜单使操作者可以快速地访问设计工程的不同部分，并且快速访问组成 Delphi 环境的不同窗口。Compile(编译)菜单含有编译和语法检查工具。Run(运行)菜单使设计者可以在开发自己的应用软件时对其进行测试，而且可以访问用来纠正(调试)问题的工具。Options(选项)菜单使操作者能够控制 Delphi 的环境(第二十三章将详细讨论调试技术)。Tools(工具)菜单含有外部的工具 ReportSmith 和 Bitmap 编辑器。最后，Help(帮助)菜单可以用来访问非常详尽的 Delphi 帮助系统。