

万水 Delphi 技术丛书

Delphi 5.0

开发多媒体应用



周 铭 邹 辉 王凌云 等编著



中国水利水电出版社
www.waterpub.com.cn

484

TP311.56

Z756

万水 Delphi 技术丛书

Delphi 5.0 开发多媒体应用

周铭 邹辉 王凌云 编著

中国水利水电出版社

内 容 提 要

近年来,得到了长足发展的多媒体技术,是计算机技术的一个重要研究领域。这一领域横跨了包括计算机、远程通信、出版、家用电子、音像产品和广播电影电视等在内的多个工业领域。每一位关心计算机的人几乎都会接触到各种各样的与多媒体相关的东西:多媒体计算机、多媒体出版物、多媒体游戏、网络多媒体……。那么如何进行多媒体应用程序的开发呢?能否成为一名成功的程序员,其开发工具是至关重要的。Delphi 5.0 是 Inprise 公司开发的全新的可视化 RAD(快速应用开发)编程环境。它是基于面向对象的 Pascal 语言。虽然它不是专门为多媒体应用开发的工具,但它可以完成几乎所有涉及多媒体的应用。

本书通过大量的实例,阐述了 Delphi 5.0 的新特点、编程思想,介绍了 Delphi 在多媒体领域各个方面的应用。本书尽量以具体的示例进行分析说明,内容丰富,条理清楚,具有很强的实用性。

图书在版编目(CIP)数据

Delphi 5.0 开发多媒体应用/周铭等编著. —北京:中国水利水电出版社, 2000.10

(万水 Delphi 技术丛书)

ISBN 7-5084-0470-X

I. D… II. 周… III. Delphi 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 52080 号

书 名	Delphi 5.0 开发多媒体应用
作 者	周铭 邹辉 王凌云 编著
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: sale@waterpub.com.cn 电话: (010) 63202266(总机)、68331835(发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 23.25 印张 526 千字
版 次	2000 年 10 月第一版 2000 年 10 月北京第一次印刷
印 数	0001—5000 册
定 价	35.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换
版权所有·侵权必究

前 言

自从微软推出 Windows，特别是 95 年 8 月 Windows 95 发布以来，PC 机上的应用程序都开始向 Windows 平台转移。无论是计算机爱好者还是专业的程序员，学习 Windows 编程是很重要且有意义的。

但是传统的程序开发工具在开发 Windows 应用程序时，需要很多的技巧和工作量，即便是 Visual C++ 和 Borland C++。因此快速应用程序开发工具（RAD）应运而生并飞速发展，如微软的 Visual Basic、PowerSoft 的 Power Builder、Inprise 的 C++ Builder 等便接近口语化语言，拖放式控件编程、大量的模板以及可重用技术和精心设计的数据库接口，使得编程难度大大简化了。而 Delphi 可以说是 RAD 工具中的佼佼者。Inprise 公司推出的 Delphi 是一个全新的可视化编程工具，使用了 Microsoft 的 Windows GUI 的许多先进特性和设计思想，并采用了先进的数据库技术和目前最快的编译器，同时继承了 Pascal 语言的代码结构清晰、可读性好和效率高等优点，一经推出，便受到广泛的好评。

本书以大量实例为基础，系统的介绍了 Delphi 在多媒体方面的应用。全书共分 8 章，第 1 章介绍了 Delphi 5 的新特性及其集成开发环境（IDE）。第 2 章介绍多媒体的原理。第 3 章通过一些实例，分析了 Delphi 中有关多媒体的各个组件的属性、方法和事件。第 4 章介绍组件的定制开发过程，这是因为本书中多次用到自定义的组件。第 5 章是有关图形、图像的处理，以及视频的播放。第 6 章介绍如何使用 Delphi 开发屏幕保护程序。第 7 章介绍了各种音频的播放，如播放 CD、播放 wav 文件和 midi 文件，并且介绍了两个组件，分别用来控制 MP3 的播放和语音比较。第 8 章介绍有关 DirectX 的有关知识，包括 DirectSound、Direct3D、DirectDraw 以及 DirectX 游戏。

书中示例中所涉及的人名、电话号码、通信地址和邮件地址等均为虚构，如有雷同，纯属巧合。

本书由方可燕策划，周铭、邹辉、王凌云编写，同时参与编写还有王长春、侯志东、王东辉、韦思亮、周兆确、叶青、邓增涛、谢威、方可军、毛诗华、方海英、唐家才、段小华、许刚、唐健、王晓兰、陈斌、杨琳等。董朝旭、向军、何晓威、王宇、李亨、张双民等提供了许多有益的参考资料和参考意见。

由于作者水平有限，且编写时间仓促，书中难免有疏漏和错误，恳请广大读者提出宝贵意见。

第 1 章 Delphi 5.0 的新特性和集成开发环境

许多 Delphi 的开发都希望 Inprise 公司新版本的 Delphi 5.0 在稳定的基础上, 会比 Delphi 4 有着很大的改进。Inprise 公司的开发者们十分清楚 Delphi 4 的不足, 并且非常认真地做了修改, 以确保这些问题不会出现在新版本中。到目前为止, Delphi 4 发展如此迅速, 所以如果希望在 Delphi 5.0 发布之前, 了解 Delphi 4 中的所有错误是不太可能的。Inprise 公司称, Delphi 5.0 更注重产品的稳定性, 而不是单纯地、无限制地添加新的功能。事实上, Delphi 5.0 原来预定在六月份发布, 多出一个多月的时间进行完善和测试。

当然, 和其他同一个系列的产品一样, Delphi 5.0 是其先前产品的超集。Delphi 5.0 具有 Delphi 4 及其更早版本所提供的所有功能。有关这些信息, 在 Delphi 4 中已经提供, 可以查看 1998 年 8 月的 Developers Review Issue 5 (也可以在 Inprise 站点上的 Review Online 中找到相关信息)。这些资料着重介绍 Delphi 5.0 的新特性。这里提供的信息依赖于预发布的软件, 所以在软件正式发布后, 可能会有一些改变。

1.1 Delphi 5.0 的新特性

安装了新版本的 Delphi 之后会发现, 新版本有了许多改进, 在本节中介绍其中一些主要的特点。

1.1.1 对象观察器 (Object Inspector)

最主要的区别在于对象观察器 (Object Inspector)。只需要单击白色的 “+”, 就可以展开可以展开的属性, 而不需要双击属性名。更重要的是, 属性具有明显的可视化的标志加以指示。例如鼠标属性和颜色属性, 如图 1.1 所示。另外一个改进是属性的概念和事件的分类。对于大部分组件来说, 在 Delphi 4.0 中组件集意味着对象观察器中会列举出许多的属性和事件, 其中的相当一部分, 普通的开发者是不会用到的。现在, 属性和事件分类了, 用户可以从所选的类别中选择显示或隐藏属性。也可以使对象观察器仅列出各个类别的可扩展的顶级入口。这样, 用户就可以不用显示出那些看似重要而实际上用不到的属性。对象观察器的状态栏会显示出当前隐藏的属性数目, 如图 1.2 所示。当然, 如果需要, 可以使分类显示无效。所有这些为使用定制属性编辑器创建组件提供了资源更加丰富的开发环境, 用户可以决定属性以怎样的方式显示出来才最有意义。

当需要进行调试时, 用户可以从正常的桌面切换到调试桌面。事实上, 在用户设置的所有桌面之外 (这些设置以 .DST 文件存储在 BIN 目录下), 可以指定自己的调试桌面。以后当开始调试一个工程时, 加载这个桌面设置。

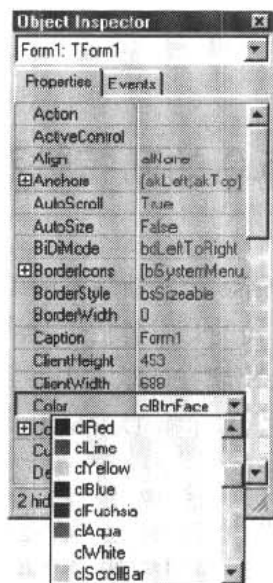


图 1.1 属性的可视化指示更加清楚

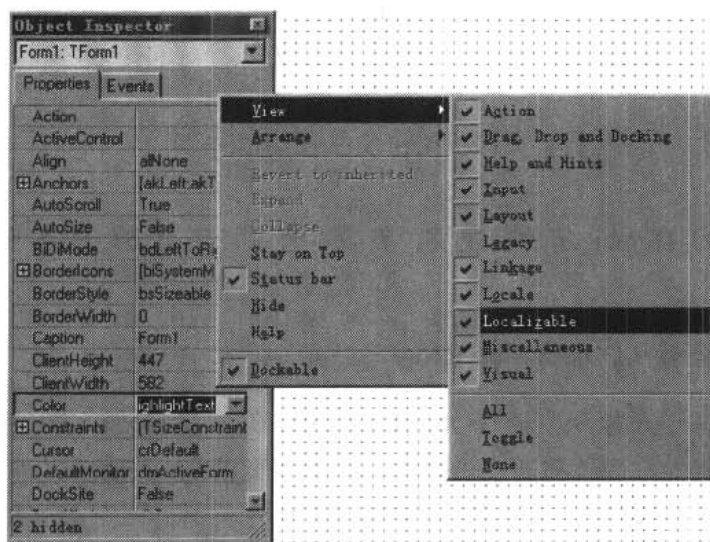


图 1.2 对象观察器的改进

1.1.2 窗体和窗体编辑器

为了使得版本控制系统更容易处理窗体，窗体编辑器将窗体以文本的格式存储。换言之，.DFM 文件默认的存储格式为文本格式。如果想存储为二进制格式。点击鼠标右键，取消 Text DFM 选项。在“环境选项（Environment Options）...”中的参数选择页中有一个文本复选框，取消这个选项，则.DFM 文件默认的存储格式将被设置为二进制格式。另外，VCL 的支持功能得到扩展，以支持二进制格式的.DFM 文件和其文本格式文件之间的转换。另一个与窗体相关的改进是相同的对话框中多出了一个选项：“自动创建窗体”选

项默认选中。这与 Delphi 的早期版本相匹配（以前的版本中，新窗体都是默认地被自动创建）。如果该选项没有被选中，则任何新窗体都无法自动创建。这意味着用户必须管理相关的窗体变量，如果程序要求访问一个窗体，这个窗体需要自动创建，但是这个选项没有被选中，用户可以使用工程选项对话框中的 Form 页进行切换。

还需要提到的 Delphi 5.0 的集成开发环境（IDE）的变化在其他的 IDE 中是很普通的，并且已经过时了：需要输入目录的对话框提供给用户一个选项，来浏览相关的目录，使得用户不用再输入一长串的路径。

1.1.3 桌面

Delphi 5.0 的一个重要的新特征是整体桌面的概念（相对于工程桌面）。一旦按照自己的方式布置了 IDE 窗口，用户可以通过视图（View）菜单中的一个菜单项将它存储为一个新的桌面。存储在桌面中的信息包括：哪些窗口是可视的；在什么位置；在哪儿可以“靠岸”（dock）。同时，对象观察器中选中的页、有效的属性以及加速条（SpeedBar）的所有状态都记录了下来。如果要载入一个特定的桌面设置，可以在视图菜单中选择。更好的方法是使用新桌面加速条中的 combobox。加速条在 Delphi 的主窗口中，如图 1.3 所示。当用户需要进行调试时，可以从正常的工作桌面切换到调试桌面，停靠（docked）在最大化编辑器边的是标准的调试窗口。事实上，在用户安装的所有桌面中（以.DSF 文件保存在 BIN 目录下），可以选择其中一个作为桌面，在启动调试工程时载入它。

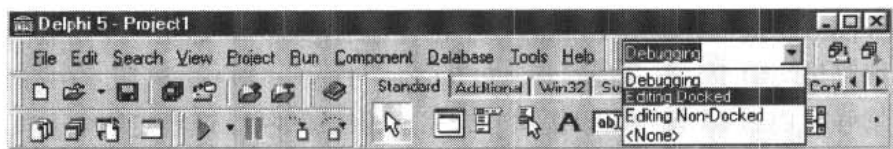


图 1.3 可定制的桌面

1.1.4 工程管理器和工程浏览器（Project Browser）

工程管理器和以前的版本相比较稍有改变。当对工程组操作时，用户可以使用工程管理器顶部的下拉选择器在各个工程之间进行切换。还可以在工程之间拷贝和粘贴文件。此外在 Windows 浏览器中，文件可以被拖放到工程中。

资源描述文件（.RC 文件）能够被添加到一个工程中，并且将在 IDE 内部编译为.RES 文件，而不用象以前那样需要外部编译。

旧的对象观察器（从 Delphi 1 起就未做改动）已经做了改进。如图 1.4 所示的一个工程示例。如果这个工程没有编译，工程浏览器会在它调用时自动进行编译。可以选择浏览器仅对工程内的符号有效，或者对整个 VCL 有效。双击新浏览器中的符号打开一个新的带有标签（tab）窗口。如果双击一个类，可以得到该类的 Symbol Explorer，其中包括 scope、inheritancy 和 references 三个不同的页。然而当双击一个类成员时，只出现一个 references 页。图 1.4 显示出可以通过全局符号以及单元来浏览，就像通过浏览类一样。

设置属性，完成事件句柄。当框架放置到窗体中，单个的组件可以定制和移动等等，但是不能被删除。当把一个框架放到窗体中，就创建了框架类的一个实例。这意味着任何对于框架的改动，都会直接反映到窗体上。此外，和窗体继承一样，框架实例的任何属性的改变将不影响其父类的属性。通过在窗体中创建事件句柄可以扩充框架组件的功能。这些事件句柄默认调用框架类中的相关方法。可以添加代码，也可以用自己的代码替代这些调用。图 1.5 显示了一个浏览框架，这是一个开发好的框架，可以在图像浏览器窗体中使用。框架中的文件列表由它的 OnChange 事件的事件句柄控制。

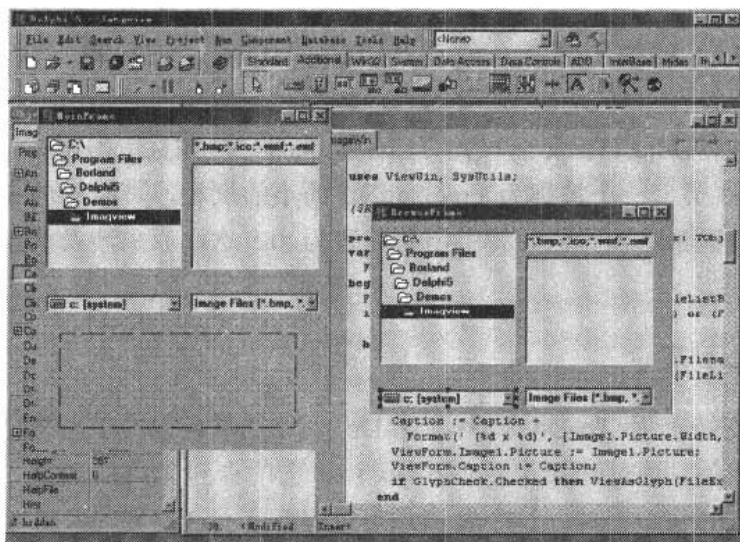


图 1.5 嵌入到窗体中的框架

1.1.7 语言与 VCL

作为一个新产品，Delphi 5.0 在语言上没有多大的改变。唯一要提到的一点是系统必须具有锁机制指令，以确保在读引用字符串和动态数组，或将其作为参数的时候，操作是安全的。与此相对应，可视组件库更新为版本 5（编译器本身的版本是 13）。需要指出的是，主窗体在最小化或恢复时毫无生气的问题得以解决了。由于 VCL 应用程序的内部配置，在 Delphi 2、3 和 4 中当主窗体最小化时，其他所有窗体都隐藏了。事实上，任务栏图标是应用程序对象窗口的最小化。这意味着，用户无法看到主窗体最小化为工具栏的过程。现在这个困扰用户很长时间的问题得以解决。应用程序对象的事件可以使用新的 TApplicationEvents 组件进行自动安装。所有的应用程序对象的事件都有这个特性，所以用户可以使用对象观察器来完成基本事件句柄框架。每一个窗体都有一个 ApplicationEvents 组件，当触发应用程序的相关事件，每个 ApplicationEvents 相应的事件按顺序执行。

用户可以通过调用激活的方法，决定一个特定的 ApplicationEvents 在其他事件之前执行其事件。事实上是一个窗体的 OnActivate 事件句柄将调用窗体的 ApplicationEvents Activate 方法，使其成为第一个执行者。使用 CancelDispatch 方法可以使更多的 ApplicationEvents 组件停止执行它们的事件句柄。TAppletApplication 类使得用户可以按照自己的意

愿在创建控制面板工程时（.CPL 文件）使用任意多的控制面板程序。可以从 File|New...对话框访问这些类，还可以从一个或多个.CPL 文件向控件面板中添加多个额外的选项。在 Delphi 5.0 中，有许多工具可用来在 TToolBar 中定制绘图，并且在 THeader-Control 中记录标题。

在编写 web 应用程序时，web 操作具有一个 Producer 属性，所以在这些操作执行时，自动修改响应消息的内容。

在新的 Contnrs 单元中，有一系列有用的容器类。TObjectList、TComponentList 以及 TClassList 是由标准的 TList 继承而来，只是扩充了一些方法和属性。从根本上说，这是一些安全的对 TList 的类型封装。放置在 TObjectList 上的对象隶属于 TObjectList，TComponentList 能够在组件上获得，可以自己删除和修改。

TStack 类也象 TObjectStack 一样由 TList 继承而来，并且衍生出 TQueue 类，进而衍生出 TObjectQueue 类。一个被称为 FreeAndNil 的小的例程，其唯一的参数是一个对象的引用变量。这个例程删除该对象并将引用参数置为 0。这样原来涉及两条语句，现在减少为一个过程调用。

1.1.8 To-Do 列表

Delphi 现在支持工程 To-Do 列表。To-Do 列表选项可以在工程范围之内，也可以和个人的资源文件相关。通过选择 View|To-Do List 选项看到 To-Do 列表，此时会弹出一个菜单，和大多数 IDE 窗口一样，这个窗口是可以停靠（docked）到编辑器，如图 1.6 所示。

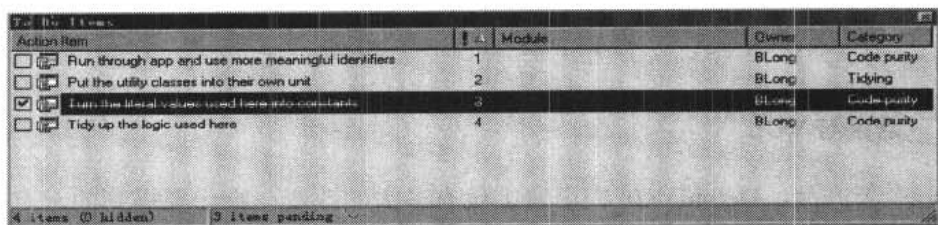


图 1.6 To-Do 列表

To-Do 列表的菜单项可以直接添加到这个窗口中，并且被保存为后缀为.TODO、文件名与工程文件相同的文件。也可以右击鼠标在下拉菜单中选择 Add To-Do 选项（Shift+Ctrl+T）来添加列表选项。此时会弹出一个对话框，并且在用户的源程序中添加特定的注释，说明该选项是将要执行，还是已经完成。同时还指明优先级（1 到 5 五个优先级）、所有者、种类以及该项的文本，例如：

```
{ TODO 1 -oBLong -cStop-ship :Fix the Access Violation problem in this
routine }
```

这说明一个优先级最高（为 1）的 To-Do 选项由 BLong 登录到 Stop-ship 类别，与一个 Access violation 相关。一旦用户在源程序中输入某一个 To-Do 列表项的注释，则直接添加到 To-Do 列表中。当完成一个 To-Do 列表项，该表项的状态会变成高亮的。此外，如果一个 To-Do 表项在当前工程的一个文件中，并且此文件没有被打开，该列表项是灰

色的。双击窗口中的源代码 To-Do 表项, 会显示相应源代码中的注释。基于工程的 To-Do 列表和基于源代码的 To-Do 列表是不同的, 它们在列表窗口中使用不同的位图, 如图 1.6 所示。通过单击 column 或使用右键快捷菜单, 可以按照列对 To-Do 列表进行排序。用户可以使用标准文本格式或者 HTML 表格格式将 To-Do 列表复制到剪贴板上。使用哪种格式由用户自己选择, 如图 1.7 所示。

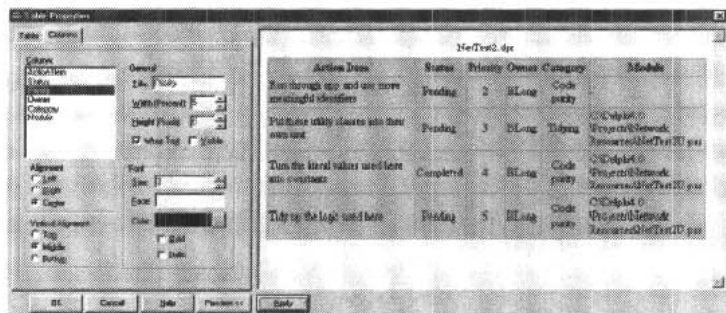


图 1.7 导出到 HTML 的 To-Do 列表

1.1.9 调试

在 Delphi 5.0 中, 不但可以通过 CPU 窗口查看程序的机器代码、堆栈、内存以及 CPU 寄存器等, 还有一个 FPU 窗口, 用来显示浮点单元的状态及信息或 MMX 信息。在调试状态下, IDE 支持一些拖放操作, 用户可以将任何表达式从编辑器中拖到监视列表中, 这将对语句进行跟踪。还可以将表达式拖到观察器中方便地查看。另外将表达式拖放到 CPU 窗口的堆栈中, 可以查看该表达式的地址。

运行 (Run) 菜单增加了一个选项, 叫做 Run Until Return (Shift+F8), 它执行子程序, 并且在返回到调用者时停止执行。这在原本想跳过, 却不小心进入了一个很长例程时是很有用的。运行菜单的另外一个选项是 Attach to Process..., 这使得用户可以附加一个外部的运行的应用程序并进行调试。

Evaluate/Modify 对话框增加了新的 Speedbutton, 使得用户可以对当前的表达式求值, 将其添加到监视窗口, 或者加载到观察器中。另外一个与监视有关的改进是监视窗口可以为高亮的表达式启动一个调试观察器。在使用这个窗口来跟踪加载到当前进程地址空间的模块时, 如果选中一个模块, 则它的入口点会被列出, 或者按名字顺序, 或者按地址顺序。新版本的 Delphi 中, 断点属性对话框 (如图 1.8 所示) 具有了更多的选项, 用户可以设置在断点处不再中断程序。

现在, 断点可以分成组, 其中一个断点的操作可以打开或关闭同组中的其他断点。用鼠标右键单击编辑器左边代表断点的红色圆, 可以得到断点的属性。如果将鼠标指针放置在圆上, 几秒钟后会出现断点的主要属性, 如图 1.9 所示。在调试器选项对话框中, 新增了一些复选框, 使得用户可以调试任意一个由应用程序启动的进程, 同时在监视窗口中得到表达式的值, 并禁止多个求值程序的自动使用。在调试多进程应用程序 (有可能是多种语言写成) 时, 默认情况下 Delphi 会根据当前调试器跟踪的语句, 采用合适的表达式求

值程序。在异常列表中还有额外的入口, Delphi 中是可以忽略的。异常可能来自微软的 ADO 库, 也可能是 CORBA 系统的内部 VisiBroker 异常或用户异常。对于多个进程的应用程序的调试, 可以在线程窗口中的某一个进程上, 单击鼠标右键, 然后在快捷菜单中设置临时调试选项。

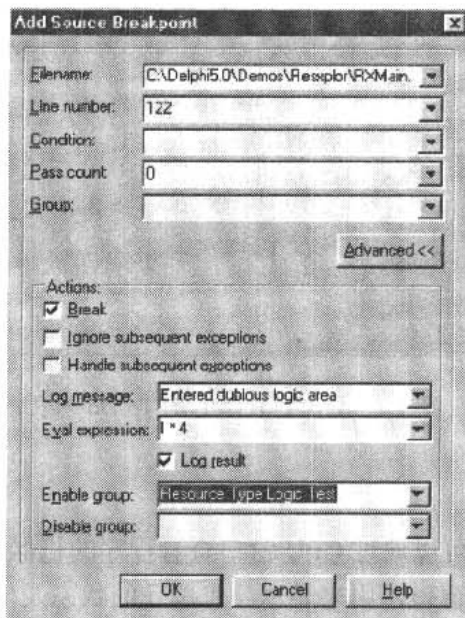


图 1.8 跨过断点的设置

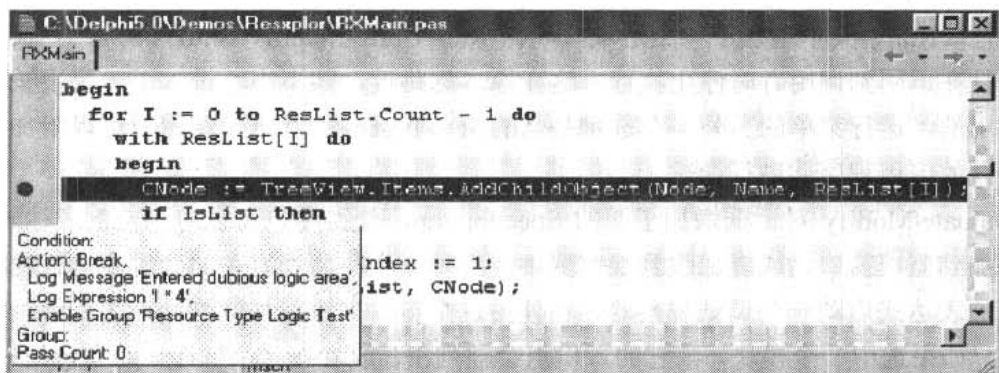


图 1.9 查看断点详细信息

1.1.10 COM/ActiveX

从 File|New... 的 ActiveX 页, 可以创建 Active Server 对象。这是一个自动控制的对象, 可以通过 IIS Web 服务器的 ASP 访问。当用户请求一个 Active Server 对象时, 向导会提供一个是否要生成一个简单的 ASP 测试脚本。当在 Delphi 4 中使用类库支持在应用程序创建一个新的 COM 对象时, 会有向导帮助用户创建。然而在缺省的状态下并不检查产生

的类库 OLE 自动控制标志。如果这个标志没有设置, COM 编程者必须编写所有的调度代码, 控制数据流入和流出 COM 服务器, 这是一件很烦琐的事。Delphi 4 的用户会因为客户端应用程序无法与 COM 对象接口连接而得到错误信息: Interface not supported。如果搞不清错误的原因, 这些用户也许会放弃或是使用其他的自动控制对象。Delphi 5.0 的 COM 对象向导解决了这个问题。同时默认提供类库支持, 而不需要用户的检验。

另外一个改进是在开发 COM/Automation 对象时, 类库编辑器会刷新实现代码。会有一个更新对话框告诉用户系统将要做了些什么。这使得用户可以决定是否进行这些改动, 如图 1.10 所示。在 Delphi 5.0 中导入类库时, 可以将 COM 服务器作为组件安装到组件面板中。服务器的所有合法属性在运行时都可获得, 尽管在对象观察器中没有列出。Automation 服务器的组件化对于 Delphi 4.0 中有限的事件支持来说是一个很好的改进。

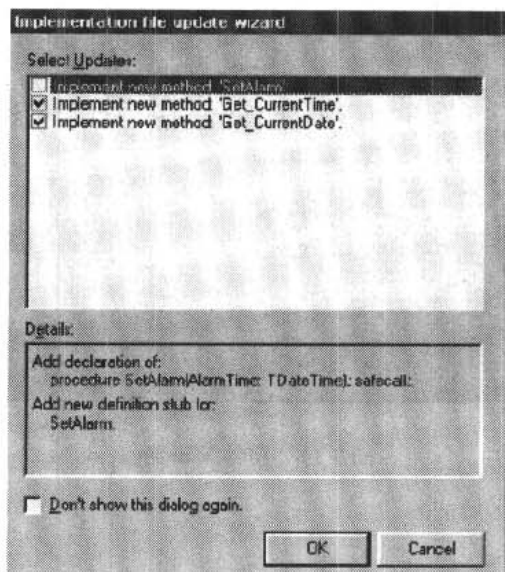


图 1.10 类库编辑器

许多 Microsoft Office 服务器已经安装到组件面板的服务器页, 所以在 Delphi 5.0 中, 可以很容易创建 Office Automation。Visual Basic 可以使用 vtables 来创建 Active 服务器和 Active 控件, 这在 Delphi 3.0 和 Delphi 4.0 中都会产生问题, 但是在 Delphi 5.0 中得到解决。一旦检查到 vtable, 类库导入程序将在导入单元的接口定义部分添加伪入口。

1.2 Delphi 5.0 集成开发环境

Delphi 是专门为 Windows 95/NT 设计的应用程序开发工具。任何一台能够运行 Windows 95/NT 的计算机都可以运行 Delphi。完全安装 Delphi 5.0 并生成 Borland 根目录共有 179MB 的内容, 加上临时文件占用空间, 大约需要 275MB 左右的硬盘空间。此外, 无论编写多么小的应用程序, 生成的临时文件都将占用几十 KB 的硬盘空间。

要使用 Delphi 5.0 开发应用程序, 首先必须在计算机上安装 Delphi 5.0。可以使用安

装程序安装 Delphi 5.0, 而且安装程序还可以根据实际情况对安装进行裁减。

1.2.1 窗体编辑器

Delphi 5.0 的窗体编辑器, 如图 1.11 所示为创建窗体提供了方便可靠的方法, 包括可视的和非可视的组件。在向一个工程中增加新的窗体或使用对象库创建窗体时使用窗体编辑器。窗体是 Delphi 系统中最基本的组件, 是与用户交互最多的部分。在大多数 Delphi 应用程序中, 窗体都用作其他组件的容器。窗体具有自己的一些属性, 如标题、系统菜单、最小化和最大化按钮、可变大小的边界等, 还可以具有工具条、菜单、状态条等控件。

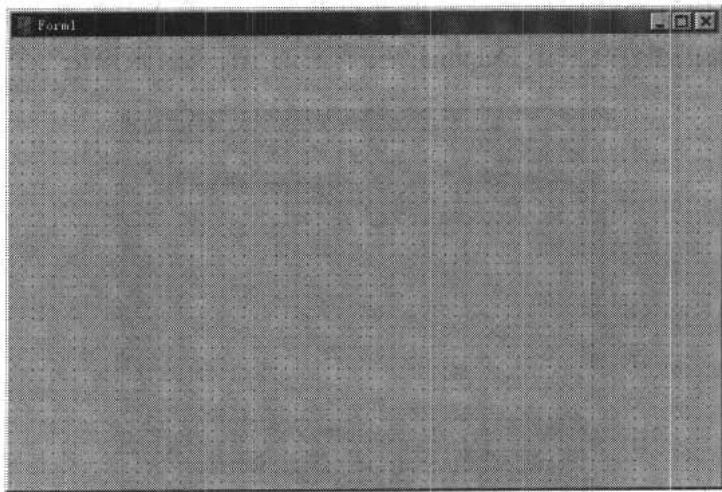


图 1.11 窗体编辑器

1.2.2 代码编辑器

Delphi 提供了方便可靠的方法来检查和修改源代码。Delphi 在以下几种情况使用代码编辑器, 如图 1.12 所示。

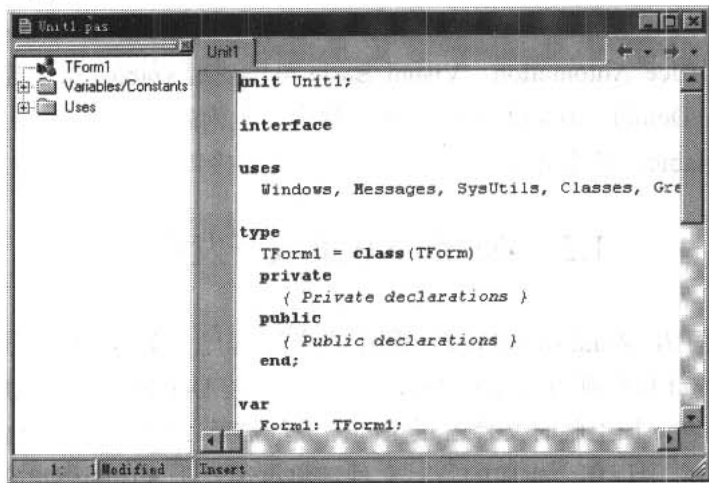


图 1.12 代码编辑器

- 创建新的工程
- 向一个工程中添加一个窗体或组件
- 打开文件时，即使并不将它添加到工程文件中也可以使用代码编辑器打开文本文件查看和修改，其操作步骤包括：

1. 打开代码编辑器页

在向工程文件中添加新的窗体、组件或其他文件时，需要激活代码编辑器页。打开一个代码编辑器窗口，可进行如下操作：

- 打开一个文件
- 选择 View/New Edit Window
- New Window 命令打开当前代码编辑器中页的拷贝

2. 选择代码编辑器页

在代码编辑器窗口中，单击要查阅或修改页相关的标签。

在一个代码编辑器窗口中可以打开多个文件。每个文件打开一个新的代码编辑器页，每一个页由该窗口上方的一个标签所表示。例如，当你打开一个工程，它成为该窗口中的第一个标签。而其他打开的文件，例如组件文件，依次成为窗口中其他的标签。

3. 在代码编辑其中修改文本

- 将光标定位在需要修改的地方
- 输入新的文本，并键入回车键

4. 插入代码模板

- 按下<ctrl>-j，出现模板的弹出列表

5. 添加组件

当向一个窗体中添加组件时，Delphi 产生该组件的一个实例变量或是域，并将其加入到窗体的类型定义中。

使用以下方法可以查看添加到代码编辑器中的代码。

- 拖拉窗体标题栏直到可以看到完全看到代码编辑器
- 在监视代码编辑器有何事件发生时，添加组件到窗体中

注意：不要对任何 Delphi 生成的代码进行编辑。

6. 代码编辑器帮助

几乎在代码编辑器中的任何位置都可以获得上下文相关帮助。上下文由光标的当前位置决定。将光标放到希望获得帮助的代码处，按下 F1 键。

7. 查看代码编辑器窗口中的页

当代码编辑器中显示了一页，可以查看它所包含的所有数据而不仅仅是可以查看自己的代码，可通过下列方式查看：

- 如果代码编辑器就是活动窗口，单击与要查看的页相关的标签
- 选择 View/Units
- 从工程管理器 SpeedMenu 中选择 View Unit

8. 关闭一个代码编辑器页

- 在代码编辑器窗口中单击鼠标右键，选择 Close Page
- 选择 File/Close

代码编辑器窗口包含一个或多个代码编辑器页、一个装订线以及一个消息窗口。代码编辑器窗口不能是空的，一旦关闭了最后一个页，代码编辑器窗口会被关闭。

9. 关闭所有代码编辑器页和工程

- 选中需要关闭的代码编辑器页，选择 File/Close
- 单击关闭按钮

如果已经修改了代码而没有保存，Delphi 弹出 Save File As 对话框。如果修改了工程而没有保存，Delphi 弹出 Save File 对话框。

1.2.3 菜单

File 菜单

使用 File 菜单（如图 1.13 所示）打开、保存、关闭和打印新的或已存在的工程和文件。文件菜单包含的命令如下：

- New: 打开新的工程对话框，包括新的对象和对象库中的对象，可以创建一个新的对象，或是从先前存储在对象库中的对象开始。
- New Application: 创建包含一个空窗体的新工程。
- New Form: 创建一个空窗体并添加到当前的工程中。
- Open: 打开 Open 对话框以向代码编辑器中装入已经存在的工程、窗体、组件或文本文件。
- Open Project: 使用 Open Project 对话框在如工程（.BPR 或.BPK 文件）。
- Reopen: 显示最近关闭的工程或模块以供打开。
- Save: 使用当前文件名保存当前文件。
- Save As: 使用新的文件名保存当前文件。
- Save Project As: 使用新的名字保存当前工程。
- Save All: 保存所有打开的文件，包括工程和模块。
- Close: 关闭当前工程以及相关的组件和窗体。
- Close All: 关闭所有文件。
- Use unit: 使用 use 语句将所选文件添加到活动模块中。
- Print: 打印有效文件。
- Exit: 关闭打开的工程并退出 Delphi。

Edit 菜单

使用 Edit 菜单，如图 1.14 所示中的命令在设计时操作文本和组件，命令如下：

- Undelete: 取消最近的操作。
- Redo: 取消 UnDelete 操作。
- Cut: 将选中的对象剪切，存放到剪贴板。

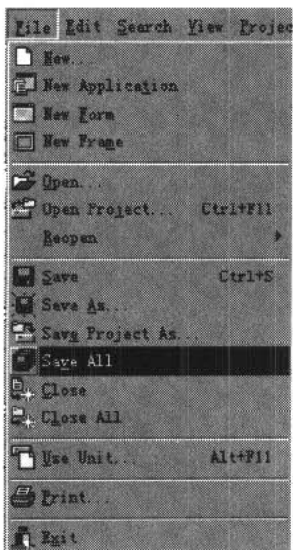


图 1.13 File 菜单

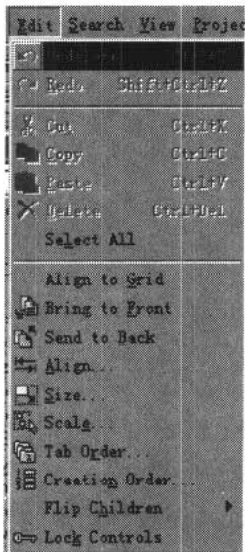


图 1.14 Edit 菜单

- Copy: 复制选中的对象, 并存放到剪贴板中。
- Paste: 将剪贴板中的内容复制到代码编辑器中或窗体中。
- Delete: 删除选中的对象。
- Select All: 选中窗体中的所有内容。
- Align to Grid: 将选中的组件以最近的网格点作为基准来定位。
- Bring to Front: 将选中的组件放到最前面。
- Send to Back: 将选中的组件放到最后面。
- Align: 组件定位。
- Size: 调整组件大小。
- Scale: 调整窗体中所有组件的大小。
- Tab Order: 修改活动窗体中组件的标签顺序。
- Creation Order: 修改非可视组件创建的顺序。
- Flip Children: 允许用户左右镜像翻转当前窗体中的控件。这使得开发者可以很快创建一个适合有从右到左阅读习惯的用户浏览的窗体。该菜单项还有两个二级菜单项:

All: 翻转窗体中所有的子控件, 同时改变所有控件的对齐方式。

Selected: 翻转窗体中选中的子控件, 同时改变选中控件的对齐方式。

- Lock Controls: 将窗体中所有组件锁定到当前位置。

Search 菜单

使用 Search 菜单, 如图 1.15 所示在代码编辑器中定位文本、错误、对象、组件、变量和符号。该菜单中包括如下命令:

- Find: 查找指定字符串, 并将代码编辑器中出现该字符串的区域加亮。