

BUILDER

C++ BUILDER 跟我学计算机语言系列

BUILDER

BUILDER

鹏达工作室 编 著

4.0 入门与技巧

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

BUILDER

武汉大学出版社

跟我学计算机语言系列

C++ Builder 4.0 入门与技巧

鹏达工作室 编著

武 汉 大 学 出 版 社

图书在版编目(CIP)数据

C++Builder4.0 入门与技巧/鹏达工作室编著. —武汉: 武汉大学出版社, 2001. 2

ISBN 7-307-03107-8

I. C… II. 鹏… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 52267 号

责任编辑: 陈 刚

责任校对: 黄添生

版式设计: 支 笛

出版: 武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件: wdp4@whu.edu.cn 网址: www.wdp.whu.edu.cn)

发行: 新华书店湖北发行所

印刷: 湖北省通山县印刷厂

开本: 787×1092 1/16 印张: 12.375 字数: 239 千字

版次: 2001 年 2 月第 1 版 2001 年 2 月第 1 次印刷

ISBN 7-307-03107-8/TP·92 定价: 16.50 元

版权所有, 不得翻印; 凡购买我社的图书, 如有缺页、倒页、脱页等质量问题者, 请与当地图书销售部门联系调换。

前 言

目前，各种语言编程资料和教程书本难以数计，但大部分由于内容繁杂、概念抽象空洞、专业词汇层出不穷，使一般的初学者敬而远之，遇到问题时无所适从。为了给一般的入门者提供一个行之有效的方法和途径，使之能尽快提升自己的编程水平，我们编写了《跟我学计算机语言》系列丛书，并且精心制作了《编程经典》系列和《用多媒体学》系列教学光盘（另售）。在这套丛书中，我们本着实用、即时、快学的原则，以精辟详实的理论讲解和丰富经典的例子剖析使您迅速步入编程之神秘殿堂。

本书共分为十章，第一章到第五章是一些 C++ Builder 的基础知识，包括 C++ Builder 的基本概念、菜单、窗体、组件等；第六章具体叙述了事件及事件响应；第七章介绍了对象储藏技术；第八章和第九章讲解了 C++ Builder 的应用工具和程序调试技术；第十章具体剖析了几个 C++ Builder 的典型例子。此外，本书还备有大量的附录，内容经典详实，以供读者查询。

该系列丛书适合入门者和面临应用的初、中级用户，也适合大学、中专计算机专业和非计算机专业学生自学使用。本书由武汉鹏达计算机系统开发有限公司“鹏达工作室”舒军、刘辉等编著，在此特别感谢相关工作人员和技术编辑。

由于时间仓促，错误和不当之处敬请一并指出！

鹏达工作室

2000 年 4 月

目 录

第一章 C++ Builder 简介	1
1.1 C++ Builder 特点	1
1.2 快速浏览 C++ Builder IDE	2
1.3 编制一个简单的程序	4
第二章 C++ Builder 窗体	6
第三章 C++ Builder 组件	9
3.1 文本输入组件	9
3.2 按钮类组件	13
3.3 列表类组件	17
3.4 特殊的输入控制类组件	19
3.5 图形类组件	24
3.6 图形列表组件 TreeView	29
3.7 图形列表组件 ListView	34
3.8 多页组件	36
3.9 菜单组件	40
3.10 工具条组件	43
3.11 状态条组件	46
3.12 对话框组件	47
第四章 菜单	50
第五章 组件一览	61
5.1 Standard 组件页	61
5.2 Additional 组件页	62
5.3 Win32 组件页	63
5.4 System 组件页	64
5.5 Internet 组件页	64

5.6	Data Access 组件页	65
5.7	Data Control 组件页	66
5.8	MIDAS 组件页	66
5.9	Decision Cube 组件页	67
5.10	QReport 组件页	67
5.11	Dialogs 组件页	68
5.12	Win31、Samples 和 ActiveX 组件页	69
 第六章 事件及事件响应		70
 第七章 对象储藏		77
 第八章 C++ Builder 工具		83
 第九章 程序调试技术		88
 第十章 C++ Builder 编程实例		91
10.1	建立一个 MDI 应用程序, 使其能打开并显示图像文件	91
10.2	建立一个 MDI 应用程序, 使其能打开、显示、编辑文本文件	96
10.3	窗口菜单的完善	103
10.4	建立一个标准的 WINDOWS 帮助应用程序	120
10.5	创建一个动态连接库	123
10.6	创建组件	135
10.7	建立一个单层数据库应用程序	138
10.8	建立一个多层数据库应用程序	140
10.9	使用决策支持组件	144
10.10	简单的 Web 服务器应用程序	146
10.11	建立一个事件交互式的数据库查询	152
10.12	一个网上交谈 (Chat) 程序	160
10.13	用 POP3 组件编制电子邮件的接收程序	166
 附录 A 函数库		171
附录 B C++ Builder 操作技巧		188
附录 C 警告信息		189

第一章 C++ Builder 简介

1.1 C++ Builder 特点

C++ Builder 是一个怎样的程序开发工具？

C++ Builder 4.0 是 Inprise 公司 (Borland 公司) 最新推出的快速应用程序的开发工具, 支持最完整的 C++ 语言规则, 是一个真正可视化 C++ 语言的开发工具, 其主要特点有:

1. 包含众多的组件

C++ Builder 有一个可视化组件库, 包含 130 多个可重构组件, 它的功能范围包括完整的 Windows95 用户界面元素、数据库感知能力、多媒体工具、报表功能和 Internet 元素, 可以帮助用户在极短的时间内开发出复杂的 Windows 应用程序。

2. 具有强大的数据库功能

它包含全新的 32 位数据库引擎 BDE, 可以支持桌面型数据库 dbase 和 Paradox; 与 ODBC 具有良好的接口, 可以直接存取 Access 和 Foxpro 等数据库; 高性能的 32 位 SQL, 可以访问 Oracle、Interbase、Sybase、MS SQL Server、DB2 和 Informix 等大型数据库, 由于采用了丰富的数据库可视化组件, 使数据库应用程序的开发极其方便。

3. 包含实用的 Internet 组件

Internet 组件可以使用户很容易地创建与 Internet 有关的应用程序。如 Web browsers、Internet 电子邮件管理、档案传输、网络论坛连接等。C++ Builder 支持 Microsoft、Sun、Netscape 等公司开发的所有 Internet 标准, 如 ActiveX、CGI、ISAPI、Winnet 和 NSAPI 等。

4. 具有平滑的 ActiveX/OCX 接口

C++ Builder 平滑地把 ActiveX/OCX 控制元结合到自身的编程环境中, 用户不仅可以利用现有的 VCL 组件, 而且还可以采用第三方的控制组件来扩充用户应用软件的功能。

5. 支持多媒体的 OLE 编程

由于 C++ Builder 包含大量的可视化组件, 用户可以使用多媒体组件开发如声

音、图像和动画一体的多媒体应用程序，使用 OLE 组件能够自动创建或控制其他 Windows 应用程序，如 Microsoft Word、Excel、Powerpoint、AutoCAD 和 Lotus1-2-3 等。

6. 具备扩展的中文编程能力

C++ Builder 能够支持中文应用程序的开发，与中文具有良好的兼容性，提供了一些附加的中文控制能力，如中文输入法控制、Editmask 和双位元组代码编辑器等。如果用户具有支持中文的 BDE(数据库引擎)，C++ Builder 还可以创建支持汉字的数据库应用软件。

1.2 快速浏览 C++ Builder IDE

C++ Builder 的界面是怎样布局的呢？

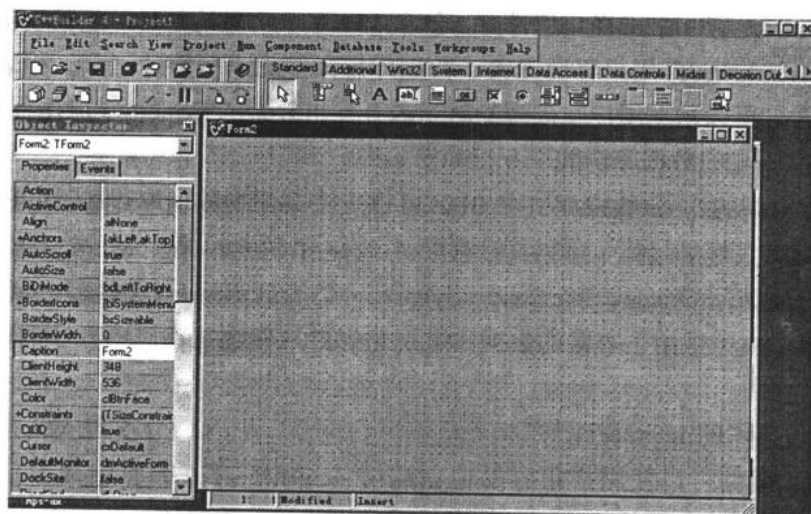


图 1-1 C++ Builder 集成界面开发环境界面

C++ Builder 集成开发环境可以分为三部分：

图 1-1 的窗口可以认为是主窗口，它的左边是工具栏，这些按钮在我们工作的时候，可以节省我们宝贵的时间。每一个按钮执行一个操作命令，如打开文件、保存工程、运行程序等等。我们可以在工具条上单击鼠标右键，然后选择“Properties”菜单项来按照我们的要求定制这个工具条。

右边是组件调色板。工具栏允许你通过单击来完成诸如打开和编译项目等任务，组件调色板包括一个宽的组件陈列，你可将这些组件放到你的表格中去。为方便起

见, 组件又分为不同的组。你是否注意到在组件调色板顶部的标志? 单击这些标志可以得到不同的可用组件。要把一个组件放到表单里, 你只需单击组件调色板中该组件的按钮, 然后再单击你打算放组件的表单的相应位置。

最左边是对象观察器, 通过它可以修正组件的属性和事件, 组件的属性用来控制组件特征, 例如改变一个组件的 Color 属性将改变组件的背景颜色。所有可见组件有几个共同属性如 Width 和 Height 等, 各个组件的属性列表一般是各不相同的。在 C++ Builder 中常常会用到对象观察器, 它有一或两个标签, 根据当前选择的组件而定。它总有一个 Properties 标签。

在对象观察器的右边是 C++ Builder 工作空间。工作空间最初显示 Form Editor。它允许你创建表单。在 C++ Builder 中的一个表单代表程序中的一个窗口。表单可以是程序中的主窗口、一个对话框或任何其他类型的窗口。作为表单创建过程的一部分, 可以用 Form Editor 放置、移走组件和调整组件尺寸。

为了关闭一个表单编辑器, 可以选择想要关闭的表单编辑器页, 用鼠标左键点击表单编辑器窗口右上部分的关闭按钮。要关闭所有的表单编辑器页和工程, 可以选择文件“File”菜单下的“Close All”菜单项。同样, 当应用程序的修改没有保存时, C++ Builder 会自动弹出文件保存对话框, 提醒用户保存文件。

隐藏在 Form Editor 之后的是 Code Editor。Code Editor 是你编写程序时输入代码的地方。

C++ Builder 提供的代码编辑器为我们的应用程序设计提供了一个方便可靠的查看、修改程序源代码的途径。

代码编辑器窗口包含了一个或多个代码编辑器页、一个槽和一个消息窗口。当创建一个新的工程、加入一个表单或者一个单元、打开一个文件的时候, C++ Builder 在代码编辑器中都将产生一个新页。我们也可以使用代码编辑器打开文本文件来查看、修改。为了打开代码编辑器的页, 可加入一个新的表单、单元或者其他文件到工程中。

在代码编辑器窗口中, 我们可以用鼠标左键点一下我们想要查看、修改页所对应的标签来选择代码编辑器的页。我们也可以打开任何代码编辑器中一个页的拷贝, 即打开一个分离的窗口。其方法是: 打开一个文件, 然后选择查看“View”菜单下的新建编辑窗口“New Edit Window”菜单项。这个命令打开一个代码编辑器中当前页的拷贝。

为了关闭代码编辑器的页, 可以用鼠标右键点击代码编辑器的窗口, 然后选择关闭页“Close Page”菜单项。当要关闭的代码页还没有保存时, 代码编辑器会弹出文件保存对话框, 提醒用户保存文件。当我们关闭了代码编辑器的最后一个页的时候, 代码编辑器将会自动关闭。

在编辑代码的时候，最为方便的是用户通过按下 `ctrl+j` 键就可以插入相应的代码模板。

另外，在代码编辑器中，我们要获取上下文相关帮助，可以用光标选取单词，然后，按下“F1”键来取得帮助信息。

和 Borland C++ 系列开发平台类似，在 C++ Builder 的开发平台上，当我们编辑应用程序的时候，我们也可以显示上下文相关的命令。其方法是通过将鼠标移动到一定的区域，然后单击鼠标右键，这时，一个包含了与我们所做工作有关的菜单命令就会显示出来。这个菜单通常是上下文相关菜单（context menu）或者是加速菜单（Speed Menu）。在必要的时候，我们也可以定制这个上下文相关菜单，使得这个菜单包含有我们所需要的特殊指定的功能。在 C++ Builder 的集成开发平台上，我们可以在很多的地方通过单击鼠标右键来得到上下文相关菜单。这些地方主要包括：表单、组件板、对象监视器、工具条、工程管理器、调试以及代码编辑器等。

1.3 编制一个简单的程序

现在，我们正运行 C++ Builder，正在面对空表单，空表单的缺省名为 `Form1`，在表单的左边，对象观察器显示表单的属性。单击对象观察器的标题条，点击 `Caption` 属性，这时，`Caption` 属性变亮，光标停留在那儿等待你的指令。输入 `Hello World!` 以改变表单的标题。如图 1-2 所示，现在单击工具栏上的 `Run` 按钮，C++ Builder



图 1-2 标准的属性

开始建立程序。此时会显示编译器状态对话框。你可以看到 C++ Builder 正在构造建立程序所需文件。稍候, 编译器状态框消失了, 出现了表单, 并显示标题 Hello world! 如图 1-3 所示。此时正在运行的程序几乎与空表单完全一样。是不是很容易?

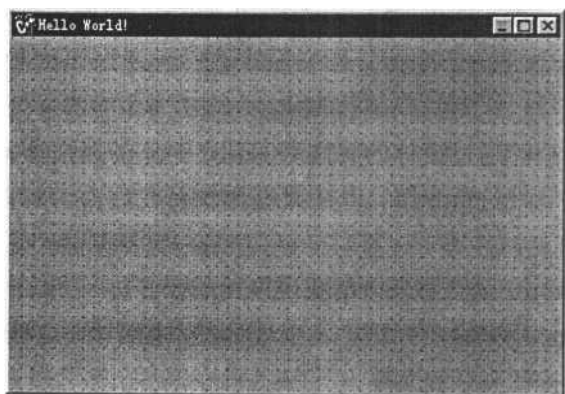


图 1-3 改变标题运行后的窗体

第二章 C++ Builder 窗体

C++ Builder 是怎样处理窗体的呢？

窗体的一部分功能通过窗体的属性来实现。窗体的属性非常多，在这里我们只对其中比较重要的属性进行介绍，其中某些属性将在后面讲解。这里只简单地介绍一下。

1. Active Control 属性用来指定获得焦点的控件；
2. Border Icons 用来设定标题栏上出现哪些系统图标。这些图标有最大化按钮、最小化按钮、帮助按钮和系统菜单；
3. Icon 属性用来指定窗体左上角显示的图标；
4. Key Preview 属性用来确定键盘事件发生时是否先传给窗体，然后再传给产生键盘事件的控件；
5. Menu 属性用来指定窗体的主菜单；
6. Visible 属性用来确定窗体是否可见。

此外还有一些属性，这些属性只有在程序运行的时候才能通过编程访问。这种属性主要有：

1. Active 属性用来确定窗体是否获得输入焦点；
2. Active MDI Child，使用该属性可以得到获得了输入焦点的 MDI 子窗体，如果窗体不是 MDI 父窗体，那么这个属性将返回 nil；
3. Drop Target 属性用来指定窗体是否是拖放操作的对象；
4. Parent 属性用来设定包含窗体的窗口控件，通常是另一个窗体。如果窗体没有包含它的控件，那么它的 Parent 属性为 nil。

如何控制窗体的边界样式？

属性 Border Style 可以指定窗体边界的外观和行为。它一共有六种可能的取值：

1. Bs Dialog 表示大小不可变的标准的对话框边界；
2. Bs Single 表示大小不可变的单线边界；
3. Bs None 表示大小不可变的非可见边界；
4. Bs Sizeable 表示标准的可变边界；
5. Bs Tool Window 与属性值取 bsSingle 时相似，但是标题较小；

6. Bs Size Tool Win 与属性值取 bsSizeable 时相似, 但是标题较小。

要注意的是, 对于 MDI 子窗体, sDialog 或 bsNone 将不起作用。

只从文字的角度理解属性 Border Style, 很难对这些属性值有印象, 我们可以通过程序来说明这个问题。其步骤如下:

首先点击组件调色板的 Standard 标签, 选择 Radio Group 组件, 在窗体上我们希望放置这个组件的地方点击一下, 调整这个组件和窗体的大小和位置。选中 RadioGroup1 组件, 在对象观察器中选择 Items 属性, 点击小按钮, 弹出一个编辑器, 在编辑器中输入几行文字。点击 OK 键, 这时, 收音机按钮组中就出现了六个收音机按钮。

这样我们就在窗体上建立了一个单选按钮组, 它的选择项就是这六种属性值。选择不同的选择项就可以生成边界样式不同的表单了。

在对象观察器中选择 Item Index 属性, 将它修改成 0, 这样, 在缺省情况下将选中对话框风格。

在对象编辑框上选择 Events 标签页, 双击 On Click 的 Value 列, 加入一段代码。

这段程序根据用户的选择, 将当前窗体设置为指定的边界风格。

要注意的是在访问单选按钮组的属性时, 使用了成员指针运算符。这是因为, C++ Builder 的每一个控件都是用动态分配的内存保存的。所以 RadioGroup1 只是一个对象的指针。

运行这个程序, 分别选择六个单选按钮, 可以看到他们的不同之处。

属性 Form Style 有什么作用?

属性 Form Style 可以确定窗体的类型, 它一共有四种属性值:

1. fsNormal 表示窗体既不是 MDI 父窗口也不是 MDI 子窗口, 只可能是单文档 SDI 窗口或者对话框。这个属性值是缺省值;

2. fsMDIChild 表示这个窗体是一个 MDI 子窗口;

3. fsMDIForm 表示这个窗体是一个 MDI 父窗口;

4. fsStayOnTop 表示窗体始终保持在窗体所属工程中其他窗体的上面, 除非还有别的窗体的 Form Style 属性也被设置成了 fsStayOnTop。

要特别注意的是, 在程序运行期间不要修改窗体的 Form Style 属性。

属性 Position 有什么作用?

属性 Position 可以确定程序运行时出现在屏幕上的窗体的大小和位置, 它有五种可能的取值:

1. `poDesigned` 表示使用设计时指定的尺寸和位置属性显示窗体。尺寸属性是指 `Height` 和 `Width`，位置属性是指 `Left` 和 `Top`。这个属性值是缺省值；

2. `poDefault` 表示窗体在屏幕上显示的位置和大小由 `Windows` 决定。每次运行应用程序时，窗体都会稍微向下向右移动；

3. `poDefaultPosOnly` 表示窗体以设计时的窗体尺寸显示窗体，但是在屏幕上显示的位置由 `Windows` 确定。每次运行应用程序时，窗体都会稍微向下向右移动。当无法再向下移动时，就又会从屏幕的左上角开始显示；

4. `poDefaultSizeOnly` 表示窗体的位置由设计时指定的值确定，而尺寸由 `Windows` 确定；

5. `poScreenCenter` 表示使用设计时指定的尺寸和位置属性，在屏幕的中央显示窗体。

怎样最大化或最小化窗体？

属性 `WindowState` 与窗体的显示有关，它有三种可能的取值：

1. `WsNormal` 表示窗体既不是最大化显示也不是最小化显示。这个值是缺省值；

2. `WsMinimized` 表示窗体最小化显示；

3. `WsMaximized` 表示窗体最大化显示。

属性 `Client Height` 和 `Client Width` 有什么作用？

这两个属性用来定义窗体用户区域的高和宽。所谓用户区域是指窗体的实际工作区域，就是除了窗体的标题栏、菜单条、工具条和状态条以及边框之外的窗体区域。所以，对于新创建的窗体，它的用户区域比窗体的大小要稍微小一些，相对而言，窗体的大小是由属性 `Height` 和 `Width` 确定的。

第三章 C++ Builder 组件

3.1 文本输入组件

在 C++ Builder 中如何实现文本输入？

C++ Builder 常用文本输入组件来实现文本输入，常用的文本输入组件有编辑框 Edit 和 Memo。他们的主要不同在于编辑框用于输入单行文本，而 Memo 可以输入多行文本。此外，Label 组件也可用来进行文本显示。编辑框是一个窗口控件，它可以获得输入焦点。当用户需要输入单行文本时，就应该使用编辑框。它通常与标签组件一起使用。

编辑框常用的属性有哪几个？

1. Text 属性是一个 String 类型的数据，它决定了在编辑框中出现的文本字符串。在编程中，我们经常要通过 text 属性获取编辑框中的文本字符串。

2. Max Length 是一个 Integer 类型的数据，它指定编辑框所能容纳的最大字符数。缺省情况下为 0，表示长度不限。

3. 编辑框还可以用做口令输入。具体方法是，把 Password Char 属性设置为除 #、0 之外的任何字符。这时，无论你在编辑框中输入什么字符，都只显示 Password Char 包含的字符。我们经常把 Password Char 设置为星号*，当用户输入口令时只显示星号。如输入#0 表示编辑框正常显示。Password Char 是一个字符类型的数据。

4. 如果要限制用户对编辑框写的权力，可以使用属性 Read Only，当 Read Only 的值为 True 时只读。

标签的常用属性有 Caption 和 Focus Control。

Caption 属性是字符串类型，用来指定标签的标题，也就是标签的显示内容。

Focus Control 属性是窗口控件类，用来指定一个与标签相连的窗口控件，从而允许这个控件使用快捷键来获得输入焦点。

标签是一个典型的非窗口控件，它不能获得输入焦点，所以经常被用来给一些没有 Caption 属性的组件做标签。标签还能同时给这些组件提供快捷键的功能，允许用户通过快捷键获得输入焦点。

现在，我们就为编辑框实现刚才介绍的两个功能。

如图 3-1，首先，在输入标题时需要指定一个快捷键，这可以用在一个字母前面加上一个连字符&来实现。单击 Caption 属性输入栏，输入&N 姓名，这里字母 N 就被指定为快捷键，要注意的是中文是不能被指定为快捷键的，只有 26 个英文字母才可以。

然后，把标签的 Focus Control 属性与编辑框 Name Edit 相连，选择 Focus Control 属性，从属性值字段列举的窗体上所有的窗口控件中选取 Name Edit。这样，当用户按下 Alt+N 时，输入焦点就会转移到编辑框 Name Edit。

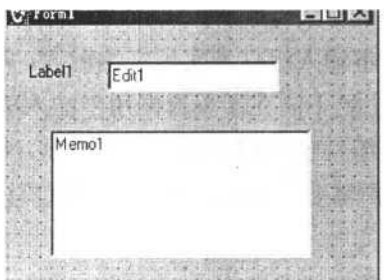


图 3-1 Memo 组件的窗体布局

键盘事件的处理：

当我们按下回车键时，产生了 On Key Press 事件。所以，如果要在程序中处理这种事件，就要编写 On Key Press 事件处理程序。

还有三种编辑框常用的事件：On Change 事件、On Enter 事件和 On Exit 事件。每当编辑框中的文本发生改变时都会触发 On Change 事件。当编辑框获得输入焦点时会触发 On Enter 事件，而失去焦点时会产生 On Exit 事件。

编辑框只能编辑单行的文本，如果要编辑多行文本怎么办？

这就必须使用 Memo 组件了。Memo 组件与编辑框组件相比有一些不同的属性。Lines 属性是一个 TStrings 类的一个对象，它是由多个字符串组成的，每一个字符串就是 Lines 中的一个元素。Memo 组件的每一行文本都是 Lines 中的一个字符串。

在设计阶段，如果要给 Memo 组件增加一些显示内容，可以在对象编辑器中选择属性 Lines，单击 Value1 列上的省略号按钮，这时会打开一个字符串编辑窗口。

在这个编辑器中输入文本。编辑器中显示的所有内容都会出现在 Memo 组件中。

属性 Scroll Bar 是一个常见的属性，它确定着滚动组件滚动条的行为。Scroll Bar

有四种可能的取值:

1. ssNone 表示没有滚动条;
2. ssHorizontal 表示只有水平滚动条;
3. ssVertical 表示只有垂直条;
4. ssBoth 表示既有水平滚动条也有垂直滚动条。

当显示的内容比较多时, 应该选择 ssBoth, 这样才能够使用户看到所有的内容。

Memo 组件属性中:

- Alignment 用来确定 Memo 组件显示文本的对齐方式;
- Word Wrap 确定文本到达右边界时是否自动换行;
- Want Return 确定用户是否可以在文本中插入回车符;
- Want Tabs 确定用户是否可以在文本中插入 Tab 字符。

如果要在程序运行期间修改 Memo 组件的显示内容, 应该怎么办?

这就需要使用 TString 类的某些属性和方法。例如要给 Memo 组件增加一句话“鹏达软件”, 那么可以使用方法 Add 来完成。首先新建一个表单, 放进一个 Memo 和一个 Button, 在 Button 上双击鼠标, 加入下面一行话:

```
Memo1->Lines->Add("鹏达软件");
```

然后运行, 结果如图 3-2 所示。

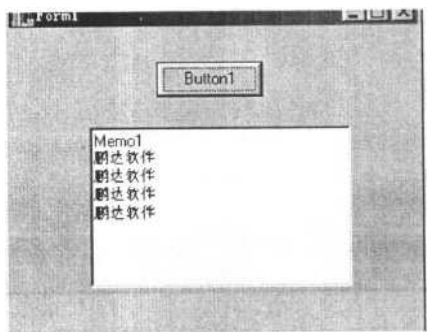


图 3-2 Memo 组件的运行结果

要寻找 Memo 中的某一行, 可以使用 Memo->Lines->Strings[n] 的形式, 其中 n 是行号减一。这时取出的实际上是 Lines 的第 n+1 个字符串, 不过因为 Lines 的字符串是从 0 开始编号的, 所以在与 Memo 中的实际行号对应时, 行号需要减 1。