BORLAND® C++BUILDER™ 编程指南

C++BUILDER™ 问题的明确解答

[美] John Miano
Tom Cabanski 著
Harold Howe
郝 杰 崔晓东 龚 蕙 等译
严 程 审校

上百种的C++BUILDER实用方法，包括：

- Delphi调用序列与C++Builder代码的混合编程
- 在运行期间创建BDE别名
- 生成可被翻译为多种语言的窗体
- 利用Tstring Grid组件创建Windows 95滚动条
- 绘制垂直方向的y坐标轴
- 掌握C++Builder的Internet功能及其它功能

BORLAND®
C++BUILDER™ HOW-TO

WAITE GROUP PRESS™

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL:http://www.phei.com.cn

译 者 前 言

本书介绍了一些关于 Borland® C++ Builder™的编程方法与技巧。C++ Builder 是 Borland 的可视化 C++开发工具。本书假设读者已经懂得 C++,并且具有 C++ Builder 的基本知识。

本书采用问答的格式,书中每一种方法由问题描述、解决问题的技术、方法的各步细节和对方法的解释构成。本书的目的是帮助你解决在开发 Windows 应用时经常遇到的问题。对使用 C++ Builder 的程序员来说,本书是对手册和帮助文件的一个重要的补充。

本书第 1 章到第 3 章由龚蕙翻译,第 4 和第 5 章由崔晓东翻译,第 7 章到第 9 章及附录由严程翻译,第 10 章和第 11 章由李德胜翻译,第 6、12、13、14、15、16 和 17 章由郝杰翻译,全书由郝杰整理并且定稿,严程审校。另外,辛广、田骏、吴松平、曹伟、彭勇、杨景农和刘家海也参与了部分整理与校对工作,在此一并表示感谢。

译者

作 者 介 绍

John Miano 是 Colosseum 公司的负责人，现在从事与广播和娱乐业有关的系统开发。他毕业于 Wooster 学院，并获得数学学士学位。在过去的十一年中，他从事过软件咨询、基于 UNIX 和 Windows 的系统开发等工作。John 与他的名为 Mycroft 和 Irene 的猫住在新泽西的 Summit。他的电子邮件地址是 miano@worldnet.att.net。

Tom Cabanski 经营着他自己的 Common Sense Consulting 公司，现在从事多层客户机/服务器和 Internet/Intranet 应用的开发。他最初是个 COBOL 程序员，曾使用从 APL 到 Z(一种专用的打印机编程语言)的各种工具开发过系统。他现在住在德克萨斯的 Kingwood，他的电子邮件地址是 tomc@vonl.com。

Harold Howe 作为一个设计工程师和一个计算机程序员，工作于 Technology Resource Group，这是一个工程与咨询公司。他在依阿华州立大学获得了电气工程专业的学士学位。Harold 在高中就开始用 BASIC 和 Pascal 编程，后来又用 C 和 C++ 编写嵌入式系统。从那以后，他开始在 Windows 环境中编程，使用 C++ 和 Borland 的 OWL 类库编了一些共享软件。他的电子邮件地址是 hhowe@trgnet.com。

引　言

“什么要花这么长时间呢?”这是大多数 C++ 程序员在第一次尝试 C++ Builder 时的反应。直到现在,用 C++ 制作 Windows 应用仍是一场恶梦。开发应用时,可以选用编程接口晦涩的 Windows API,也可以选用类库,选用后者更多地缘于它们的复杂性,而不是它们的易用性。编写 Windows 应用是如此之难,以至于有经验的 C++ 程序员也求救于 Visual Basic 和 PowerBuilder,而不顾这些工具不适合开发大的应用。多少年过去了,Windows 的使用持续增长,C++ 开发工具仍然处在黑暗时代。终于,C++ Builder 的 C++ 开发环境使 Windows 编程变成一件有趣的事。开发者再不必对使用 C++ 来开发 Windows 应用有其他想法。

《Borland ® C++ Builder™编程指南》一书的目的是帮助你解决在开发 Windows 应用时经常遇到的问题。这本书并不是手册或帮助文件的替代品,而是对它们的一个补充。当你读这本书的时候,你可能会不时地参考这些相关材料。这本书并不是编程的入门读物。我们假设读者已经懂得 C++ 并且具有 C++ Builder 的基本知识。

这本书被组织成一系列的问题或者说是方法,采用一种问答的格式。每一个方法包括问题的描述、解决问题的技术、方法的各步细节和对方法的解释。方法的复杂度范围包括从非常简单到非常难(从入门到高级)。为了方便读者,所有各种方法的源码都被包括在随书提供的光盘中。光盘还包括一些扩展的范例以及所有必需的附加工具。

在开始阅读本书之前,需要记住两条约定。在源码中,由程序员输入的代码用黑体表示,区别于由 C++ Builder 自动生成的代码。大多数方法有一个或多个表格,用来表示如何为范例中的控制设置属性值。这些表格只列出需要改变其值的一些属性。由于视频显示上的差别,可能需要调整一些位置与尺寸属性,使应用在系统中的显示与图中给出的一样。

第 1 章,“窗体”,描述了如何解决在使用 C++ Builder 窗体时经常遇到的问题。使用窗体是用 C++ Builder 进行可视化编程时最基本的任务。不像大多数可视化编程工具,C++ Builder 允许在运行期间创建窗体或组件。学会如何在一个 MDI 窗体上画背景和拖动一个无标题栏的窗口。

第 2 章,“标准组件和类”,讲述如何使用 VCL 类库中的许多基类。克服标准组件中的一些限制,并且学习如何使用 VCL 类库的复杂特性来制作看起来很巧妙并具专业水平的应用。

第 3 章,“文本控件”,包括了处理键盘输入的各种控件。详细描述了 Rich Edit 控件的用法。

第 4 章,“鼠标和菜单”,包含如何创建特殊的菜单的方法。创建使用不同字体或使用图形的菜单,甚至在运行期间创建菜单。

第 5 章,“图形”,讲述如何在应用中使用高级的图形特性。学习动画的显示、图像的淡入淡出和使用画布的高级特性。

第 6 章,“环境与系统”,讲述如何与操作系统交互作用。除了最简单的程序,所有程序都需要获取关于它们正在运行的环境的信息。学习如何确定硬盘空间与可用内存的多少、如何在任务栏的“托盘”中放置一个图标以及如何拖放文件。

第 7 章,“外设”,描述如何控制并且获取关于外设的信息。学习如何确定软驱中是否有盘片以及如何在一个应用中使用调制解调器。

第 8 章,“Internet”,描述如何使用 C++ Builder 带的 Internet 控件。可以很容易地用

C++ Builder制作具有 Web 浏览或 FTP 功能的应用。

第 9 章,“多媒体”,包括 C++ Builder 应用中的音频、视频和图形。由于计算机的功能越来越强大,用户已经开始期望像这样复杂的功能。利用 C++ Builder 制作多媒体应用很轻松。

第 10 章,“打印”,包含在 C++ Builder 中如何使用打印机的方法。几乎每个应用都需要能够打印数据,这章讲述几种不同的方法,包括使用油布、打印文本行以及使用 Quick Reports。

第 11 章,“数据库”,讲述如何利用 C++ Builder 中的数据库控件来制作针对数据的应用。大多数可视化开发工具允许开发者很容易地创建数据库的前端,C++ Builder 也不例外。学习如何在数据库中观察和编辑数据以及如何在运行期间创建 BDE 别名。

第 12 章,“线程”,包含在应用中如何使用多个线程的方法。Windows 95 和 Windows NT 都允许在一个进程中创建多个线程。学习如何创建利用多线程同时完成多项任务的 C++ Builder 应用。

第 13 章,“对象的链接与嵌入(OLE)”,描述如何在应用中使用 OLE。利用 OLE 使多个应用互相通信,编制能够并入其他应用中去的 C++ Builder 应用,以及编制一个 OLE 服务器应用供其他应用使用。

第 14 章,“异常”,描述如何在应用中使用异常处理。为了编写稳健的应用,必须考虑到异常情况。异常处理允许应用从错误中恢复。

第 15 章,“定制组件”,是对创建定制组件的介绍。能够容易地将新组件增加到组件板中的能力是 C++ Builder 最强大的功能之一。创建自己的组件,并且在设计期间使用它们。

第 16 章,“洗炼的应用”,描述如何实现经常在专业的应用软件中发现的一些特性。学习如何创建一个看上去很专业的 About 屏幕或者当应用启动时显示的封面屏幕。

第 17 章,“秘密和技巧”,包含如何做一些不太寻常的事的方法。如何从一个毁坏的窗体文件中恢复?如何在应用中加入分析功能?弄清楚如何实现这些及其他功能。

安　装

随本书提供的光盘包含了生成《Borland® C++ Builder™编程指南》一书中所有的源码和所需的工具以及一些扩展的范例和辅助 C++ Builder 的工具。另外，因为 Borland 的C++ Builder 与 Delphi 高度集成，所以我们提供给你一个 Delphi 2 的试用版。

光盘包括了每一节中的应用，而且是已经编译好的，可以直接从光盘上运行它们。但是，在 C++ Builder IDE 中运行它们时，必须先将它们安装到硬盘上。

将光盘上的内容安装到硬盘上的时候，运行根目录中的 WSETUP 程序。WSETUP 程序将提示安装哪些文件以及安装到何处。

安装 FLEX 和 BISON

按照以下各步安装 17.1 节中用到的 GNU FLEX 和 BISON 程序：

1.创建一个目录，名为 C:\BISON，或是你另取的一个名字。

2.将光盘上目录\3RDPARTY \BISON 下的内容复制到目录 C:\BISON 或你在上一步创建的目录下。

3.创建一个目录，名为 C:\FLEX，或是你另取的一个名字。

4.将光盘上目录\3RDPARTY \FLEX 下的内容复制到目录 C:\FLEX 或你在上一步创建的目录下。

5.编辑 AUTOEXEC.BAT 文件，在文件尾增加下面一行：

PATH = C:\FLEX;C:\BISON;%PATH%

打开 MS - DOS 窗口，在命令行中敲 BISON 或 FLEX 就可以运行 BISON 或 FLEX 了。

光盘内容

书中各种方法的源码文件以不压缩的格式存储在光盘上，因此不需运行 WSETUP 程序就可以浏览或复制这些文件。光盘的内容安排如图 I-1 所示。

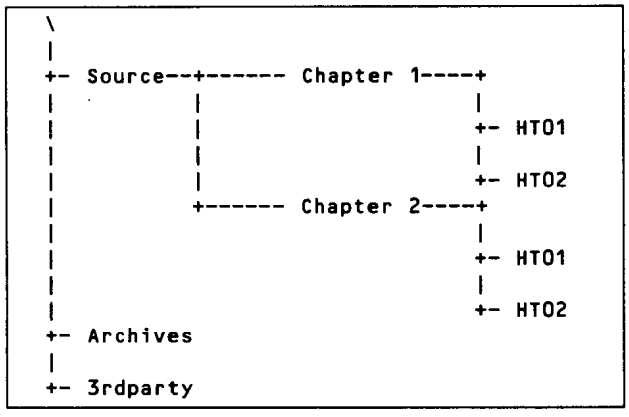

图 I-1　光盘的内容安排

目　录

第1章 窗 体

如何...

C++ Builder™应用程序是基于窗体的。每一个窗口和每一个对话框是一个窗体。窗体允许你为自己的程序可视化地设计用户界面。不使用窗体就可以创建应用程序,如单一模式的程序和 DLL 动态连接库,但是对于大多数的 GUI 设计来说,应利用包含在C++ Builder 窗体中的可视化成分。

窗体的作用就像一个容器,以某种逻辑方式将控件编组,这对程序员和用户都有益处。本章的技巧和技术展示了如何在应用程序设计中最大限度地使用窗体。读者将会学到如何确定窗体的位置、更改窗体的缺省设置和制定窗体的外观及更多的知识。

如果用户更熟悉传统的编译器例如 Borland C++ 、Microsoft Visual C++ 或 Watcom C++ 等,使用窗体的思想对你来说也许很新鲜。Delphi 和 Visual Basic 的程序员应该熟悉基于窗体的可视化程序设计。对于读者来说,学习 C++ Builder 的最大挑战也许是学习C++ 语言本身。本章介绍了C++ Builder 基于窗体的方法和编程背景。

1.1 使窗体自动地出现在应用程序中央

窗体有一个被称做 Position 的参数,它可使用户控制窗体的显示位置。设置 Position 为 poScreenCenter 即将窗体放在屏幕中央,对于担当普通对话框的窗体来说,这是个方便的选项。然而,有时用户想要放置对话框在主窗体中央,而不是屏幕中央。这一节教给用户无论主窗体在屏幕任何位置,用户都可以将副窗体放在应用程序的主窗体中央。

1.2 在运行期间从头创建一个窗体

偶尔也需要在运行期间完整地创建和显示窗体,在运行期间创建窗体需要费些功夫,但付出的努力是值得的。这一节教给用户如何在运行期间构造窗体,然后把控件增加到基于用户输入的窗体中。

1.3 拖动一个无标题的窗口

有时需要小窗体起到系统工具的作用,例如时钟、桌面计算器或资源监视器。也许用户想

移走窗体的标题栏以使应用程序所占的空间最小。然而，移走窗体的标题栏会阻碍用户在桌面上拖动窗体。这一节讲述了如何能使用户拖动无标题的窗体。

1.4 为窗体设置最大最小尺寸

用户希望可以修改窗体尺寸，但有时也需要对窗体的尺寸进行限制以防止它变得十分难看。假如一个窗体太小，它会隐藏掉重要的信息，这一节介绍了如何才能利用 Windows 限制窗体的最大最小尺寸。

1.5 在窗体中使 ENTER 发挥像 TAB 一样的作用

对于一些计算机用户来说，通过敲击 ENTER 键完成一个编辑框项目是很自然的。如果这样做，WINDOWS 仅在其表面背后发出正常的蜂鸣声。这一节解释了该如何提供给那些想要使用 ENTER 键完成一个编辑框项目并且能移到下一个的用户。

1.6 在 MDI 父窗体上绘制一个有意思的背景

C++ Builder 允许改变所有窗体的背景颜色，包括在 fsMDIForm 中具有 FormStyle 特征的窗体。然而，在 MDI 背景中绘一张用户图片并不容易，因为 MDI 客户窗口会阻止绘图。这一节描述了如何摆脱 MDI 客户窗口，绘一张自己的图形 MDI 背景。

1.7 定制带图示的窗口标题栏

通常在非客户区域应用程序不能有图形。但是 Microsoft Office 应用程序以标题栏装饰，所以必有一种方法能做到同样的事情。这一节描述了如何做出通过增加客户图形到主窗体的标题栏同样的专业外观。

1.8 询问用户是否真的想要关闭窗体

关闭窗体的按钮(X)置于醒目位置会使用户很容易意外关闭窗体。这一节显示了如何能使用户在他们没有充分准备关闭窗体和可能会丢失对文件的修改之前迅速做出确认。

1.9 在程序开始运行时显示一个醒目的屏幕

所有窗口和对话框在C++ Builder 中都是窗体，引人注目的屏幕也不例外。这一节介绍如何增加有专业外观的醒目屏幕到程序中的过程。

1.10 显示具有自己的风格的对话框

如果从另一个C++ 开发环境提升到C++ Builder 中，所有这些窗体的事物一定会使人大吃一惊。如果试着创建一个对话框，RC 文件发生的事情会使人很疑惑。在C++ Builder 中使用窗体创建对话框而不是用 RC 文件。这一节讲述了如何使用C++ Builder 的基于窗体的方法设计具有自己风格的对话框。

1.1 使窗体自动地出现在应用程序中央

问题

TForm 包含一个 Positon 参数，能被设置成 poScreenCenter，把窗体放置在屏幕的中央。有时宁愿把窗体放置在应用程序的边界内。如何才能放置窗体在应用程序主窗体的中央，而不是屏幕中央呢？

技术

TForm 包含两个参数，Top 和 Left，用来确定窗体左上角的位置。在显示副窗体前，可通过计算副窗体的 Top 和 Left 数据成员的值，把窗体放置在应用程序主窗体的中央。

步骤

运行 CD-ROM 上的 CENTER.EXE 例程。可在屏幕的任何地方调节主窗体的大小，设置其位置。选择 File | Show，注意到子窗体出现在主窗体边框内中央。如图 1-1 所示。关闭子窗体，重设主窗体尺寸大小，再打开子窗体，会发现它仍像原来一样在主窗体中央。

1.选择 File | New Application 创建一个新的工程文件。选择 File | Save Project As 来保存工程文件。保存程序单元为 MAINFORM.CPP，保存工程文件为 CENTER.MAK。

2.使用 File | New Form 增加副窗体到工程文件中。选择 File | Save As，新窗体保存为 CENTFORM.CPP。直到第 8 步才用到 Form2，现在可将其最小化。

3.调节 Form1 的尺寸大小，使它如图 1-1 所示。

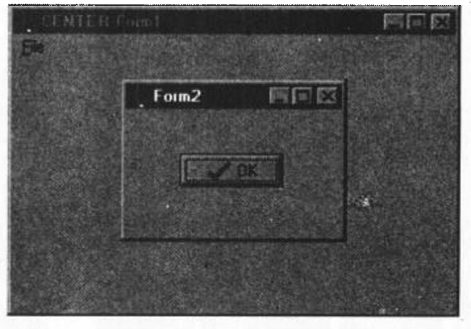

图 1-1 运行期间的 CENTER 程序

4.从 Component Palette 的 Standard tab 中使用 MainMenu 控件，在 Form1 上放置一菜单。双击 MainMenu 组件，进入 Menu Designer。创建 File 菜单项，并增加两个 Show 和 Exit 子项。

5. 为 Exit 菜单项创建 OnClick 事件句柄，在事件句柄的函数中增加以下代码。

提示

为菜单项创建 OnClick 句柄可从 Menu Designer 中双击菜单项或从窗体浏览器中选择菜单项或在 Object Inspector 中双击邻近的 OnClick。C++ Builder 将选择一函数名，但也可使用 Object Inspector 改变名字。

```
void __fastcall TForm1::Exit1Click(TObject *Sender)
{
  Close();
}
```

注意

TForm::Close 关闭一窗体。如果窗体也是应用程序主窗体则中断程序。

6. 为 Form1 的 Show 菜单项创建 OnClick 事件句柄。增加代码,在显示 Form2 前将其放置到中央。

```
void __fastcall TForm1::Show1Click(TObject *Sender)
{
  // Determine the size of the application frame.
  TRect Rect = Application->MainForm->BoundsRect;

  // Calculate the top and left edges of Form2.
  Form2->Left =((Rect.Right - Rect.Left)-Form2->Width)/2 + Rect.Left ;
  Form2->Top = ((Rect.Bottom - Rect.Top)-Form2->Height)/2 + Rect.Top ;
  Form2->ShowModal();
}
```

注意

Form2->ShowModal 在形式上启动 Form2。ShowModal 函数直到用户关闭 Form2 才返回。使用 Show 函数代替 ShowModal 可真正启动 Form2。

7. 第 6 步的代码访问一个指向子窗体 Form2 的指针。在此,Form1 并不知道哪个是 Form2。保持 Code Editor 的状态,移到 MAINFORM.CPP 的顶部。增加 # include 声明:

```
#include <vcl\vcl.h>
#pragma hdrstop

#include "MAINFORM.h"
#include "CENTFORM.h"  // Add include for Form2.
```

注意

CENTFORM.H 头文件包含一个对于 Form2 指针的 extern 原型。

8. Form1 已完成使命。启动 Form2,从 Component Palette 的 Additional tab 中放置 BitBtn 组件,用 Object Inspector 改变 BitBtn 的 Kind 参数为 bkOK。调节 Form2 的尺寸大小,使之与图 1-1 的子窗体匹配。

9. 双击 BitBtn1 以创建 OnClick 事件句柄。增加代码以使用户按下按钮时关闭子窗体。

```
void __fastcall TForm2::BitBtn1Click(TObject *Sender)
{
  Close();
}
```

10. 编译并测试程序。

原理

在第 6 步创建了名为 TForm1::Show1Click 的函数。用户从主菜单中选择 Form | Show,此函数显示 Form2。使用 BoundsRect 参数计算主窗体的坐标,Show1Click 开始运行。再利用主窗体的坐标和窗体的 Height 和 Width 参数,它可计算 Form2 的 Top 和 Left 的边界值。

注意

Application- > MainForm 指向应用程序的主窗体。在这个例子中,Form1 是主窗体,可使用 BoundsRect 代替 Application- > MainForm- > BoundsRect。增加 Application- > MainForm 以保证对于任何类,代码都可正常运行。

说明

TForm1::Show1Click 假设 Form2 指针已经指向一个有效的对象。由于缺省状态,程序开始时C++ Builder 应用程序自动创建它们的窗体。可在 CENTER.CPP 的 WinMain 函数内看到窗体的创建代码。如果从自动创建的窗体中移走 Form2,则需要增加代码来创建 Form2 指针。

如果主窗体部分在屏幕外,怎么办? 再次运行程序,向着屏幕底部向下移动主窗口,以至菜单完全显示出来。选择 File | Show,并注意发生的现象。Form2 跳出屏幕,现在应用程序完全被死锁了。为防止这种情况发生,使用 GetSystemMetrics 以保证 Form2 保持在屏幕上。

复杂度

高级

1.2 在运行期间从头创建一个窗体

问题

现在打算写一个应用程序,使它能基于运行期间的信息创建一个窗体。可以使用C++ Builder 在运行期间从头创建一个窗体吗?

技术

可视组件库(Visual Component Library,VCL)提供了完成这项任务的所有工具。需要用到 TForm 和包括 VCL 组件的类。从头创建一个窗体包括说明窗体指针、创建控件并安排其位置,调用 Show 或 ShowModal 以显示窗体。

注意

这一节完全从头用代码创建了一个窗体。最好动态地创建和释放利用集成开发环境创建的窗体。参考 1.9 和 1.10 节 ,看看如何实现这项技术。

步骤

运行 RUNTIME.EXE。从 Combo box 中选择一个值,敲击下拉按钮来迅速创建窗体。新的窗体动态地创建一套 Check boxes 以至 Check boxes 的数目与在 Combo box 中被选择的数目相符。在执行期间图 1-2 显示 RUNTIME。使用在主窗体中 Combo box 的不同数目试着再次打开动态窗体。

1. 创建一个新工程文件。保存程序单元为 MAINFORM.CPP,保存工程文件名为 RUNTIME.MAK。

2. 使用 File | New Unit 增加一个新程序到工程文件中。选择 File | Save As,并将程序保存为 DYNAFORM.CPP。

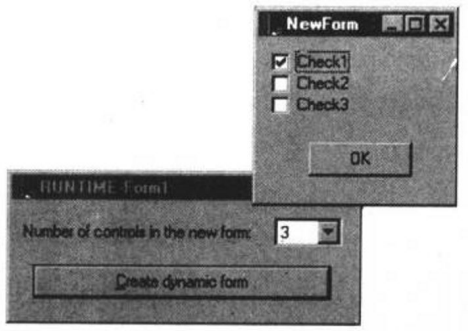

图 1-2　运行期间的 RUNTIME 程序

3. 增加一个 Combo box、一个标签和一个下拉按钮到 Form1 中，以至它能像图 1-2 所示。相应调节窗体的尺寸大小。设置属性与表 1-1 一致。

表 1-1　在 RUNTIME 中 Form1 的组件、属性和设置

COMPONENT	PROPERTY	SETTING
Form1		
Label1	Caption	Number of controls in the new form:
Button1	Caption	&Create dynamic form
ComboBox1	Items	0,1,2,3,4, & 5
	Style	csDropDownList

注意

敲击在 Object Inspector 的 Items 属性中的省略号(…)按钮，可增加字符到 Combo box。这会启动 String List Editor。在每一个 Combo box 行上键入其选项，并按回车键。

4. 双击 Button1，并增加 OnClick 事件句柄代码：

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
  TNewForm *NewForm = new TNewForm(Application,ComboBox1->ItemIndex);
  NewForm->ShowModal();
  delete NewForm;
}
```

5. 第 4 步的 OnClick 句柄需要知道 TNewForm 类。增加 # include 声明到 MAINFORM.CPP 中：

```
#include <vcl\vcl.h>
#pragma hdrstop

#include "MAINFORM.h"
#include "DYNAFORM.h"
```

6. 打开 DYNAFORM.H，并输入类声明和对于 TNewForm 的 # include 语句。

```
#ifndef DynaFormH
#define DynaFormH
//---------------------------------
#include <vcl\Classes.hpp>
#include <vcl\Controls.hpp>
```