

C++Builder 6

实用编程

Programming
Practically

魏俊鹏 于秋生 编著

Programming
Practically

100例

Programming
Practically

Programming
Practically

- ◎ 内容由浅入深、循序渐进；知识点全面，覆盖了C++Builder开发的方方面面。
- ◎ 理论与实践紧密结合，注重实用性和实际问题的解决方案。
- ◎ 结构清晰、层次分明、分类合理，按照实例说明、技术要点、实现步骤、归纳注释的顺序进行阐述。
- ◎ 所附光盘包含书中所有的程序源代码，以方便读者学习。

Programming
Practically

TP312
W5852

C++ Builder 6 实用编程 100 例

魏俊鹏 于秋生 编著

TP312
W5851

中国铁道出版社

2004·北京

66 72 890

内 容 简 介

本书通过 100 个实例由浅入深地介绍了如何使用 C++Builder 6 来进行 Windows 应用程序的开发。全书共分成 11 个部分,其中包括:基本组件应用、功能组件应用、对话框设计、基本绘图、图像处理、文件与系统操作、硬件设备检测、多媒体编程、网络编程、数据库编程及其他应用技巧。

本书实例的程序源代码均包含在随书赠送的光盘中。其中,除了实例 71 与实例 100 需要向 C++Builder 中添加 ActiveX 组件,以及数据库部分的实例需要进行数据源的配置之外,其余程序全部能够在 C++Builder 6 中直接运行。

本书适合于初、中级读者(最好是了解 C 或 C++语法的读者)使用,对于高级程序员也具有一定的参考价值。

图书在版编目(CIP)数据

C++ Builder 6 实用编程 100 例/魏俊鹏,于秋生编著. —北京:中国铁道出版社,2004.5

(实用编程百例丛书)

ISBN 7-113-05966-X

I. C. … II. ①魏… ②于… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 047029 号

书 名: C++ Builder 6 实用编程 100 例

作 者: 魏俊鹏 于秋生

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街8号)

策划编辑: 严晓舟 郭毅鹏

责任编辑: 苏茜 王占清 卜照斌

封面设计: 薛为

印 刷: 北京鑫正大印刷有限公司

开 本: 787×1092 1/16 印张: 23 字数: 562 千

版 本: 2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-05966-X/TP·1222

定 价: 40.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

在面向对象的可视化编程技术领域，由 Borland 公司所开发的 C++Builder 以其强大的 RAD (Rapid Application Development, 快速应用程序开发) 功能而独树一帜。它能够节省开发人员设计程序界面和实现底层支持的时间，可以将更多的精力花在程序的逻辑设计上。目前最新的产品是 C++Builder 6，它不但继承了以前版本的所有优点，而且还扩展并增强了一系列新的属性：如网络与数据库技术、Debug 调试功能等。

在 C++Builder 中所使用的是 VCL (Visual Component Library) 类库。VCL 类库使用简单，功能强大，而且充分体现了面向对象的特点。另外，VCL 类库是用 Pascal 语言编写的，所以在 C++Builder 中就同时具有了 C++ 和 Pascal 的强大功能。

本书针对 C++Builder 6 中的应用程序开发，共设计了 100 个实例，按照功能可将其分成 11 个部分，具体安排如下：

在第一和第二部分中，介绍了用 C++Builder 开发应用程序的一般步骤和一些常用组件的使用方法。

在第三部分中，详细地介绍了 C++Builder 中的通用对话框组件和消息对话框的使用。

在第四和第五部分中，介绍了在 C++Builder 中绘图和一些简单图像处理的方法。

在第六和第七部分中，着重介绍了 Windows API 函数的使用方法，并用它来获取和设置文件、系统和硬件的各种信息。

第八部分是多媒体编程部分，主要介绍了如何播放几种常见的多媒体文件。

第九部分是网络编程部分，介绍了几种常用的与网络相关的组件。

第十部分是数据库编程部分，简要介绍了在用 C++Builder 进行数据库编程时所使用的一般方法。

在最后一个部分中，介绍了在 C++Builder 中的一些其他常用的应用技巧。

本书从基本知识开始讲解，逐步深入，覆盖了用 C++Builder 进行程序设计的大部分领域，以实用性见长。

本书主要由清华大学魏俊鹏编写，另外，于秋生、周家新、张俊岭、李志伟、李子婷、张海霞、李龙、吴金华、管佩森、侯高岚、许社村、赵磊、张素卿、周松建、仇谷峰、沈应逵、吴建文、徐日强、于华芸、俞雷、钱秀槟和张龙卿等也参加了部分内容的编写及素材整理工作，在此一并表示感谢。

由于编写的时间仓促，再加上编者的个人能力有限，书中难免会出现一些不足或纰漏之处。竭诚欢迎广大读者对本书提出批评和建议。

编 者
2004 年 5 月

目 录

第一部分 基本组件应用

实例 1	Hello World.....	3
实例 2	动态创建窗体.....	6
实例 3	窗体设计	10
实例 4	按钮设计	14
实例 5	单选与复选按钮.....	18
实例 6	文本框与格式文本框.....	21
实例 7	备忘录组件——C++Builder 中的记事本	24
实例 8	多文本格式组件——C++Builder 中的写字板	27
实例 9	列表框	31
实例 10	组合框	34

第二部分 功能组件应用

实例 11	菜单设计.....	39
实例 12	控制菜单	42
实例 13	工具栏和状态栏.....	45
实例 14	单选按钮组设计.....	49
实例 15	复选列表框设计.....	52
实例 16	多页界面设计（一）	56
实例 17	多页界面设计（二）	59
实例 18	使用栅格	63
实例 19	定时器应用	67
实例 20	树状视图组件.....	70
实例 21	列表视图组件.....	73

第三部分 对话框设计

实例 22	打开与保存对话框.....	79
实例 23	字体与颜色对话框.....	83
实例 24	打印与打印设置对话框.....	87
实例 25	查找与替换对话框.....	90
实例 26	输入对话框	93

实例 27	路径选择对话框.....	96
实例 28	消息对话框.....	99

第四部分 基本绘图

实例 29	绘制直线.....	105
实例 30	二维绘图.....	108
实例 31	绘制多边形.....	112
实例 32	绘制文本.....	115
实例 33	画笔与画刷.....	117
实例 34	OpenGL 绘图.....	121

第五部分 图像处理

实例 35	滚动与推拉效果.....	127
实例 36	百叶窗效果.....	131
实例 37	雨滴效果.....	133
实例 38	爆炸效果.....	136
实例 39	旋转及翻转图像.....	139
实例 40	图像淡入淡出.....	142
实例 41	柔化图像.....	146
实例 42	锐化图像.....	150
实例 43	中值滤波.....	153
实例 44	边缘增强.....	157
实例 45	边缘检测.....	159
实例 46	扩散图像.....	162
实例 47	图像格式转化.....	164

第六部分 文件及系统操作

实例 48	获取文件日期信息.....	171
实例 49	复制文件.....	174
实例 50	查看临时文件.....	177
实例 51	图像浏览器.....	180
实例 52	按键监测.....	183
实例 53	鼠标拖放操作.....	186
实例 54	更改应用程序图标.....	188
实例 55	运行外部应用程序.....	191
实例 56	隐藏桌面图标及任务栏.....	194
实例 57	获取系统信息.....	197

实例 58	获取及设置系统时间.....	201
实例 59	操作注册表.....	205
实例 60	关闭系统.....	209

第七部分 硬件设备检测

实例 61	获取 CPU 类型.....	215
实例 62	检测内存空间.....	218
实例 63	获取磁盘类型.....	221
实例 64	检测硬盘空间.....	223
实例 65	控制光驱.....	227
实例 66	检测显示器和打印机.....	231
实例 67	设置鼠标参数.....	234

第八部分 多媒体编程

实例 68	检测声卡.....	239
实例 69	播放视频文件.....	243
实例 70	CD 播放器.....	246
实例 71	MP3 播放器.....	249
实例 72	播放动画.....	253

第九部分 网络编程

实例 73	获取本机的 IP 地址.....	259
实例 74	检测网络连通性.....	262
实例 75	ECHO 操作.....	266
实例 76	获取远程主机日期时间.....	269
实例 77	FINGER 操作.....	272
实例 78	发送与接收消息.....	275
实例 79	发送与接收数据.....	278
实例 80	FTP 下载工具.....	282
实例 81	接收电子邮件.....	286
实例 82	发送电子邮件.....	290
实例 83	Web 浏览器.....	294

第十部分 数据库编程

实例 84	建立数据库.....	299
实例 85	查询数据库.....	302
实例 86	SQL 查询.....	306




C++ Builder 6 **实用编程100例**

实例 87	修改数据库	309
实例 88	使用数据模块.....	312
实例 89	主从表连接	315
实例 90	查看数据字典.....	318
实例 91	设计报表	321
实例 92	使用决策支持组件.....	324
实例 93	ADO 编程.....	327

第十一部分 其他应用技巧

实例 94	运行 C 程序.....	333
实例 95	运行 C++程序	336
实例 96	制作日历	340
实例 97	多文档界面应用程序.....	342
实例 98	多线程程序设计.....	346
实例 99	异常处理	350
实例 100	使用 COM 对象编程.....	353
附录 A	ASCII 码对照表	357
附录 B	虚拟键值对照表.....	359



第一部分 基本组件应用

C++Builder 是一款基于 Windows 操作系统的面向对象的程序设计开发环境。它具有强大的可视化开发能力，含有众多的可以直接使用的 VCL 组件。所以说，使用 C++Builder 进行程序设计的关键在于各种组件的使用。

在这一部分中向大家介绍其中最基本的一些组件及其使用方法，如文本框、标签和按钮等。用 C++Builder 来进行程序设计就必须熟练掌握这些组件的使用方法，这也是设计一个大的应用程序所必备的基础。

在这一部分中，将向大家介绍以下几个实例：

- ◆ Hello World
- ◆ 动态创建窗体
- ◆ 窗体设计
- ◆ 按钮设计
- ◆ 单选与复选按钮
- ◆ 文本框与格式文本框
- ◆ 备忘录组件
- ◆ 多文本格式组件
- ◆ 列表框
- ◆ 组合框



实例 1

Hello World

实例说明

Hello World, 本书就从这里开始向大家介绍用 C++Builder 来开发应用程序的一般步骤。在下面的这个例子中, 设计了一个用于显示字符的小程序。其程序运行的结果如图 1-1 与图 1-2 所示。



图 1-1 显示字符 (一)

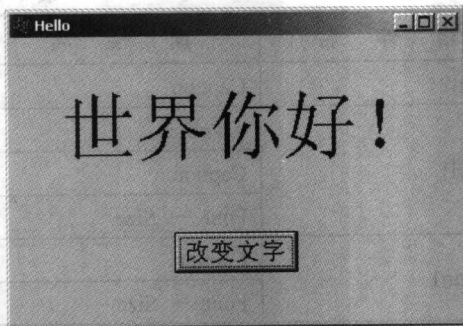


图 1-2 显示字符 (二)

技术要点

- 新建一个应用程序, 在窗体上放置组件并用 Object Inspector (对象观察器) 来设定初始值;
- 调用按钮 TButton 组件的 OnClick 事件并向其中添加事件响应代码;
- 使用标签 TLabel 组件来显示字符串信息;
- 保存源文件和整个工程文件, 运行及调试程序。

实现步骤

1. 新建项目

启动 C++Builder, 单击菜单“File | New | Application”, 新建一个 Application 项目。屏幕上就会出现一个窗体 Form1, 改变窗体大小, 调整其在屏幕上的位置。

2. 设计程序界面

从 Component Palette 工具栏的 Standard 工具箱中选择并向窗体中添加一个 TLabel 组件 **A** 和一个 TButton 组件 **B**, 并适当调整其大小。最后窗体的整体布局如图 1-3 所示。



图 1-3 程序界面设计

3. 设置组件属性

按【F11】键调出 Object Inspector 对话框，选择窗体以及各个组件，依次设置其关键属性，如表 1-1 所示。

表 1-1 组件属性设置

组 件 名	属 性 名	属 性 意 义	新 内 容
Form1	Caption	标题名	Hello
Labell	AutoSize	自动大小	False
	Caption	标题名	Hello, World!
	Font - Size	字体大小	36
Button1	Caption	标题名	改变文字
	Font - Size	字体大小	12

4. 添加代码

双击“改变文字”按钮，调用该按钮的 OnClick 事件，并向其中添加如下所示的程序代码：

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    if(Labell->Caption=="Hello,World!")
        Labell->Caption="世界你好! ";
    else
        Labell->Caption="Hello,World!";
}
```

然后，看一下自动生成的头文件部分的程序代码，它包含了一些必备的 HPP 文件、各个组件的定义和一些组件的事件响应函数的声明。

```
#define Unit1H
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
class TForm1 : public TForm
{
    _published: // IDE-managed Components
```


```

TLabel *Label1;
TButton *Button1;
void _fastcall Button1Click(TObject *Sender);
private: // User declarations
public: // User declarations
    _fastcall TForm1(TComponent* Owner);
};
extern PACKAGE TForm1 *Form1;
#endif

```

5. 保存及运行程序

程序设计完毕之后，就要保存整个工程文件了。单击“File | Save All”菜单，从弹出的“Save Unit1 As”保存对话框中选取合适的保存路径，默认采用 C++Builder unit (*.cpp)保存类型，输入存储文件名，单击“保存”按钮；接着将弹出“Save Project1 As”保存对话框，选取合适的保存路径，默认采用 C++Builder project (*.bpr)保存类型，输入适当的文件名，单击“保存”按钮即完成整个应用程序项目的保存。

运行及调试程序，可以通过按下键盘上的【F9】键或单击 Debug 工具栏的  图标来完成。



归纳注释

在这个例子中，显示字符串时使用的是标签 TLabel 组件，该组件所显示的信息是它的 Caption 属性项的值，程序利用按钮 TButton 组件的 OnClick 事件来实现所显示的中英文字符串之间的变换。

通过上面的这个例子，我们可以看出，使用 C++Builder 能够既方便又快捷地开发一个应用程序。接下来，给大家介绍一下在整个工程项目目录下各个文件的作用。

Project1.bpr ——工程文件，它包含了工程所需的选项与规则；

Project1.exe ——调试运行时所生成的应用程序文件；

Project1.res ——工程资源文件，它包含了应用程序的图标、版本号等信息；

Project1.cpp ——工程主源文件，程序运行时自动生成的文件，它包含了应用程序的起始（入口）代码及各种定义的宏；

Unit1.cpp ——窗体主文件，需要编写相应程序代码的文件；

Unit1.h ——窗体声明头文件，包含一些 HPP 文件、组件定义以及部分事件响应函数的声明，它也是程序自动生成的；

Unit1.dfm ——保存窗体及各个组件属性设置值的文件。

而其他的一些，如*.obj、*.tds 文件及扩展名以“~”开头与窗体或工程文件同名的文件等都是临时文件，这些文件是在程序调试阶段所生成的中间文件或者是临时的备份文件。

实例 2

动态创建窗体

实例说明

在开发应用程序的图形界面时，可能会用到多于一个的窗体界面。在默认的情况下，在程序开始运行时，程序用到的所有窗体都是被自动创建的，这样在程序运行时会耗费系统大量的内存，严重的时候则导致程序无法正常运行。在程序设计时，通常用动态创建窗体来解决这一问题。所谓的动态创建窗体，就是在当用到窗体的时候才创建，而在窗体用完之后马上释放这部分内存，这样程序就不会占用过多的内存。

下面我们就编写一个动态创建窗体的例子，其运行结果的效果图如图 2-1 所示。

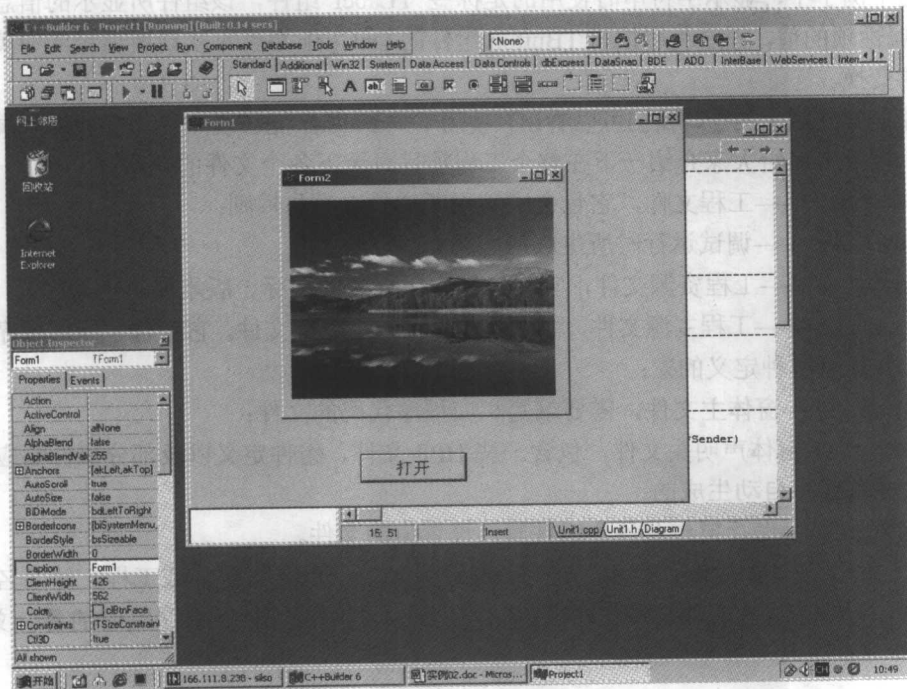



图 2-1 动态创建窗体

技术要点

- 新窗体的建立及窗体之间的动态调用；
- 窗体 TForm 组件的基本响应事件。

STEP 实现步骤

1. 新建项目

启动 C++Builder，单击 Standard 工具栏的  图标，弹出 New Items 对话框，选择 New 选项卡中的 Application 项，新建一个项目，如图 2-2 所示。

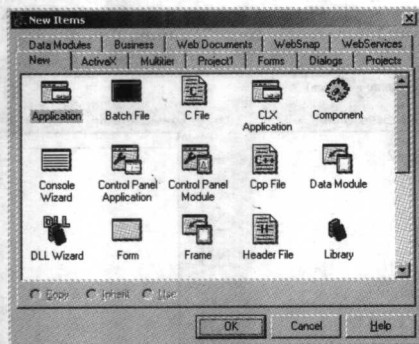



图 2-2 新建应用程序

2. 设计程序界面

在第一个窗体上放置从 Component Palette 工具栏中 Standard 工具箱选出的一个 TButton 组件。

然后，通过单击菜单“File | New | Form”来新建一个窗体，调整窗体 Form2 的大小及其在屏幕上的位置，最后在工具栏的 Additional 工具箱中选择一个 TImage  组件放入到 Form2 中。

3. 设置组件属性

两个窗体中组件的关键属性设置如表 2-1 所示。

表 2-1 组件属性设置

窗体名	组件名	属性名	属性意义	新内容
Form1	Button1	Caption	标题名	打开
Form2	Image1	AutoSize	自动大小	true

其后就是设置 Image1 的 PicTure 属性，单击该属性项右端的“...”按钮，打开 Picture Editor 对话框，添加 image 文件夹下的一张图片，如图 2-3 所示。

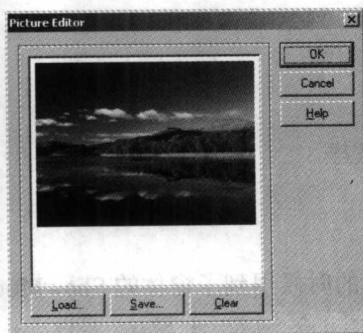


图 2-3 设置 Image1 的 PicTure 属性

4. 添加代码

在程序中要用到两个窗体，两个窗体默认都是在程序开始运行时建立，现在要将窗体 Form2 设置为动态创建的窗体。单击菜单“Project | Options...”或者按下【Shift+Ctrl++F11】组合键，打开 Project Options for Project1.exe 对话框，切换到 Forms 选项卡，如图 2-4 所示。

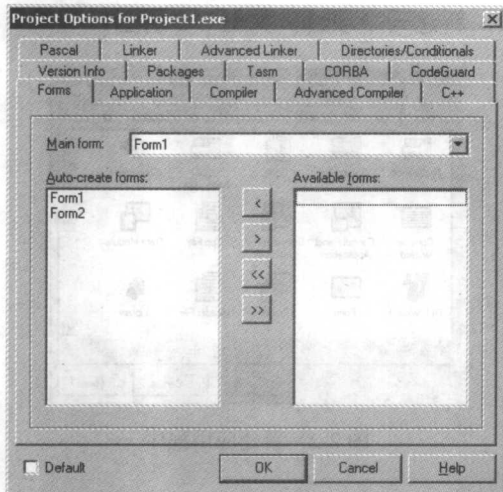


图 2-4 设置窗体的自动创建功能

在 Main Form 的下拉列表框中选择程序运行时的主窗体。然后，在 Auto-create forms 列表框中列出了程序初始运行时所自动创建的窗体 Form1 与 Form2。接着在选中 Form2 之后，单击“>”按钮，Form2 将会转移到 Available forms 列表框中。而在 Available forms 列表框中所显示的窗体是不会被程序自动创建的。

在这一个程序中一共有两个窗体，而要在 Form1 中调用 Form2 窗体，所以在 Form1 窗体声明头文件 Unit1.h 中添加如下代码：`#include "Unit2.h"`。另一种方法是单击菜单“File | Include Unit Hdr...”或者按下【Alt+F11】组合键，这时会弹出 Use Unit 对话框，从中选择所要包含的文件，再单击 OK 按钮，这样在代码中就会自动添加`#include"Unit2.h"`语句。

最后，双击窗体 Form1 中的“打开”按钮，并在它的 OnClick 事件中添加如下所示的代码：

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    TForm2* form2=new TForm2(this);
    form2->ShowModal();
    delete form2;
}
```

5. 保存工程文件，运行程序。



归纳注释

在上面的例子中，显示窗体的时候用到了窗体的 ShowModal 方法，另外显示窗体的时候还可以用窗体的 Show 方法，二者的区别如下：Show 是以非模态方式显示窗体，即当窗体显示时可以访问应用程序中的其他窗体；ShowModal 则是以模态方式显示窗体，则只有在该窗

体被关闭后才可以访问应用程序中的其他窗体。

接下来，列出一些在窗体 `TForm` 组件中所常用的事件，在今后的程序界面设计中也会常常用到这些事件的。

OnCreate ——在创建窗体时发生的事件，它经常用于对程序进行初始化；

OnShow ——窗体显示时发生的事件；

OnActive ——当窗体变为活动窗体时的事件；

OnPaint ——当窗体重画时发生的事件，在窗体发生变化时也会发生这一事件；

OnHide ——当窗体隐藏时发生的事件；

OnClose ——当窗体关闭时发生的事件。

最后，对窗体的 `OnClose` 事件响应函数进行一下说明，在这个函数中只有一个参数 `Action`，这个参数用来确定窗体关闭的方式。该参数的取值是一个枚举类型：

```
enum TCloseAction { caNone, caHide, caFree, caMinimize };
```

其中各个取值所代表意义如下：`caNone` 表示不允许关闭窗体；`caHide` 表示隐藏窗体但不关闭；`caFree` 表示关闭窗体并释放窗体所占内存；`caMinimize` 表示将窗体最小化但不关闭。