C++ Builder 6

彻底研究

陈灿煌 著

内附范例光盘

■ C++ Builder 6是Borland公司的最新版本应用开发工具

本书从基础开发进行介绍，

由浅入深地向读者展示C++ Builder 6的强大功能

■ 本书详细地介绍了C++ Builder 6在数据库开发中的方法和技巧

■ 结合实际案例

讲解如何使用C++ Builder 6构建传统的客户/服务器应用程序以及如何快速开发Web应用程序

■ 本书内容丰富、资料全面，适合中高级读者或开发人员使用

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

C++ Builder 6 彻底研究

陈灿煌 著

中国铁道出版社
2003·北京

（京）新登字 063 号

北京市版权局著作合同登记号：01-2002-4463 号

版 权 声 明

图书在版编目（CIP）数据

C++ Builder 6 彻底研究/陈灿煌著. —北京：中国铁道出版社，2002.11

ISBN 7-113-05019-0

Ⅰ.C… Ⅱ.陈… Ⅲ.C 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 092047 号

书　　名：C++ Builder 6 彻底研究
作　　者：陈灿煌
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
封面设计：孙天昭
印　　刷：河北省遵化市胶印厂
开　　本：787×1092　1/16　印张：51.75　字数：1267 千
版　　本：2003 年 2 月第 1 版　　2003 年 2 月第 1 次印刷
印　　数：1～4000 册
书　　号：ISBN 7-113-05019-0/TP·821
定　　价：84.00 元

出 版 说 明

C++ Builder 6 作为一个功能强大且易于上手的程序开发工具，得到越来越多的编程人员的青睐。用 C++ Builder 6 可以轻易地通过拖放组件来组建自己的应用程序，或一些编程人员调用 Windows API 编写 DLL 程序等开发出专业级别的高级软件。本书覆盖面甚广，共分为四大篇：Windows 程序设计、数据库程序设计、互联网程序设计及高级程序设计，内容独到经典，是一本学习和处理 C++ Builder 6 问题的参考好书。

本书由台湾博硕文化股份有限公司提供版权，经中国铁道出版社计算机图书中心审选，由彭木根、贾英茂、游广志、刘宇、朱远波、张新东、张琦、王东、刘小娜、陈贤淑及廖康良等同志参与了本书的整稿及编排工作。

2003 年 1 月

目　录

第一部分　Windows 程序设计

第二部分 数据库程序设计

第四部分 高级程序设计

第一部分 Windows程序设计

第 1 章

认识 C++ 语言

- ◆你的第一个 C++程序
- ◆基本数据类型（Basic Data Type）
- ◆运算符（Operators）
- ◆数组（Array）
- ◆指针（Pointer）
- ◆条件语句（Conditional Statement）
- ◆循环（Loop）
- ◆函数（Function）

1-1 你的第一个C++程序

本书的书名为《C++ Builder 6 彻底研究》，所以本书的重点还是在C++ Builder这个开发工具，本书会深入探讨如何彻底利用C++ Builder这套工具来开发各种应用程序。不过，在开始深入了解C++ Builder这个好用的C++开发工具之前，不可避免的，我们还是先来回顾一下C++本身语法的一些基本概念及应用。

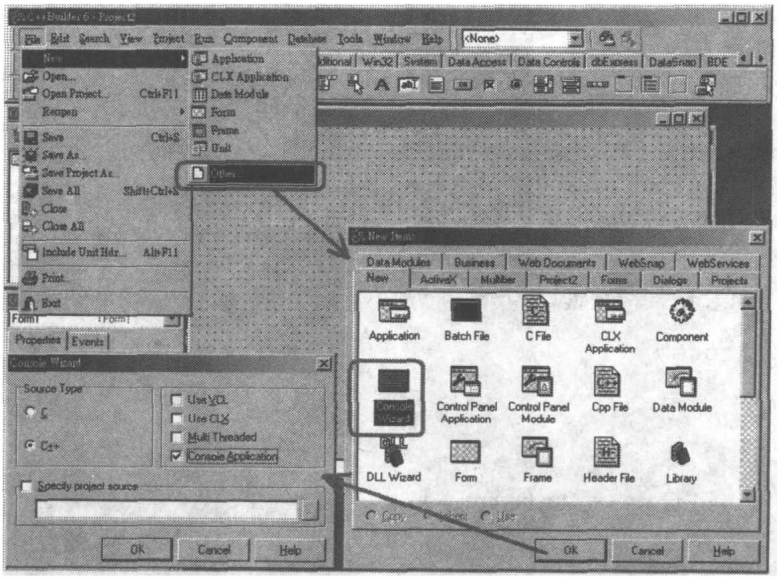

图 1-1 C++ Builder提供的Console Wizard

由于C++ Builder本身是一套Windows程序的开发工具，所以它所提供的Framework主要还是偏重在图形界面的Windows程序，不过它仍旧提供文字的console模式，由于本章的目的主要在于让读者了解C++语法的应用，所以不必大费周章地采用图形界面，因此本章的范例程序都是console模式，在C++ Builder中要产生console模式的程序结构很简单，只要点击选中[File]→[New]→[Other]打开Object Repository，选择[New]→[Console Wizard]即可启动向导程序，由图1-1可以看出此向导有许多选项，请你点击选中C++及Console Application两项即可，最后点击选中[OK]，C++ Builder将会自动帮你产生Console程序，如下所示：

```
//---------------------------------------------------------------------
#pragma hdrstop
//---------------------------------------------------------------------
#pragma argsused
int main(int argc, char* argv[])
{
        return 0;
}
//---------------------------------------------------------------------
```

基本上，这个程序只是个空框架，没做任何事，此时你就可以在这个空框架上开始做文

章，假设我们希望能显示一段文字，那么你就可以把程序改写成：

```
//---------------------------------------------------------------------
// C++ Builder 彻底研究范例
//---------------------------------------------------------------------
#include <iostream.h>
#include <conio.h>

#pragma hdrstop
//---------------------------------------------------------------------
#pragma argsused
int main(int argc, char* argv[])
{
    /* 第一个 C++ 程序 */
     cout<<"Hello, C++ World";
     getchar();
    return 0;
}
//---------------------------------------------------------------------
```

如果你运行上面这个程序后，将会得到如图 1-2 所显示的结果。Windows 会打开一个 DOS 窗口，此程序将“Hello, C++ World”的信息显示在 DOS 窗口里，接下来，你只要按一下[Enter]键，程序就会结束运行。接着，让我们仔细的来看看此范例程序中的每一行代码所代表的意义。

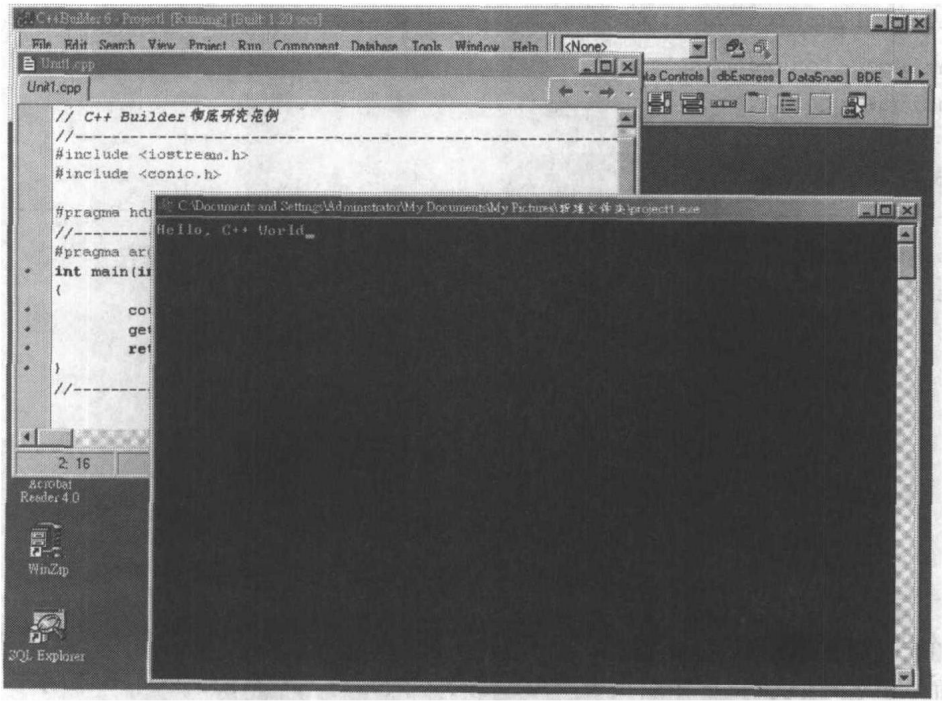

图 1-2　第一个 C++程序的运行结果（console）

首先看到的是含有#include 指令的#include <iostream.h>及#include <conio.h>，它们的作用在于引入头文件（header file）。当运行编译时，#include 指令会指示 C++ Builder 编译器把括号内的头文件 include 到我们的代码中。一般的头文件均是以.h 或.hpp 作为文件的扩展文件名。多数的头文件是编译器的厂商所提供，有的则是程序设计师预先设计好供人们（或自己）使用。依照多数程序设计习惯，我们通常会使用#include <文件名>来引入编译器厂商内置的头文件（系统默认路径下），至于自己所设计的头文件，则使用#include "文件名"（用户程序路径下）。

至于程序中使用双斜线//及/* */符号来标识的部分，代表注释的意思，其后同一行的文字仅作为注释用，与程序内容无关；另外一个用来注释的符号是/*，它的用法与//略有不同，/*必须与*/成对出现，注释文字则放在 /* 和 */之间，它的范围不限于同一行。

C++程序由 main()开始运行，这是任何一个 C++程序的开始运行的地方或称为入口点（Entry point）。在这程序中，跟随着关键字 main 的小括号 “(”及“)” 之间并没有任何的参数（argument），这表示程序运行时我们不需要由程序外部输入任何参数值。而大括号“{”及“}”则代表 main()程序的开始与结束。

cout 指令则是负责把字符串输出到屏幕上，此指令的定义放在 iostream.h 头文件内，这也是我们为什么要把 iostream.h 文件 include 进来的原因。而由于运行环境是在 Windows 下，所以如果 console 程序只显示字符串，那么运行结果将一闪而过，你必须加上 getchar()函数，让程序运行完显示字符串后停下来等待输入，才能清楚看到程序的运行结果。而 getchar()函数则是被定义在 conio.h 头文件内。

由于本章的范例程序只会用到 console 的输入及输出，所以下面再介绍一个包含 console 输入及输出指令的范例程序：

```
//---------------------------------------------------------------------
#include <iostream.h>
#include <conio.h>

#pragma hdrstop
//---------------------------------------------------------------------
#pragma argsused
int main(int argc, char* argv[])
{
        int  n;
        cout<<"输入一个整数 :";
        cin>>n;
        cout<<"\n 你输入的整数是 : "<<n<<endl;
        cin>>n;
}
//---------------------------------------------------------------------
```

这个程序中，首先定义一个整数名称为 n，接着使用 cout 指令输出字符串提示输入一个整数，而 cin 指令则会让程序接收用户由键盘输入的整数，并且把值放入 n 变量中。运行结果如图 1-3 所示。

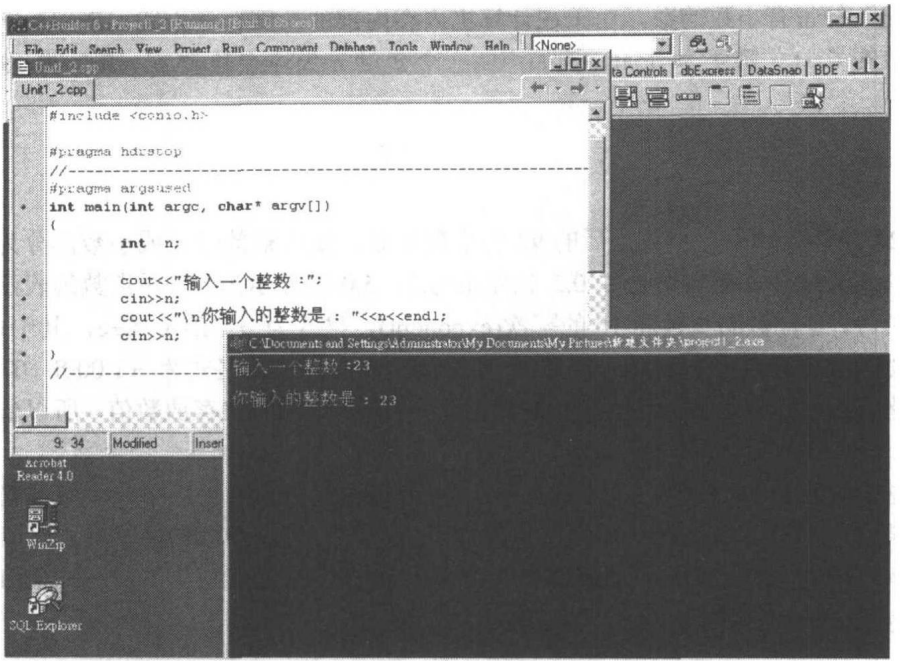

图 1-3　console 输入及输出指令的程序运行结果

到此你应该了解如何打开及运行一个 C++ Console 程序，也大略了解 C++最基本的结构了吧，接下来我们再来看看 C++的基本语法。

1-2　基本数据类型(Basic Data Type)

变量是程序中用来保存数据的内存块的标识名称(Identifier)，指定一个标识名称的 C++命令称为变量声明(Variable Declaration)，每一个变量声明的指令至少要包含变量的数据类型(data type)及变量名称(name)，变量的数据类型用来决定将存放在内存的变量值的类型（如 int、char 等），变量名称则是作为内存位置的标识来用。

你可以使用任意的字符和数字及下划线（ _ ）的组合作为变量名称，但是有两个限制，一个是不得使用 C++的保留字作为变量名称，另一个则是不得在名称中使用特殊字符（如@、$和空格键）或算术运算符符号(+,−,*,/,%)。在 C++中，变量名称的大小写是有区别的，例如 name、Name、NAME 代表三个不同的变量。

当我们要将一个数据放入变量所指的内存中，必须使用 Assignment 的指令，上述的 Assignment 指令就是"="符号，要特别注意的是"="符号在 C++里是被用作 Assignment，而不是相等（Equality）的意思。这是许多初学者常混淆的地方。

C++提供有不少的基本数据类型，其中最常使用到的四个基本的数据类型分别是整数(Integer)、浮点数(Floating Point)、双精度浮点数(Double)和字符(Character)。

整数类型包括所有自然数（int、正数、负数及零）、没有符号的整数(unsigned int)、短整数(short)及长整数(long)，用来定义整数类型的关键字为 int。在一个 32 位的操作系统（如 Unix、Linux 或 Windows 2000）中，一个整数的变量使用 32 位(4 bytes)的存储空间，其值的范围是从 2,147,483,647 到-2,147,483,648。