



康向东 汪洁 张骁兵 肖欣 等编著

C++Builder 新起点

✦ 4.0 实用教程



机械工业出版社
China Machine Press

本书通过实例,讲解了C++Builder 4.0的基础知识、编程方法和技巧,全书通过介绍C++Builder 4.0新特征及VCL组件的一些基础知识,以及实例讲解了C++Builder 4.0的编程技巧、异常处理机制及应用等。

本书配有大量实例,讲解清晰,叙述深入浅出,对广大编程人员尽快掌握C++Builder 4.0的强大功能提供了有益的帮助。

本书由机械工业出版社出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

版权所有,翻印必究。

图书在版编目(CIP)数据

C++Builder 新起点——4.0 实用教程 / 康向东等编著 -北京 机械工业出版社, 2000.2

ISBN 7-111-07805-5

I C… II 康… III. C语言-程序设计 IV.TP 312

中国版本图书馆CIP数据核字(1999)第57811号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:郭东青

北京市密云县印刷厂印刷·新华书店北京发行所发行

2000年2月第1版第1次印刷

787mm×1092mm 1/16·24 5印张

印数:0 001-5 000册

定价 39.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前 言

C++Builder是Borland 公司（现已更名为Inprise）继Delphi后推出的基于C语言的快速应用程序开发(Rapid Application Development, RAD)工具。它同时具有C++语言的灵活和可视化编程的便利，功能丰富强大，界面友好，使用方便，是C/C++程序员首选的编程工具。C++Builder 4.0是其最新的版本。与以前的版本相比，它在ActiveX、数据库等方面增加了许多新的功能，同时也对编程环境做了进一步地完善，使用户在使用时更加得心应手。笔者曾长期从事C/C++程序的开发工作，使用过从早期的Turbo C、MS C到Visual C++等多种编程工具。但当笔者初次接触C++Builder时，便被其轻松方便的编程方式和丰富强大的功能所吸引。就笔者认为，C++Builder可以说是Windows 下功能最为强大的C/C++ 语言编程工具。笔者编写这本书，主要是将自己使用C++Builder 4.0的一些经验与大家分享，书中有理论知识，但更多地提供了综合性很强的实例，并且给出了所有程序的清单。希望能够对广大编程人员，尤其是C++Builder的爱好者尽快地掌握这一功能强大的编程工具有所帮助。本书从结构上来说分为两大部分，第1章和第2章概括地介绍了C++Builder 4.0的新特征及VCL组件的一些基础知识。从第3章起开始介绍利用C++Builder 4.0编写各类应用程序的方法，第3章通过一个简单的例子介绍了编程的基本方法；第4章简要介绍C++Builder 4.0中多文档程序的编写；第5章深入讨论了界面设计的方法；第6章介绍图形图像在C++Builder 4.0程序中的应用；第7章介绍多媒体应用程序的编制；第8章讨论数据库应用程序的开发；第9章较为详细地介绍了C++Builder 4.0中网络组件的特性及网络应用程序的开发；第10章讨论了在C++Builder 4.0中如何创建用户自己的组件；第11章简要地阐述了C++Builder 4.0中异常处理的机制及应用。

参加本书编写工作的主要有康向东、汪浩、张骁兵、肖欣，此外，顾郁莲、王睿、蔡汇锦、赵文、王双、涂玲、施杰、周锐、于朝晖、王宝柱、李全玉、黄敏燕、王联庆、丁戈、易延波、文革、王跃宇、刘华松、郑联语等，也参加了编写工作，由于笔者水平有限，且时间仓促，书中难免存在一些错误和不足，热诚希望广大读者朋友在阅读和使用本书的过程中提出宝贵的意见，也真诚希望您能尽快成为C++Builder 4.0的专家。

目 录

前言

第1章 C++Builder 4.0的新特征1

1.1 C++Builder 4.0的新特性及功能1

1.1.1 新的工程管理器1

1.1.2 类浏览器2

1.1.3 新的编译器2

1.1.4 新的运行库3

1.1.5 支持MTS4

1.1.6 提供扩展的Active X支持4

1.1.7 与微软产品的兼容性4

1.1.8 包管理4

1.1.9 增强的数据库功能4

1.1.10 增强的调试功能5

1.1.11 其他的新增功能5

1.2 C++Builder 4.0集成开发环境5

1.2.1 开发环境概述5

1.2.2 主菜单栏6

1.2.3 工具栏6

1.2.4 组件面板7

1.2.5 对象观察器13

1.2.6 窗体编辑器13

1.2.7 代码编辑器15

1.3 工程和工程组20

第2章 VCL组件24

2.1 组件概述24

2.1.1 VCL组件的基本概念及分类24

2.1.2 VCL的结构25

2.2 常用组件32

2.2.1 标准组件32

2.2.2 附加组件36

2.2.3 Win32组件37

第3章 如何开始应用程序设计42

3.1 从一个简单实例的分析开始42

3.1.1 明确程序的目的42

3.1.2 建立应用程序整体框架42

3.1.3 建立一个工程43

3.1.4 进一步了解工程43

3.2 定制应用程序界面45

3.2.1 设计应用程序主窗口45

3.2.2 菜单的设计46

3.2.3 增强的菜单和工具栏47

3.2.4 加入表格及状态栏48

3.2.5 加入图像与说明49

3.3 编制程序代码51

第4章 多文档应用程序实例54

4.1 单文档和多文档54

4.1.1 Windows应用程序的基本类型54

4.1.2 单文档和多文档用户界面54

4.1.3 打开单文档和多文档用户界面应用程序55

4.2 多文档用户界面实例分析55

4.2.1 明确应用程序目的55

4.2.2 建立应用程序整体框架55

4.3 设计应用程序界面56

4.3.1 创建多文档用户界面应用程序56

4.3.2 程序界面设计59

4.4 编制代码实现应用功能62

4.4.1 实现“保存”命令和工具栏“保存”按钮的功能62

4.4.2 打印功能的实现64

4.5 实例程序清单65

第5章 应用程序界面的深入设计68

5.1 界面设计的基本类68

5.1.1 TApplicaton类68

5.1.2 TScreen类77

5.1.3 TForm类85

5.2 深入讨论应用程序界面设计88

5.2.1 窗体设计88

5.2.2 创建和管理菜单89

5.2.3 工具栏及CoolBar92

5.2.4 应用ActionList组件93

5.2.5 应用其他组件96

5.3 应用程序界面设计示例	97	7.4 多媒体程序编写实例	156
5.3.1 目的分析及整体设计	97	7.4.1 明确程序的功能	156
5.3.2 界面设计	97	7.4.2 应用程序的整体框架	157
5.4 实例程序清单	106	7.4.3 应用程序的界面设计	157
第6章 引入图形图像	115	7.4.4 组件响应事件程序代码	161
6.1 常用的图形类	115	7.5 小结	179
6.1.1 TCanvas类	115	7.6 实例程序清单	179
6.1.2 TPen类	116	第8章 数据库应用程序	190
6.1.3 TBrush类	117	8.1 数据库应用程序基本结构	190
6.1.4 常用类的使用	117	8.1.1 利用BDE使用数据库	190
6.2 常用的图像类	118	8.1.2 数据库应用程序基本结构	190
6.2.1 Bitmap对象	118	8.1.3 简单的数据库应用程序	192
6.2.2 Icon对象	118	8.2 设计数据库应用程序界面	192
6.2.3 Metafile 对象	118	8.2.1 数据控制组件	192
6.2.4 Clipboard 对象	118	8.2.2 显示单一字段	193
6.2.5 Picture 对象	118	8.2.3 显示多个字段	195
6.2.6 Image 对象	119	8.3 TDataSet与TTable的应用	200
6.3 画图的综合实例	119	8.3.1 TDataSet组件	200
6.3.1 明确程序的目的	119	8.3.2 应用TTable	204
6.3.2 程序的整体规划	119	8.4 使用字段对象	208
6.3.3 开始建立新程序	120	8.4.1 TField类及其派生类	208
6.3.4 程序的界面设计	120	8.4.2 动态字段对象	209
6.3.5 控件事件设计	123	8.4.3 创建固定字段对象	209
6.4 小结	136	8.4.4 设置新的固定字段对象	210
6.5 实例程序清单	137	8.4.5 使用计算字段	210
第7章 设计多媒体程序	146	8.4.6 使用查找字段	210
7.1 关于多媒体	146	8.5 使用TDataSource	211
7.1.1 文本	146	8.5.1 TDataSource的属性	211
7.1.2 图形图像	147	8.5.2 使用TDataSource的事件	211
7.1.3 音频	148	8.6 理解TSession类及TDataBase类	212
7.1.4 视频	149	8.6.1 TSession类与会话的管理	212
7.1.5 动画	149	8.6.2 TDatabase与数据库	215
7.2 组件TMediaPlayer及使用	150	8.7 创建数据库应用程序	218
7.2.1 TMediaPlayer组件的按钮	150	8.7.1 数据库应用程序类型	218
7.2.2 TMediaPlayer组件的属性	151	8.7.2 一或两层的基于BDE的数据库 应用程序	218
7.2.3 TMediaPlayer组件的方法	153	8.7.3 创建数据库应用程序实例	219
7.2.4 TMediaPlayer组件响应的事件	153	8.8 实例程序清单	233
7.3 组件TAnimate及使用	154	第9章 网络应用程序	242
7.3.1 TAnimate组件的属性	154	9.1 C++Builder 4.0网络组件概述	242
7.3.2 TAnimate组件的方法	155	9.1.1 NetMasters的网络组件	243
7.3.3 TAnimate组件响应的事件	155		

9.1.2 WebBroker的网络组件	246	10.3 创建第二个组件TMoveLabel: 会移动的Label组件	345
9.2 NetMasters网络组件的公共特性	248	10.3.1 分析TMoveLabel	345
9.2.1 TPowersock的属性	248	10.3.2 编写TMoveLabel组件	346
9.2.2 TPowerSock的方法	250	10.3.3 测试TMoveLabel组件	353
9.2.3 TPowerSock的事件	252	10.3.4 安装TMoveLabel组件	357
9.3 网络组件实例分析	253	10.4 创建第三个组件TLabelEdit: 利用 两个已有组件创建新的组件	360
9.3.1 TNMFTP组件	254	10.4.1 创建TLabelEdit组件	360
9.3.2 HTML组件	269	10.4.2 用组件模板创建TLabelEdit组件	365
9.3.3 TNMSMTP组件	278	10.5 小结	366
9.3.4 TNMPOP3组件	286	10.6 实例程序清单	366
9.3.5 TClientSocket组件	292	第11章 异常处理	376
9.3.6 TServerSocket组件	294	11.1 C++Builder 4.0异常处理概述	376
9.4 小结	301	11.2 异常处理的基本形式	376
9.5 实例程序清单	301	11.2.1 try...catch模块	376
第10章 创建组件	325	11.2.2 try....catch模块的使用	377
10.1 创建组件的基本过程	325	11.3 Exception类	379
10.1.1 创建新的TMyEdit组件	325	11.3.1 Exception类及其派生类	379
10.1.2 TMyEdit组件的说明	331	11.3.2 Exception类的基本属性与方法	380
10.2 组件的基本理论	333	11.3.3 使用Exception类自定义异常	380
10.2.1 组件的创建方法	333	11.4 实例程序清单	382
10.2.2 组件的属性	334		
10.2.3 组件的方法	339		
10.2.4 组件的事件	340		
10.2.5 创建组件的步骤	342		

第1章 C++Builder 4.0的新特征

1.1 C++Builder 4.0的新特性及功能

C++Builder 4.0相对于C++Builder 3.0来说,增加了许多特征,例如支持Windows 98等。下面将详细地介绍这些新的特性和功能。

1.1.1 新的工程管理器

对于一个包括许多文件的大型应用程序和一个包含多个工程的大型工程组来说,新的工程经理的下述新特性将使得应用程序的开发者对它们的管理明显简化:

- 工程激活
- 工程选择器。
- 拖拉和下拉拷贝。
- 拷贝和粘贴。
- 移去按钮。

1. 工程激活

当你对一个工程进行操作时,被激活的工程在工程管理器中以高亮大写形式出现。而当前激活的工程也会在工程选择器中出现。

2. 工程选择器

通过工程管理器顶端的工程选择器,可以轻而易举地在一个工程组中选择一个工程。一个下拉式的菜单将会列出当前工程组中所有的工程。而你所选中的工程将成为当前激活的工程。

3. 拖拉和下拉式拷贝

通过工程管理器,可以将任何文件夹中的文件的一个或多个条目拖到所选中的工程中来,具体步骤如下:

- 1) 激活或选中想要拷贝的工程。
- 2) 在任何文件夹中选中一个或多个你想要拷贝的选项。
- 3) 在工程管理器中将选中的文件夹中的条目拖到想要拷贝进的工程名上来(此时该工程名是高亮显示的)。
- 4) 当被问及是否拷贝该项或这些项时,点击“Yes”,这些条目就会被拷贝入当前激活的工程中。

5) 储存当前修改过的工程。

4. 拷贝和粘贴

通过工程管理器,可以将各种条目在工程和工程之间拷贝和转移,具体步骤如下:

- 1) 选中想要拷贝的工程条目。
- 2) 在工具条上选择下拉菜单“EditlCopy”(或“Ctrl+C”)。
- 3) 选中想要将复制的条目拷入的工程。
- 4) 通过工具条下拉菜单中“EditlPaste”(或“Ctrl+V”)将复制的条目粘贴到想拷入的工程中。

5) 储存修改后的工程。

5. 移去按钮

工程管理器工具条上的移去(Remove)按钮实现将选择的工程条目移走的功能。如果一个工程被选中后按下移去按钮,那么这个工程及其所包含的一切条目都会被从包含它的工程中删除。但只是从工程中消失,并不是被从硬盘中删除。

1.1.2 类浏览器

类浏览器使得你在各类和工程文件之间的浏览与切换更加容易,它包含一个树形结构来显示所有的类型、类、方法和变量等所有你定义的东西。

1.1.3 新的编译器

C++Builder 4.0中新的编译器包括以下新特性。

- 语法的改变。
- 模板。
- 位域。
- 新的C语言特性。
- 新的C++语言特性。
- 缺省队列。

1. 语法的改变

下述的一些对C++语言的改变,导致了用户应用程序在编译时与用以前版本的编译器编译有些不同:

- “# Pragma Pack”语法的改变。
- 增加了两个新的附注项(# Pragma),即“#Pragma alignment”和“#Pragma Checkoption”。
- 在一个模块中一个标准功能(function)的地址基本相同的。

2. 模板

(1) 模板生成语义的改变

在以前的版本中,一个程序所能用到的应用于所有模板实例的方法都会生成,而现在只生成以下部分:

- 那些真正有用的方法。
- 一个实例的虚拟方法。
- 明确例示的类的所有方法。

这些新的改进好处就在于命名为.OBJ文件、.LIB文件和.EXE的文件明显地减小了,当然,也可以使用“-Ja”开关来产生所有的方法。

(2) 类模板

类模板也叫做一般类或类产生器,它使你定义一个类定义的样板。

(3) 输入和输出模板

一个模板函数或模板类的说明必须具有相当的通用性和灵活的适用性,它不仅可以用在DLL文件中也可以用在EXE文件中。同样的模板说明也应该适于作入口和/或出口。为了充分通用,头文件模板的说明就不应使用“_export”或“_import”之类的前缀修饰。

下面的步骤示范了模板的输入和输出,源代码组织在三个文件中。应用头文件、代码在DLL中产生,一个DLL库被创建和链接到一个EXE文件。

(1) 可输入/可输出的模板说明

头文件包含所有模板类和模板函数的说明，在编译的时候可以通过定义适当的宏来示例模板的输入/输出版本。

(2) 编译所输出的模板

编写DLL的源代码，当编译时，这个DLL就具有模板可再用的输出代码。

(3) 使用入口模板

现在可编写一个使用模板的调用函数。这个文件将链接入DLL。在头文件中只有对象没有说明，但可以在主函数中被示例并使得编译器产生新的代码。新建的示例对象代码就被写入MAIN.OBJ文件。

(4) 模板体的解析

在老版本中编译器并不检查模板体的语法，除非这个模板被引用。而在C++Builder 4.0中模板体应该进行解析。

(5) 模板编译器的开关选项

“_Jg”系列开关控制着编译器如何生成模板实例。每一个模板实例都由编译器计数，而每一个模板实例都在编译器发现在某个开关值处某个实例出现后被影响并发生作用。

3. 位域

位域现在在C++Builder 4.0中可以被定义为任意整数大小的宽度。

4. 新的C语言特性

- 无类型的结构在C和C++中都支持。
- 无类型的联合先前只有C++支持，现在C也支持。
- 过去只在C++中用“inline”关键字表示的“inline”函数，现在在C中也可以用“_inline”来表示。

5. 新的C++语言特性

- 原来只有C语言支持的“_try”关键字，现在在C++中也支持。
- “try”仍然是一个只有C++支持的关键字。
- 无类型的结构在C和C++中都支持。

6. 缺省队列

编译器的缺省数据长度由过去C++Builder 3.0的4字节变为现在C++Builder 4.0的8字节。这种改变并不影响VCL代码，而VCL代码中仍沿用4字节的缺省数据长度。这样，在C++Builder 3.0的IDE和C++Builder 4.0的IDE之间就存在差异了。因为C++Builder 3.0的IDE的缺省数据位长度就是4字节，不需另外指出，而在C++Builder 4.0的IDE中就要为工程的生成文件指出是用8字节的缺省长度还是用4字节的缺省长度来编译和生成。当用C++Builder 4.0的IDE更新一个C++Builder 3.0的工程时，它就会使用默认的8字节缺省长度。

1.1.4 新的运行库

新的运行库的特点包括新的IEEE浮点、线程和字符串类型函数，更新的标准C++库和附加的MS头文件，具体包括：

- 新的运行库函数。
- 下面就是新的运行库函数的几个种类：
- 新的字符串函数(包括新的TCHAR.H函数)。
 - 新的IEEE浮点函数。

- 新的线程函数。
- 新的长字符和64位“Findxxx”函数。
- 新的多字节和多功能函数。
- 更新的标准模板库。
- 新的MS头文件。

在C++Builder 4.0中附加了400个Microsoft Windows头文件(在C++Builder 3.0中只有约200个), 绝大多数的新文件都是取自SDK平台。

- 增加了线程本地存储。

线程本地存储量由原来的23 ~ 24个扩展为没有限制。这种储存量的增加意味着动态运行库(RTL.DLL)在每一个进程中将只需使用一个线程本地存储槽。而不是以前的每一个进程模块就需一个线程本地存储槽。

1.1.5 支持MTS

MTS即Microsoft Transaction Server, 微软事务处理服务器。

1.1.6 提供扩展的Active X支持

C++Builder 4.0具有许多新的和数据的Active X特性, 包括有:

- 自控向导生成支持所有自控服务器对象事件的代码。
- 数据绑定Active X控制与VCL数据集通信。
- C++Builder产生支持自控对象的事件就像COM对象一样简单。
- 增加了客户服务支持, 生成的Active X类可以很好地合成进OLE服务器中。
- 支持MTS对象。

1.1.7 与微软产品的兼容性

C++Builder 4.0所具有的新的编译器和运行库, 使得编译和链接微软的Microsoft Visual C++5.0的工程文件.DSP、.DSW和源文件更加简单。C++Builder 4.0还包含一个可以将Microsoft Visual C++5.0的工程转换成对等的Borland C++Builder文件的工具——VCTOBPR, 要使用它只需在C++Builder工具菜单中选中或在命令行中键入VCTOBPR即可。

1.1.8 包管理

包是在C++Builder 3.0中引入的一个概念。在C++Builder 4.0中包就像C++Builder 3.0中一样用来安装Active X组件和其他组件。它使你可以编辑、编译和安装包工程文件。可以使你将程序模块独立出来、独立交付, 并在运行时共享代码, 它不仅极大缩小了EXE文件的大小, 方便其在Internet上的发布, 并可以缩短应用程序的编译时间。

1.1.9 增强的数据库功能

C++Builder 4.0的数据访问组件在得到增强后, 将使你可以在构造窗体或数据模块时调整数据模型, 新的“Borland数据库引擎(BDE)”支持访问97版和新的Oracle 8及SQL, 包括ADTS(抽象数据类型)、数据组和嵌入式的表(只限于企业版)。Visual Query Builder也已被一个新的智能的SQL Builder所取代(只限于企业版)。

1.1.10 增强的调试功能

在C++Builder 4.0中集成的调试器具有许多新特性。包括远程和多进程调试改进的断点设置、调试器细节菜单等。其中，远程调试支持用调试符号建立的EXE文件、DLL文件和包文件；多进程调试支持本地和远程的多个进程。多种求值计数程序使调试信息更准确，当调试一个包含C++和Pascal的DLL文件时，调试器会使用C++的本质程序计算处理C++代码，而用Pascal程序处理Pascal代码。另外新的调试器不支持分布式调试，而局部变量观察器将可以在调试模式下使你看到当前函数的局部变量的值的变化。

1.1.11 其他的新增功能

由于新增功能很多，不可能在此一一详细说明，现简列如下：

- 代码自动识别。
- 增强的VCL。
- VCL支持2000年问题。
- 动态数组。
- 支持CDRBA(Common Object Request Broker Architecture, 通用对象请求代理机构)。
- 增强的客户端数据集。
- 支持多层多级结构的应用。

1.2 C++Builder 4.0集成开发环境

1.2.1 开发环境概述

C++Builder 4.0的强大功能，集中体现在它的集成开发环境(IDE)中。当打开C++Builder 4.0时，一个主窗口就展现在了用户的面前，使用户就像坐在一个规划完善、工具齐备的实验室的工作台前，在它的主窗口中包含着以下几个可以让用户在屏幕上任意拖动、可见可隐藏的部分：窗体编辑器、对象观察器、代码编辑器、组件面板、菜单栏和工具栏，而控制所有这些组件的命令则包含在菜单栏的View选项中，用户可以选择每个组件是否出现在屏幕上。C++Builder 4.0的用户界面，如图1-1所示。

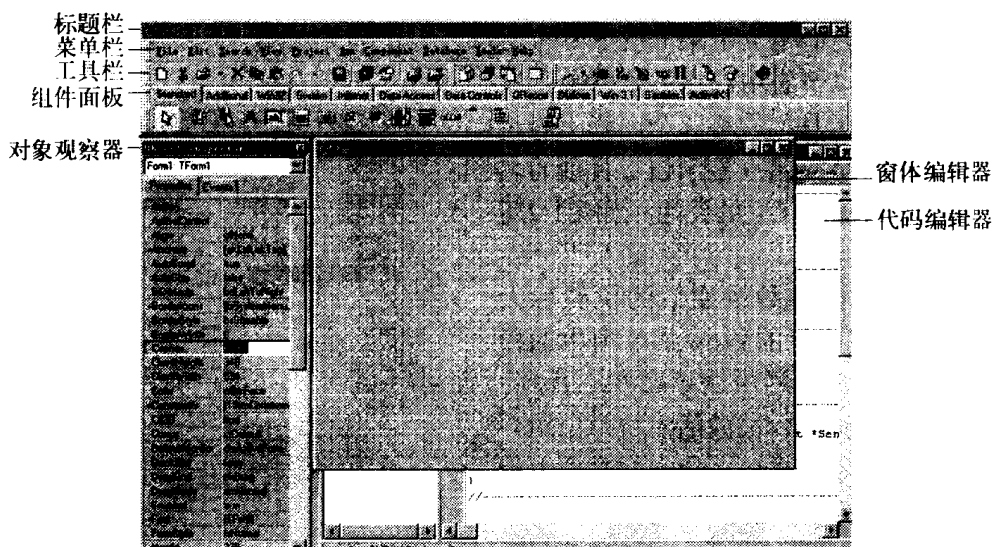


图1-1 C++Builder 4.0 用户界面

1.2.2 主菜单栏

当用户工作在C++Builder 4.0的IDE环境中时,无论是编辑、编译、链接还是进行其他操作,大部分的功能都是通过主菜单中的各个菜单选项来完成的。

File(文件)菜单: 用户可以通过点击菜单栏上的File选项或键入Alt+F弹出下拉菜单,它的主要功能是完成各种文件操作。

Edit(编辑)菜单: Edit菜单的各功能选项用于设计期间对文本和组件等的操作。

Search(搜索)菜单: 用于在代码编辑器中进行文本、对象或选定的项目的定位与查找。

View(视图)菜单: 其主要功能是用来显示和隐藏IDE中各功能组件。

Project(工程)菜单: 用于一个工程的创建、编译、链接等各种操作。

Run(运行)菜单: Run菜单中的许多选项主要用于程序的调试执行。

Component(组件)菜单: 这个菜单项完成对C++Builder中组件的控制、操作,包括:

- New Component——打开一个组件向导,生成新的组件。
- Install Component——安装一个新的组件。
- Import ActiveX Control——向C++Builder 4.0的工程中增加ActiveX组件的类型库。
- Create Component Template——创建一个新的组件模板。
- Install Package——指定用户工程文件形成包。
- Configure Palette——打开并设置面板对话框。

DataBase(数据库)菜单:

- Explorer——打开数据库浏览器,使用户在开发应用程序时与一个远程数据库服务器保持链接并创建和编辑BDE别名与metadata对象。
- Form Wizard——使用窗体向导创建一个显示本地或远程数据库中数据的窗体。

Tool(工具)菜单: 工具菜单用于设置用户编程环境、设置对象库及把外部程序作为工具加入到C++Builder 4.0的开发环境中。

Help(帮助)菜单: 主要用于向用户提供在线帮助。

1.2.3 工具栏

C++Builder 4.0的工具栏就是许多常用菜单选项的快捷方式,它们是位于菜单栏下方的几个小小图标,它可以节省用户点击菜单操作的时间。将光标移到工具栏的每一个图标上静止片刻,都会出现一个提示,显示该工具项的名称,用户也就自然知道了它的作用,如图1-2所示。

可以通过鼠标的拖拽改变它的位置和大小,也可以在主菜单栏的View菜单项中选择Toolbars项来决定工具栏中显示哪些工具项,在View\toolbars\customize被激活后将弹出用户定义对话框,在对话框的Commands选项中用户可以将任意选定的想增加的工具拖放到工具栏上,反之亦然,如图1-3所示。



图1-2 工具栏

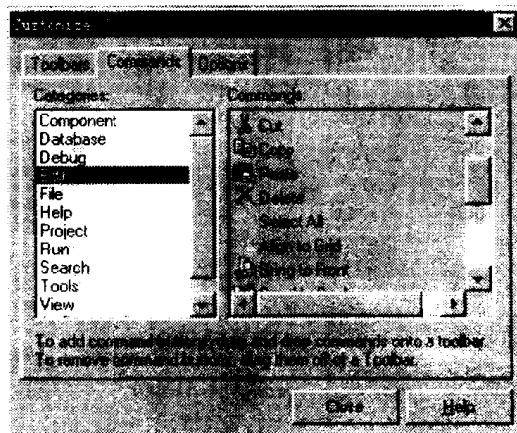


图1-3 定制工具栏

1.2.4 组件面板

C++Builder 4.0的可视化开发体现在它强大的丰富的组件上，而C++Builder 4.0的组件在功能上与C++Builder 3.0相比有了许多提高和增强，它的组件共分为以下几类：Standard(标准组件)、Additional(附加组件)、W32(Win32组件)、System(系统组件)、Internet组件、DataAccess(数据访问组件)、Data Control(数据控制组件)、QReport(快速报表组件)、Dialogs(对话框组件)、Windows3.1组件、Samples(样本组件)和ActiveX组件。它们可以在组件面板的不同页面上看到。下面将大体介绍各类组件中的内容和各个组件的大概功能，详细的内容将在以后的相关章节中具体介绍。

1. Standard(标准)组件

MainMenu——主菜单组件，为窗体建立菜单条。

Popup Menu——弹出式菜单组件，创建由用户鼠标右键击活的弹出式菜单。

Label——标签组件，显示用户不可选和不可更改的文本，常用作标题或标签。

Edit——编辑组件，显示一个文本编辑区域，用户可以输入或修改一单行文本。

Memo——备忘录组件，产生一个多行文本编辑器。

Button——按钮组件，建立按钮。

Checkbox——用于建立复选框，用户可以同时选择多个选项。

RadioButton——单选按钮组件，建立一个单选按钮，用户通过单项按钮来决定选项的是/否、真/假。

ListBox——列表框组件，产生一个带滚动条的列表框。

ComboBox——组合框组件，产生一个由编辑框和列表框结合在一起的组合框，用户不仅可以输入文本和选项，也可以选择选项。

ScrollBar——滚动条组件，用于建立一个滚动条，以改变用户在窗体中的视角或视野，即可见范围。

GroupBox——组控制组件，提供一个将窗体中多个有组关联选项组合在一起的框体。

RadioGroup——单项按钮组件，建立一个成组的单选按钮。

Panel——面板组件，创建一个窗体分割的部分，可以将窗体中部分组件组合在一个面板上，和其余组件区分开来。

ActionList——行为列表组件，创建一列表显示应用程序对用户行为的响应。

2. Additional(附加)组件

BitBtn——位图按钮组件，创建一个可显示位图的按钮，即按钮上可带有位图。

SpeedButton——用于创建一个可显示字形但不是标题的按钮。

MaskEdit——掩模文本编辑框组件，创建一个编辑框，允许用户输入和编辑，但它与文本编辑框的不同在于它可以限制用户输入文本的格式，如电话号码等，以避免用户输入无效的或非法格式的文本。

StringGrid——字符串栅格组件，创建使字符串可以行列方式显示的栅格。

DrawGrid——类似于String Grid，创建栅格可以显示数据。

Image——图像组件，用于显示位图、图标和媒体文件。

Shape——形状组件，显示几何图形，如椭圆、圆、三角形等。

Bevel——创建具有3D和雕刻效果的线与框。

ScrollBar——建立一个可改变大小的框，并在需要的时候自动出现滚动条。

CheckListBox——建立一个以滚动浏览的复选框选项为单位的列表框。

Splitter——分割条组件，用于在两个组件或两个不同区域之间加上分割条，用户可通过拖动分割条来改变两部分的大小。

StaticText——静态文本框组件，创建一个类似Label组件的不可编辑的文本。

ControlBar——控制条组件，一个工具条组件的设计管理器。

Chart——图表组件，类似于TTable，在窗体中放置图表组件并右击它将出现第三方的帮助主题，主要用于显示图表。

3. Win32 组件

TabControl——非常类似于文件柜或笔记本中的分隔板或分类标签，它建立一组互不相同的笔记类型标签。

PageControl——用于建立对话框类型的多页控制。它与TabControl的不同在于组成PageControl组件的单项都作为独立的对象存在。

ImageList——图像列表组件，是一个相同大小图像的集合列表，可通过索引查找，就是对大量的图标或位图文件进行有效的管理。

RichEdit——一个功能强大的文本编辑器。

TrackBar——建立轨迹条，就像一个调节器，可以定义范围和程序，告知用户现在的状态或由用户来改变当前的值。

ProgressBar——进度条组件，用来显示当前操作的进度。

UpDown——上下箭头组件，用于控制当前值的增减。

HotKey——热键组件，用于建立一个组件的热键。

Animate——建立一个播放无声AVI文件或一套连续的位图文件窗口。

DateTimePicker——建立一个允许用户输入日期和时间的列表框，用户也可能通过滚动条在日历中选择日期和时间，但必须有最新版本的COMCTL32.DLL文件，它位于Windows\System或Windows\System32目录下。

MonthCalender——日历组件，显示单独一个月的日历。

TreeView——建立一个对象之间多层次关系的树状显示。

ListView——提供一个分栏显示列表的方法。

HeaderControl——在文本或数字栏上加一个表头。

StatusBar——状态条组件，建立状态栏，一般是在屏幕的底部。

ToolBar——工具条组件，用于管理工具按钮，限定它们的范围和排列，可以自动调节工具条的大小和位置。

CoolBar——显示一个窗口化组件(CoolBand对象)的集合。

PageScroller——将对象包含在一个委托区域并且可以通过上下左右滚动条浏览。

4. System(系统)组件

Timer——定时器组件，按照指定的时间间隔产生指定的事件。

PaintBox——绘图框组件，用于建立一个可以绘图的矩形框。

MediaPlayer——显示一个VCR类型的面板用来播放和录制多媒体视频和音频文件。

Olecontainer——在窗体中创建一个OLE代理区域，生成OLE客户程序。

DbeClientConv——建立一个与DDE(动态数据交换)服务的客户链接。

DbeClientItem——指定在DDE对话中要传输的DDE客户数据。

DbeServerConv——建立一个DDE客户应用的服务器链接。

DbeServerItem——指定在DDE对话中要传输的DDE服务器数据。

5. Internet 组件

Internet组件为应用程序提供多种Internet访问协议。

ClientSocket——将应用加入到一个窗体或模块中，使该应用变成一个TCP/IP客户程序，ClientSocket指定一个到TCP/IP服务器的理想链接，管理打开的链接和中断连接。

ServerSocket——加入到一个窗合格或模块中使一个应用变成TCP/IP服务器，它监听别的机器的TCP/IP连接请求并在接收到请求后建立这个连接。

WebDispatcher——用于将一个普通的数据模块转换为Web数据模块并可以使Web服务器应用程序对HTTP请求作出响应。

PageProducer——将HTML模板转换成能被一个例如Web浏览器之类的客户应用程序所接受和翻译的HTML命令字符串。这些命令和HTML转换页面可以在OnHTMLTag事件中定制。

QueryTableProducer——将一个HTML命令序列组装成为TQuery对象中的记录，并用表格显示出来。

DataSetTableProducer——将一个HTML命令序列组装成为TDataSet对象中的记录，并用表格方式显示。

DataSetPageProducer——将一个包含域参考的HTML模板转换成一个客户应用程序可接受和翻译的HTML命令字符串。

NMDayTime——从Internet或intranet中日期时间服务器中取得时间和日期。

NMEcho——发送文本到Internet回应服务器，并得到回应的信息。

NMFinger——从一个Internet指示服务器中取得一个用户的信息。

NMFTP——是一个不可见的ActiveX组件，执行文件传输协议，使得在一个本地机和一个远程机之间传输文件和数据以及访问Internet FTP服务变得很容易。

NMFTTP——是一个不可见的ActiveX组件，它充当HTTP协议客户机，使用户在没有浏览器或图像处理工具时直接得到HTTP文档。

NMMsg——通过TCP/IP协议在Internet或intranet上传输简单的ASCII文本。

NMMsgServ——接收通过NMMsg组件发送的信息。

NMMTP——不可见的ActiveX客户机组件，允许应用程序访问NNTP(网络新闻传输协议)的新闻服务器，它提供新闻的阅读和邮发功能。

NMPOP3——一个不可见的ActiveX组件；用于从支持POP3(邮件协议)的UNIX系统或其他服务器上取得邮件。

NMUUProcessor——对MIME和UUEncodes格式文件加解密。

MNSMTP——ActiveX组件，使应用程序访问SMTP(简单邮件传输协议)的邮件服务器并且兼顾邮发邮件。

NMStrm——通过Internet或intranet向流服务器发送流。

NMStrmServ——接收通过NMStrm组件发送的流。

NMTime——从Internet时间服务器上获得时间和日期。

NMUDP——不可见的WinSock ActiveX组件，提供用户对数据报(UDP)协议网络服务的轻松访问。向客户机和服务器都可以提供WinSock并且可以描述一个利用UDP网络服务的通讯点，也可以用来发送和接收UDP数据。

PowerSock——是创建处理其他协议的组件的基础，或用来创建用户自定义协议。

NMGeneralServer——是一个开发多线程的Internet服务器的基类。

HTML——不可见的ActiveX组件，用于建立一个HTML浏览器，对所选择的HTML页进行

语法分析和设计,也可以作为一个非可视化的HTML分析器或HTML文本处理器来分析和处理HTML文档。

NMURL——将URL数据解码生成可读的字符串,或将标准字符串加密为URL格式。

6. Data Access(数据访问)组件

DataSource——该组件在数据集和与数据集相关的组件之间充当一个通道的作用,将两者联系起来,每个被数据访问组件和数据控制组件使用的数据集都必须要有有一个DataSource组件。

Table——属于数据集组件,它可以通过BDE(Borland数据库引擎)取得某个物理数据库中的数据并通过DataSource组件将数据提供给多个数据控制组件;反过来也可以将数据控制组件中的数据通过BDE返回到数据库中。

Query——查询组件,可以使用SQL语句通过BDE从一个物理数据库中得到数据,并将数据通过TDataSource组件提供给一个或多个数据控制组件,反过来也可以使用SQL语言,通过BDE将组件中的数据传回到数据库。

StoredProc——该组件使应用程序可以访问服务器预存储进程,并通过BDE将组件中的数据送回到物理数据库中。

DataBase——数据库组件,它使应用程序和数据库之间建立一个持久稳定的连接,特别表现在那些需要用户登录和口令的远程数据库。

Session——提供一个对数据库组件组的全局管理,针对每一个C++Builder 4.0数据库应用。一个缺省的TSession组件会自动被创建,只有在创建一个多线程的数据库应用时必须用到TSession组件,每一个数据库线程都需要一个它自己的Session组件。

BatchMove——拷贝一个数据库表的结构或它的数据,它主要用于将整个数据库表的格式转换成另一个数据库格式。

UpdateSQL——允许使用缓冲更新功能更改只读数据库。

HestedTable——从一个嵌套数据集中取得数据并通过一个DataSource组件将数据提供给数据控制组件。

7. Data Controls(数据控制)组件

所有的组件都用于数据库中的数据,但必须和数据访问组件配合使用。

DBGrid——以类似电子表格的形式显示和编辑数据,还可以决定一栏或一行数据的可见性、显示格式和排列顺序等。

DBNavigator——数据控制漫游按钮可以向前向后移动一个表格的当前记录指针,并且可以将一个表格置为插入、编辑或浏览状态,生成新的或修改记录或用新的数据刷新表中内容。

DBText——将当前字段标注为不可编辑的文本。

DBEdit——将当前记录的数据以一个文本编辑框显示出来并允许对其进行编辑。

DBMemo——将当前记录的数据用备忘编辑器形式显示出来,并允许进行编辑,主要针对Memo类型的字段。

DBImage——打开Image编辑框对当前记录的位图进行显示、剪切或粘贴等操作。

DBListBox——将当前记录中的字段以列表框的形式显示出来。

DBComboBox——将当前记录中的字段以组合框的形式显示出来。

DBCheckBox——将当前记录中的字段以复选框的形式显示出来。

DBRadioGroup——以单选按钮组的形式对数据库进行操作。

DBLookupListBox——用列表框的形式显示当前数据库Lookup栏的数据。

DBLookupComboBox——用组合框的形式显示当前数据库Lookup栏的数据。

DBRichEdit——多行编辑组件，用于显示多格式的Memo字段的内容。

DBCtrlGrid——该组件功能与DBGrid功能相似，但可以对它的每个单元设置属性。

DBChart——在窗体放置该组件，右击将显示第三方开发者的帮助主题。DBChart和Chart的不同在于DBChart需要BDE支持。

8. 快速报表组件

快速报表组件主要用于快速建立可视化快速报表。

QuickRep——用于建立自定义报表的基本报表形式，它是一个带有当前选定纸张形式的可视化组件。

QRSubDetail——用于将一个附加的数据集链接到报表中。

QRStringsBand——在报表中加入包含字符串的边框。

QRBand——在TQuickRep组件中画出边框，并通过设定BandType属性设定边框类型显示在报表生成中边框的行为。

QRChildBand——如果你有一个内部包含组件的边框，并且希望其他组件放在它们下面，就可以使用该组件。

QRGroup——可以让用户将所有边框组织到一个组中并提供对头、注脚和分页的控制，用于建立明细报表。

QRLabel——输出静态的或无数据库关系的文本。

QRDBText——输出数据库字段值的TQRLabel组件的数据敏感形式。包括字符串字段、可变数目字段、日期字段和Memo字段。

QRExpr——输出显示数据库字段、计算和静态文本，可以在其表达式属性中放置一个有效的快速报表表达式。

QRSysData——打印输出系统信息，如报表标题、当前页号等，可在数据属性中选择要打印输出的数据等。

QRMemo——用于打印输出一个非数据库字段的文本。它可能是一个静态文本或是在报表生成时可改变的文本。

QRExprMemo——允许用户使用快速报表表达式编程生成内容。

QRRichText——允许在报表中嵌入RichText。

QRDBRichText——在用户的报表中提供一个访问DBRichText字段的快速报表封面。

QRShape——允许在报表中绘出简单的几何图形如圆、方、三角形等。

QRImage——在报表中显示一幅图像，支持TPicture类所支持的所有图形格式。

QRDBImage——打印输出存储在数据库二进制字段中的图形，它可以输出所有C++Builder 4.0支持的图形格式。

QRCompositeReport——允许用户将一个以上的报表合成，用以生成复杂报表。

QRPreview——用于定制报表预览的模式。

QRTextFilter——允许用户将报表内容以文本形式输出。

QRCSVFilter——允许用户将报表内容以逗号划分的数据库源文件形式输出。

QRHTMLFilter——允许用户将报表内容以HTML格式输出。

QRChart——用于创建图表格式的快速报表。

9. 对话框组件

这些组件生成Windows标准对话框，这些标准对话框为诸如打开、关闭、打印、存储等文