杭州应用工程技术学院学报,第12卷第1期,2000年3月
Journal of Hangzhou Institute of Applied Engineering
Vol.12 No.1, March 2000

C++ Builder中用户自定义组件的实现

潘俊强
(杭州应用工程技术学院信电系　杭州 310012)

摘　要　介绍在C++ Builder环境中用户自定义组件的实现,包括自定义组件的属性、事件和方法.

关键词　C++ Builder　用户自定义组件　属性　事件　方法
中图分类号　TP368.5

C++ Builder是Borland公司继Delphi之后开发的又一个完全面向32位的、具有通用客户/服务器结构的开发工具,使用它可以很容易地编写Windows应用程序,是目前继Visual Basic、Delphi之后在32位Windows环境下最具吸引力的开发工具.

C++ Builder的集成开发环境溶入了更多的、Delphi所没有的Windows控制组件,建立的组件分类也更为详细,包括标准组件、Win32组件、Internet组件、数据访问组件、数据控制组件等,程序员使用组件可以很容易地完成各种界面、操作等的设计和编程,如信息录入界面的设计、各种Internet的操作等. 但有时用户为了完成特殊的任务需要自己设计一个组件,这时就要使用用户自定义组件,用户自定义组件包括可视的和非可视的,本文介绍的是可视用户组件的实现. 一个组件包括属性、事件和方法,下面就如何实现用户自定义组件的属性、事件和方法作一分析,本文以C++ Builder环境中Samples组件页下的Ccalendar(日历)组件为例来说明.

1　创建一个新组件

在C++ Builder主菜单的Component菜单项中点击New Component,弹出一New Components对话窗口,让你选择或填入祖先类型、新组件的类名、新组件将添加在哪一组件页等信息;在本文所举例中,Ccalendar组件以TCustomGrid为祖先类型,组件类名为TCCalendar,组件添加在Samples组件页中,确定正确后单击OK,C++ Builder将自动在当前的项目中为你生成ccalendar.cpp和ccalendar.h. 其中,ccalendar.h文件包含了对TCCalendar类的定义,ccalendar.cpp包含了对TCCalendar类具体实现的代码.

收稿日期:1999-07-20

2 用户自定义组件属性的实现

从应用编程的角度来看,组件的属性就象是变量,开发人员可以象对数据成员操作一样设置和读取,但是属性比数据成员的功能更强大,第一,它可以在程序设计阶段赋值,在对象观察器(Object Inspector)[1]的属性页中可以看到相应组件的所有属性,在这里,开发人员可以设置和读取各个属性值;第二,它可以隐藏实现细节,在表现值和实际存储值之间可以存在复杂的算法. 作为用户自定义组件的属性,如何做到以上两点呢? 现以组件 Ccalendar 的 Day 属性(用来设置 CCalendar 组件实例的当前日)来说明.

2.1 让 Day 属性存在于对象观察器的属性页中

在 ccalendar.h 中有如下说明:

```
class PACKAGE TCCalendar:public TCustomGrid
{
private:
… //此处为其他说明的省略,以下同
TDateTime FDate;//记录日历的当前日期
…
Integer _ fastcall GetDateElement(Integer Index);//读取日期的函数
…
void _ fastcall SetDateElement(int Index,int Value);//设置日期的函数
protected:
…
public:
…
_ published:
…
_ property Integer Day = {read = GetDateElement,write = SetDateElement,stored = false,index = 3,nodefault};
…
};
```

_ property Integer Day 说明 Day 将作为 Ccalendar 的属性出现在对象观察器的属性页中,类型为整型;read = GetDateElement 说明属性 Day 的读取操作由函数 GetDateElement 来完成;write = SetDateElement 说明属性 Day 的设置操作由函数 SetDateElement 来完成;index = 3 说明 GetDateElement 和 SetDateElement 两个函数中的形参 index 的值等于 3.

2.2 让 Day 属性与具体存储值对应

在以上的说明中我们已经知道,属性 Day 的读取和设置操作分别由函数 GetDateElement 和 SetDateElement 来实现,那么通过对这两个函数的编程就可以做到属性与具体存储值的对应;下面是函数的源代码:

```
int _ fastcall TCCalendar:GetDateElement(int Index)
{
  Word AYear,AMonth,ADay;
```

```
  int result;
DecodeDate(FDate, AYear, AMonth, ADay);//DecodeDate 负责将当前日期 FDate 分解为 Year,
                                        Month, Day
  switch(Index)
  {
    case 1://Index = 1 为取 Year
      result = AYear; break;
    case 2://Index = 2 为取 Month
      result = AMonth; break;
    case 3://Index = 3 为取 Day
      result = ADay; break;
    default:
      result = -1;
  };
  return result;
};
void _fastcall TCCalendar::SetDateElement(int Index, int Value)
{
  Word AYear, AMonth, ADay;
  bool Update = false;
  if (Value > 0)
  {
    DecodeDate(FDate, AYear, AMonth, ADay);
    switch(Index)
    {
      case 1://设置 Year
        if (AYear! = Value)
        {
          AYear = Value;
          Update = true;
        }
        break;
      case 2://设置 Month
        if ((Value < = 12) && (Value! = AMonth))
        {
          AMonth = Value;
          Update = true;
        }
        break;
      case 3://设置 Day
        if ((Value < = DaysThisMonth()) && (Value! = ADay))
```

```
            {
              ADay = Value;
              Update = true;
            }
            break;
          }
          if (Update)
          {
            FDate = EncodeDate(AYear, AMonth, ADay);//将 Year, Month, Day 组合成日期类型
            UpdateCalendar();//更新日历
            Change();//调用 OnChange 事件,见事件实现中的说明
          }
        };
      };
```

当程序员向应用插入 Ccalendar 组件时,Day 属性出现在对象观察器的属性页中,最初该属性显示的值为组件初始化时赋予当前日期 FDate 的 Day 部分,由函数 GetDateElement 取得;当向 Day 属性框中输入合法数值后,组件将调用 SetDateElement 函数重新设置当前日期 FDate 的值.

3 用户自定义组件事件的实现

C++ Builder 中的每一个可视组件,在对象观察器的事件页中均可发现若干事件项,如 OnClick 等,在其中可以定义事件函数(事件触发时执行的代码)的句柄. 对于用户自定义组件,如果需要定义一个用户事件,则必须做以下两步工作,本文以 Ccalendar 组件中的 OnChange 事件(每当 Ccalendar 组件实例的当前日期改变后将触发该事件)为例来说明:第一,必须定义用户事件,在 TCCalendar 类定义的 private 段说明一类型为 TNotifyEvent 的变量 FOnChange,用以记录事件函数的句柄;同时类似于属性的实现,在 TCCalendar 类定义的_ published 段中说明_ property TNotifyEvent OnChange = {read = FOnChange, write = FOnChange},使 OnChange 事件项出现对象观察器的事件页中,同时定义了对 OnChange 事件项的操作就是对变量 FOnChange 的读取和设置,这样,在应用中程序员一旦完成了对 OnChange 事件的设置,变量 FOnChange 也就获得了事件函数的句柄. 第二,必须在合适的地方调用用户定义的事件函数,在本文所举例中,每当 Ccalendar 组件的当前日期发生变化后,就调用 Change 函数,Change 函数实现如下:

```
void _ fastcall TCCalendar::Change()
{
  if(FOnChange)//如果用户已对 OnChange 事件项定义
    FOnChange(this);//则调用事件函数(FOnChange 保存了用户定义的事件函数的句柄)
}
```

4 用户自定义组件方法的实现

组件的方法有时也称为组件的行为,是指该组件可以执行的动作或操作. C++ Builder 为组件提供了许多缺省方法,如 Button 组件的 Click 方法(可以使程序员在程序中仿真鼠标的单击操作)

等. 用户自定义组件方法的实现很简单,只要.h 文件中组件类定义的 public 段中加入方法原型说明,然后在.cpp 文件中加入方法的具体实现代码即可,如本文所举例中,Ccalendar 组件提供了 NextMonth(使 Ccalendar 组件实例的当前日期往后移一个月)的方法,它的实现如下:首先在 TCCalendar 类定义的 public 段中加入 void _ fastcall NextMonth();的方法原型说明,然后在 ccalendar.cpp 文件中加入以下实现代码:void _ fastcall TCCalendar::NextMonth()

```
{
ChangeMonth(1);//往后移一个月的用户自定义函数
}
```

在应用中,程序员可以通过以下形式调用该组件的方法:组件实例名 -> NextMonth().

5 安装用户定义组件

在完成了以上所有的工作后,就要安装用户自己定义的组件了,在 C++ Builder 主菜单的 Component 菜单项中点击 Install Component,弹出一 Install Components 对话窗口,让你输入组件的源文件名、搜索路径以及安装到哪一个组件包等信息,在本文所举例中,源文件名为 ccalendar.cpp,组件包文件名为 bcbsmp35.bpk(此包为 C++ Builder 自带,提供了 Samples 组件页下的所有组件),搜索路径为缺省路径,确定正确后单击 OK,C++ Builder 会根据用户输入信息生成(即编译连接)或重新生成组件包文件,生成成功后,用户自定义组件将自动挂在用户指定的组件页中,在本文所举例中,Ccalendar 组件将挂在 Samples 组件页下.

6 结束语

通过用户自定义组件,程序员可以很容易地扩展和维护自己的组件库,为应用程序的开发提供极大的便利. 笔者只是简单介绍了一个用户自定义组件的实现,除了文中所述,还有许多技术要点和实现技巧,有兴趣者可以参考 C++ Builder 的随机文档和例程,并进行深入研究.

参 考 文 献

1　广正工作室 .C++ Builder 实用教程 . 北京:机械工业出版社,1998.23

How to create custom components in C++ Builder

Pan Junqiang

(Information & Electronic Engineering, Hangzhou Institute of Applied Engineering, Hangzhou 310012)

Abstract　This paper introduces how to create properties, methods and events for custom components in C++ Builder.

Key words　C++ Builder　custom _ components　property　event　method