PowerBuilder
高级应用技术

北京市晓通网络数据库研究所

PowerBuilder
高级应用技术

作者 侯志平

北京晓通网络数据库研究所

前　言

用 PowerBuilder 开发应用系统确实是一件令人振奋的事情，她让开发人员体会到了美与和协的含义，使应用系统的生产率得以成倍的提高。目前，PowerBuilder 以其优良的性能和普及率领导着数据库应用技术的发展潮流，每个应用开发人员都应掌握她。

第一册《PowerBuilder 使用方法与实例 4.0/5.0》讲述 PowerBuilder 的安装步骤，开发方法，以及各种开发工具的使用方法和技巧。

本册资料以一个精心设计的实例贯穿始终。在讲完 PowerBuilder 开发环境的每一部分后，都配有完成此实例的操作步骤，并详细讲解实例的设计思路和设计方法。

应用实例以当前流行的MDI风格作为窗口界面，包含多媒体动画、OLE对图象的处理、动态查询、树状浏览、远程拨号、中国式报表制作等许多用户感兴趣的问题；涉及到了数据管道、用户对象、用户事件、继承等开发技术，为广大读者提供了一套非常有价值的应用范例，只要适当修改就可以变成自己的应用。

本册资料还配有完成实例的源代码盘．源代码盘是按照应用开发的实际步骤设计的，从最初开发到完善应用的每一阶段都有相应的源代码．读者可以跟着书中的实例学习，然后再与相应阶段的源代码比较．初学 PowerBuilder 的读者可以跟着实例快速地掌握 PowerBuilder 的开发方法，并可模仿这个实例开发出有一定水平的应用系统。

第二册（含 4.0 和 5.0 上下两册）《PowerBuilder 语言、事件、函数和属性4.0/5.0》用大量的例子讲解了 PowerScript 语言、函数、对象属性和事件．特别是对在 PowerBuilder 中起重要作用的函数和事件投入了大量的笔墨．本书的特点是例子丰富实用，有很多在实际开发中常用到的程序段，读者可以借鉴使用。

第三册《PowerBuilder 高级应用技术 4.0/5.0》讲述了开发高级应用的方法．本册资料以专题的形式讲解了多文档界面（MDI），对象连接与嵌入（OLE），动态数据交换(DDE)，动态连接库（DDL），动态数据窗口，动态统计图，数据窗口的列校验，子数据窗口，树状浏览，数据转换技术，微软邮件系统（MAPI），面向对象的开发技术，中国式报表，基础类库，安装

盘制作工具，开发工具包等的使用方法。对于那些有一定 PowerBuilder 使用经验的读者来说，可以通过本册资料掌握深层次的开发方法，学会用更巧更快的方法开发出高水平的应用。

第四册《珠联璧合- PowerBuilder与数据库配合开发技术》强调数据库设计的重要性，讲解了数据库实体关系模型设计方法；讲述了数据库规则、存储过程、触发器和视图在开发中的作用；讨论了并发控制和事务处理方法；给出了 PowerBuilder 应用在不同数据库间移植的方法。本册资料通过大量实例表明：数据库设计是一只看不见的手，在很大程度上决定着信息系统的成败；而良好的设计会使 PowerBuilder 的开发更简捷、更有效，是应用成功的基石。因此，对追求高品味和一流质量的开发人员来说，这是一本难得的好书。

目 录

第一章 多文档界面(MDI)窗口的使用

1.1 MDI 简介

1.1.1 什么是 MDI

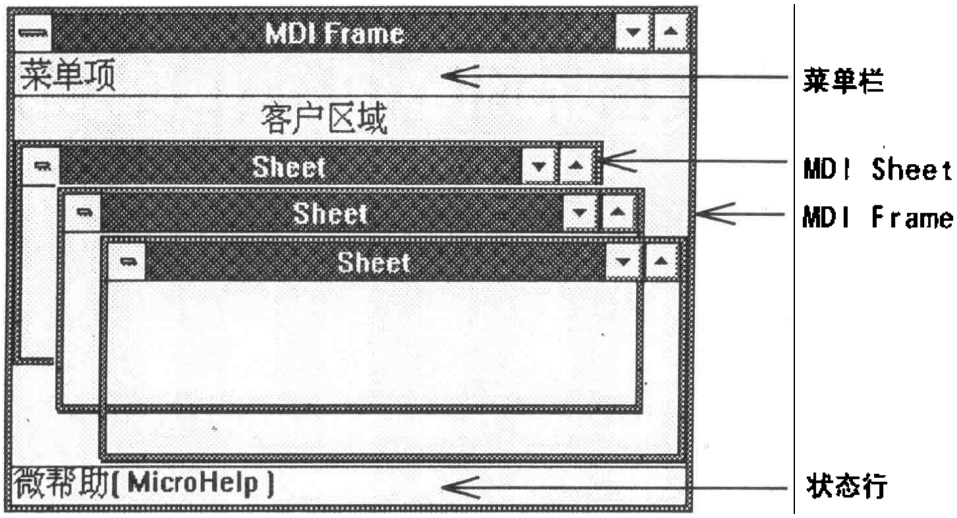
MDI 的意思是“Multiple Document Interface”，其含义是“多文档界面”。它是 Windows 上常见的一种应用窗口风格，这种风格目前较为流行，Windows 上许多软件的主窗口用的都是这种风格，例如 PowerBuilder 软件的主窗口。采用这种窗口风格的软件会给用户一种整体感，可以使用户感觉在一个窗口里就可以完成许多工作。当你的应用需要打开一组外观、类型和功能相似的窗口时，就可以考虑使用 MDI 风格，它可以使应用的界面简单、美观、灵活和方便。本套丛书的应用实例采用的就是 MDI 风格，是一个典型的 MDI 应用，你可以参照它设计你的 MDI 应用。

要想建立 MDI 应用，就要定义一个类型为“MDI Frame”或“MDI Frame with Microhelp”的窗口，并且在这个窗口中打开一些其它的窗口（Sheet）。

MDI 应用风格的特点是用户可以在一个 MDI 类型的窗口中打开多个窗口（Sheet），而且还可以在这些打开的窗口之间任意切换。一般在 MDI 窗口上打开的窗口类型是一致的，并且有类似的特点或用途。

1.1.2 MDI 类型窗口的框架结构

MDI 类型窗口的框架如下图所示:



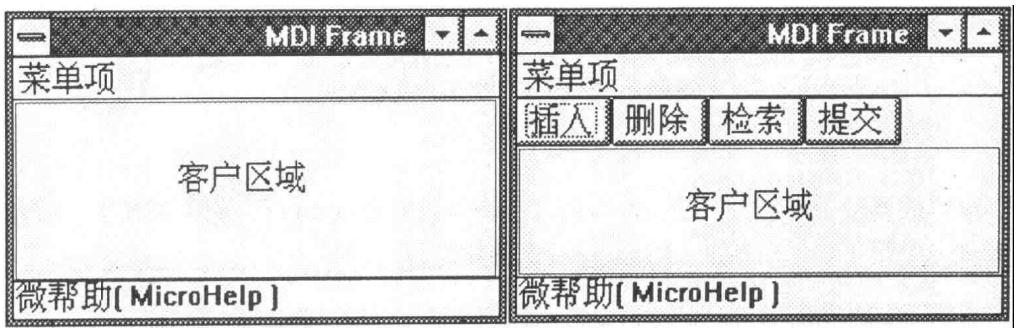
可以看出，MDI 类型窗口可分为五个部分：Frame、菜单栏、客户区域、Sheet 和状态行。Frame 指的是 MDI 窗口的轮廓，其上有菜单栏，其下有微帮助，中间是客户区域；客户区域指的是窗口的工作空间，在这个空间上可以打开多个窗口；Sheet 指的是打开在 MDI 客户区域上的窗口。除了 MDI 类型的窗口之外，所有其它类型的窗口都可以用作 Sheet 打开在 MDI 窗口上，但用 Open 函数不行，要用 OpenSheet 函数打开 Sheet （有关函数 opensheet 的使用请参见本套丛书的函数部分）。

通常 MDI Frame 有两种类型，一种是标准的，另一种是定制的。

标准 MDI 窗口有可选的菜单栏和状态行（用于显示微帮助），客户区域空着，等待打开窗口（Sheet）。Sheet 可以有自己的菜单，也可以继承 MDI 窗口的菜单。典型的 MDI 应用通常在菜单栏中用一个菜单项列出所有打开的 Sheet ，另一个菜单项用于选择显示 Sheet 的方式（按平铺、层叠还是排列图标的方式显示）。

定制的 MDI 窗口与标准 MDI 窗口类似，也有菜单栏和状态行，但客户区域不同。 标准 MDI 窗口的客户区域仅仅包含打开的 Sheet，它的客户区域会充满整个 MDI 窗口的工作空间。例如，MDI 窗口有菜单栏和状态栏，客户区域将是菜单栏以下，状态行之上的所有空间，见下图；而定制的 MDI 窗口的客户区域除了包含打开的 Sheet 之外还包含其它对象，诸如按钮，编辑器，静态文本等等，它的客户区域的范围取决于 MDI 窗口上的菜单栏和状态行以外的对象大小。例如，假如在 MDI 窗口的顶端放置一组按钮以完成应用的其它功能，那么，MDI 窗口的客户区域将是在按钮之下，状态行之上的区域，见下图。在标准的 MDI 窗口中，Power Builder 会自动定制客户区域的尺寸，并将打开的 Sheet 显示在客户区域中；定制 MDI 窗口的客户区域

上包含一些对象，因此，必须自己定制客户区域（通过编写程序实现），如果不定制客户区域的尺寸，客户区域上的 Sheet 会打开但将是不可视的。具体的定制方法会在本章的后面部分讲述。



标准 MDI 窗口 MDI_Win1　　客户 MDI 窗口 MDI_Win2

MDI 窗口除了可以有菜单栏之外，还可以有工具栏。你可以用 PowerBuilder 自动建立的基于当前菜单的工具栏，也可以创建自己的定制工具栏（通常是用户对象），而且还可以自己定制客户区域尺寸。有关 MDI 窗口的工具栏的建立，后面会讲述。

有两点要注意:

第一点，MDI 风格的应用的菜单栏永远显示在 MDI Frame 上的，而不会显示在 Sheet 上。

第二点，MDI 窗口上最好有菜单，当用户关闭了所有的 Sheet ，菜单是关闭 MDI 窗口的最佳途径。

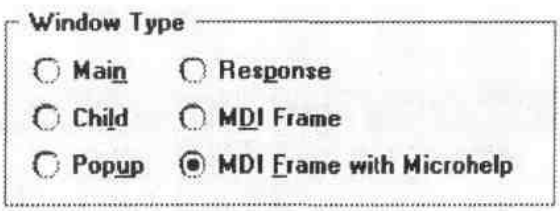
1.2 创建 MDI 窗口

建立 MDI 窗口的基本步骤如下:

1. 创建一个新的窗口或是打开一个已经建立好的窗口。

2. 双击打开的窗口，“窗口的风格”（Window Style）窗口显示。

3. 选择窗口类型为“MDI Frame”或“MDI Frame with Microhelp”，参见下图:



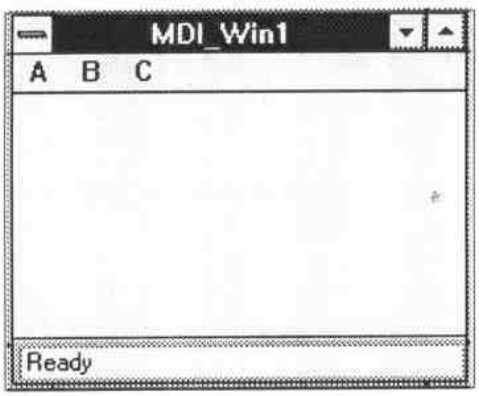
窗口的类型“MDI Frame”和“MDI Frame with Microhelp”的区别在于，后者比前者多一个用于显示微帮助（Microhelp）的状态行。

4. 再选中“菜单”选项，并为 MDI 窗口选择一个要连接的菜单（ 要建立 MDI 窗口，建议用户使用菜单）。

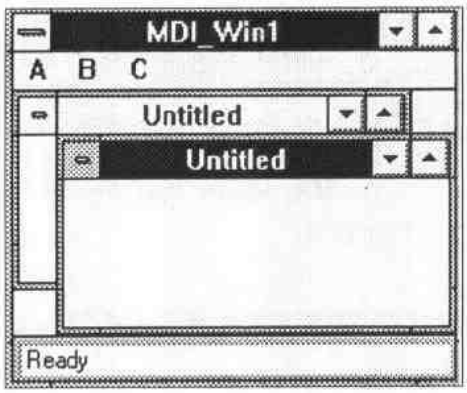
5. 点取“Ok”按钮返回。

6. 编写适当的程序以打开 MDI 窗口以及执行其它处理。

如果你创建的 MDI 窗口是标准的，按上述的步骤就可以创建出前面显示过的 MDI 窗口 mdi_win1，当它运行时，它会显示如下:



标准 MDI 窗口打开时的外观



在标准MDI窗口中打开Sheet时的外观

如果你要建立的 MDI 窗口是定制的（还要在 MDI 窗口上增加一些对象，例如按钮和编辑器等），那么在完成上述操作后，还需要编写程序来定制客户区域的尺寸，否则将不显示你打开在 MDI 窗口上的 Sheet。这段定制客户区域尺寸的程序要编写在 MDI 窗口上的 Resize （重定义尺寸）事件下，编程步骤如下（以上图显示的定制

MDI 窗口 MDI_Win2 为例，其上的四个按钮的名字分别为 cb_1，cb_2，cb_3，cb_4）：

1. 首先，打开这个定制的 MDI 窗口 MDI_Win2 并选中这个窗口，点取“编程”按钮，“编程”（Script）窗口打开；

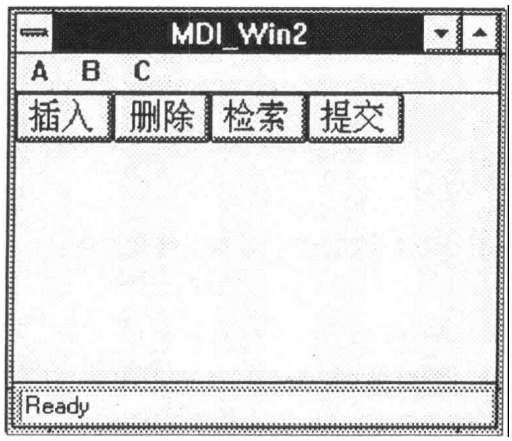
2. 先选择 Resize （重定义尺寸）事件，然后在“编程”（Script）窗口上输入下列程序：

```
int WSwidth, WSheight

WSwidth = workspacewidth (mdi_win2)
WSheight = workspaceheight (mdi_win2)

//上面这两条语句用于获取 MDI 窗口 mdi_win2 的工作空间的宽度和高度.

WSheight = WSheight - (cb_1.y + cb_1.height)
WSheight = WSheight - (mdi_1.microhelpheight)

//上面这两条语句是用于计算按钮和微帮助之间的工作空间的高度。
//其中 cb_1.y 指的是窗口 mdi_win2 上最左边的“命令”按钮的纵坐标。
//mdi_1.microhelpheight 是窗口 mdi_win2 上用于显示微帮助的状态行
//的高度. 如果 MDI 的窗口类型不是“MDI Frame with Microhelp”而
//是“MDI Frame ”这个状态行的高度将为 0.

mdi_1.Move ( 0, cb_1.height)

//上面这条语句是将 MDI 窗口的客户区域的右上角移动到横坐标为0，纵坐标为
//cb_1.height 的位置上。

mdi_1.Resize ( WSwidth, WSheight)

//上面这条语句将 MDI 窗口的客户区域的尺寸按宽度为 WSwidth ，高度为
//WSheight 定制。
```

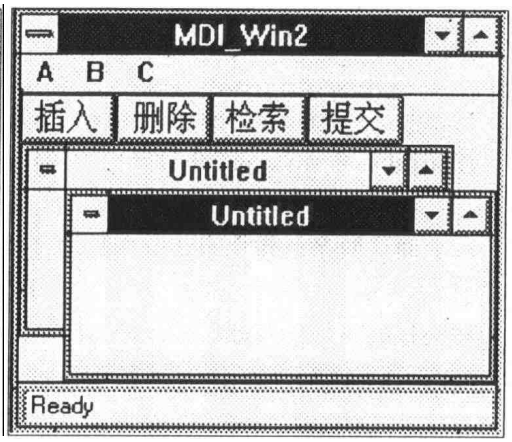
上面这段程序将定制 MDI 窗口的客户区域的尺寸重新定制了。在这段程序中出现过一个控制名字“mdi 1 ”，在这里解释一下其含义。当你建立了 MDI 窗口，Power

Builder 就创建一个控制，叫 MDI_1，它是用于标识 MDI 窗口的客户区域的．上面程序中的“ mdi_1.midrohelpheight ”（MDI 窗口的微帮助高度）是 MDI_1 控制的属性．除此之外，MDI_1 还有其它一些属性，例如，MDI_1.height（MDI 窗口的客户区域高度），MDI_1.width（MDI 窗口的客户区域宽度）等等．在标准的 MDI 窗口中，Power Builder 会自动管理 MDI_1；而在定制的 MDI 窗口中，需要你写程序来管理．

按上述的步骤编写好定制 MDI 窗口 mdi_win1 的定制客户区域的程序后，这个窗口在运行时会显示如下:



客户 MDI 窗口打开时的外观



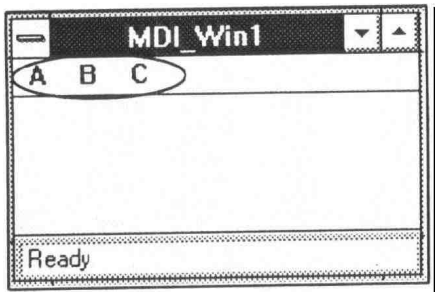
在客户MDI窗口中打开Sheet时的外观

这样，就完成了 MDI 窗口的创建工作．

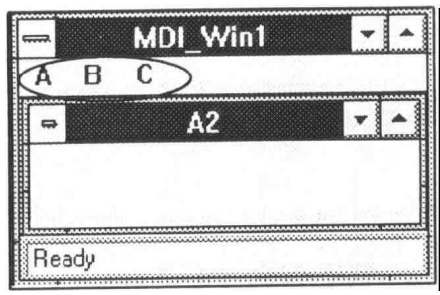
1.3 MDI 窗口的菜单栏

菜单对于 MDI 窗口是非常重要的，当你为应用建立 MDI 窗口时，你应该为这个窗口连接一个菜单．这样，当用户关闭了 MDI 窗口中的所有 Sheet 后，菜单是用户关闭 MDI 窗口的一个途径．在这部分中，我们主要讲一讲 MDI 窗口上的菜单和打开在该窗口上的 Sheet 的菜单之间的关系，这对于建立 MDI 风格的应用是十分重要的．如果搞不清它们的关系，会使你的 MDI 应用流程出错．下面介绍它们之间的关系:

1. 当一个没有菜单的 Sheet 打开在 MDI 窗口中时，它会继承 MDI 窗口上的菜单．见下图:

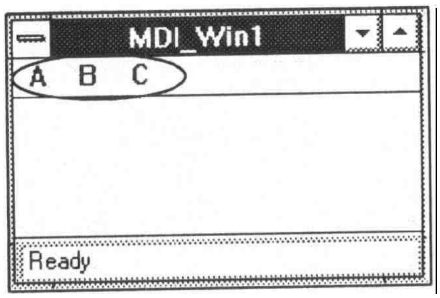


MDI 窗口上的菜单

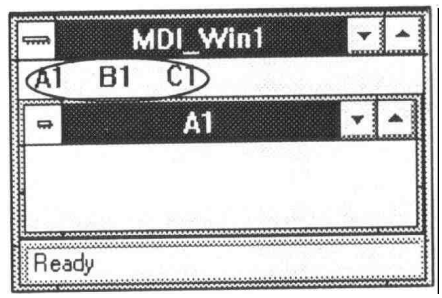


不带菜单的 Sheet 被打开后，它会继承 MDI 窗口上的菜单

2. 当一个有菜单的 Sheet 打开在 MDI 窗口中时，它的菜单会取代 MDI 窗口上的菜单，见下图：



MDI 窗口上的菜单



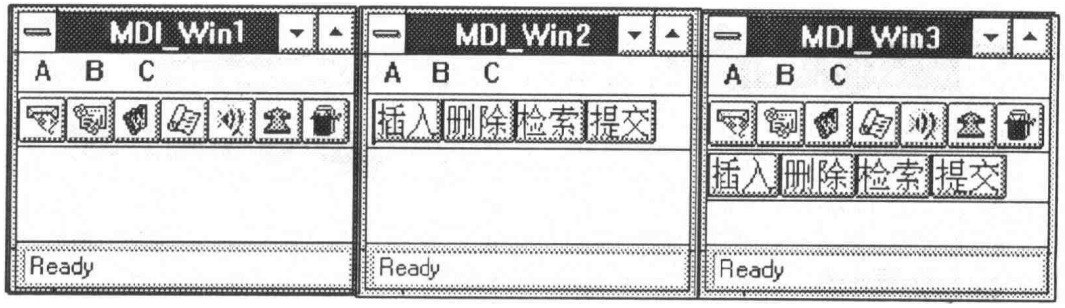
带菜单的 Sheet 被打开后 MDI 窗口上的菜单换成了 Sheet 的菜单

3. 当一个有菜单的 Sheet 先打开在 MDI 窗口，不关闭，然后再打开一个没有菜单的 Sheet，这时，这个没有菜单的 Sheet 会继承那个有菜单的 Sheet 的菜单，而不会再继承 MDI 窗口上的菜单了。而且，只要这个 Sheet 不关闭，不管你再打开多少个 Sheet，当这个 Sheet 动作时，它就会显示出它继承的那个菜单；当你关闭了这个 Sheet 然后再打开它时，它将继承当前的菜单。简言之，sheet 打开时，继承当前打开窗口的菜单。

细心的读者可能会想到，如果在 MDI 窗口的菜单项上编写了打开多个 Sheet 的程序，那么，当带菜单的 Sheet 被打开后，其菜单替换了 MDI 窗口上的菜单，这时要想再打开其它的 Sheet 就成为了不可能的事情了。那么，怎样解决这个问题呢，这就要借助基于菜单的工具栏了，请看下一部分“工具栏”讲解。

1.4 MDI 窗口的工具栏

在使用 MDI 风格的应用时，你可能常常会觉得仅仅有下拉、级联菜单还不够，要是能有 PowerBuilder 上的那种工具栏会更好、更方便。在 MDI 应用中，你可以享用 Power Builder 自动创建、管理的基于你的菜单的工具栏（在定义菜单时，指定了工具栏上的按钮）。使用这种工具栏，你无需重定制客户区域的尺寸。你也可以创建自己的定制工具栏（通常作为用户对象来使用），然而在这种情况下需要自己定制客户区域尺寸，就象前面列举过的带一排“命令”按钮的例子那样。如果需要，还可以使这两种工具栏并存，不过就更要注意客户区域的尺寸定制了，否则将不会按你想象的那样显示。这三种情况的举例见下图：



基于菜单的工具栏　　客户工具栏　　两种工具栏并存

定制工具栏的建立取决于你的想象和习惯，只要定制好客户区域的尺寸就可以使用，这里不再赘述。下面就这种基于菜单的工具栏，讲述 PowerBuilder 创建它的方法和特点。

这种工具栏与我们在 Power Builder 上见过的工具栏风格相同，使用方法也相同。由上图可以看出它位于菜单栏的下面，工具栏中的每个图标按钮对应于菜单中的某个菜单项，因此称它为基于菜单的工具栏。当你为 MDI 窗口创建了工具栏，你的用户就可以象使用 Power Builder 的工具栏那样，按自己的习惯将工具栏显示在 MDI 窗口的上端、下端、左端、右端或悬浮。使用这种基于菜单的工具栏，无需人来管理，Power Builder 会自动管理这种工具栏的。

这种工具栏的创建是用下图标出的“change”按钮完成的，具体步骤请参见本套丛书第一册的“菜单”一章。