

PowerBuilder 6.0/6.5 技术丛书

刘红岩 张步达 主编

PowerBuilder 高级开发技术

袁刚 黄辉 李雷 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要论述如何开发和维护大型的 PowerBuilder 的应用程序。

本书共分四篇、十八章,全面而详细地介绍了 PowerBuilder 基本类库(PFC)、C++ 类构造器以及版本控制系统 ObjectCycle、版本控制接口 PVCS 和 PowerBuilder 中的各种高级分析和开发工具的使用。对于熟练的 PowerBuilder 开发人员来说,使用这些工具能使您的应用程序锦上添花。

本书是数据库应用开发人员首选的使用手册和培训教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

PowerBuilder 高级开发技术/袁刚等编著. - 北京:电子工业出版社,1999.6

(PowerBuilder6.0/6.5 技术丛书)

ISBN 7-5053-5261-X

I. P… II. 袁… III. 数据库系统-软件工具,PowerBuilder IV. TP311.56

中国版本图书馆 CIP 数据核字(1999)第 12530 号

丛 书 名: PowerBuilder 6.0/6.5 技术丛书

书 名: PowerBuilder 高级开发技术

主 编: 刘红岩 张步达

编 著: 袁刚 黄辉 李雷

策 划: 郭 立

责任编辑: 黄志瑜

特约编辑: 陈 熠

排版制作: 电子工业出版社计算机排版室

印 刷 者: 北京东光印刷厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 18.75 字数: 480 千字

版 次: 1999 年 6 月第 1 版 1999 年 8 月第 2 次印刷

书 号: ISBN 7-5053-5261-X
TP·2629

印 数: 4000 册 定价: 36.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

《PowerBuilder 6.0/6.5 技术丛书》编委会

顾 问：徐国祥 SYBASE(中国)公司副总经理 技术总监
杨孝如 SYBASE(中国)公司高级工程师

主 编：刘红岩 张步达

编 委：何 军 王 蓉 肖 斌 杨 慧 石京民
邵建生 龚伟志 孟凡军 袁 刚 黄 辉
李 雷 王怀君 刘红丽 何雅丽 戴建荣
何 杰 王 宇 陈 青

策 划：郭 立

前 言

一、PowerBuilder 6.0/6.5 的优异性能

应用软件的开发生产率与计算机硬件的巨大进步相比,已成为限制计算机应用发展的瓶颈。因而,从事应用系统开发的广大软件人员迫切需要适用、高效的工具。作为一种可视化的、面向对象的快速应用开发(RAD)工具,PowerBuilder 已被广大数据库应用开发人员所使用,并获得好评。相对于其他任何应用开发工具,PowerBuilder 可以使开发人员的工作速度更快、成本更低、质量更高、功能更强。PowerBuilder 全面支持面向对象编程;是集成强大并易于使用的第四代编程语言(4GL);内置包括数据窗口在内的多种对象类,可以方便地访问数据库。

1994 年推出的 PowerBuilder 4.0 和 1996 年推出的 PowerBuilder 5.0 版非常适用于 Client/Server 结构的应用系统,但是,随着应用系统结构的发展,对于目前的 Web 应用结构和瘦客户端的应用模式来说具有一定的局限。SYBASE 公司在 PowerBuilder 6.0 中做了很大改进,提供了强大的组件生成器,主要是提供了多种方式以支持 Web 应用。PowerBuilder 还支持 Java Proxy 组件,这大大扩展了 PowerBuilder 的适用范围,PowerBuilder 6.0 几乎支持目前通用的 Web 应用模式,但在客户端编程方面有些不足,PowerBuilder 6.5 在这方面又有了很大进步。PowerBuilder 6.5 几乎完全可在 Web 上进行所有的工作,它在 6.0 的基础上增加了 PowerSite Web 应用开发技术,成为建立 Client/Server 和多层分布式应用的一个完整的 4GL RAD 环境。PowerBuilder 6.5 使用的是业界熟悉的 Web 标准(HTML 和 Java),通过端对端的设计、调试和提交,可以控制 Web 应用的整个生命周期。

二、本套丛书的特点

1996 年 5 月,电子工业出版社出版了我们编写的《PowerBuilder 5.0 原理与应用开发指南》一书。该书较全面地向读者介绍了 PowerBuilder 5.0 的基本概念和主要功能,成为许多开发人员和读者了解和使用 PowerBuilder 的主要参考书,先后重印四次,深受读者欢迎。

面对 PowerBuilder 6.0/6.5 这样一个内容博大精深的应用开发产品,广大应用开发人员迫切需要一套更加系统而全面地介绍 PowerBuilder 的丛书。应广大用户的热情要求,我们编写了这套《PowerBuilder 6.0/6.5 技术丛书》。

本丛书在内容的编写和结构的安排上做了较为精心的设计。如果您不熟悉 PowerBuilder,利用本套丛书能很快入门并循序渐进;如果您对 PowerBuilder 有所了解,使用本套丛书能尽快提高应用水平;如果您长期使用 PowerBuilder,也能从本套丛书中获得很多应用开发的高级技巧。

本丛书包括以下分册:

- 1.《PowerBuilder 原理与应用指南》是利用 PowerBuilder 开发应用程序或学习如何使用 PowerBuilder 时所必读的入门书,该书全面介绍了 PowerBuilder 开发应用的步骤、各种工具和使用方法。

2.《PowerScript 语言、事件与函数》是利用 PowerBuilder 开发应用程序,或遇到诸如 PowerScript 语句、事件或函数的使用而不清楚时,所必备的参考书。

3.《PowerBuilder 应用开发技术详解》针对那些已经掌握用 PowerBuilder 进行应用开发的基本方法的用户,围绕着一组常用的应用实现技术,指导读者进一步掌握开发、分发与维护 PowerBuilder 应用程序的方法。

4.《PowerBuilder 数据窗口技术详解》全面详尽地介绍了开发人员常常涉及到的数据窗口的有关技术,更重要的是,通过对数据窗口技术内部机制的深入剖析和对大量程序片段的解释,使得开发人员能够快速而全面地掌握数据窗口的各种技术。

5.《PowerBuilder 控件技术详解》主要介绍一些功能强大而略微复杂的控件以及其高级使用方法,并介绍如何使用 PowerBuilder 以外的应用程序所提供的控件,如 OLE、Active X 等,来丰富和增强 PowerBuilder 应用程序的功能。

6.《PowerBuilder Internet 技术详解》由浅入深、详尽地介绍了用 PowerBuilder 6.0/6.5 开发 Internet 应用的方法。此外,还花费了相当的篇幅讨论 Web 站点及其应用开发的规划问题,这对打算建立 Web 站点的用户会有较大的帮助。

7.《PowerBuilder 高级开发技术》全面而详细地介绍了 PowerBuilder 基础类库(PFC)、C++ 类构造器以及版本控制系统 ObjectCycle、版本控制接口 PVCS 和 PowerBuilder 中的各种高级分析和开发工具的使用,对于熟练的 PowerBuilder 开发人员来说,使用这些工具能使您的应用程序锦上添花。

三、编写人员简介

这套丛书由刘红岩博士(清华大学经管学院)、张步达先生(北方计算中心)担任主编,何军副教授(中国人民大学数据库所)在本丛书的组织及编写工作中起了重要作用。

本套丛书是在征求了 SYBASE 公司培训和技术支持专家的意见并进行了多次讨论后,由本书主编与电子工业出版社的人士共同策划和组织的。参加本书编写的多数人员是国内应用 PowerBuilder 历史最久的数据库专家,他们在研究、使用和开发 PowerBuilder 中积累了较为丰富的经验。

四、致谢及其他

本套丛书在编写过程中得到了许多 PowerBuilder 用户的热情支持,他们是我们编写这套丛书的动力;SYBASE(中国)公司给我们提供了大量的技术资料和宝贵建议,他们的帮助是高质量编写本套丛书的保证;电子工业出版社的领导和编辑也对本丛书的编写给予了极大的关心,并付出了艰辛的劳动,在此一并表示感谢。

由于时间仓促,书中难免会有不足和错误之处,敬请广大读者提出宝贵意见,以便我们在下一版中修正。

刘红岩 张步达
1999 年 3 月于北京

本书序

随着社会的进步和发展,需要借助计算机来进行管理的事务和现实问题越来越庞杂,从而使得相应的应用系统复杂度也越来越高。如何进一步扩充现有开发工具的开发能力,如何对大型的应用进行团队开发并进行有效地组织和管理以及如何对已完成的程序进行卓有成效的维护和复用,都是摆在开发人员面前的问题。

作为 PowerBuilder 产品系列的一部分,PowerBuilder 基础类库(PFC)、C++ 类构造器以及版本控制系统 ObjectCycle、版本控制接口 PVCS 和 PowerBuilder 中的各种高级分析和开发工具的使用,使得开发和维护大型的 PowerBuilder 的应用程序不那么困难了。本书全面而详细地介绍了这些工具包的使用,对于熟练的 PowerBuilder 开发人员来说,使用这些工具能使您的应用程序锦上添花。

全书共有四篇,18 章。第一篇详述了 PFC 基础类库的概念和使用,着重介绍使用 PowerBuilder 基本类库进行应用程序开发的一些基本知识和常用技巧,并以示例作为主线,介绍了从利用 PFC 进行编程到分发一个 PFC 应用的全过程,PowerBuilder 提供了各种类库和它们的源代码,程序员可以很方便地对类中的程序和控件进行修改和扩充,使之符合自己的需求,从而提高应用开发的效率,增强代码和组件的可重用性;第二篇结合 C++ 的特点详细介绍了 C++ 类构造器的使用和利用 C++ 编程的注意事项,使开发人员能够既充分利用 PowerBuilder 编程方法的简单易用和面向对象的能力,又能使自己的程序具有 C/C++ 语言的灵活、高效的强大功能;第三篇介绍了如何利用版本控制系统进行团队开发和对应用程序进行维护的方法,介绍了 PowerBuilder 的通用版本控制接口 SCC API 和基于网络的版本控制及配置产品 ObjectCycle,以及 PowerBuilder 的 PVCS 与版本控制工具 PVCS 的配置和使用;第四篇介绍了一套扩展 PowerBuilder 开发环境的 PowerBuilder 高级实用开发工具,包括四个分析工具(Cross Reference、DWEAS、DataWindow SQL Verifier 和 PEAR)和两个开发工具(Stored Procedure Update 和 Object Search),这使得对 PowerBuilder 应用程序的开发和管理维护更方便。通过这些工具,可以分析 PowerBuilder 应用中各对象之间的定义和引用参照关系,可以修改数据窗口的扩展属性和 PowerBuilder 库中各表的扩展属性并产生相应的分析报表;当应用程序在不同的 DBMS 之间移植时,可以检查数据窗口的 SQL 语句在新的 DBMS 中的合法性;还可以利用存储过程来实现对数据库的更新,以加强数据库的安全性、完整性和高性能。

目 录

第一篇 利用 PowerBuilder 的基础类库建立应用

第 1 章 概述	(3)
1.1 基础类库与 PowerBuilder	(3)
1.1.1 了解基础类库(PFC)	(3)
1.1.2 了解 PowerBuilder	(7)
1.2 基础类库的面向对象特点	(8)
1.2.1 继承	(9)
1.2.2 封装	(10)
1.2.3 多态性	(10)
1.3 基础类库的扩展层结构及其特性	(10)
1.4 基础类库的部件	(11)
1.4.1 PFC 基础类库的库结构及其存放位置	(12)
1.4.2 PFC 部件简介	(14)
本章小结	(15)
第 2 章 设计 PFC 类库	(16)
2.1 利用 PFC 设计类库	(16)
2.2 选择扩展策略	(16)
2.2.1 创建中间扩展层	(16)
2.2.2 使用已有的 PFC 扩展层	(18)
2.3 定义一个新的服务	(19)
2.3.1 如何定义一个配套的服务(associative service)	(19)
2.3.2 如何定义一个综合的服务(aggregate service)	(20)
2.4 库扩展器	(21)
本章小结	(21)
第 3 章 PFC 的编程技术	(23)
3.1 PFC 编程基础	(23)
3.1.1 建立应用管理器	(23)
3.1.2 应用的创建	(25)
3.1.3 属性对象(attribute object)的使用	(27)
3.1.4 PFC 中的常量	(28)
3.1.5 消息路由(Message Router)	(28)
3.1.6 PFC 的事务对象管理	(30)
3.1.7 调用祖先层的函数和事件	(30)
3.1.8 为应用程序增加联机帮助	(31)

3.1.9 安装 PFC 的升级版	(31)
3.2 使用 PFC 提供的服务	(32)
3.2.1 应用服务	(32)
3.2.2 窗口服务	(34)
3.2.3 数据窗口服务	(35)
3.3 使用 PFC 可视化控件	(39)
3.3.1 使用基础的控制功能	(39)
3.3.2 使用高级的控制功能	(41)
3.3.3 定制可视用户对象的使用	(48)
3.4 使用 PFC 窗口和菜单	(62)
3.4.1 使用 PFC 窗口	(62)
3.4.2 使用 PFC 菜单	(64)
3.5 PFC 实用工具	(67)
3.5.1 数据窗口的属性控制工具	(67)
3.5.2 SQL 监测工具(SQL SPY)	(70)
3.5.3 安全管理(Security)	(71)
3.6 分发一个 PFC 应用	(78)
3.6.1 选择分发策略	(78)
3.6.2 使用 PFC 的 DLL 动态链接库文件	(78)
3.6.3 使用 PFC 的 PSR 资源文件	(79)
3.6.4 配置数据库	(79)
3.6.5 配置 PFC 对话框的帮助文件	(79)
本章小结	(80)
第 4 章 PFC 实例	(81)
4.1 修改应用对象	(81)
4.1.1 更新库的搜索路径	(81)
4.1.2 使用 n_tr 对象与 SQLCA 事务对象建立通信	(83)
4.1.3 创建应用管理器	(84)
4.1.4 定义全局变量并查看事件	(86)
4.2 创建一个框架(frame)窗口	(87)
4.2.1 通过继承 w_frame 创建一个后代 frame 窗口	(88)
4.2.2 添加代码,打开 sheet 窗口	(88)
4.2.3 定义 pfc_Preopen 和 pfc_Posopen 事件的处理程序	(90)
4.2.4 修改应用管理器	(90)
4.2.5 运行应用程序	(91)
4.3 创建菜单	(92)
4.3.1 创建一个后代窗口	(92)
4.3.2 增加和修改菜单项	(93)
4.3.3 创建一个 frame 菜单	(97)
4.3.4 将菜单链接到 frame 窗口上	(98)
4.3.5 为 sheet 窗口 w_produces 创建一个菜单	(100)
4.3.6 为 sheet 窗口 w_produce_report 创建一个菜单	(101)

4.4 建立第一个 sheet 窗口	(102)
4.4.1 通过继承 w_sheet 创建一个后代 sheet 窗口	(102)
4.4.2 增加一个数据窗口控件	(104)
4.4.3 激活数据窗口服务	(106)
4.4.4 检索数据行	(107)
4.4.5 运行应用程序	(107)
4.5 建立第二个 sheet 窗口	(111)
4.5.1 通过继承 w_sheet 创建一个后代 sheet 窗口	(111)
4.5.2 增加一个数据窗口控件	(113)
4.5.3 激活报表和打印预览服务	(116)
4.5.4 检索数据行	(116)
4.5.5 运行应用程序	(116)
本章小结	(118)

第二篇 构造 C++ 类

第 5 章 C++ 类构造器概述	(121)
5.1 C++ 类构造器简介	(121)
5.1.1 C++ 和 PowerBuilder	(121)
5.1.2 C++ Class Builder	(121)
5.2 DLL 及其部件	(122)
5.2.1 什么是 DLL	(122)
5.2.2 DLL 部件	(123)
5.3 内存模型	(124)
5.3.1 代码模型	(124)
5.3.2 数据模型	(124)
本章小节	(125)
第 6 章 创建 C++ 源代码	(126)
6.1 创建 C++ 源代码的方法	(126)
6.1.1 建立用户对象	(126)
6.1.2 声明用户对象函数	(127)
6.1.3 定义变量	(129)
6.1.4 生成 C++ 源代码	(130)
6.1.5 错误处理	(131)
6.2 C++ 文件	(131)
6.3 C/C++ 编程须知	(132)
6.3.1 继承和函数重载	(132)
6.3.2 保留字	(133)
6.3.3 PowerBuilder 和 C++ 中相似的数据类型	(134)
本章小结	(135)
第 7 章 编译和链接 C++ 类	(136)
7.1 IDE 概述	(136)

7.2 使用 IDE	(137)
7.2.1 简介	(137)
7.2.2 编辑 C++ 源代码	(138)
7.2.3 编译和链接 DLL	(140)
7.3 Watcom 调试器	(142)
7.3.1 Watcom Debugger 特点	(143)
7.3.2 Watcom Debugger 菜单项简介	(144)
本章小结	(146)
第 8 章 示例程序	(147)
8.1 介绍	(147)
8.2 C++ 代码	(148)
8.2.1 类层次	(148)
8.2.2 C++ 源代码	(149)
本章小结	(160)

第三篇 利用版本控制系统进行团队开发

第 9 章 版本控制系统概述	(163)
9.1 版本控制系统简介	(163)
9.1.1 版本控制系统(源代码控制系统)	(163)
9.1.2 PowerBuilder 库	(163)
9.1.3 版本及版本标签	(163)
9.1.4 版本控制接口(version control interfaces)	(164)
9.1.5 使用版本控制系统	(165)
9.2 设置环境	(166)
9.2.1 为 PowerBuilder SCC API 设置环境	(166)
9.2.2 为其他版本控制接口设置环境	(166)
9.3 使用 PowerBuilder SCC API 管理版本控制	(166)
9.3.1 版本控制功能	(166)
9.3.2 设置 PowerBuilder SCC API	(167)
9.3.3 使用 PowerBuilder SCC API	(169)
本章小结	(173)
第 10 章 使用 ObjectCycle 进行版本控制	(174)
10.1 ObjectCycle 简介	(174)
10.1.1 什么是 ObjectCycle	(174)
10.1.2 ObjectCycle 接口简介	(175)
10.2 软件安装	(175)
10.2.1 安装配置 Windows 3.11 客户机	(176)
10.2.2 准备安装 ObjectCycle Server	(176)
10.2.3 安装 ObjectCycle Server 软件	(177)
10.2.4 安装 ObjectCycle Manager	(177)

10.3	设置开发项目	(177)
10.4	链接到 ObjectCycle	(178)
10.4.1	初次在 PowerBuilder 中链接 ObjectCycle	(178)
10.4.2	建立并保存配置文件	(179)
10.5	定义项目节点	(180)
10.6	创建工作库	(181)
10.6.1	创建工作库	(181)
10.6.2	向库搜索路径中追加工作库	(182)
10.7	注册 PowerBuilder 对象	(183)
10.7.1	注册对象	(183)
10.7.2	列表显示注册对象	(184)
10.7.3	撤消对象的注册	(184)
10.7.4	打开已注册对象的只读版本	(185)
10.8	ObjectCycle 中对象的检入和检出	(185)
10.8.1	ObjectCycle 中对象的检出	(185)
10.8.2	ObjectCycle 中对象的检入	(187)
10.9	创建一个新版本	(188)
10.9.1	为新文档建立选项	(188)
10.9.2	指定一个开始版本号	(188)
10.9.3	创建版本	(189)
10.10	使用版本标签	(190)
10.10.1	为一组对象指定版本标签	(190)
10.10.2	用版本标签筛选版本列表	(191)
10.10.3	取消筛选	(191)
10.10.4	构建项目时指定版本标签	(192)
10.11	查看对象的版本历史	(192)
10.12	显示报表	(193)
10.12.1	显示文档报表	(193)
10.12.2	显示版本报表	(194)
10.12.3	将一个报表复制到一个文件中	(195)
10.13	恢复早期版本对象	(195)
10.13.1	决定恢复哪个版本	(195)
10.13.2	恢复早期版本	(195)
10.13.3	用版本标签恢复版本	(196)
10.14	恢复库	(196)
10.14.1	在项目中列表显示出对象	(196)
10.14.2	恢复库的方法	(197)
10.14.3	从 ObjectCycle 中检索项目对象	(198)
10.14.4	用新名称保存对象	(200)
10.15	使对象同步化	(200)
10.15.1	同步化的时机	(200)
10.15.2	同步化所作的工作	(200)
本章小结		(201)

第 11 章 使用 PVCS 进行版本控制	(202)
11.1 PVCS 简介	(202)
11.1.1 设置 PVCS 接口概述	(202)
11.1.2 源代码控制活动概述	(202)
11.2 配置 PVCS 接口	(203)
11.2.1 确认安装	(203)
11.2.2 首次链接到 PVCS	(203)
11.2.3 指定一个配置文件	(203)
11.2.4 创建工作库	(205)
11.2.5 注册对象	(206)
11.2.6 查看已注册对象的列表	(208)
11.2.7 清除对象的注册状态	(208)
11.3 PVCS 对象的检入和检出	(209)
11.3.1 检出对象	(209)
11.3.2 修改对象	(210)
11.3.3 检入对象	(211)
11.4 使用版本标签	(212)
11.4.1 为一组对象指定一个版本标签	(212)
11.4.2 使用版本标签筛选版本列表	(213)
11.4.3 使用版本标签检索版本	(213)
11.5 创建新的版本	(213)
11.6 查看对象的变化历史	(215)
11.7 查看和打印报表	(216)
11.7.1 查看或打印一个版本报表	(216)
11.7.2 查看或打印一个文档报表	(216)
11.8 恢复对象的早期版本	(217)
11.8.1 决定恢复哪个版本级别	(218)
11.8.2 进行恢复	(218)
11.9 恢复库	(219)
11.9.1 在一个项目中查看对象的列表	(219)
11.9.2 恢复库的两种方法	(220)
11.10 使对象同步变化	(222)
本章小结	(223)

第四篇 PowerBuilder 高级开发工具

第 12 章 概述	(227)
12.1 简介	(227)
12.2 使用发射平台	(227)
本章小结	(229)
第 13 章 Cross Reference	(230)
13.1 Cross Reference 简介	(230)

13.2 使用 Cross Reference	(231)
13.2.1 设置 Cross Reference 的工作参数	(231)
13.2.2 运行 Cross Reference	(233)
13.2.3 分析 Cross Reference 报表	(235)
13.3 Cross Reference 数据库	(237)
13.3.1 App_classes 表	(237)
13.3.2 App_info 表	(237)
13.3.3 Xref_info 表	(238)
本章小结	(238)
第 14 章 Data Window Extended Attribute Synchronizer	(239)
14.1 概述	(239)
14.2 使用 DWEAS	(240)
14.2.1 设置工作参数	(240)
14.2.2 运行 DWEAS	(241)
14.2.3 分析 DWEAS 报表	(243)
本章小结	(245)
第 15 章 DataWindow SQL Verifier	(246)
15.1 简介	(246)
15.2 使用 DataWindow SQL Verifier	(247)
15.2.1 设置工作参数	(247)
15.2.2 运行 DataWindow SQL Verifier	(249)
15.2.3 分析报表	(251)
本章小结	(252)
第 16 章 PowerBuilder Extended Attribute Reporter	(253)
16.1 简介	(253)
16.2 使用 PEAR	(254)
16.2.1 设置 DBMS 链接参数	(254)
16.2.2 运行 PEAR	(255)
16.2.3 查看 PEAR 报表	(255)
16.2.4 设置数据库参数的缺省值	(257)
本章小结	(258)
第 17 章 Stored Procedure Update	(259)
17.1 简介	(259)
17.1.1 为什么要使用存储过程	(259)
17.1.2 Stored Procedure Update 的工作方式	(259)
17.2 使用 Stored Procedure Update	(261)
17.2.1 设置 DBMS 链接参数	(261)
17.2.2 运行 Stored Procedure Update	(264)
17.2.3 更新数据窗口的 SQLPreview 事件	(272)

本章小结	(272)
第 18 章 Object Search	(273)
18.1 简介	(273)
18.2 使用 Object Search	(274)
18.2.1 运行 Object Search	(274)
18.2.2 查看 Object Search 报表	(276)
18.2.3 设置库选择参数	(277)
本章小结	(278)
附录 C++ 源代码	(279)

第一篇

利用 PowerBuilder 的基础类库建立应用

对于一个数据库应用系统来说,选择 PowerBuilder 开发工具进行程序设计与开发,是一种快速、高效、省时、省力的方法。PowerBuilder 以简洁、直观、实用的编程方式为程序员节约了大量的时间和精力,使得程序员可以把大部分精力放在提高程序执行效率和完善系统功能上,这样做的结果使程序开发时间缩短,开发成本降低,程序编制质量也会大大提高。

开发一个小型的数据库应用系统,使用 PowerBuilder 提供的菜单编辑器、窗口编辑器、数据窗口生成器等可视化编程工具,结合 PowerBuilder Script 脚本语言直接开始程序的编制工作,并不会影响程序的开发效率。然而,对于一个大(中)型数据库应用系统,使用这种原始的编程方式进行开发与充分利用 PowerBuilder 提供的基础类库进行开发相比,在程序开发的时间、效率、成本等方面,都会有明显的差别。

PowerBuilder 基础类库是基于面向对象程序设计的思想而设计的,它包含了大多数应用共有的内容,例如各种形式的菜单、窗口、用户对象、数据窗口以及错误处理、报表预览和行印、主细目录和查询等等。通过继承的方法,程序员可以方便地把这些类加载到自己的程序中去,从而达到提高开发效率的目的。

本篇着重介绍使用 PowerBuilder 基础类库进行应用程序开发的一些基本知识和常用技巧,因此,在阅读本篇内容之前,读者应具备以下能力:对 Windows 下面的窗口编程风格有所了解;使用过或正在使用 PowerBuilder 开发应用程序,理解一些基本概念和专用术语;对数据库管理系统有一定了解。

本篇包括:

- 第 1 章 概述
- 第 2 章 设计 PFC 类库
- 第 3 章 PFC 的编程技术
- 第 4 章 PFC 实例

第 1 章 概 述

1.1 基础类库与 PowerBuilder

1.1.1 了解基础类库(PFC)

PowerBuilder 基础类库(PowerBuilder Foundation Class Library, 简称 PFC)是 PowerBuilder 提供给高级程序员的一组方便、灵活、功能强大的素材库。它采用了先进的 PowerBuilder 面向对象的代码技术和面向服务的设计方法,可使应用程序耗费最少的计算机资源而发挥最大的功效,充分体现了面向对象编程方法所具备的众多优点。在程序的开发过程中,熟练地使用 PFC,可大大提高程序开发效率,节约开发成本。

PFC 的结构是一种基于服务的结构(Service-Based Architecture),PowerBuilder 提供了各种类库和它们的源代码,程序员可以很方便地对类中的程序和控件进行修改和扩充,使之符合自己的需求。使用 PFC 创建 PowerBuilder 的应用程序为我们提供了多方面的优点,如下所示:

(1)显著提高应用开发的效率和准确性。PFC 封装了各种类型的基础类对象,在应用系统的开发中,只需做少许的修改,便可直接将基础类引入应用的设计和实现中,这样可以为程序员节约大量的开发时间,提高开发效率。此外,由于 PFC 的内部设计较为合理,引用 PFC 进行程序设计有助于提高程序的执行效率,增强应用程序的准确性与可靠性。

(2)应用结构使用可扩充的部件取代了以往使用的难以进行扩展的部件。

(3)增强了代码和组件的可重用性。PFC 是一组定制的 PowerBuilder 对象,它包含了各种控件定义在这些控件中的代码,以完成特定的服务。通过继承这些对象,程序员可以使用 PFC 提供的各种对象和服务,而不必重新开发。

(4)一致的命名规则和代码标准。PFC 内部的脚本语言采用了统一的命名规则和编程格式,使程序更加通俗易懂,便于进行修改扩充。命名规则简介如下:

基础类对象的命名规则为: PFC 对象类型_对象名。其中_对象类型的含义如表 1.1 所示。

表 1.1 对象类型的命名

前缀	说 明
m_	菜单(Menu)
n_	标准类用户对象(Standard class user object)