



# 应用程序开发指南：构建和运行应用程序

版本 8.2





# 应用程序开发指南：构建和运行应用程序

版本 8.2

在使用本资料及其支持的产品之前，请务必阅读『声明』中的一般信息。

本文档包含 IBM 的专利信息。它是根据许可协议提供的，并受版权法保护。本出版物包含的信息不包括任何产品保证，且本手册提供的任何声明不应作如此解释。

可以用在线方式或通过您当地的 IBM 代表订购 IBM 出版物。

- 要以在线方式订购出版物，可访问 IBM 出版物中心 (IBM Publications Center)，网址为 [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order)。
- 要查找您当地的 IBM 代表，可访问 IBM 全球联系人目录 (IBM Directory of Worldwide Contacts)，网址为 [www.ibm.com/planetwide](http://www.ibm.com/planetwide)。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

© Copyright International Business Machines Corporation 1993 - 2004. All rights reserved.

# 目录

欢迎使用 DB2 应用程序开发 . . . . .	vii
DB2 开发者版产品 . . . . .	vii
关于本书 . . . . .	viii

## 第 1 部分 应用程序开发环境 . . . . . 1

### 第 1 章 DB2 环境支持 . . . . . 3

DB2 应用程序开发客户机 . . . . .	3
数据库管理器实例 . . . . .	5
DB2 支持的服务器 . . . . .	7
DB2 支持的开发软件 . . . . .	7
AIX 支持的开发软件 . . . . .	8
HP-UX 支持的开发软件 . . . . .	10
Linux 支持的开发软件 . . . . .	12
Solaris 支持的开发软件 . . . . .	17
Windows 支持的开发软件 . . . . .	19

### 第 2 章 设置 . . . . . 23

一般设置信息 . . . . .	23
设置应用程序开发环境 . . . . .	23
重新构建 DB2 例程共享库 . . . . .	24
更新数据库管理器配置文件 . . . . .	25
设置 Java 环境 . . . . .	26
设置 DB2 WebSphere MQ 函数 . . . . .	28
UNIX . . . . .	30
设置 UNIX 应用程序开发环境 . . . . .	30
UNIX 环境变量设置 . . . . .	31
设置 UNIX Java 环境 . . . . .	32
设置 AIX Java 环境 . . . . .	34
设置 HP-UX Java 环境 . . . . .	34
设置 Linux Java 环境 . . . . .	36
设置 Solaris Java 环境 . . . . .	37
Windows . . . . .	38
设置 Windows 应用程序开发环境 . . . . .	38
设置 Windows Java 环境 . . . . .	41
样本数据库 . . . . .	43
设置样本数据库 . . . . .	43
创建样本数据库 . . . . .	44
在主机或 AS/400 和 iSeries 服务器上创建样本数据库 . . . . .	45
对样本数据库进行编目 . . . . .	46
绑定样本数据库实用程序 . . . . .	46
迁移应用程序 . . . . .	48
将应用迁移至 DB2 版本 8 . . . . .	48
迁移 Java 应用程序、例程和 applet . . . . .	49
将应用程序从 32 位环境迁移至 64 位环境 . . . . .	50
确保应用程序可移植性 . . . . .	52
在两个版本的 DB2 上运行应用程序 . . . . .	53
下一步 . . . . .	56

## 第 3 章 样本程序和相关文件 . . . . . 57

样本文件 . . . . .	57
按语言和应用程序接口分类的样本程序 . . . . .	62
C 样本 . . . . .	62
C++ 样本 . . . . .	65
C# 样本 . . . . .	67
CLI 样本 . . . . .	69
命令行处理器 (CLP) 样本 . . . . .	70
COBOL 样本 . . . . .	71
动态重新配置样本 . . . . .	75
JDBC 样本 . . . . .	75
SQLJ 样本 . . . . .	78
Java WebSphere 样本 . . . . .	79
Java 插件样本 . . . . .	80
日志管理用户出口样本 . . . . .	81
对象链接与嵌入 (OLE) 样本 . . . . .	82
对象链接与嵌入数据库 (OLE DB) 表函数样本 . . . . .	83
Perl 样本 . . . . .	83
PHP 样本 . . . . .	84
REXX 样本 . . . . .	85
安全性插件样本 . . . . .	87
SQL 过程样本 . . . . .	87
Visual Basic 样本 . . . . .	89
Visual Basic .NET 样本 . . . . .	91
Visual C++ 样本 . . . . .	92
Windows 管理规范样本 . . . . .	92
构建文件、makefile 和错误检查实用程序 . . . . .	93
构建文件 . . . . .	93
Makefile . . . . .	95
错误检查实用程序 . . . . .	98

## 第 2 部分 构建和运行独立于平台的应用程序 . . . . . 101

### 第 4 章 Java . . . . . 103

Java 样本程序 . . . . .	103
Java applet 注意事项 . . . . .	104
JDBC . . . . .	106
构建 JDBC applet . . . . .	106
构建 JDBC 应用程序 . . . . .	107
构建 JDBC 例程 . . . . .	108
SQLJ . . . . .	110
构建 SQLJ 程序 . . . . .	110
构建 SQLJ applet . . . . .	111
构建 SQLJ 应用程序 . . . . .	112
SQLJ 应用程序和 applet 的 UNIX 构建脚本 . . . . .	113
UNIX 的 SQLJ 应用程序和 applet 选项 . . . . .	115
SQLJ 应用程序和 applet 的 Windows 批处理文件 . . . . .	115
Windows 的 SQLJ 应用程序和 applet 选项 . . . . .	117

构建 SQLJ 例程 . . . . .	117
SQLJ 例程的 UNIX 构建脚本 . . . . .	119
UNIX 的 SQLJ 例程选项 . . . . .	120
SQLJ 例程的 Windows 批处理文件 . . . . .	120
Windows 的 SQLJ 例程选项 . . . . .	122
<b>第 5 章 命令行处理器 . . . . .</b>	<b>123</b>
运行命令行处理器 (CLP) 脚本 . . . . .	123
从命令行处理器 (CLP) 调用过程 . . . . .	124
<b>第 6 章 SQL 过程 . . . . .</b>	<b>127</b>
创建 SQL 过程 . . . . .	127
使用客户机应用程序调用 SQL 过程 . . . . .	128
为 SQL 过程定制预编译和绑定选项 . . . . .	129
备份和复原在 DB2 8.2 之前创建的 SQL 过程 . . . . .	130
重新绑定 SQL 过程 . . . . .	131
<b>第 7 章 Perl . . . . .</b>	<b>133</b>
构建 Perl 应用程序 . . . . .	133
<b>第 8 章 PHP . . . . .</b>	<b>135</b>
构建 PHP 应用程序 . . . . .	135

### 第 3 部分 构建和运行特定于平台的应用程序 . . . . . 139

<b>第 9 章 UNIX . . . . .</b>	<b>141</b>
构建 UNIX C 应用程序 . . . . .	141
构建 UNIX C 多连接应用程序 . . . . .	142
构建 UNIX C 例程 . . . . .	144
构建 UNIX C++ 应用程序 . . . . .	147
构建 UNIX C++ 多连接应用程序 . . . . .	149
构建 UNIX C++ 例程 . . . . .	151
构建 UNIX Micro Focus COBOL 应用程序 . . . . .	154
构建 UNIX Micro Focus COBOL 例程 . . . . .	155
<b>第 10 章 AIX . . . . .</b>	<b>159</b>
重要注意事项 . . . . .	159
例程的 AIX 导出文件 . . . . .	159
AIX 例程和 CREATE 语句 . . . . .	160
替换 AIX 共享库 . . . . .	161
在 AIX 上安装 COBOL 的注意事项 . . . . .	161
IBM C . . . . .	161
C 应用程序的构建脚本 . . . . .	161
AIX C 应用程序编译和链接选项 . . . . .	162
C 例程的构建脚本 . . . . .	163
AIX C 例程编译和链接选项 . . . . .	164
在 AIX 上构建 C 多线程应用程序 . . . . .	165
VisualAge C++ . . . . .	165
C++ 应用程序的构建脚本 . . . . .	166
AIX C++ 应用程序编译和链接选项 . . . . .	166
C++ 例程的构建脚本 . . . . .	167
AIX C++ 例程编译和链接选项 . . . . .	168
在 AIX 上构建 C++ 多线程应用程序 . . . . .	169
VisualAge C++ 配置文件 . . . . .	170

使用配置文件构建 VisualAge C++ 程序 . . . . .	170
使用配置文件构建 C++ DB2 API 应用程序 . . . . .	171
使用配置文件构建 C++ 嵌入式 SQL 应用程序 . . . . .	171
使用配置文件构建 C++ 存储过程 . . . . .	172
使用配置文件构建 C++ 用户定义的函数 . . . . .	173
IBM COBOL Set AIX 版 . . . . .	174
在 AIX 上配置 IBM COBOL 编译器 . . . . .	174
在 AIX 上构建 IBM COBOL 应用程序 . . . . .	175
IBM COBOL 应用程序的构建脚本 . . . . .	176
AIX IBM COBOL 应用程序编译和链接选项 . . . . .	177
在 AIX 上构建 IBM COBOL 例程 . . . . .	178
IBM COBOL 例程的构建脚本 . . . . .	179
AIX IBM COBOL 例程编译和链接选项 . . . . .	179
Micro Focus COBOL . . . . .	180
在 AIX 上配置 Micro Focus COBOL 编译器 . . . . .	180
Micro Focus COBOL 应用程序的构建脚本 . . . . .	181
AIX Micro Focus COBOL 应用程序编译和链接选项 . . . . .	181
Micro Focus COBOL 例程的构建脚本 . . . . .	182
AIX Micro Focus COBOL 例程编译和链接选项 . . . . .	183
REXX . . . . .	184
在 AIX 上构建 REXX 应用程序 . . . . .	184

### 第 11 章 HP-UX . . . . . 187

HP-UX C . . . . .	187
C 应用程序的构建脚本 . . . . .	187
HP-UX C 应用程序编译和链接选项 . . . . .	188
C 例程的构建脚本 . . . . .	189
HP-UX C 例程编译和链接选项 . . . . .	190
在 HP-UX 上构建 C 多线程应用程序 . . . . .	191
HP-UX C++ . . . . .	192
C++ 应用程序的构建脚本 . . . . .	192
HP-UX C++ 应用程序编译和链接选项 . . . . .	193
C++ 例程的构建脚本 . . . . .	195
HP-UX C++ 例程编译和链接选项 . . . . .	196
在 HP-UX 上构建 C++ 多线程应用程序 . . . . .	197
Micro Focus COBOL . . . . .	198
在 HP-UX 上构建 Micro Focus COBOL 编译器 . . . . .	198
Micro Focus COBOL 应用程序的构建脚本 . . . . .	199
HP-UX Micro Focus COBOL 应用程序编译和链接选项 . . . . .	199
Micro Focus COBOL 例程的构建脚本 . . . . .	200
HP-UX Micro Focus COBOL 例程编译和链接选项 . . . . .	200

### 第 12 章 Linux . . . . . 203

Linux C . . . . .	203
C 应用程序的构建脚本 . . . . .	203
Linux C 应用程序编译和链接选项 . . . . .	204
C 例程的构建脚本 . . . . .	205
Linux C 例程编译和链接选项 . . . . .	206
在 Linux 上构建 C 多线程应用程序 . . . . .	207
Linux C++ . . . . .	208
C++ 应用程序的构建脚本 . . . . .	208
Linux C++ 应用程序编译和链接选项 . . . . .	209
C++ 例程的构建脚本 . . . . .	210

Linux C++ 例程编译和链接选项 . . . . .	210
在 Linux 上构建 C++ 多线程应用程序 . . . . .	211
Micro Focus COBOL . . . . .	212
在 Linux 上配置 Micro Focus COBOL 编译器 . . . . .	212
Micro Focus COBOL 应用程序的构建脚本 . . . . .	214
Linux Micro Focus COBOL 应用程序编译和链接选项 . . . . .	214
Micro Focus COBOL 例程的构建脚本 . . . . .	215
Linux Micro Focus COBOL 例程编译和链接选项 . . . . .	215

## 第 13 章 Solaris . . . . . 217

Solaris C . . . . .	217
C 应用程序的构建脚本 . . . . .	217
Solaris C 应用程序编译和链接选项 . . . . .	218
C 例程的构建脚本 . . . . .	219
Solaris C 例程编译和链接选项 . . . . .	220
在 Solaris 上构建 C 多线程应用程序 . . . . .	221
Solaris C++ . . . . .	222
C++ 应用程序的构建脚本 . . . . .	222
Solaris C++ 应用程序编译和链接选项 . . . . .	223
C++ 例程的构建脚本 . . . . .	224
Solaris C++ 例程编译和链接选项 . . . . .	225
在 Solaris 上构建 C++ 多线程应用程序 . . . . .	226
Micro Focus COBOL . . . . .	227
在 Solaris 上配置 Micro Focus COBOL 编译器 . . . . .	227
Micro Focus COBOL 应用程序的构建脚本 . . . . .	228
Solaris Micro Focus COBOL 应用程序编译和链接选项 . . . . .	228
Micro Focus COBOL 例程的构建脚本 . . . . .	229
Solaris Micro Focus COBOL 例程编译和链接选项 . . . . .	230

## 第 14 章 Windows . . . . . 231

WCHARTYPE CONVERT 预编译选项 . . . . .	231
对象链接与嵌入数据库 (OLE DB) 表函数 . . . . .	232
Windows 管理规范 (WMI) . . . . .	233
Microsoft Visual Basic . . . . .	233
使用 Visual Basic 构建 ADO 应用程序 . . . . .	233
使用 Visual Basic 构建松散耦合事务 . . . . .	236
Visual Basic 松散耦合事务项目故障诊断 . . . . .	237
使用 Visual Basic 构建 RDO 应用程序 . . . . .	238
使用 Visual Basic 的对象链接与嵌入 (OLE) 自动化 . . . . .	240
.NET . . . . .	240
构建 C# .NET 应用程序 . . . . .	240
C# .NET 应用程序的批处理文件 . . . . .	241
C# .NET 应用程序编译和链接选项 . . . . .	242
构建 Visual Basic .NET 应用程序 . . . . .	243
Visual Basic .NET 应用程序的批处理文件 . . . . .	244
Visual Basic .NET 应用程序编译和链接选项 . . . . .	245
构建公共语言运行时 (CLR) .NET 例程 . . . . .	246
C# .NET 例程的批处理文件 . . . . .	249
Visual Basic .NET 例程的批处理文件 . . . . .	250
CLR .NET 例程编译和链接选项 . . . . .	250
Microsoft Visual C++ . . . . .	251
使用 Visual C++ 构建 ADO 应用程序 . . . . .	251

使用 Visual C++ 的对象链接与嵌入 (OLE) 自动化 . . . . .	253
在 Windows 上构建 C/C++ 应用程序 . . . . .	254
C/C++ 应用程序的批处理文件 . . . . .	256
Windows C/C++ 应用程序编译和链接选项 . . . . .	256
在 Windows 上构建 C/C++ 例程 . . . . .	257
C/C++ 例程的批处理文件 . . . . .	260
Windows C/C++ 例程编译和链接选项 . . . . .	261
在 Windows 上构建 C/C++ 多连接应用程序 . . . . .	262
IBM VisualAge COBOL . . . . .	264
在 Windows 上配置 IBM COBOL 编译器 . . . . .	264
在 Windows 上构建 IBM COBOL 应用程序 . . . . .	264
IBM COBOL 应用程序的批处理文件 . . . . .	266
Windows IBM COBOL 应用程序编译和链接选项 . . . . .	267
在 Windows 上构建 IBM COBOL 例程 . . . . .	268
IBM COBOL 例程的批处理文件 . . . . .	269
Windows IBM COBOL 例程编译和链接选项 . . . . .	270
Micro Focus COBOL . . . . .	271
在 Windows 上配置 Micro Focus COBOL 编译器 . . . . .	271
在 Windows 上构建 Micro Focus COBOL 应用程序 . . . . .	272
Micro Focus COBOL 应用程序的批处理文件 . . . . .	273
Windows Micro Focus COBOL 应用程序编译和链接选项 . . . . .	273
在 Windows 上构建 Micro Focus COBOL 例程 . . . . .	274
Micro Focus COBOL 例程的批处理文件 . . . . .	275
Windows Micro Focus COBOL 例程编译和链接选项 . . . . .	276
Object REXX . . . . .	276
在 Windows 上构建 Object REXX 应用程序 . . . . .	276

## 第 4 部分 附录 . . . . . 279

### 附录 A. DB2 通用数据库技术信息 . . . . . 281

DB2 文档和帮助 . . . . .	281
DB2 文档更新 . . . . .	281
DB2 信息中心 . . . . .	282
DB2 信息中心安装方案 . . . . .	283
使用“DB2 安装”向导来安装 DB2 信息中心 (UNIX) . . . . .	285
使用“DB2 安装”向导来安装 DB2 信息中心 (Windows) . . . . .	287
调用 DB2 信息中心 . . . . .	289
更新安装在计算机或内部网服务器上的 DB2 信息中心 . . . . .	290
以首选语言显示 DB2 信息中心中的主题 . . . . .	290
DB2 PDF 和印刷文档 . . . . .	291
核心 DB2 信息 . . . . .	291
管理信息 . . . . .	292
应用程序开发信息 . . . . .	292
商业智能信息 . . . . .	293
DB2 Connect 信息 . . . . .	293
入门信息 . . . . .	294
教程信息 . . . . .	294
可选组件信息 . . . . .	295

发行说明 . . . . .	295	与辅助技术的兼容性 . . . . .	301
从 PDF 文件打印 DB2 书籍 . . . . .	296	文档的辅助功能 . . . . .	301
订购印刷的 DB2 书籍 . . . . .	296	点分十进制语法图 . . . . .	302
从 DB2 工具调用上下文帮助 . . . . .	297	DB2 通用数据库产品的 Common Criteria 认证 . . . . .	303
从命令行处理器调用消息帮助 . . . . .	298	<b>附录 B. 声明 . . . . .</b>	<b>305</b>
从命令行处理器调用命令帮助 . . . . .	298	商标 . . . . .	307
从命令行处理器调用 SQL 状态帮助 . . . . .	299	<b>索引 . . . . .</b>	<b>309</b>
DB2 教程 . . . . .	299	<b>与 IBM 联系 . . . . .</b>	<b>315</b>
DB2 故障诊断信息 . . . . .	300	产品信息 . . . . .	315
辅助功能 . . . . .	300		
键盘输入和导航 . . . . .	301		
界面显示的辅助功能 . . . . .	301		



---

## 欢迎使用 DB2 应用程序开发

DB2 开发者版产品 . . . . . vii | 关于本书 . . . . . viii

本前言提供了着手进行 DB2 应用程序开发（尤其是 DB2 开发者版产品的开发）所需的信息。它还提供了所有三卷《应用程序开发指南》的概述。

有关如何安装随 DB2 开发者版提供的产品的指示信息，参阅相应的《快速入门》一书（可从 PDF CD 获取），或者检查产品 CD 本身以获取安装指示信息。

如果您想要访问您的计算机上的 DB2 文档但尚未安装 DB2 信息中心，则参阅第 285 页的『使用“DB2 安装”向导来安装 DB2 信息中心（UNIX）』或第 287 页的『使用“DB2 安装”向导来安装 DB2 信息中心（Windows）』。DB2 信息中心包含 DB2 通用数据库和 DB2 相关产品的文档。

---

## DB2 开发者版产品

DB2<sup>®</sup> 通用数据库提供了两个用于应用程序开发的产品程序包：DB2 个人开发者版和 DB2 通用开发者版。个人开发者版提供了在 Linux 和 Windows<sup>®</sup> 操作系统上运行的 DB2 Universal Database<sup>™</sup>（DB2 通用数据库）和 DB2 Connect<sup>™</sup> 个人版产品。DB2 通用开发者版提供了在这些平台以及 AIX<sup>®</sup>、HP-UX 和 Solaris Operating Environment 上运行的 DB2 产品。请与 IBM<sup>®</sup> 代表联系以获取有关受支持平台的完整列表。

使用这些产品附带的软件，您可以开发和测试在一个操作系统上运行而存取在同一操作系统或另一操作系统上的数据库的应用程序。例如，您可以创建在 Windows 操作系统上运行但存取在 UNIX<sup>®</sup> 平台（如 AIX）上的数据库的应用程序。有关使用开发者版产品的条款和条件，请参阅“许可协议”。

个人开发者版包含几个 CD-ROM，提供了开发和测试应用程序所需的所有代码。在每个“介质包”中，将可找到：

- 用于 Linux 和 Windows 操作系统的 DB2 通用数据库产品 CD-ROM。每个 CD-ROM 包含用于受支持操作系统的 DB2 服务器和“应用程序开发客户机”。这些 CD-ROM 仅供用于测试应用程序。如果需要安装并使用数据库，则必须通过购买通用数据库产品来获得有效的许可证。
- DB2 Connect 个人版。如果需要安装并使用此产品，必须通过购买 DB2 Connect 个人版来获得有效的许可证。
- 包含 PDF 格式的 DB2 书籍的 DB2 出版物 CD-ROM
- 包含 HTML 格式的 DB2 文档的 DB2 信息中心 CD-ROM
- DB2 Net Search Extender（仅适用于 Windows）
- DB2 Spatial Extender（仅适用于 Windows）
- VisualAge<sup>®</sup> for Java<sup>™</sup>, Entry Edition

通用开发者版包含 DB2 支持的所有操作系统的 CD-ROM，并包括下列内容：

- DB2 通用数据库个人版、工作组服务器版和企业服务器版
- DB2 Connect 个人版和 DB2 Connect 企业版

- 所有平台的管理客户机。这些客户机包含用于管理数据库的工具，如控制中心和事件分析器。这些客户机还允许您在任何系统上运行应用程序。
- 所有平台的应用程序开发客户机。这些客户机具有应用程序开发工具、样本程序和头文件。每个 DB2 AD 客户机都包含了开发应用程序所需的所有内容。
- 所有平台的运行时客户机。可在任何系统上从运行时客户机运行应用程序。运行时客户机没有管理客户机的某些功能部件，如 DB2 控制中心和事件分析器，因此占用的空间会小一些。
- 包含 PDF 格式的 DB2 书籍的 DB2 出版物 CD-ROM
- 包含 HTML 格式的 DB2 文档的 DB2 信息中心 CD-ROM
- DB2 Net Search Extender
- DB2 Spatial Extender

另外，两个开发者版都提供了对于开发应用程序非常有用的其它软件。此软件可能会经常变更，同时可使用的许可协议也会发生变化。

#### 相关概念:

- 『DB2 UDB 企业服务器版』 (《DB2 服务器快速入门》)
- 『DB2 运行时客户机』 (《DB2 客户机快速入门》)
- 『DB2 管理客户机』 (《DB2 客户机快速入门》)
- 『DB2 工作组服务器版』 (《DB2 服务器快速入门》)

#### 相关任务:

- 『安装 DB2 个人版 - 概述 (Windows)』 (《DB2 个人版快速入门》)
- 『安装 DB2 个人版 - 概述 (Linux)』 (《DB2 个人版快速入门》)

#### 相关参考:

- 第 3 页的『DB2 应用程序开发客户机』

---

## 关于本书

《应用程序开发指南》这一本书包含三卷，描述您需要了解的有关编写、调试、构建和运行 DB2 应用程序的信息:

- *Application Development Guide: Programming Client Applications* 包含您需要了解的有关编写在 DB2 客户机上运行的独立 DB2 应用程序的信息。它包含有关下列各项的信息:
  - DB2 支持的编程接口。为下列各项提供了高级描述: DB2 开发者版、受支持的编程接口、用来创建 Web 应用程序的设施以及 DB2 提供的编程功能部件，如例程和触发器。
  - DB2 应用程序应遵循的一般结构。对如何维护数据库中的数据值与关系提出了建议，描述了权限注意事项，并提供了有关如何测试和调试应用程序的信息。
  - 嵌入式 SQL，包括动态和静态 SQL。描述了嵌入式 SQL 的一般注意事项，以及适用于在 DB2 应用程序中使用静态和动态 SQL 的特定问题。
  - 受支持的主语言和解释语言，如 C/C++、COBOL、Perl 和 REXX，以及如何在使用这些语言编写的应用程序中使用嵌入式 SQL。
  - DB2 .NET Data Provider 以及 OLE DB .NET Data Provider 和 ODBC .NET Data Provider。

- Java (JDBC 和 SQLJ) 以及构建 Java 应用程序以供在 WebSphere Application Server 上使用注意事项。
- IBM OLE DB Provider for DB2 Server。提供有关对 OLE DB 服务、组件和属性的 IBM OLE DB Provider 支持的一般信息。同时提供了有关对 “ActiveX 数据对象” (ADO) 使用 OLE DB 接口的 Visual Basic 和 Visual C++ 应用程序的特定信息。
- 本地语言支持问题。描述一般主题，如整理顺序、派生代码页和语言环境，以及字符转换。同时描述了其它特定问题，如 DBCS 代码页、EUC 字符集，以及适用于日语和繁体中文 EUC 和 UCS-2 环境的问题。
- 事务管理。描述适用于执行多站点更新的应用程序以及执行并发事务的应用程序的问题。
- 分区数据库环境中的应用程序。描述分区数据库环境中的直接 DSS、本地忽略、缓冲插入以及故障诊断应用程序。
- 经常使用的应用程序技巧。提供有关如何使用生成列和标识列、已声明临时表以及如何使用保存点来管理事务的信息。
- 支持在嵌入式 SQL 应用程序中使用的 SQL 语句。
- 访问主机和 iSeries 环境的应用程序。描述有关访问主机和 iSeries 环境的嵌入式 SQL 应用程序的问题。
- EBCDIC 二进制文件整理的仿真。
- *Application Development Guide: Programming Server Applications* 包含您需要了解的有关使用服务器端对象 (包括例程、大对象、用户定义的类型和触发器) 进行编程的信息。它包含有关下列各项的信息：
  - 例程 (存储过程、用户定义的函数和方法)，包括：
    - 例程性能、安全性、库管理注意事项和限制。
    - 创建例程 (包括外部例程) 和 CREATE 语句。
    - 过程参数方式和参数控制。
    - 过程结果集。
    - SQL 过程，包括调试和条件控制。
    - 用户定义的标量函数和表函数。
    - 用户定义的标量函数调用和表函数调用 (FIRST 调用，FINAL 调用，...) 以及暂存区。
    - 方法。
    - 权限和外部例程的绑定。
    - 有关 C、Java、.NET 公共语言运行时和 OLE 自动例程的特定于语言的注意事项。
    - 调用例程。
    - 函数选择。
    - 将单值类型和 LOB 传送至函数。
    - 代码页和例程。
  - 大对象，包括 LOB 的用法和定位器、引用变量和 CLOB 数据。

- 用户定义的单值类型，包括强类型、定义和删除 UDT、创建带有结构化类型的表、对特定应用程序使用单值类型和类型表、处理单值类型及其之间的转型，以及执行单值类型的比较与赋值，包括对相异类型列的 UNION 运算。
- 用户定义的结构化类型，包括存储实例和实例化、结构化类型层次结构、定义结构化类型行为、方法的动态分派、结构化类型的比较、转型、构造函数以及 mutator 和 observer 方法。
- 类型表，包括创建、删除、替代、存储对象、定义系统生成的对象标识以及对对象标识列的约束。
- 引用类型，包括类型表中的各个对象之间的关系、引用之间的语义关系以及引用完整性与范围引用。
- 类型表和类型视图，包括结构化类型（如列类型）、变换函数与变换组、主语言程序映射以及结构化类型主变量。
- 触发器，包括 INSERT、UPDATE 和 DELETE 触发器，与引用约束之间的交互作用、创建基准、粒度、激活时间、转换变量和表、触发操作、多个触发器以及触发器、约束和例程之间的协作。
- 《应用程序开发指南：构建和运行应用程序》包含您需要了解的有关在 DB2 支持的操作系统上构建和运行 DB2 应用程序的信息。
  - AIX
  - HP-UX
  - Linux
  - Solaris
  - Windows
 它包含有关下列各项的信息：
  - DB2 支持的用来构建应用程序的服务器和软件，包括支持的编译器和解释器。
  - DB2 样本程序文件、makefile、构建文件以及错误检查实用程序文件。
  - 如何设置应用程序开发环境，包括有关 Java 函数和 WebSphere MQ 函数的特定指示信息。
  - 如果设置样本数据库。
  - 如何从 DB2 的先前版本迁移应用程序。
  - 如何构建和运行 Java applet、应用程序和例程。
  - 如何构建和运行 SQL 过程。
  - 如何构建和运行 C/C++ 应用程序和例程。
  - 如何构建和运行 IBM 和 Micro Focus COBOL 应用程序和例程。
  - 如何在 AIX 和 Windows 上构建和运行 REXX 应用程序。
  - 如何在 Windows 上构建和运行 C# 应用程序、Visual Basic .NET 应用程序和 CLR .NET 例程。
  - 如何在 Windows 上使用 Visual Basic 和 Visual C++ 对“ActiveX 数据对象”（ADO）构建和运行应用程序。
  - 如何在 Windows 上使用 Visual C++ 对远程数据对象构建和运行应用程序。

---

## 第 1 部分 应用程序开发环境



---

## 第 1 章 DB2 环境支持

DB2 应用程序开发客户机 . . . . .	3	HP-UX 支持的开发软件 . . . . .	10
数据库管理器实例. . . . .	5	Linux 支持的开发软件 . . . . .	12
DB2 支持的服务器 . . . . .	7	Solaris 支持的开发软件 . . . . .	17
DB2 支持的开发软件. . . . .	7	Windows 支持的开发软件. . . . .	19
AIX 支持的开发软件. . . . .	8		

《应用程序开发指南》的这一卷描述了对应用程序开发的 DB2 支持。它提供了设置开发 DB2 应用程序的环境所需的信息，并给出了在此环境中编译、链接和运行这些应用程序的逐步指示信息。它说明如何在下列平台上使用“DB2 应用程序开发 (DB2 AD) 客户机” DB2 通用数据库版本 8.2 来构建应用程序：

- AIX
- HP-UX
- Linux
- Solaris Operating Environment
- Windows

---

### DB2 应用程序开发客户机

“DB2 应用程序开发 (DB2 AD) 客户机”提供了工具和环境支持，以便开发访问实现“分布式关系数据库体系结构” (DRDA) 的 DB2 服务器和应用程序服务器的应用程序。

可使用安装的 DB2 AD 客户机来构建和运行 DB2 应用程序。也可在以下 DB2 客户机上运行 DB2 应用程序：

- DB2 运行时客户机
- DB2 管理客户机

适用于受支持平台的 DB2 AD 客户机包括：

- **C/C++、COBOL 和 Fortran 的预编译器**（假定该平台支持该语言）。
- **嵌入式 SQL 应用程序支持**（包括编程库）包含文件和代码样本。
- **DB2 调用级接口 (DB2 CLI) 应用程序支持**（包括编程库）包含文件和代码样本，它们用于开发易于移植到 ODBC 并可用 ODBC SDK 编译的应用程序。可从 Microsoft 公司获得用于 Windows 操作系统的 ODBC SDK，而从其它供应商那里获得用于许多其它受支持平台的 ODBC SDK。对于 Windows 操作系统，DB2 客户机包含一个 ODBC 驱动程序，它支持用 Microsoft ODBC Software Developer's Kit 开发的应用程序。对于所有其它平台，DB2 客户机包含一个可选择安装的 ODBC 驱动程序，它支持可用该平台的 ODBC SDK 开发的应用程序（如果有的话）。只有用于 Windows 操作系统的“DB2 客户机”才包含 ODBC 驱动程序管理器。
- **DB2 Java Enablement**，包括用于开发 Java 应用程序和 applet 的“DB2 Java 数据库连接” (DB2 JDBC) 支持，以及用于开发 Java 嵌入式 SQL 应用程序和 applet 的 DB2 Java 嵌入式 SQL (DB2 SQLJ) 支持。

- 随 DB2 一起为所有受支持的操作系统提供了 **Java Development Kit** 或者同等产品。有关特定版本的详细信息，请参阅适用于您的操作系统的『受支持的开发软件』主题：
  - 第 8 页的『AIX 支持的开发软件』
  - 第 10 页的『HP-UX 支持的开发软件』
  - 第 12 页的『Linux 支持的开发软件』
  - 第 17 页的『Solaris 支持的开发软件』
  - 第 19 页的『Windows 支持的开发软件』
- Windows 操作系统上的 **ActiveX 数据对象 (ADO)** 和“对象链接与嵌入” (**OLE**) **自动化例程 (UDF 和存储过程)**，包括用 Microsoft Visual Basic 和 Microsoft Visual C++ 实现的代码样本。还有使用“远程数据对象” (RDO) 的代码样本，RDO 是用 Microsoft Visual Basic 实现的。
- Windows 操作系统上的**对象链接与嵌入数据库 (OLE DB)** 表函数。
- Windows 操作系统上的 **C# 应用程序、Visual Basic .NET 应用程序和 CLR .NET 例程**。
- **DB2 开发中心**，支持例程（存储过程和用户定义的函数）快速开发和结构类型的图形应用程序。开发中心提供了一个开发环境，它支持从工作站到 z/OS 的整个 DB2 系列。可以将开发中心作为独立应用程序启动，或者从 DB2 通用数据库中心（如控制中心、命令编辑器或任务中心）启动它。开发中心是使用 Java 来实现的，且所有数据库连接都是通过使用“Java 数据库连接 (JDBC) API”来管理的。开发中心还为下列每一开发环境提供了一个 DB2 Development Add-In：
  - Microsoft Visual C++ V6
  - Microsoft Visual Basic V6
  - Microsoft Visual InterDev V6
- **交互式 SQL**，可通过命令编辑器或命令行处理器 (CLP) 使用它创建 SQL 语句的原型或对数据库执行立即查询。
- **一组已归档的 API**，它允许其它应用程序开发工具直接在其产品内实现 DB2 的预编译器支持。例如，AIX 上的 IBM COBOL 将使用此接口。有关“预编译器服务 API”组的信息，可从 DB2 应用程序开发 Web 站点提供的 PDF 文件 prepapi.pdf 中获得，网址为：
 

<http://www.ibm.com/software/data/db2/udb/ad>
- **SQL92 和 MVS 一致性标志**，它标识应用程序中不符合“ISO/ANSI SQL92 入门级”标准或不受 DB2 UDB z/OS 和 OS/390 版支持的嵌入式 SQL 语句。如果将在某台工作站上开发的应用程序迁移到另一个平台，则该标志会说明它们的语法是否兼容以节省您的时间。

#### 相关参考：

- 『PRECOMPILE Command』 (*Command Reference*)
- 第 8 页的『AIX 支持的开发软件』
- 第 10 页的『HP-UX 支持的开发软件』
- 第 12 页的『Linux 支持的开发软件』
- 第 17 页的『Solaris 支持的开发软件』
- 第 19 页的『Windows 支持的开发软件』



## 数据库管理器实例

DB2® 支持在同一机器上有多个数据库管理器实例。数据库管理器实例有它自己的配置文件、目录和数据库。

每个数据库管理器实例可管理几个数据库。但是，一个给定的数据库只属于一个实例。下图显示了此关系。

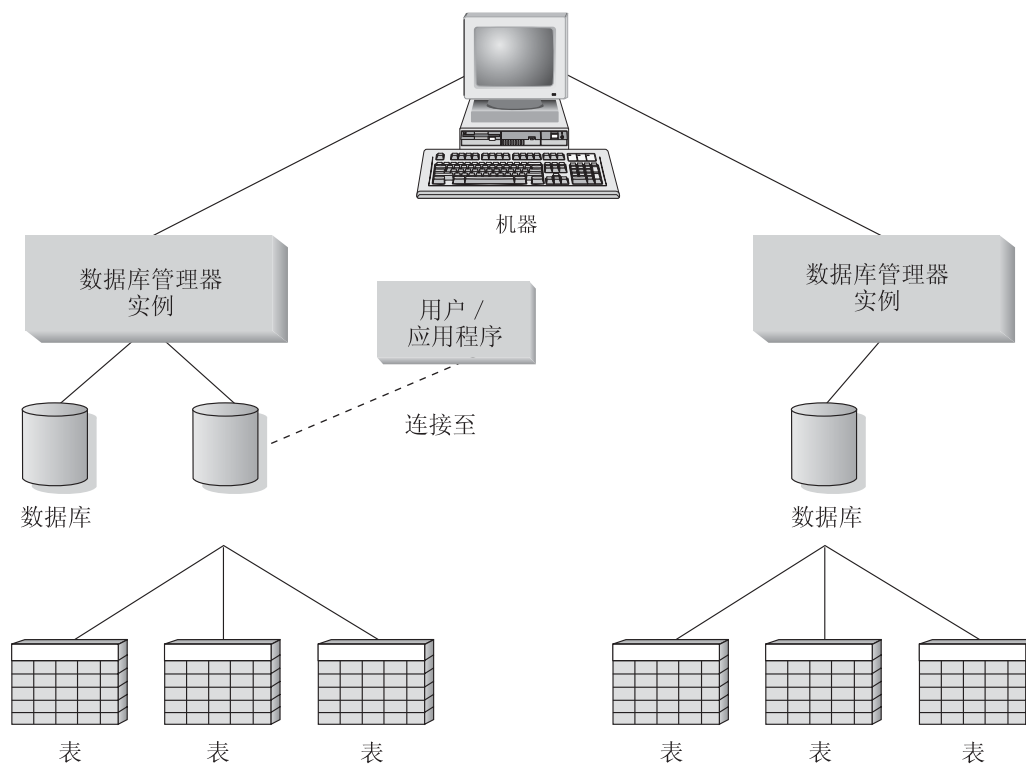


图 1. 数据库管理器实例

数据库管理器实例增大了灵活性，使您在同一机器上可有多个数据库环境。例如，可有一个数据库管理器实例用于开发环境，而另一个实例用于生产环境。

借助 UNIX® Enterprise Server Edition (ESE) 服务器，可以在不同的数据库管理器实例上具有不同的 DB2 版本。例如，您可以有一个数据库管理器实例运行 DB2 通用数据库版本 7.1，而另一个运行 DB2 通用数据库版本 8.2。在 DB2 版本 8 之前，在一个版本级别内只支持一个发行版和修订包级别。例如，DB2 版本 7.1 和 DB2 版本 7.2 不能在一个 UNIX 服务器上共存。对于 DB2 版本 8，多个修订包级别可共存于同一个 UNIX 服务器上，例如 DB2 版本 8.1 和 DB2 版本 8.2。

对于 Windows® 服务器，在每个数据库管理器实例上必须具有相同的 DB2 版本、发行版和修订包级别。不能让一个数据库管理器实例运行 DB2 通用数据库版本 7.1，而另一个实例运行 DB2 通用数据库版本 8.2。

对于所使用的每个实例，您需要了解下列信息：

**实例名** 对于 UNIX 平台，这是在创建数据库管理器实例时指定的有效用户名。

对于 Windows 操作系统，这是一个最多为 8 个字符的字母数字字符串。在安装期间会为您创建一个名为“DB2”的实例。

## 实例目录

实例所在的主目录。

对于 UNIX 平台，实例目录是 \$HOME/sql1lib，其中 \$HOME 是实例所有者的主目录。

对于 Windows 操作系统，实例目录为 %DB2PATH%\instance\_name。变量 %DB2PATH% 确定将 DB2 安装在何处。%DB2PATH% 的缺省安装值是 \Program Files\IBM\SQLLIB，所以，根据 DB2 安装在哪一个驱动器，%DB2PATH% 将指向 drive:\Program Files\IBM\SQLLIB（除非更改了缺省值）。

根据下列内容创建 Windows 服务器上的实例路径：

```
%DB2PATH%\%DB2INSTANCE%
```

（例如，C:\Program Files\IBM\SQLLIB\DB2）

或者，如果定义了 DB2INSTPROF，则根据以下内容创建：

```
%DB2INSTPROF%\%DB2INSTANCE%
```

（例如，C:\PROFILES\DB2）

DB2INSTPROF 环境变量在 Windows 服务器上用来支持在客户机对其具有只读存取权的网络驱动器上运行 DB2。在此情况下，DB2 将设置为指向 drive:\Program Files\IBM\SQLLIB，且 DB2INSTPROF 将设置为指向本地路径（例如，C:\PROFILES），该路径将包含所有特定于实例的信息（如目录和配置），因为 DB2 需要对这些文件有更新存取权。

## 相关概念：

- 『数据库管理器的多个实例』（《管理指南：实现》）
- 『UNIX 操作系统上的多个实例』（《管理指南：实现》）
- 『Windows 操作系统上的多个实例』（《管理指南：实现》）

## 相关任务：

- 第 25 页的『更新数据库管理器配置文件』
- 『设置当前实例』（《管理指南：实现》）
- 『创建附加实例』（《管理指南：实现》）

## 相关参考：

- 『GET DATABASE MANAGER CONFIGURATION Command』（*Command Reference*）
- 『UPDATE DATABASE MANAGER CONFIGURATION Command』（*Command Reference*）

---

## DB2 支持的服务器

可以使用 DB2 AD 客户机来开发将在特定操作系统上运行的应用程序。然而，您的应用程序可以存取下列操作系统服务器上的远程数据库：

- DB2 AIX 版
- DB2 HP-UX 版
- DB2 Linux 版
- DB2 OS/2 版
- DB2 NUMA-Q 版
- DB2 Solaris 版
- DB2 Windows NT 版
- DB2 Windows 2000 版
- DB2 Windows XP 版
- DB2 Windows Server 2003 版
- 符合“分布式关系数据库体系结构”（DRDA）的应用程序服务器，例如：
  - DB2 z/OS 和 OS/390 版
  - DB2 AS/400 版和 iSeries 版
  - DB2 服务器 VSE 版和 VM 版（以前是 SQL/DS VM 和 VSE 版）
  - 由 IBM 之外的数据库开发商提供的与 DRDA 兼容的应用程序服务器。

注：

1. DB2 版本 8 HP-UX 64 位服务器不支持运行 DB2 版本 7 的 64 位本地应用程序。
2. DB2 OS/2 版不可用于 DB2 版本 8。
3. DB2 NUMA-Q 版在 PTX 操作系统上运行，且只可用于 DB2 版本 7。
4. DB2 版本 8 Windows 64 位服务器支持来自 DB2 版本 6 和版本 7 的 32 位客户机的连接（仅对于 SQL 请求）。不支持来自版本 7 的 64 位客户机的连接。

相关任务：

- 『配置对服务器数据库的远程存取』（《安装与配置补充手册》）

相关参考：

- 第 7 页的『DB2 支持的开发软件』

---

## DB2 支持的开发软件

DB2 版本 8 支持下列操作系统的编译器、解释器和相关软件：

- AIX
- HP-UX
- Linux
- Solaris Operating Environment
- Windows

DB2 支持以上每个操作系统的 32 位和 64 位版本。大多数情况下，在这些操作系统上的 32 位与 64 位环境中构建应用程序会有些区别。DB2 不支持在 64 位操作系统环境中运行 COBOL 应用程序或例程（存储过程和用户定义的函数）。对于其它语言，DB2 支持在所有受支持的 64 位操作系统环境（除了 Linux IA64 和 Linux zSeries 之外）上运行 32 位应用程序和例程。

除了 Windows 上的 IBM COBOL 之外，为以上每个操作系统给出的编译器信息假定：您正在对该操作系统使用 DB2 预编译器，而不是使用可能已构建到其中一个所列示编译器中的预编译器支持。对于 Windows 上的 IBM COBOL，DB2 支持 IBM COBOL 预编译器和 DB2 预编译器。

有关最新 DB2 编译器信息和相关软件更新内容，可访问 DB2 应用程序开发 Web 站点：

<http://www.ibm.com/software/data/db2/udb/ad>

关于软件支持，要注意以下几点：

- **Fortran 和 REXX** 在 DB2 通用数据库版本 5.2 中，DB2 对 Fortran 和 REXX 的功能部件的增强将不超过这两种语言的支持级别。
- **Perl** 在交付印刷时，“Perl 数据库接口”（Perl DBI）版本 0.93 或更新版本的 DB2 UDB 驱动程序（DBD::DB2）发行版 0.76 可用于 AIX、HP-UX、Linux、Solaris 和 Windows。最新的驱动程序可从以下站点下载：

<http://www.ibm.com/software/data/db2/perl>

- **PHP** PHP 可用作一种从基于 Web 的应用程序或命令行存取 DB2 的方法，并且可用于 AIX、HP-UX、Linux、Solaris 和 Windows。在交付印刷时，最新版本是 PHP 4.3.4。可从以下站点下载 PHP 的最新版本：

<http://www.php.net>

**相关参考：**

- 第 8 页的『AIX 支持的开发软件』
- 第 10 页的『HP-UX 支持的开发软件』
- 第 12 页的『Linux 支持的开发软件』
- 第 17 页的『Solaris 支持的开发软件』
- 第 19 页的『Windows 支持的开发软件』

---

## AIX 支持的开发软件

DB2 AIX 版支持下列操作系统：

### **AIX V4.3.3**（仅适用于 32 位）

具有维护级别 11

加上 2003 年 3 月发布的 C++ 运行时 PTF（请参阅下面的 C++ 部分以获取下载链接）和：

对于 **JFS** 文件系统：

APAR IY49385

对于 **Java**：

- OpenGL.OpenGL\_X.rte.base
- OpenGL.OpenGL\_X.rte.soft
- X11.adt.lib

### **AIX V5.1.0**（32 位和 64 位）

具有维护级别 5

加上 2003 年 3 月发布的 C++ 运行时 PTF（请参阅下面的 C++ 部分以获取下载链接），以及：

对于 **JFS** 文件系统：

APAR IY48735

对于 **JFS2** 文件系统：

APAR IY49254

对于 **Java**：

建议的维护包 AIX 5100-04 和 APAR IY46667

对于运行超过 **1000** 个 **db2agent**：

APAR IY49220，并在 db2start 之前或在 AIX 引导时指定 “vmtune -T 0”

### **AIX V5.2.0 (32 位和 64 位)**

具有维护级别 2，以及：

对于已安装并发 **I/O (CIO)** 和直接 **I/O** 的卷：

APARs IY49129 和 IY49346

对于 **JFS** 文件系统：

APAR IY48339

对于 **JFS2** 文件系统：

APAR IY49304

对于 **Java**：

建议的维护包 AIX 5200-01 和 APAR IY46668

对于运行超过 **1000** 个 **db2agent** 及使用 **32** 位 **AIX** 内核：

APAR IY49885，并在 db2start 之前或在 AIX 引导时指定 “vmo -o pta\_balance\_threshold=0”

注：您可以查询您的系统以查看是否使用以下命令安装了特定 APAR：

```
instfix -v -i -k <APAR>
```

例如：instfix -v -i -k IY31254

DB2 AIX 版支持下列编程语言和编译器：

**C** IBM C for AIX V5.0.2.3

IBM C for AIX V6.0

**C++** 带有 2003 年 3 月发布的 C++ 运行时 PTF 的 IBM VisualAge C++ V5.0.2.3

<http://www-1.ibm.com/support/docview.wss?rs=0&q=x1C.rte&uid=swg24004427>

带有 2003 年 3 月发布的 C++ 运行时 PTF 的 IBM VisualAge C++ V6.0:

<http://www-1.ibm.com/support/docview.wss?rs=0&q=x1C.rte&uid=swg24004427>

### **COBOL**

IBM COBOL Set AIX 版版本 1.1

带有 Service Pack 1 的 Micro Focus COBOL Server Express V2.2

注：COBOL 支持仅适用于 32 位。

## Fortran

IBM XL Fortran for AIX V4.1 (仅适用于 32 位) 和 V5.1.0 (适用于 32 位和 64 位)

**Java** IBM 的 Java Developer Kit V1.3.1 和 Java 运行时环境版本 1.3.1 AIX 版

IBM Developer Kit for AIX, Java Technology Edition, V1.4.1 Service Release 1 (仅适用于 AIX 5.1 和 5.2)

**注:** 如果尚未安装 Developer Kit 的最新受支持版本, 则将安装它, 除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装, 则必须从 CD-ROM 手工安装受支持的 Developer Kit。

**Perl** Perl 5.004\_04 或更高版本以及 DBI 0.93 或更高版本

**PHP** PHP 4.3.4 或更新版本

**REXX** IBM AIX REXX/6000 AISPO 产品号: 5764-057

IBM Object REXX for AIX V1.1

REXXSAA 4.00

**注:** REXX 支持仅适用于 32 位。

有关 DB2 AIX 版软件支持更新的内容, 请访问 DB2 应用程序开发 Web 站点:

<http://www.ibm.com/software/data/db2/udb/ad>

## 相关参考:

- 第 7 页的『DB2 支持的开发软件』
- 第 162 页的『AIX C 应用程序编译和链接选项』
- 第 164 页的『AIX C 例程编译和链接选项』
- 第 166 页的『AIX C++ 应用程序编译和链接选项』
- 第 168 页的『AIX C++ 例程编译和链接选项』
- 第 177 页的『AIX IBM COBOL 应用程序编译和链接选项』
- 第 179 页的『AIX IBM COBOL 例程编译和链接选项』
- 第 181 页的『AIX Micro Focus COBOL 应用程序编译和链接选项』
- 第 183 页的『AIX Micro Focus COBOL 例程编译和链接选项』
- 『DB2 服务器的安装需求 (AIX)』 (《DB2 服务器快速入门》)

---

## HP-UX 支持的开发软件

DB2 HP-UX 版支持以下操作系统:

### HP-UX on PA-RISC

具有以下项的 HP-UX V11i(11.11) for PA-RISC 2:

2003 年 6 月发布的 GOLDBASE11i 和 2003 年 6 月发布的 GOLDAPPS11i 捆绑软件以及补丁 PHSS\_26560、PHKL\_28489、PHCO\_27434 和 PHCO\_29960

### HP-UX on IA64

带有补丁 PHKL\_30065 的 HP-UX V11i V2 (B.11.23) for Intel Itanium

注: 不支持尝试运行用 aries 二进制转换程序为 PA-RISC 构建的 HP-UX on IA64 的 DB2 应用程序或例程。

**DB2 HP-UX on PA-RISC 版支持下列编程语言和编译器:**

**C** HP C Compiler VB.11.11.02

**C++** HP aC++ VA.03.52

**COBOL**

带有 Service Pack 1 的 Micro Focus COBOL Server Express V2.2

注: COBOL 支持仅适用于 32 位。

**Fortran**

HP-UX f90 B.11.01.06

**Java** HP-UX 32 位: Hewlett-Packard 公司的用于 HP-UX 11.0 和 11i PA-RISC 的 Software Developer's Kit and Runtime Environment 1.3.1 和 1.4.2.01

HP-UX 64 位: Hewlett-Packard 公司的用于 HP-UX 11.0 和 11i PA-RISC 的 Software Developer's Kit and Runtime Environment 1.4.2.01

注:

1. 只有 Software Developer's Kit 1.3.1 支持 32 位 Java 例程。
2. 如果尚未安装 Developer Kit 的最新受支持版本, 则将安装它, 除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装, 则必须从 CD-ROM 手工安装 Developer Kit。

**Perl** Perl 5.004\_04 或更新版本以及 DBI 0.93 或更新版本

**PHP** PHP 4.3.4 或更新版本

**DB2 HP-UX on IA64 版支持下列编程语言和编译器:**

**C** HP C Compiler VA.05.52

**C++** HP (aCC) aC++ VA.05.52

注: DB2 HP-UX on IA64 版不支持用建议不要使用的选项 -AP 构建的或依赖于 libstd (而不是 libstd\_v2) 的任何 C++ 应用程序或第三方 C++ 库。

**Fortran**

HP-UX F90 B.11.23

注:

1. 要生成 64 位代码, 需要 +DD64 编译选项。
2. 要生成 32 位代码, 需要 +DD32 编译选项。

**Java** 用于 HP-UX 11 PA-RISC 和基于 Itanium 的系统的 Software Developer's Kit and Runtime Environment 1.4.2.01 或更高版本

注: 如果尚未安装 Developer Kit 的受支持版本, 则将安装它, 除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装, 则必须将 Developer Kit 从 CD-ROM 手工安装到 /opt/java1.4 中。

**Perl** Perl 5.004\_04 或更新版本以及 DBI 0.93 或更新版本

**PHP** PHP 4.3.4 或更新版本

有关 DB2 HP-UX 版软件支持更新的内容, 请访问 DB2 应用程序开发 Web 站点:

<http://www.ibm.com/software/data/db2/udb/ad>

**相关参考:**

- 第 7 页的『DB2 支持的开发软件』
- 第 188 页的『HP-UX C 应用程序编译和链接选项』
- 第 190 页的『HP-UX C 例程编译和链接选项』
- 第 193 页的『HP-UX C++ 应用程序编译和链接选项』
- 第 196 页的『HP-UX C++ 例程编译和链接选项』
- 第 199 页的『HP-UX Micro Focus COBOL 应用程序编译和链接选项』
- 第 200 页的『HP-UX Micro Focus COBOL 例程编译和链接选项』
- 『DB2 服务器的安装需求 (HP-UX)』 (《DB2 服务器快速入门》)

---

## Linux 支持的开发软件

DB2 Linux 版支持以下操作系统体系结构:

- **Linux on Intel x86 (32 位)**
- **Linux on s/390 and zSeries**
- **Linux on AMD64**
- **Linux on PowerPC**
- **Linux on IA64**

有关受支持分发以及这些体系结构的每一个的内核级别和库级别的信息, 请访问:

<http://www.ibm.com/db2/linux/validate>

下表描述在交付印刷时的 DB2 Linux 体系结构支持。强烈建议您检查上述验证 Web 站点以获取对此支持的更新:

表 1. Linux on Intel x86 (32 位)

分发	内核	库	注释
Conectiva Linux Enterprise Edition (CLEE)	2.4.19	glibc 2.2.5	Powered by United Linux 1.0
LINX Rocky Secure Server 2.1	2.4.21	glibc 2.2.5	
Red Flag Advanced Server 4.0	2.4.21-as.2	glibc 2.2.93-5	
Red Flag Function Server 4.0	2.4.20-8smp	glibc 2.2.93-5	
Red Hat Enterprise Linux 2.1 AS/ES/WS	2.4.9-e16	glibc 2.2.4	
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	
SCO Linux 4.0	2.4.19	glibc 2.2.5	Powered by United Linux 1.0
SuSE Pro 8.0	2.4.18	glibc 2.2.5	



表 1. Linux on Intel x86 (32 位) (续)

分发	内核	库	注释
SuSE Pro 8.1	2.4.19	glibc 2.2.5	
SuSE Linux Enterprise Server (SLES) 7	2.4.7	glibc 2.2.2	
SuSE Linux Enterprise Server (SLES) 8	2.4.19	glibc 2.2.5	验证至 SuSE Service Pack 2 级别
Turbolinux 7 Server	2.4.9	glibc 2.2.4	
Turbolinux 8 Server	2.4.18-5	glibc 2.2.5	
Turbolinux Enterprise Server 8	2.4.19	glibc 2.2.5	
United Linux 1.0	2.4.19	glibc 2.2.5	

表 2. Linux on Intel x86 (32 位) 非企业分发 (承包商不再支持)

分发	内核	库	注释
Red Hat 7.2	2.4.9-34	glibc 2.2.4	
Red Hat 7.3	2.4.18	glibc 2.2.5	
Red Hat 8.0	2.4.18-14	glibc 2.2.93-5	
SuSE 7.3	2.4.10	glibc 2.2.4	

表 3. Linux on s/390 and zSeries (在 s/390 上支持 31 位内核版本; 在 zSeries 上支持 64 位内核版本)

分发	内核	库	注释
Red Hat 7.2	2.4.9-38	glibc 2.2.4	
SuSE Linux Enterprise Server (SLES) 7	2.4.7-58	glibc 2.2.4	compat.rpm 包含 libstdc++ 6.1。使用 JDK 1.3.1 SR 1 for Java
SuSE Linux Enterprise Server (SLES) 8	2.4.19	glibc 2.2.5	Powered by United Linux 1.0
Turbo Linux Enterprise Server (TLES) 8	2.4.19	glibc 2.2.5	Powered by United Linux 1.0
United Linux 1.0	2.4.19	glibc 2.2.5	

表 4. Linux on AMD64

分发	内核	库	注释
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	
SuSE Linux Enterprise Server (SLES) 8.0	2.4.19-SMP	glibc 2.2.5-16	

表 5. Linux on PowerPC (iSeries 和 pSeries)

分发	内核	库	注释
Red Hat Enterprise Linux (RHEL) 3 AS	2.4.21-7.EL	glibc-2.3.2-95.3	

表 5. Linux on PowerPC (iSeries 和 pSeries) (续)

分发	内核	库	注释
SuSE Enterprise Server (SLES) 8	2.4.19-16	glibc 2.2.5	Powered by United Linux 1.0
Turbolinux Enterprise Server 8	2.4.19-16	glibc 2.2.5	Powered by United Linux 1.0
United Linux 1.0	2.4.19	glibc 2.2.5	

表 6. Linux on IA64

分发	内核	库	注释
Red Hat Enterprise Linux 2.1 AS/ES/WS	2.4.18-e.12smp	glibc	
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	
SuSE Linux Enterprise Server (SLES) 8	2.4.19-SMP	glibc 2.2.5	Powered by United Linux 1.0
United Linux 1.0	2.4.19	glibc 2.2.5	

用于 Intel x86 的 DB2 Linux 版支持下列编程语言和编译器:

**C** GNU/Linux gcc V2.95.3 和 V2.96

**C++** GNU/Linux g++ V2.95.3 和 V2.96

**COBOL**

带有 Service Pack 1 的 Micro Focus COBOL Server Express V2.2

**Java** IBM Developer Kit and Runtime Environment for Linux, Java 2 Technology Edition, V1.3.1 和 1.4.1 Service Release 1, 32 位版本

注: 如果尚未安装 Developer Kit 的最新受支持版本, 则将安装它, 除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装, 则必须从 CD-ROM 手工安装受支持的 Developer Kit。

**Perl** Perl 5.004\_04 或更高版本以及 DBI 0.93 或更高版本

**PHP** PHP 4.3.4 或更新版本

**REXX** Object REXX Interpreter for Linux V2.1

**DB2 Linux on s/390 版 或 DB2 Linux on zSeries 版** 上的 32 位实例支持下列编程语言和编译器:

**C** GNU/Linux gcc V2.95.3

**C++** GNU/Linux g++ V2.95.3

**COBOL**

带有 Service Pack 1 的 Micro Focus COBOL Server Express V2.2

**Java** IBM zSeries Developer Kit for Linux Java 2 Technology Edition (级别为 Sun 1.3.1 和 1.4.1 Service Release 1 SDK)

注：如果尚未安装 Developer Kit 的最新受支持版本，则将安装它，除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装，则必须从 CD-ROM 手工安装受支持的 Developer Kit。

**Perl** Perl 5.004\_04 或更高版本以及 DBI 0.93 或更高版本

**PHP** PHP 4.3.4 或更新版本

**REXX** Object REXX 2.2.0 for Linux/390

**DB2 Linux on zSeries** 版上的 64 位实例支持下列编程语言和编译器：

**C** GNU/Linux gcc V3.2

**C++** GNU/Linux g++ V3.2

**Java** IBM zSeries Developer Kit for Linux, Java 2 Technology Edition (级别为 Sun 1.4.1 Service Release 1 SDK)

注：如果尚未安装 Developer Kit 的最新受支持版本，则将安装它，除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装，则必须从 CD-ROM 手工安装受支持的 Developer Kit。

**Perl** Perl 5.8

**DB2 Linux on AMD64** 版上的 32 位实例支持下列编程语言和编译器：

**C** GNU/Linux gcc V3.2 和 V3.3

注：“-m32”编译器选项必须用来生成 32 位应用程序或例程（存储过程和用户定义的函数）。

**C++** GNU/Linux g++ V3.2 和 V3.3

注：

1. 这些版本的 GNU/Linux g++ 编译器不接受某些 fstream 函数的整数参数。有关更多信息，请参阅编译器文档。
2. “-m32”编译器选项必须用来生成 32 位应用程序或例程（存储过程和用户定义的函数）。

**Java** IBM Developer Kit and Runtime Environment for Linux x86, Java 2 Technology Edition, V1.3.1 Service Release 4, 32 位版本和 V1.4.1 Service Release 1, 32 位版本。

注：

1. 如果尚未安装 Developer Kit 的最新受支持版本，则将安装它，除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装，则必须从 CD-ROM 手工安装受支持的 Developer Kit。
2. 如果已经安装了由 SuSE SLES 8 提供的 Developer Kit 1.3.1 Service Release，则在安装 DB2 之前必须卸载它，否则 DB2 将不能安装建议的 Developer Kit。如果已经安装了 DB2，但是未卸载由 SuSE SLES 8 提供的 Developer Kit，则请参阅自述文件中的指示信息来手工更新 Developer Kit。

**Perl** Perl 5.8

**PHP** PHP 4.3.4 或更新版本

**DB2 Linux on AMD64** 版上的 64 位实例支持下列编程语言和编译器：

**C** GNU/Linux gcc V3.2 和 V3.3

**C++** GNU/Linux g++ V3.2 和 V3.3

注: 这些版本的 GNU/Linux g++ 编译器不接受某些 fstream 函数的整数参数。  
有关更多信息, 请参阅编译器文档。

**Java** DB2 当前不支持任何 64 位 Java Developer Kit for Linux on AMD64。

**Perl** Perl 5.8

**PHP** PHP 4.3.4 或更新版本

**DB2 Linux on PowerPC** 版支持下列编程语言和编译器:

**C** GNU/Linux gcc V3.2

**C++** GNU/Linux g++ V3.2

注: 此版本的 GNU/Linux g++ 编译器对于某些 fstream 函数不接受整数参数。  
请参阅编译器文档以获得更多信息。

**Java** 对于 32 位实例: 用于 PowerPC 的 IBM Developer Kit and Runtime Environment for Linux Java 2 Technology Edition, V1.3.1 和 1.4.1 Service Release 1, 32 位版本。

对于 64 位实例: 用于 PowerPC 的 IBM Developer Kit and Runtime Environment for Linux Java 2 Technology Edition, V1.4.1 Service Release 1, 64 位版本。

注: 如果尚未安装 Developer Kit 的最新受支持版本, 则将安装它, 除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装, 则必须从 CD-ROM 手工安装受支持的 Developer Kit。

**Perl** Perl 5.8

**PHP** PHP 4.3.4 或更新版本

**DB2 Linux on IA64** 版支持下列编程语言和编译器:

**C** GNU/Linux gcc V3.2

**C++** GNU/Linux g++ V3.2

注: 此版本的 GNU/Linux g++ 编译器对于某些 fstream 函数不接受整数参数。  
请参阅编译器文档以获得更多信息。

**Java** IBM Developer Kit and Runtime Environment for Linux, Java 2 Technology Edition, V1.3.1, 64 位版本。要使用此 Developer Kit, 还必须安装 gcc 3.2 和 gcc3 libstdc++ 运行时库。

注: 如果尚未安装 Developer Kit, 则将安装它, 除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装, 则必须从 CD-ROM 手工安装 Developer Kit。

**Perl** Perl 5.8

注: Linux IA64 不支持运行 DB2 32 位应用程序或例程 (存储过程和用户定义的函数)。

有关 DB2 Linux 版软件支持更新的内容, 请访问 DB2 应用程序开发 Web 站点:

<http://www.ibm.com/software/data/db2/udb/ad>

**相关参考:**

- 第 7 页的『DB2 支持的开发软件』
- 第 204 页的『Linux C 应用程序编译和链接选项』
- 第 206 页的『Linux C 例程编译和链接选项』
- 第 209 页的『Linux C++ 应用程序编译和链接选项』
- 第 210 页的『Linux C++ 例程编译和链接选项』
- 『DB2 个人版的安装需求 (Linux)』 (《DB2 个人版快速入门》)
- 『DB2 服务器的安装需求 (Linux)』 (《DB2 服务器快速入门》)
- 第 214 页的『Linux Micro Focus COBOL 应用程序编译和链接选项』
- 第 215 页的『Linux Micro Focus COBOL 例程编译和链接选项』

---

## Solaris 支持的开发软件

DB2 Solaris 版支持下列操作系统:

**Solaris**

DB2 工作组服务器版在下列 Solaris Operating Environment 版本上受支持:

Solaris 7 (32 位), 带有 “Recommended & Security Patches” + 107226-17 + 107153-01 + 106327-10

Solaris 8 (32 位), 带有 “Recommended & Security Patches” + 108921-12 + 108940-24 + 108434-03 + 108528-12

Solaris 9 (32 位)

DB2 企业服务器版在下列 Solaris Operating Environment 版本上受支持:

Solaris 7 (32 位), 带有 “Recommended & Security Patches” + 107226-17 + 107153-01 + 106327-10

Solaris 7 (64 位), 带有 “Recommended & Security Patches” + 107226-17 + 107153-01 + 106300-11 + 106327-10

Solaris 8 (32 位), 带有 “Recommended & Security Patches” + 108921-12 + 108940-24 + 108434-03 + 108528-12

Solaris 8 (64 位), 带有 “Recommended & Security Patches” + 108921-12 + 108940-24 + 108435-03 + 108434-03 + 108528-12

Solaris 9 (32 位)

Solaris 9 (64 位)

“Recommended & Security Patches” 可从以下站点获得:

<http://sunsolve.sun.com>

从 SunSolve Online Web 站点上, 单击左面板中的 “Patches” 菜单项并从 “Downloads” 部分中选择 “Recommended & Security Patches”。

还需要 J2SE Solaris Patch Clusters。可从 <http://sunsolve.sun.com> Web 站点获得它们。从 SunSolve Online Web 站点上，单击左面板中的“Patches”菜单项并从“Browse &Download Patches”部分中选择“Recommended Patch Clusters”。

需要 SUNWlibC 软件来在 Solaris 上安装 DB2。

对于 64 位 Fujitsu PRIMEPOWER 系统上的 DB2，还需要下列软件：

- Solaris 8 内核更新补丁 108528-16 或更新版本，以获取对补丁 912040-01 的修正
- Solaris 9 内核更新补丁 112233-01 或更新版本，以获取对补丁 912041-01 的修正

可以通过以下链接从 FTSI 中下载用于 Solaris 的 Fujitsu PRIMEPOWER 补丁：

<https://download.ftsi.fujitsu.com/>

DB2 Solaris 版支持下列编程语言和编译器：

**C** Forte C V5.0、V6、V6.1 和 V6.2  
Sun ONE Studio 7 和 8 Compiler Collection

**C++** Forte C++ V5.0、V6、V6.1 和 V6.2  
Sun ONE Studio 7 和 8 Compiler Collection

#### **COBOL**

带有 Service Pack 1 的 Micro Focus COBOL Server Express V2.2

注：COBOL 支持仅适用于 32 位。

#### **Fortran**

SPARCompiler Fortran V4.2 和 V5.0

**Java** Solaris 32 位：Sun 公司的 Java Development Kit (JDK) V1.3.1 和 1.4.2 for Solaris

Solaris 64 位：Sun 公司的 Java Development Kit (JDK) V1.4.2 for Solaris

注：如果尚未安装 JDK 的最新受支持版本，则将安装它，除非此 DB2 安装是对先前 DB2 版本 8 安装的更新。如果要更新先前的 DB2 版本 8 安装，则必须从 CD-ROM 手工安装 JDK。

**Perl** Perl 5.004\_04 或更高版本以及 DBI 0.93 或更高版本

**PHP** PHP 4.3.4 或更新版本

有关 DB2 Solaris 版软件支持更新的内容，请访问 DB2 应用程序开发 Web 站点：

<http://www.ibm.com/software/data/db2/udb/ad>

#### **相关参考：**

- 第 7 页的『DB2 支持的开发软件』
- 第 218 页的『Solaris C 应用程序编译和链接选项』
- 第 220 页的『Solaris C 例程编译和链接选项』
- 第 223 页的『Solaris C++ 应用程序编译和链接选项』

- 第 225 页的『Solaris C++ 例程编译和链接选项』
- 第 228 页的『Solaris Micro Focus COBOL 应用程序编译和链接选项』
- 第 230 页的『Solaris Micro Focus COBOL 例程编译和链接选项』
- 『DB2 服务器的安装需求 (Solaris Operating Environment)』 (《DB2 服务器快速入门》)

---

## Windows 支持的开发软件

**DB2 Windows 版 32 位操作系统支持下列操作系统:**

### Microsoft Windows XP

### Microsoft Windows Server 2003

### Microsoft Windows 2000

Service Pack 2 对于 Windows Terminal Server 是必需的。

### Microsoft Windows NT

带有 Service Pack 6a 或更高版本的 V4.0。

### Microsoft Windows ME

### Microsoft Windows 98

DB2 Windows 版 32 位操作系统支持下列编程语言:

#### **Basic** Microsoft Visual Basic 6.0 Professional Edition

Microsoft Visual Basic .NET 7.0 和 7.1 (分别适用于 Microsoft .NET Framework V1.0 和 V1.1)

**注:** 在使用 DB2 Install 程序来安装 DB2 .NET Data Provider 之前必须安装 .NET Framework。

#### **C#** Microsoft Visual C# .NET Compiler V7.0 和 V7.1 (分别适用于 Microsoft .NET Framework V1.0 和 V1.1)

**注:** 在使用 DB2 Install 程序来安装 DB2 .NET Data Provider 之前必须安装 .NET Framework。

#### **C/C++** Microsoft Visual C++ V6.0

Microsoft Visual C++ .NET 2002 和 2003

Intel C++ Compiler for 32-bit applications V6 或更高版本

#### **COBOL**

Micro Focus COBOL V4.0.20

Micro Focus COBOL Net Express V3.1.0

IBM VisualAge COBOL V3.0.4 或更高版本

#### **Java** IBM Developer Kit and Runtime Environment for Windows Java 2 Technology Edition V1.3.1 和 1.4.1 Service Release 1, 32 位版本

**注:** 如果尚未安装 Developer Kit 的最新受支持版本, 则将安装它。

Sun 公司的 Java Development Kit (JDK) 1.3.1 for Win32

**Perl** Perl 5.004\_04 和 DBI 0.93

**PHP** PHP 4.3.4 或更新版本

**REXX** IBM Object REXX for Windows NT/95 V1.1

有关获取 IBM Object REXX for Windows 的信息, 请访问:

<http://www.ibm.com/software/ad/obj-rexx/>

**Microsoft Windows 脚本编制主机**

版本 5.1

**注:** 对于 Windows 2000 和 Windows XP: 要运行让 COM+ 对象使用 ODBC 的 DB2 应用程序或运行将 OLE DB 提供程序用于 ODBC 且禁用 OLE DB 资源合用的应用程序, 必须使用 Windows 2000 service pack 3 或 Windows XP service pack 1。如果不确定应用程序环境是否合适, 建议您安装适当的 Windows 服务级别。有关更多信息, 参阅以下 Microsoft Knowledge Base 文章:

<http://support.microsoft.com/default.aspx?scid=KB;EN-US;306414>

对于 DB2 服务器本身或作为 DB2 产品一部分交付的应用程序, 这些服务包不是必需的。

**DB2 Windows 版 64 位操作系统支持下列操作系统:**

**Microsoft Windows XP 64-bit Edition**

**Microsoft Windows Server 2003**

DB2 Windows 版 64 位操作系统支持下列编程语言:

**C/C++** 带有 Microsoft Software Developer's Kit 3790 或更新版本的 Intel C++ Compiler for Itanium V7.1。

用于 Intel Itanium 体系结构的 Microsoft 的 C/C++ 编译器 (通过 Microsoft Software Developer's Kit 3790 提供)

**Java** IBM Developer Kit and Runtime Environment for Windows, Java 2 Technology Edition, V1.4.1 Service Release 1, 64 位版本

**注:** 如果尚未安装 Developer Kit, 则 DB2 将安装它。

**Perl** Perl 5.004\_04 和 DBI 0.93

**PHP** PHP 4.3.4 或更新版本

**Microsoft Windows 脚本编制主机**

版本 5.1

**注:** Windows Server 2003 支持包括下列各项:

- Windows Server 2003, Standard Edition
- Windows Server 2003, Enterprise Edition
- Windows Server 2003, Datacenter Edition

有关 DB2 Windows 版软件支持更新的内容, 请访问 DB2 应用程序开发 Web 站点:

<http://www.ibm.com/software/data/db2/udb/ad>

**相关参考:**



- 第 7 页的『DB2 支持的开发软件』
- 『DB2 服务器的安装需求 (Windows)』 (《DB2 服务器快速入门》)
- 第 256 页的『Windows C/C++ 应用程序编译和链接选项』
- 第 261 页的『Windows C/C++ 例程编译和链接选项』
- 第 267 页的『Windows IBM COBOL 应用程序编译和链接选项』
- 第 270 页的『Windows IBM COBOL 例程编译和链接选项』
- 第 273 页的『Windows Micro Focus COBOL 应用程序编译和链接选项』
- 第 276 页的『Windows Micro Focus COBOL 例程编译和链接选项』
- 『DB2 个人版的安装需求 (Windows)』 (《DB2 个人版快速入门》)
- 第 245 页的『Visual Basic .NET 应用程序编译和链接选项』
- 第 242 页的『C# .NET 应用程序编译和链接选项』
- 第 250 页的『CLR .NET 例程编译和链接选项』



## 第 2 章 设置

一般设置信息 . . . . .	23	设置 Windows Java 环境 . . . . .	41
设置应用程序开发环境 . . . . .	23	样本数据库 . . . . .	43
重新构建 DB2 例程共享库 . . . . .	24	设置样本数据库 . . . . .	43
更新数据库管理器配置文件 . . . . .	25	创建样本数据库 . . . . .	44
设置 Java 环境 . . . . .	26	在主机或 AS/400 和 iSeries 服务器上创建样本数 据库 . . . . .	45
设置 DB2 WebSphere MQ 函数 . . . . .	28	对样本数据库进行编目 . . . . .	46
UNIX . . . . .	30	绑定样本数据库实用程序 . . . . .	46
设置 UNIX 应用程序开发环境 . . . . .	30	迁移应用程序 . . . . .	48
UNIX 环境变量设置 . . . . .	31	将应用迁移至 DB2 版本 8 . . . . .	48
设置 UNIX Java 环境 . . . . .	32	迁移 Java 应用程序、例程和 applet . . . . .	49
设置 AIX Java 环境 . . . . .	34	将应用程序从 32 位环境迁移至 64 位环境 . . . . .	50
设置 HP-UX Java 环境 . . . . .	34	确保应用程序可移植性 . . . . .	52
设置 Linux Java 环境 . . . . .	36	在两个版本的 DB2 上运行应用程序 . . . . .	53
设置 Solaris Java 环境 . . . . .	37	下一步 . . . . .	56
Windows . . . . .	38		
设置 Windows 应用程序开发环境 . . . . .	38		

### 一般设置信息

有关 DB2 CLI 设置信息，请参阅 *CLI Guide and Reference*。

### 设置应用程序开发环境

为了构建和运行 DB2 应用程序，必须对操作系统所支持的其中一种编程语言使用编译器或解释器，除非您使用命令行处理器（CLP）脚本编制或 SQL 过程（请参阅下面的内容）。必须设置 DB2 环境并配置它以满足开发需求。要从 DB2 的先前版本迁移 DB2 应用程序，需要遵循某些过程。另外，您可能想要为进行测试而创建 DB2 样本数据库。

#### 先决条件:

确保通过首先构建非 DB2 应用程序正确设置了计划使用的 DB2 支持的编译器或解释器的环境。然后，如果遇到问题，请参阅与该编译器或解释器一起提供的文档。

将“应用程序开发”客户机安装在正在使用的客户机或服务器工作站上。如果要从远程客户机开发应用程序，则应确保客户机可访问到 DB2 数据库服务器所驻留的机器。还应确保客户机可以成功连接至数据库。可以使用命令行处理器（CLP）或客户机配置助手（CCA）来测试连接。

#### 过程:

要设置应用程序开发环境:

1. 除非缺省值是可接受的，否则应遵循第 25 页的『更新数据库管理器配置文件』中的指示信息
2. 如果将使用 DB2 CLI、Java 或 WebSphere MQ 函数来编程，则必须配置环境。在执行任何特定于平台的更改之前，遵循下面各主题中的指示信息：
  - 设置 CLI 环境

- 第 26 页的『设置 Java 环境』
  - 第 28 页的『设置 DB2 WebSphere MQ 函数』
3. 用下面各主题中的指示信息配置操作系统环境:
    - 第 30 页的『设置 UNIX 应用程序开发环境』
    - 第 38 页的『设置 Windows 应用程序开发环境』
  4. 可选: 第 43 页的『设置样本数据库』

## SQL 过程

从 DB2 版本 8.2 开始, 创建 SQL 过程就不要在服务器上有 C 或 C++ 编译器了, 因此不需要安装 C 或 C++ 编译器。当创建了 SQL 过程之后, 它的过程语句会被转换为存储在数据库目录中的本机表示法, 就如对其它 SQL 语句执行的转换一样。当 SQL 过程被调用时, 就会从目录中装入此表示法, 而 DB2 引擎就会执行该过程。

### 相关概念:

- 第 5 页的『数据库管理器实例』
- 第 48 页的『将应用迁移至 DB2 版本 8』

### 相关任务:

- 第 25 页的『更新数据库管理器配置文件』
- 『Setting up the CLI environment』 (*CLI Guide and Reference, Volume 1*)
- 第 26 页的『设置 Java 环境』
- 第 28 页的『设置 DB2 WebSphere MQ 函数』
- 第 30 页的『设置 UNIX 应用程序开发环境』
- 第 38 页的『设置 Windows 应用程序开发环境』
- 第 43 页的『设置样本数据库』

### 相关参考:

- 第 3 页的『DB2 应用程序开发客户机』
- 第 7 页的『DB2 支持的服务器』
- 第 7 页的『DB2 支持的开发软件』
- 第 8 页的『AIX 支持的开发软件』
- 第 10 页的『HP-UX 支持的开发软件』
- 第 12 页的『Linux 支持的开发软件』
- 第 17 页的『Solaris 支持的开发软件』
- 第 19 页的『Windows 支持的开发软件』

## 重新构建 DB2 例程共享库

一旦装入之后, DB2<sup>®</sup> 将高速缓存用于存储过程和用户定义的函数的共享库。如果要开发例程, 您可能想要测试多次装入同一共享库, 而高速缓存可防止您采用最新版本的共享库。避免高速缓存问题的方式取决于例程的类型:

1. 受防护的非线程安全的例程。数据库管理器配置关键字 KEEPFCENCED 具有缺省值 YES。这使得设防方式进程保持为活动状态。此缺省设置可能会干扰重新装入库。

开发受防护的非线程安全的例程时，最好将此关键字的值更改为 NO，然后，在装入共享库最终版本准备就绪时将该值更改回 YES。有关更多信息，请参阅第 25 页的『更新数据库管理器配置文件』。

2. **可信的或线程安全的例程。**除了 SQL 例程（包括 SQL 过程）之外，当将共享库用于可信的或线程安全的例程时确保采用了已更新版本的 DB2 例程库的唯一方法是通过在命令行上输入 db2stop 后跟 db2start 来回收 DB2 实例。对于 SQL 例程，则不需要这样做，因为当重新创建 SQL 例程时，编译器使用新的唯一库名来防止可能的冲突。

对于 SQL 例程之外的例程，还可以通过用另外命名的库（例如，foo.a 变成 foo.1.a）创建例程的新版本然后将 ALTER PROCEDURE 或 ALTER FUNCTION SQL 语句与新库配合使用来避免此高速缓存问题。

#### 相关任务:

- 第 25 页的『更新数据库管理器配置文件』

#### 相关参考:

- 『ALTER FUNCTION statement』（*SQL Reference, Volume 2*）
- 『ALTER PROCEDURE statement』（*SQL Reference, Volume 2*）

## 更新数据库管理器配置文件

本文件包含应用程序开发的重要设置。

关键字 KEEPFCED 具有缺省值 YES。对于受防护的而不是线程安全的例程（存储过程和 UDF），这会使例程进程保持为活动状态。开发这些例程时，最好将此关键字的值更改为 NO，然后，在装入共享库最终版本准备就绪时将该值更改回 YES。有关更多信息，请参阅第 24 页的『重新构建 DB2 例程共享库』。

**注：**在先前 DB2 版本中，KEEPFCED 称为 KEEPFCED。

为了进行 Java 应用程序开发，需要用 Java Development Kit 的安装路径来更新 JDK\_PATH 关键字。

**注：**JDK\_PATH 在先前 DB2 版本中称为 JDK11\_PATH。

#### 过程:

要更改这些设置，请输入：

```
db2 update dbm cfg using <keyword> <value>
```

例如，要将关键字 KEEPFCED 设置为 NO：

```
db2 update dbm cfg using KEEPFCED NO
```

要将 JDK\_PATH 关键字设置为目录 /home/db2inst/jdk13：

```
db2 update dbm cfg using JDK_PATH /home/db2inst/jdk13
```

要查看数据库管理器配置文件中的当前设置，请输入：

```
db2 get dbm cfg
```

**注：**在 Windows 上，需要在 DB2 命令窗口中输入这些命令。

#### 相关概念:

- 第 24 页的『重新构建 DB2 例程共享库』
- 第 5 页的『数据库管理器实例』

#### 相关任务:

- 第 26 页的『设置 Java 环境』

#### 相关参考:

- 『CREATE FUNCTION statement』 (*SQL Reference, Volume 2*)
- 『CREATE PROCEDURE statement』 (*SQL Reference, Volume 2*)
- 『GET DATABASE MANAGER CONFIGURATION Command』 (*Command Reference*)
- 『RESET DATABASE MANAGER CONFIGURATION Command』 (*Command Reference*)

## 设置 Java 环境

可以使用适用于您平台的 Java Developer Kit (JDK) 来开发存取 DB2 数据库的 Java 程序。Developer Kit 包括“Java 数据库连接”(JDBC)，即，Java 的一个动态 SQL API。

DB2 JDBC 支持作为 DB2 客户机和服务器上的 Java Enablement 选项的一部分提供。利用此支持，可构建和运行 JDBC 应用程序和 applet。它们只包括动态 SQL，并使用 Java 调用接口将 SQL 语句传送至 DB2。

还提供了 DB2 Java 嵌入式 SQL (SQLJ) 支持作为 Java Enablement 的一部分。有了 DB2 SQLJ 的支持，再加上 DB2 JDBC 支持，则您可以构建和运行 SQLJ applet 和应用程序。它们包含静态 SQL，且使用与 DB2 数据库绑定的嵌入式 SQL 语句。

DB2 AD 客户机提供的 SQLJ 支持包括下列各项:

- DB2 SQLJ 转换程序 **sqlj**，它使用 Java 源代码语句来替换 SQLJ 程序中的嵌入式 SQL 语句，并生成包含有关 SQLJ 程序中找到的 SQL 操作信息的序列化概要文件。
- DB2 SQLJ 概要文件定制程序 **db2sqljcustomize**，它对序列化概要文件中存储的 SQL 语句进行预编译，并将这些语句定制至运行时函数调用中，且在 DB2 数据库中生成程序包。

**注:** 在 DB2 的先前版本中，DB2 SQLJ 概要文件概要表称为 **db2prof.c**。

- DB2 SQLJ 概要文件打印程序 **db2sqljprint**，它以明文格式打印概要文件的 DB2 定制版本中的内容。

**注:** 在 DB2 的先前版本中，DB2 SQLJ 概要文件打印程序称为 **db2prof.p**。

- DB2 SQLJ 概要文件绑定程序 **db2sqljbind**，它根据先前定制的 SQLJ 程序生成程序包。

**注:** 建议不要使用基于 CLI 的类型 2 和类型 3 JDBC 驱动程序。将不会对这些驱动程序添加任何新的功能部件或增强功能，在将来的 DB2 发行版中，它们将不可用。提供了完全重新设计过的通用 JDBC 驱动程序以用进一步增强了的功能部件替换这些旧的驱动程序。建议您尽快迁移应用程序以使用新的驱动程序。

## 过程:

要使用 JDBC 通用类型 2 或 JDBC 通用类型 4 连接来构建应用程序以及使用 JDBC 通用类型 4 连接来构建 applet，TCP/IP 侦听器必须在运行。为了确保满足这种情况，执行下列操作：

1. 按如下所示将环境变量 DB2COMM 设置为 TCPIP:

```
db2set DB2COMM=TCPIP
```

2. 按 services 文件中指定的那样，使用 TCP/IP 服务名称来更新数据库管理器配置文件:

```
db2 update dbm cfg using SVCENAME <TCP/IP service name>
```

必须执行“db2stop”和“db2start”此设置才能生效。

**注:** 用于 applet 和 SQLJ 程序的端口号必须与数据库管理器配置文件中使用的 TCP/IP SVCENAME 编号相同。

要运行 DB2 Java 应用程序，必须安装并调用提供本机线程支持的 Java 虚拟机 (JVM)。要使用本机线程执行 Java 应用程序，可在命令中使用 `-native` 选项。例如，要运行 Java 样本应用程序 `DbInfo.class`，可使用以下命令：

```
java -native DbInfo
```

可以通过将 `THREADS_FLAG` 环境变量设置为“native”来将本机线程指定为一些“Java 虚拟机”的缺省线程支持。本文档假定本机线程支持为缺省值。有关使本机线程成为系统缺省线程的指示信息，请参阅 JVM 文档。

要运行 DB2 Java applet，可调用提供本机线程或绿线程支持的 Java 虚拟机。

当安装了上述各项且它们处于工作状态时，可通过遵循下列其中一个主题中的步骤来设置特定操作系统 Java 环境：

- 第 32 页的『设置 UNIX Java 环境』
- 第 41 页的『设置 Windows Java 环境』

有关最新的 DB2 Java 应用程序开发更新的内容，请访问以下 Web 页面：

```
http://www.ibm.com/software/data/db2/udb/ad/v8/java
```

## 相关任务:

- 第 32 页的『设置 UNIX Java 环境』
- 第 41 页的『设置 Windows Java 环境』
- 『Installing the DB2 Universal JDBC Driver』 (*Application Development Guide: Programming Client Applications*)

## 相关参考:

- 『db2sqljcustomize - DB2 SQLJ Profile Customizer Command』 (*Command Reference*)
- 『db2sqljprint - DB2 SQLJ Profile Printer Command』 (*Command Reference*)
- 『db2sqljbind - DB2 SQLJ Profile Binder Command』 (*Command Reference*)

## 设置 DB2 WebSphere MQ 函数

DB2 和 WebSphere MQ 可以用来构造组合消息传递和数据存取的应用程序。MQ 函数与用户定义的函数 (UDF) 相似, 在 DB2 中可选择启用它们。通过使用这些基本函数, 可支持大范围的应用程序 (从简单事件通知到数据仓储到更新联合数据源)。

### 过程:

要设置 DB2 WebSphere MQ 函数:

1. 在每台物理机器上安装 WebSphere MQ。

确保至少在 DB2 通用数据库服务器上安装了具有最新修订包的 WebSphere MQ V5.1。如果已安装此版本的 WebSphere MQ, 则跳至下一步: “安装 WebSphere MQ AMI”。DB2 版本 8 包括要与 DB2 配合使用的 WebSphere MQ 服务器的副本。用于安装 WebSphere MQ 或用于升级现有 WebSphere MQ 安装的特定于平台的指示信息可以在特定于平台的《快速入门》一书中找到, 该书网址为 <http://www.ibm.com/software/ts/mqseries/library/manuals>。完成安装过程时, 一定要设置缺省队列管理器。

2. 在每台物理机器上安装 WebSphere MQ Application Messaging Interface AMI。

这是对提供完全独立的管理和编程任务的 WebSphere MQ 编程接口的扩展。DB2 WebSphere MQ 函数需要安装此接口。如果已在 DB2 服务器上安装了 WebSphere MQ AMI, 则跳至下一步: “启用和配置 DB2 WebSphere MQ 用户定义的函数”。如果未安装 WebSphere MQ AMI, 则可以从随 DB2 提供的安装程序包安装它, 也可以通过从 WebSphere MQ SupportPacs Web 站点下载 AMI 的副本来安装它, 该站点网址为 <http://www.ibm.com/software/ts/mqseries/txppacs>。可以在『Category 3 – Product Extensions』下面找到 AMI。为方便起见, DB2 附带提供了 WebSphere MQ AMI 的副本。此文件位于 `sqlib/cfg/mq` 目录中。

文件的名称与操作系统有关:

表 7. WebSphere MQ AMI

操作系统	文件名
AIX V4.3 和更高版本	ma0f_ax.tar.Z
HP-UX	ma0f_hp.tar.Z
Solaris Operating Environment	ma0f_sol7.tar.Z 或 mq0f_sol26.tar.Z
Windows	ma0f_win.zip

注: DB2 WebSphere MQ 函数在 Linux 上不受支持

遵循象包含在压缩安装映像中的 AMI readme 文件中所概述的正常 AMI 安装过程。

3. 启用和配置 DB2 WebSphere MQ 用户定义的函数。

**enable\_MQFunctions** 实用程序是一个灵活的命令, 可执行下列操作:

- a. 检查是否设置了正确的 WebSphere MQ 环境
- b. 为 DB2 WebSphere MQ 函数安装和创建缺省配置
- c. 使用这些函数启用指定数据库
- d. 确认配置是否起作用



在 UNIX 64 位上，必须修改运行时库路径以包括 \$HOME/sqllib/lib32 才能执行 enable/disable\_MQFunctions。以下设置可达到修改目的。

### AIX

```
LIBPATH=$HOME/sqllib/lib32 enable_MQFunctions -n dbname \  
-u userid -p passwd -v 0pc [-q qMgr -force -noValidate]  
LIBPATH=$HOME/sqllib/lib32 disable_MQFunctions -n dbname \  
-u userid -p passwd -v 0pc
```

### HP-UX

```
SHLIB_PATH=$HOME/sqllib/lib32 enable_MQFunctions -n dbname \  
-u userid -p passwd -v 0pc [-q qMgr -force -noValidate]  
SHLIB_PATH=$HOME/sqllib/lib32 disable_MQFunctions -n dbname \  
-u userid -p passwd -v 0pc
```

### Solaris

```
LD_LIBRARY_PATH=$HOME/sqllib/lib32 enable_MQFunctions -n dbname \  
-u userid -p passwd -v 0pc [-q qMgr -force -noValidate]  
LD_LIBRARY_PATH=$HOME/sqllib/lib32 disable_MQFunctions -n dbname \  
-u userid -p passwd -v 0pc
```

在启用步骤期间，使用下列步骤对 DB2 WebSphere MQ 函数配置和启用数据库：

- a. 对于 Windows，转至步骤“c”。
- b. 通过将 DB2 实例所有者（通常为 db2inst1）以及与受防护的用户定义的函数相关联的用户标识（通常为 db2fenc1）添加至 WebSphere MQ 组 mqm 以便在 UNIX 上启用 WebSphere MQ 函数。这对于存取 WebSphere MQ 的 DB2 函数来说是必需的。
- c. 将 AMT\_DATA\_PATH 环境变量添加至 DB2 所知的列表。可以编辑文件 \$HOME/sqllib/profile.env（UNIX）或 %DB2PATH%\profile.env（Windows），并将 AMT\_DATA\_PATH 添加至 DB2ENVLIST。还可以使用 **db2set** 命令。
- d. 重新启动数据库实例以使环境变量更改生效。
- e. 将目录切换至 \$HOME/sqllib/cfg（UNIX）或 %DB2PATH%\cfg（Windows）。
- f. 运行 **enable\_MQFunctions** 命令以对 DB2 WebSphere MQ 函数配置和启用数据库。在 DB2 ESE 环境中，仅对目录节点执行此步骤。有关此命令的完整描述，参考主题“enable\_MQFunctions”。下面给出一些常见示例。在成功完成之后，会启用指定数据库并测试配置。
- g. 要使用“命令行处理器”测试这些函数，在连接至启用的数据库之后发出下列命令：

```
values DB2MQ.MQSEND('a test')  
values DB2MQ.MQRECEIVE()
```

第一个语句将“a test”消息发送至 DB2MQ\_DEFAULT\_Q 队列，而第二个语句收回它。

**注：**在运行 enable\_MQFunctions 之后，实用程序建立缺省 WebSphere MQ 环境。该实用程序还创建 WebSphere MQ 队列管理器 DB2MQ\_DEFAULT\_MQM 和缺省队列 DB2MQ\_DEFAULT\_Q。如果文件 amt.xml、amthost.xml 和 amt.dtd 在 AMT\_DATA\_PATH 指向的目录中尚不存在，则该实用程序会安装这些文件。如果 amthost.xml 文件存在，但是不包含连接 DB2MQ 的定义，则会带有适当信息的一行添加至该文件。原始文件的副本另存为 DB2MQSAVE.amthost.xml。

- | 4. 如果想要使用事务性 MQ UDF，则确保已配置数据库以便执行联合操作。使用以下命令完成此任务

| `update dbm cfg using federated yes`

- | 5. 要使用 DB2 WebSphere MQ 函数提供的发布 / 预订功能，还必须在每台物理机器上安装 MQSeries Integrator 或 WebSphere MQ Publish/Subscribe 产品扩展。有关 MQSeries Integrator 的信息可以在以下网址找到

| <http://www.ibm.com/software/ts/mqseries/integrator>

| 有关 WebSphere MQ Publish/Subscribe 功能部件的信息，可在以下网址找到

| <http://www.ibm.com/software/ts/mqseries/txppacs>

| 相关概念:

- | • 『MQSeries Enablement』 ( *Application Development Guide: Programming Client Applications* )
- | • 『WebSphere MQ Functional Overview』 ( *Application Development Guide: Programming Client Applications* )
- | • 『WebSphere MQ Messaging』 ( *Application Development Guide: Programming Client Applications* )
- | • 『Sending Messages with WebSphere MQ Functions』 ( *Application Development Guide: Programming Client Applications* )
- | • 『Retrieving Messages with WebSphere MQ Functions』 ( *Application Development Guide: Programming Client Applications* )
- | • 『WebSphere MQ Application-to-application Connectivity』 ( *Application Development Guide: Programming Client Applications* )
- | • 『Request/Reply Communications with WebSphere MQ Functions』 ( *Application Development Guide: Programming Client Applications* )
- | • 『Publish/Subscribe with WebSphere MQ Functions』 ( *Application Development Guide: Programming Client Applications* )
- | • 『如何在 DB2 中使用 WebSphere MQ 函数』 ( 《IBM DB2 Information Integrator 应用程序开发者指南》 )

| 相关参考:

- | • 『db2mqdsn - MQ Listener Command』 ( *Command Reference* )
- | • 『enable\_MQFunctions』 ( *Command Reference* )
- | • 『disable\_MQFunctions』 ( *Command Reference* )

---

## UNIX

有关 UNIX DB2 CLI 设置信息，请参阅 *CLI Guide and Reference*。

### 设置 UNIX 应用程序开发环境

需要设置数据库实例的环境变量。每个数据库管理器实例有两个文件，db2profile 和 db2cshrc，它们是设置该实例的环境变量的脚本。

过程:

为正在使用的 shell 运行正确的脚本:

对于 **bash** 或 **Korn shell**:

```
. $HOME/sqlllib/db2profile
```

对于 **C shell**:

```
source $HOME/sqlllib/db2cshrc
```

其中, \$HOME 是实例所有者的主目录。

如果将此命令包括在 .profile 或 .login 文件中, 则命令将在您登录时自动运行。

如果将使用 ODBC、DB2 CLI 或 Java, 则执行下列主题之一中的步骤:

- 设置 UNIX ODBC 环境
- 设置 UNIX Java 环境

相关概念:

- 第 31 页的『UNIX 环境变量设置』

相关任务:

- 『Setting up the UNIX ODBC environment』 (*CLI Guide and Reference, Volume 1*)
- 第 32 页的『设置 UNIX Java 环境』

相关参考:

- 第 8 页的『AIX 支持的开发软件』
- 第 10 页的『HP-UX 支持的开发软件』
- 第 12 页的『Linux 支持的开发软件』
- 第 17 页的『Solaris 支持的开发软件』

## UNIX 环境变量设置

根据所在的 UNIX<sup>®</sup> 平台, 在 db2profile (对于 bash 或 korn shell) 或 db2cshrc (对于 C shell) 中设置了下列环境变量的值, 并且对这些文件的调用将置于实例所有者的 .profile (bash 或 korn shell) 或 .login (C shell) 文件中。

**AIX<sup>®</sup>:**

- PATH, 将包括几个 DB2<sup>®</sup> 目录 (sqlllib/bin 也在内)
- LIBPATH, 包括目录 sqlllib/lib (请参阅下面的注释)

**HP-UX:**

- PATH, 将包括几个 DB2 目录 (sqlllib/bin 也在内)
- SHLIB\_PATH (32 位和 64 位) 或 LD\_LIBRARY\_PATH (64 位), 包括目录 sqlllib/lib (请参阅下面的注释)

**Linux 和 Solaris:**

- PATH, 将包括几个 DB2 目录 (sqlllib/bin 也在内)
- LD\_LIBRARY\_PATH, 包括目录 sqlllib/lib (请参阅下面的注释)

**注:** 如果正在 64 位 DB2 实例中运行本地 32 位应用程序, 则请参阅第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』。

sqllib/userprofile 和 sqllib/usercshrc 空白文件是创建实例期间创建的，它允许用户放置他们自己的实例环境设置。在任何 DB2 修订包或以后版本安装的实例更新 (db2iupdt) 中这些文件都不会被修改。如果不要 db2profile 或 db2cshrc 脚本中的新环境设置，可以使用相应的“用户”脚本来覆盖它们，在 db2profile 或 db2cshrc 脚本的末尾调用了该“用户”脚本。在实例迁移过程 (db2imigr) 中，由于复制了用户脚本，所以仍然可以使用修改过的环境。

#### 相关任务:

- 第 30 页的『设置 UNIX 应用程序开发环境』
- 第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』

## 设置 UNIX Java 环境

为了在具有 DB2 JDBC 支持的 UNIX 上运行 JDBC 和 SQLJ 程序，在数据库管理器文件 db2profile 和 db2cshrc 中包括可更新您的 Java 环境的命令。当创建 DB2 实例时，修改 .bashrc、.profile 和 .cshrc，以便:

1. 将 THREADS\_FLAG 设置为“native”。（仅在 Solaris 上）
2. CLASSPATH 包括:
  - “.”（当前目录）
  - 文件 sqllib/java/db2java.zip
  - 文件 sqllib/java/db2jcc.jar
  - 文件 sqllib/java/db2jcc\_license\_cu.jar

**注:** db2jcc\_license\_cisuz.jar 还包括在 DB2 Connect 个人版、DB2 Connect 企业版和 DB2 ESE 的 CLASSPATH 中。这提供了与 DB2 z/OS 和 OS/390 版、DB2 AS/400 版和 iSeries 版以及 DB2 VSE 和 VM 版的附加连接。

要构建 SQLJ 程序，同时更新 CLASSPATH 以包括以下文件:

sqllib/java/sqlj.zip

要使用 Java Developer Kit 1.3 或 1.4 来构建“数据源”程序，还必须获得并安装下列各项:

#### JNDI 1.2.1 类库 (jndi.jar 和 providerutil.jar)

<http://java.sun.com/products/jndi/#download>

#### 文件系统服务提供者 1.2 (fscontext.jar)

<http://java.sun.com/products/jndi/#download>

对于 Java Developer Kit 1.3，还必须获得并安装下列各项:

#### JDBC 2.0 可选程序包

<http://java.sun.com/products/jdbc/download.html#spec>

**注:** 使用 Java Developer Kit 1.4 构建“数据源”程序并不是一定需要“JDBC 2.0 可选程序包”。

对于“数据源”程序，还必须将 CLASSPATH 更新为包括下列文件:

- jndi.jar
- fscontext.jar
- providerutil.jar

对于 Java Developer Kit 1.3, 还必须将 CLASSPATH 更新为包括下列文件之一:

- jdbc2\_0-stdext.jar
- j2ee.jar

注:

1. 对于 Java Developer Kit 1.3, 如果您已经使用 j2ee.jar 更新了 CLASSPATH, 则不需要 jdbc2\_0-stdext.jar。
2. 对于 Java Developer Kit 1.4, jdbc2\_0-stdext.jar 和 j2ee.jar 并不是必须存在于 CLASSPATH 中。

“数据源” 样本程序包括在 sqllib/samples/java/sqlj 目录中。有关详细信息, 请参阅 sqllib/samples/java 中的样本自述文件。

注:

1. 如果 CLASSPATH 中包括其它文件, 则确保首先指定上述文件。
2. 由于在将来的 DB2 版本中不存在 sqllib/java12 目录, 因此您应除去对该目录的任何直接引用。取而代之, 请引用 sqllib/java 目录。
3. DB2 Java Enablement 包含 JDBC 程序包绑定程序实用程序 db2jdbcbind。在 DB2 版本 8 服务器上, 自动绑定 JDBC 程序包。此实用程序用来在低级别服务器 (如 DB2 版本 6 和版本 7) 上创建 JDBC 程序包。

过程:

要运行 DB2 Java 例程 (存储过程和 UDF), 需要更新服务器上的 DB2 数据库管理器配置以包括该机器上 Java Developer Kit 的路径安装。为执行此操作, 可在服务器命令行上输入以下命令:

```
db2 update dbm cfg using JDK_PATH /home/db2inst/jdk13
```

其中 /home/db2inst/jdk13 是 Java Developer Kit 的安装路径。

可以通过在服务器上输入以下命令来检查 DB2 数据库管理器配置以验证 JDK\_PATH 字段的正确值:

```
db2 get dbm cfg
```

为便于查看, 您可能希望将输出重定向至文件。JDK\_PATH 字段在靠近输出开始处出现。

当安装了上述各项且它们处于工作状态时, 可通过遵循下列其中一个主题中的步骤来设置特定 UNIX 操作系统环境:

- 设置 AIX Java 环境
- 设置 HP-UX Java 环境
- 设置 Linux Java 环境
- 设置 Solaris Java 环境

相关任务:

- 第 34 页的『设置 AIX Java 环境』
- 第 34 页的『设置 HP-UX Java 环境』
- 第 36 页的『设置 Linux Java 环境』
- 第 37 页的『设置 Solaris Java 环境』
- 第 25 页的『更新数据库管理器配置文件』

- 『Installing the DB2 Universal JDBC Driver』 (*Application Development Guide: Programming Client Applications*)

相关参考:

- 『GET DATABASE MANAGER CONFIGURATION Command』 (*Command Reference*)
- 『RESET DATABASE MANAGER CONFIGURATION Command』 (*Command Reference*)
- 『db2jdbcbind - DB2 JDBC Package Binder Command』 (*Command Reference*)

## 设置 AIX Java 环境

在执行这些指示信息之前，执行第 32 页的『设置 UNIX Java 环境』中的设置。

过程:

要在具有 DB2 JDBC 支持的 AIX 上构建 Java 应用程序，您需要:

1. 其中一个受支持的 Developer Kit 列示在第 8 页的『AIX 支持的开发软件』中。
2. 在 DB2 通用数据库版本 8 AIX 版客户机和服务器上提供的 DB2 Java Enablement。

相关概念:

- 第 103 页的『Java 样本程序』
- 第 104 页的『Java applet 注意事项』

相关任务:

- 第 32 页的『设置 UNIX Java 环境』
- 第 43 页的『设置样本数据库』

相关参考:

- 第 8 页的『AIX 支持的开发软件』

## 设置 HP-UX Java 环境

在执行这些指示信息之前，执行第 32 页的『设置 UNIX Java 环境』中的设置。

过程:

要在具有 DB2 JDBC 支持的 HP-UX 上构建 Java 应用程序，需要在开发机器上安装和配置:

1. 其中一个受支持的 Developer Kit 列示在第 10 页的『HP-UX 支持的开发软件』中。
2. 在 DB2 通用数据库版本 8 HP-UX 版客户机和服务器上提供的 DB2 Java Enablement。

对于 HP-UX 上的 HP-UX 32 位 Java 例程（存储过程和用户定义的函数），最小的 JAVA\_HEAP\_SZ 为 2048。

对于 HP-UX 64 位，DB2 将最小堆设置硬编码为等于最大堆设置。对于 DB2 HP-UX on IA64 版，数据库管理器配置变量 JAVA\_HEAP\_SZ 至少应设置为 4096。

要在 HP-UX 32 位上运行 Java 例程，请确保共享库路径类似于如下内容：

```
export SHLIB_PATH=$JAVADIR/jre/lib/PA_RISC:\
                $JAVADIR/jre/lib/PA_RISC/classic:\
                $HOME/sql/lib/lib:\
                /usr/lib:$SHLIB_PATH
```

其中 \$JAVADIR 通常设置为 /opt/java1.3 (HP-UX 32 位上的 Java SDK 的缺省位置)。

要在 HP-UX 64 位上运行 Java 例程，可通过在命令行发出以下命令来启用 db2hpjv 工具：

```
db2hpjv -e
```

以下命令则禁用此支持：

```
db2hpjv -d
```

在发出 db2hpjv -e 或 db2hpjv -d 命令后，您必须执行 db2stop 和 db2start，以便此更改生效。缺省情况下，禁用 Java 例程支持。

**注：**如果在系统上使用 db2hpjv -e 命令启用了 Java 例程支持，但 Java 已卸载，则 DB2 HP-UX 版将不会运行。

对于 HP-UX on PA-RISC 64 位，应当在 /usr/lib/pa20\_64 中创建指向下列库的符号链接，否则您可能会遇到 SQL4301N 错误：

```
/opt/java1.4/jre/lib/PA_RISC2.0W/libnet.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libzip.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/librmi.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libnio.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libverify.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libmlib_image.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libhprof.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjaas_unix.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libawt.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libcmm.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libdcpr.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libdt_socket.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libfontmanager.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libioser12.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libmawt.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjsound.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjava.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjawt.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjcov.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjcpm.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjdpw.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjpeg.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/libjsig.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/libjvm.sl
```

如果这些链接不存在，则可使用以下命令创建（这些命令需要 root 权限才能执行）：

```
ln -s /opt/java1.4/jre/lib/PA_RISC2.0W/*.sl /usr/lib/pa20_64
ln -s /opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/*.sl /usr/lib/pa20_64
```

对于 HP-UX on IA64，应当在 /usr/lib/hpux64 中创建指向以下库的符号链接，否则您可能会遇到 SQL4301N 错误：

```
/opt/java1.4/jre/lib/IA64W/hotspot/libjunwind.so
/opt/java1.4/jre/lib/IA64W/hotspot/libjvm.so
/opt/java1.4/jre/lib/IA64W/hotspot/libjsig.so
/opt/java1.4/jre/lib/IA64W/libJdbcOdbc.so
```

```
/opt/java1.4/jre/lib/IA64W/libverify.so
/opt/java1.4/jre/lib/IA64W/librmi.so
/opt/java1.4/jre/lib/IA64W/libzip.so
/opt/java1.4/jre/lib/IA64W/libawt.so
/opt/java1.4/jre/lib/IA64W/libcmm.so
/opt/java1.4/jre/lib/IA64W/libmawt.so
/opt/java1.4/jre/lib/IA64W/libjava.so
/opt/java1.4/jre/lib/IA64W/libjcov.so
/opt/java1.4/jre/lib/IA64W/libjcpm.so
/opt/java1.4/jre/lib/IA64W/libjdpw.so
/opt/java1.4/jre/lib/IA64W/libjpeg.so
/opt/java1.4/jre/lib/IA64W/libjsound.so
/opt/java1.4/jre/lib/IA64W/libmlib_image.so
/opt/java1.4/jre/lib/IA64W/libnet.so
/opt/java1.4/jre/lib/IA64W/libnio.so
/opt/java1.4/jre/lib/IA64W/libjaas_unix.so
/opt/java1.4/jre/lib/IA64W/libioserl2.so
/opt/java1.4/jre/lib/IA64W/libhprof.so
/opt/java1.4/jre/lib/IA64W/libfontmanager.so
/opt/java1.4/jre/lib/IA64W/libdt_socket.so
/opt/java1.4/jre/lib/IA64W/libdcpr.so
/opt/java1.4/jre/lib/IA64W/libjawt.so
```

如果这些链接不存在，则可使用以下命令创建（这些命令需要 root 权限才能执行）：

```
ln -s /opt/java1.4/jre/lib/IA64W/*.so /usr/lib/hpux64
ln -s /opt/java1.4/jre/lib/IA64W/hotspot/*.so /usr/lib/hpux64
```

#### 相关概念:

- 第 103 页的『Java 样本程序』
- 第 104 页的『Java applet 注意事项』

#### 相关任务:

- 第 32 页的『设置 UNIX Java 环境』
- 第 43 页的『设置样本数据库』

#### 相关参考:

- 第 10 页的『HP-UX 支持的开发软件』

## 设置 Linux Java 环境

在执行这些指示信息之前，执行第 32 页的『设置 UNIX Java 环境』中的设置。

#### 过程:

要在具有 DB2 JDBC 支持的 Linux 上构建 Java 应用程序，需要在开发机器上安装和配置:

##### 1. 下列其中一项:

- 其中一个受支持的 Developer Kit 列示在第 12 页的『Linux 支持的开发软件』中。
- 在 DB2 通用数据库版本 8 Linux 版客户机和服务器上提供的 DB2 Java Enablement。

要运行 Java 存储过程或用户定义的函数，Linux 运行时链接程序必须能够存取某些 Java 共享库，且 DB2 必须既能够装入这些库又能够装入 Java 虚拟机。因为进行此装入操作的程序是以 setuid 特权运行的，它将仅查找 /usr/lib 中的从属库。



在 `/usr/lib` 中创建符号链接以指向 Java 共享库。以下是需要链接至的必不可少的共享库。根据要构建和运行的应用程序，可能会需要链接至其它共享库。

对于 IBM Developer Kit 1.3，需要指向 `libjava.so`、`libjvm.so` 和 `libhpi.so` 的符号链接。可以通过作为 root 用户运行下列命令来创建符号链接：

```
cd /usr/lib
ln -fs JAVAHOME/jre/bin/libjava.so .
ln -fs JAVAHOME/jre/bin/classic/libjvm.so .
ln -fs JAVAHOME/jre/bin/libhpi.so .
```

其中 `JAVAHOME` 是 IBM Developer Kit 的基本目录。如果 DB2 不能找到这些库，则当尝试运行 Java 例程时将会发生 -4301 错误，并且在管理通知日志中将有关于未找到的库的消息。

**注：**另一种方法是将 Java 共享库添加至 `/etc/ld.so.conf` 而不是在 `/usr/lib` 中创建链接。如果您执行添加操作，则在更改了 `/etc/ld.so.conf` 之后必须作为 root 用户运行 `ldconfig`，否则它将不能生效，因为对例程的调用将会挂起（不能完成）。此备用方法在特定的实例中也不能生效，也会导致例程挂起（不能完成）。如果遇到此问题，则请在 `/usr/lib` 目录中如上面所述那样创建链接。

#### 相关概念：

- 第 103 页的『Java 样本程序』
- 第 104 页的『Java applet 注意事项』

#### 相关任务：

- 第 32 页的『设置 UNIX Java 环境』
- 第 43 页的『设置样本数据库』

#### 相关参考：

- 第 12 页的『Linux 支持的开发软件』

## 设置 Solaris Java 环境

在执行这些指示信息之前，执行第 32 页的『设置 UNIX Java 环境』中的设置。

#### 过程：

要在具有 DB2 JDBC 支持的 Solaris Operating Environment 中构建 Java 应用程序，需要在开发机器上安装和配置下列各项：

1. 其中一个受支持的 Developer Kit 列示在第 17 页的『Solaris 支持的开发软件』中。
2. 在 DB2 通用数据库版本 8 Solaris 版客户机和服务器上提供的 DB2 Java Enablement。

#### 相关概念：

- 第 103 页的『Java 样本程序』
- 第 104 页的『Java applet 注意事项』

#### 相关任务：

- 第 32 页的『设置 UNIX Java 环境』
- 第 43 页的『设置样本数据库』

#### 相关参考:

- 第 17 页的『Solaris 支持的开发软件』

---

## Windows

有关 Windows DB2 CLI 设置信息, 请参阅 *CLI Guide and Reference*。

### 设置 Windows 应用程序开发环境

当在 Windows NT、Windows 2000、Windows XP 或 Windows Server 2003 上安装 DB2 AD 客户机时, 安装程序会使用环境变量 INCLUDE、LIB 和 PATH 更新配置注册表。安装时, 系统范围环境变量 DB2INSTANCE 会被设置为所创建的名为 DB2 的缺省实例。当 DB2 命令窗口打开时, 就在此窗口中设置 DB2PATH。当在 Windows 98 或 Windows ME 上安装 DB2 AD 客户机时, 安装程序将更新 autoexec.bat 文件。

可以覆盖这些环境变量来设置机器或当前登录的用户的值。更改这些环境变量时要慎重。不要更改 DB2PATH 环境变量。DB2INSTANCE 被定义为系统级别的环境变量。如果未设置 DB2INSTANCE, 则不需要使用 DB2INSTDEF DB2 注册表变量(它定义要使用的缺省实例名)。

#### 过程:

要覆盖环境变量设置, 使用下列任何一项:

- Windows XP 控制面板
- Windows Server 2003 控制面板
- Windows NT 控制面板
- Windows 2000 控制面板
- Windows 98 或 Windows ME 命令窗口
- Windows 98 或 Windows ME autoexec.bat 文件

当在命令中使用变量 %DB2PATH% 时, 将全路径用引号引起来, 如在设置 LIB="%DB2PATH%\lib";%LIB% 中那样。此变量的缺省安装值是 \Program Files\IBM\SQLLIB, 它包含一个空格, 所以不使用引号可能会导致错误。

另外, 为运行 DB2 应用程序, 还必须执行下列特定步骤:

- 在构建 C 或 C++ 程序时, 必须确保 INCLUDE 环境变量包含 %DB2PATH%\INCLUDE 作为第一个目录。

为此, 更新编译器的环境设置文件:

#### **Microsoft Visual C++ 6.0**

```
"C:\Program Files\Microsoft Visual Studio\VC98\bin\vcvars32.bat"
```

#### **Microsoft Visual C++ .NET**

```
"C:\Program Files\Microsoft Visual Studio  
.NET\Common7\Tools\vsvars32.bat"
```

这些文件具有下列命令:

#### **Microsoft Visual C++ 6.0**

```
set INCLUDE=%MSVCDir%\ATL\INCLUDE;%MSVCDir%\INCLUDE;  
%MSVCDir%\MFC\INCLUDE;%INCLUDE%
```

### Microsoft Visual C++ .NET

```
@set INCLUDE=%MSVCDir%\ATLMFC\INCLUDE;...;  
%FrameworkSDKDir%\include;%INCLUDE%
```

要将任一文件与 DB2 配合使用，首先将 %INCLUDE%（它设置 %DB2PATH%\INCLUDE 路径）从列表的末尾移至开头，如下所示：

### Microsoft Visual C++ 6.0

```
set INCLUDE=%INCLUDE%;%MSVCDir%\ATL\INCLUDE;  
%MSVCDir%\INCLUDE;%MSVCDir%\MFC\INCLUDE
```

### Microsoft Visual C++ .NET

```
@set INCLUDE=%INCLUDE%;%MSVCDir%\ATLMFC\INCLUDE;...;  
%FrameworkSDKDir%\include
```

- 当构建 Micro Focus COBOL 程序时，设置 COBCPY 环境变量以指向 %DB2PATH%\INCLUDE\cobol\_mf。
- 当构建 IBM COBOL 程序时，设置 SYSLIB 环境变量以指向 %DB2PATH%\INCLUDE\cobol\_a。
- 使用以下语句，确保 LIB 环境变量指向 %DB2PATH%\lib:

```
set LIB="%DB2PATH%\lib";%LIB%
```

**注：**要从 32 位环境启用交叉开发 64 位应用程序，请参阅第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』。

- 确保在远程数据库的服务器上已设置 DB2COMM 环境变量。
- 确保安全服务已在服务器上启动以便进行 SERVER 认证，而在进行 CLIENT 认证时，则应确保安全服务已在客户机上启动。

**注：**由于 CLIENT 认证在客户机端而不是在服务器端进行，所以客户机应用程序应在用户的上下文环境中运行。Win32 认证 API 需要用户可能有也可能没有的某些特权。为了确保 CLIENT 认证成功进行，会将认证请求从客户机应用程序传送到安全服务器（缺省情况下它用具有特权的帐户在本地系统中运行，并且有权调用认证 API）。

要手工启动安全服务，使用 NET START DB2NTSECSERVER 命令。

通常，仅当工作站正在充当一个连接至为“客户机认证”配置的服务器的 DB2 客户机时，您才会想要将安全服务设置为自动启动。要自动启动安全服务，执行下列操作：

### Windows NT

1. 单击“开始”按钮。
2. 单击“设置”。
3. 单击“控制面板”。
4. 在“控制面板”中，单击“服务”。
5. 在“服务”窗口中，突出显示“DB2 安全服务器”。
6. 如果它未列示设置“已启动”和“自动”，则单击“启动”。
7. 单击“自动”。

8. 单击“确定”。
9. 重新引导机器使设置生效。

### **Windows 2000 和 Windows Server 2003**

1. 单击“开始”按钮。
2. 对于 Windows 2000, 单击“设置”, 然后单击“控制面板”。  
对于 Windows Server 2003, 单击“控制面板”。
3. 单击“管理工具”。
4. 单击“服务”。
5. 在“服务”窗口中, 突出显示“DB2 安全服务器”。
6. 如果它未列示设置“已启动”和“自动”, 则从顶部菜单单击“操作”。
7. 单击“特性”。
8. 确保您在“常规”选项卡中。
9. 从“启动类型”下拉菜单中选择“自动”。
10. 单击“确定”。
11. 重新引导机器使设置生效。

### **Windows XP**

1. 单击“开始”按钮。
2. 单击“设置”。
3. 单击“控制面板”。
4. 单击“性能和维护”。
5. 单击“管理工具”。
6. 单击“服务”。
7. 在“服务”窗口中, 突出显示“DB2 安全服务器”。
8. 如果它未列示设置“已启动”和“自动”, 则从顶部菜单单击“操作”。
9. 单击“特性”。
10. 确保您在“常规”选项卡中。
11. 从“启动类型”下拉菜单中选择“自动”。
12. 单击“确定”。
13. 重新引导机器使设置生效。

Windows XP、Windows Server 2003、Windows NT 或 Windows 2000 环境上的数据库管理器是作为服务实现的, 因而当启动服务时不会返回错误或警告, 虽然可能已经发生了问题。这就意味着当您运行 `db2start` 或 `NET START` 命令时, 如果任何通信子系统启动失败, 不会返回警告信息。因此, 用户应经常检查事件日志或“DB2 管理通知”日志以找出在运行这些命令期间可能发生的任何错误。

如果将使用 DB2 CLI 或 Java, 则继续进行适当的任务:

- 设置 Windows CLI 环境
- 第 41 页的『设置 Windows Java 环境』

### 相关任务:

- 第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』
- 『Setting up the Windows CLI environment』 (*CLI Guide and Reference, Volume 1*)
- 第 41 页的『设置 Windows Java 环境』

### 相关参考:

- 第 3 页的『DB2 应用程序开发客户机』
- 第 19 页的『Windows 支持的开发软件』

## 设置 Windows Java 环境

此主题提供了在 Windows 环境中构建和运行 DB2 Java 程序所需要的信息。

为了在具有 DB2® JDBC 支持的受支持 Windows® 平台上运行 JDBC 和 SQLJ 程序, 在安装 DB2 时自动更新 CLASSPATH 以包括下列各项:

- "." (当前目录)
- sqllib\java\db2java.zip 文件
- sqllib\java\db2jcc.jar 文件
- sqllib\java\db2jcc\_license\_cu.jar 文件

注: db2jcc\_license\_cisuz.jar 还包括在 DB2 Connect 个人版、DB2 Connect 企业版和 DB2 ESE 的 CLASSPATH 中。这提供了与 DB2 z/OS 和 OS/390 版、DB2 AS/400 版和 iSeries 版以及 DB2 VSE 和 VM 版的附加连接。

要构建 SQLJ 程序, 同时更新 CLASSPATH 以包括以下文件:

sqllib\java\sqlj.zip

要使用 Java Developer Kit 1.3 或 1.4 来构建“数据源”程序, 还必须获得并安装下列各项:

### JNDI 1.2.1 类库 (jndi.jar 和 providerutil.jar)

<http://java.sun.com/products/jndi/#download>

### 文件系统服务提供者 1.2 (fscontext.jar)

<http://java.sun.com/products/jndi/#download>

对于 Java Developer Kit 1.3, 还必须获得并安装下列各项:

### JDBC 2.0 可选程序包 (jdbc2\_0-stdext.jar)

<http://java.sun.com/products/jdbc/download.html#spec>

注: 使用 Java Developer Kit 1.4 构建“数据源”程序并不是一定需要“JDBC 2.0 可选程序包”。

对于“数据源”程序, 还必须将 CLASSPATH 更新为包括下列文件:

- jndi.jar
- fscontext.jar
- providerutil.jar

对于 Java Developer Kit 1.3, 还必须将 CLASSPATH 更新为包括下列文件之一:

- jdbc2\_0-stdext.jar
- j2ee.jar

**注:**

1. 对于 Java Developer Kit 1.3, 如果您已经使用 j2ee.jar 更新了 CLASSPATH, 则不需要 jdbc2\_0-stdext.jar。
2. 当使用 Java Developer Kit 1.4 时, jdbc2\_0-stdext.jar 或 j2ee.jar 并不是必须存在于 CLASSPATH 中。

“数据源”样本程序包括在 sql11ib\samples\java\sqlj 目录中。有关详细信息, 请参阅 sql11ib\samples\java 中的样本自述文件。

**注:**

1. 如果 CLASSPATH 中包括其它文件, 则确保首先指定上述文件。
2. 由于在将来的 DB2 版本中不存在 sql11ib\java12 目录, 因此您应除去对该目录的任何直接引用。取而代之, 请引用 sql11ib\java 目录。
3. DB2 Java Enablement 包含 JDBC 程序包绑定程序实用程序 db2jdbcbind。在 DB2 版本 8 服务器上, 自动绑定 JDBC 程序包。此实用程序用来在低级别服务器 (如 DB2 版本 6 和 7) 上创建 JDBC 程序包。
4. DB2 版本 8 不支持 Microsoft Software Developer's Kit for Java。它既不能用于 SQLJ 定制, 也不能用于运行类型 2 JDBC 应用程序。

**过程:**

要在具有 DB2 JDBC 支持的 Windows 操作系统上构建 Java 应用程序, 需要在开发机器上安装和配置以下内容:

1. 其中一个受支持的 Developer Kit 列示在第 19 页的『Windows 支持的开发软件』中。
2. 在 DB2 通用数据库版本 8 Windows 版客户机和服务器上提供的 DB2 Java Enablement。

要运行 DB2 Java 例程 (存储过程和 UDF), 需要更新服务器上的 DB2 数据库管理器配置以包括该机器上 Java Developer Kit 的安装路径。为此, 可在服务器命令行上输入以下命令:

```
db2 update dbm cfg using JDK_PATH c:\jdk13
```

其中 c:\jdk13 是 Java Developer Kit 的安装路径。

如果 Java Developer Kit 的安装路径包含具有一个或多个空格的目录名, 则可以用单引号将路径引起来。例如:

```
db2 update dbm cfg using JDK_PATH 'c:\Program Files\jdk13'
```

或使用没有空格的短格式目录名:

```
db2 update dbm cfg using JDK_PATH c:\progra~1\jdk13
```

可以通过在服务器上输入以下命令来检查 DB2 数据库管理器配置以验证 JDK\_PATH 字段的正确值:

```
db2 get dbm cfg
```

为便于查看, 您可能希望将输出重定向至文件。JDK\_PATH 字段在靠近输出开始处出现。

可以将下列命令置于批处理文件中以设置 IBM Java Developer Kit 的 Java 环境。必须在 DB2 命令窗口中运行批处理文件。请确定已进行了所有必要的路径更改以适合特定环境。类似的命令可以用于其它受支持的 Java Developer Kit。

以下是设置 Sun JDK 1.3.1 环境的示例批处理文件的命令：

```
set JDKPATH=D:\JAVA\SUNjdk131
set PATH=%JDKPATH%\bin;%PATH%
set CLASSPATH=%CLASSPATH%;%JDKPATH%\lib\jdbc2_0-stdext.jar
db2 update dbm cfg using JDK_PATH %JDKPATH%
db2 terminate
db2stop
db2start
```

必须在 DB2 命令窗口中运行批处理文件。

相关概念：

- 第 103 页的『Java 样本程序』
- 第 104 页的『Java applet 注意事项』

相关任务：

- 第 43 页的『设置样本数据库』
- 『Installing the DB2 Universal JDBC Driver』（*Application Development Guide: Programming Client Applications*）

相关参考：

- 第 19 页的『Windows 支持的开发软件』

---

## 样本数据库

### 设置样本数据库

要使用与 DB2 一起交付的样本程序，需要在服务器工作站上创建 sample 数据库。这将大约需要 23.5 兆字节的硬盘空间。空的 DB2 数据库大约需要 21.5 兆字节的硬盘空间。

另外，如果将使用远程客户机来存取正在运行另一 DB2 版本或正在另一个操作系统上运行的服务器上的 sample 数据库，则需要将数据库实用程序（包括 DB2 CLI 实用程序文件）绑定至 sample 数据库。

过程：

以下是设置 sample 数据库的步骤：

1. 第 44 页的『创建样本数据库』
2. 第 46 页的『对样本数据库进行编目』
3. 第 46 页的『绑定样本数据库实用程序』

相关任务：

- 第 44 页的『创建样本数据库』
- 第 46 页的『对样本数据库进行编目』
- 第 46 页的『绑定样本数据库实用程序』

相关参考:

- 『The SAMPLE database』 ( *SQL Reference, Volume 1* )
- 『db2sampl - Create Sample Database Command』 ( *Command Reference* )

## 创建样本数据库

使用 `db2sampl` 命令在命令行上创建 `sample` 数据库。

先决条件:

必须具有“系统管理员” (SYSADM) 或“系统控制” (SYSCTRL) 权限才能创建数据库。SYSADM 和 SYSCTRL 分别是 DB2 的最高和第二高权限级别。

过程:

要创建该数据库，在服务器上执行下列操作:

1. 确保路径中包含 `db2sampl` (创建 `sample` 数据库的程序) 的位置。`db2profile` 或 `db2cshrc` 文件将把 `db2sampl` 置于您的路径中，因此，如果不加以更改，该程序将保留在您的路径中。

- 在 UNIX 服务器上，`db2sampl` 位于:

```
$HOME/sql1lib/bin
```

其中，`$HOME` 是 DB2 实例所有者的主目录。

- 在 Windows 上，`db2sampl` 位于:

```
%DB2PATH%\bin
```

其中，`%DB2PATH%` 是 DB2 的安装位置。

2. 确保将 `DB2INSTANCE` 环境变量设置为要创建 `sample` 数据库的实例的名称。如果未设置它，可使用以下命令进行设置:

- 在 UNIX 上:

如果使用 `bash` 或 `Korn shell`，则可输入以下命令进行设置:

```
DB2INSTANCE=instance_name  
export DB2INSTANCE
```

如果使用 `C shell`，则可输入以下命令进行设置:

```
setenv DB2INSTANCE instance_name
```

- 在 Windows 上，请输入:

```
set DB2INSTANCE=instance_name
```

其中 `instance_name` 是数据库实例的名称。

3. 输入 `db2sampl`，后跟您想创建的样本数据库的位置，以便创建 `sample` 数据库。在 UNIX 平台上，这是一个 `path` (即路径，如“`$HOME`”)，且输入形式如下:

```
db2sampl path
```

例如:

```
db2sampl $HOME
```

在 Windows 上，这是 `drive` (即驱动器，如“`C:`”)，且输入形式如下:



db2samp1 drive

例如:

db2samp1 C:

如果不指定路径或驱动器，安装程序将把样本表安装在数据库管理器配置文件的 DFTDBPATH 参数所指定的缺省路径或驱动器中。数据库的认证类型与在其中创建该数据库的实例的认证类型相同。

**相关任务:**

- 第 45 页的『在主机或 AS/400 和 iSeries 服务器上创建样本数据库』
- 第 46 页的『对样本数据库进行编目』
- 第 46 页的『绑定样本数据库实用程序』

## 在主机或 AS/400 和 iSeries 服务器上创建样本数据库

如果想要对“主机”服务器（如 DB2 UDB z/OS 和 OS/390 版或 AS/400 和 iSeries 服务器）运行样本程序，则需要创建一个包含在 SQL Reference 中描述的样本表的数据库。

**注:** 需要 DB2 Connect 以连接至主机服务器。

**限制:**

工作站上的 DB2 和主机系统上 DB2 的 SQL 语法和 DB2 命令有一些不同。当存取 DB2 UDB z/OS 和 OS/390 版或 DB2 AS/400 版和 iSeries 版上的数据库时，应确保程序使用的是这些数据库系统支持的 SQL 语句和预编译 / 绑定选项。

**过程:**

要创建数据库:

1. 使用 db2samp1 在 DB2 工作站服务器实例中创建 sample 数据库。
2. 连接 sample 数据库。
3. 将样本表数据导出至文件。
4. 连接至主机数据库。
5. 创建样本表。
6. 从文件（在该文件中导出了工作站服务器上的数据）导入样本表数据。

**相关概念:**

- 『Export Overview』 (*Data Movement Utilities Guide and Reference*)
- 『Import Overview』 (*Data Movement Utilities Guide and Reference*)

**相关任务:**

- 第 46 页的『对样本数据库进行编目』
- 第 46 页的『绑定样本数据库实用程序』

**相关样本:**

- 『expsamp.sqb -- Export and import tables with table data to a DRDA database (IBM COBOL)』

- 『tbmove.sqc -- How to move table data (C)』
- 『tbmove.sqC -- How to move table data (C++)』

## 对样本数据库进行编目

要从远程客户机存取服务器上的 `sample` 数据库，需要在客户机工作站上对 `sample` 数据库进行编目。

不需要在服务器工作站上对 `sample` 数据库进行编目，因为在创建时就已在服务器上对它进行了编目。

编目将使用客户机应用程序要存取的数据库的名称更新客户机工作站上的数据库目录。在处理客户机请求时，数据库管理器使用已编目的名称来查找和连接数据库。

### 过程:

要在远程客户机工作站上对样本数据库进行编目，请输入:

```
db2 catalog database sample as sample at node nodename
```

其中 *nodename* 是服务器节点的名称。

在可以连接数据库之前，还必须对远程节点进行编目。

### 相关任务:

- 『从 DB2 客户机编目 TCP/IP 节点』 (《安装与配置补充手册》)
- 『使用 CLP 从 DB2 客户机编目数据库』 (《安装与配置补充手册》)
- 『从 DB2 客户机编目 NetBIOS 节点』 (《安装与配置补充手册》)
- 『从客户机编目命名管道节点』 (《安装与配置补充手册》)
- 第 46 页的『绑定样本数据库实用程序』

## 绑定样本数据库实用程序

如果将从正在运行另一 DB2 版本的远程客户机存取服务器上的 `sample` 数据库，则需要将数据库实用程序 (包括 DB2 CLI 实用程序) 绑定至 `sample` 数据库。

绑定会创建一个程序包，当执行应用程序时数据库管理器需要用它来存取数据库。可通过对在预编译期间创建的绑定文件运行 `BIND` 命令来显式地完成绑定。

### 过程:

根据正在使用的客户机工作站平台的不同，绑定数据库实用程序的方式会有所不同。

在 UNIX 客户机工作站上:

1. 输入如下命令，连接至 `sample` 数据库:

```
db2 connect to sample user userid using password
```

其中 *userid* 和 *password* 是 `sample` 数据库所在的实例的用户标识和密码。

2. 输入如下命令，将实用程序与数据库绑定:

```
db2 bind BNDPATH/@db2ubind.lst blocking all sqlerror continue \  
messages bind.msg grant public
```

```
db2 bind BNDPATH/@db2cli.lst blocking all sqlerror continue \  
messages cli.msg grant public
```

其中 *BNDPATH* 是绑定文件所在的路径, 如 *\$HOME*/sqllib/bnd, 而 *\$HOME* 是 DB2 实例所有者的主目录。

3. 检查绑定消息文件 *bind.msg* 和 *cli.msg*, 以验证绑定成功。

在运行 Windows 操作系统的客户机工作站上:

1. 从“开始”菜单选择“程序”。
2. 从“程序”菜单(或从 Windows XP 上的“所有程序”), 选择 IBM DB2。
3. 从 IBM DB2 菜单中选择“命令行工具”。
4. 从“命令行工具”菜单中选择“DB2 命令窗口”。

将显示该命令窗口。

5. 输入如下命令, 连接至 *sample* 数据库:

```
db2 connect to sample user userid using password
```

其中 *userid* 和 *password* 是 *sample* 数据库所在的实例的用户标识和密码。

6. 输入如下命令, 将实用程序与数据库绑定:

```
db2 bind "%DB2PATH%\bnd\@db2ubind.lst" blocking all  
sqlerror continue messages bind.msg grant public
```

```
db2 bind "%DB2PATH%\bnd\@db2cli.lst" blocking all  
sqlerror continue messages cli.msg grant public
```

其中 *%DB2PATH%* 是 DB2 的安装路径。

7. 退出命令窗口, 并通过检查绑定消息文件 *bind.msg* 和 *cli.msg* 来验证绑定是否成功。

对于访问主机服务器的所有客户机, 指定下列 *.lst* 文件之一而不是 *db2ubind.lst*:

#### **ddcsmvs.lst**

用于 DB2 z/OS 和 OS/390 版

#### **ddcsvm.lst**

用于 DB2 VM 版

#### **ddcsvse.lst**

用于 DB2 VSE 版

#### **ddcs400.lst**

用于 DB2 AS/400 版和 iSeries 版

例如:

- 如果从 UNIX 客户机访问 DB2 z/OS 和 OS/390 版服务器, 则输入:

```
db2 bind BNDPATH/@ddcsmvs.lst blocking all sqlerror continue \  
messages bind.msg grant public
```

- 如果从 Windows 客户机访问 DB2 z/OS 和 OS/390 版服务器, 则输入:

```
db2 bind "%DB2PATH%\bnd\@ddcsmvs.lst" blocking all  
sqlerror continue messages bind.msg grant public
```

**相关任务:**

- 第 44 页的『创建样本数据库』
- 第 45 页的『在主机或 AS/400 和 iSeries 服务器上创建样本数据库』
- 第 46 页的『对样本数据库进行编目』

**相关参考:**

- 『BIND Command』 ( *Command Reference* )

---

## 迁移应用程序

### 将应用迁移至 DB2 版本 8

DB2<sup>®</sup> 版本 8 支持迁移下列 DB2 版本:

- DB2 版本 6
- DB2 版本 7.1
- DB2 版本 7.2
- DataJoiner<sup>®</sup> V2.1.1

当迁移至新一些的 DB2 版本时, 将会自动迁移数据库和节点目录。要从任何其它先前 DB2 版本迁移, 必须首先迁移至上述受支持版本之一, 然后从该版本迁移至 DB2 版本 8。

#### HP-UX

如果您要将 DB2 从 HP-UX V10 或更早版本迁移至 HP-UX V11, 则您的 DB2 程序必须用 HP-UX V11 上的 DB2 重新预编译 (如果这些程序包括嵌入式 SQL), 且必须重新编译。这包括所有 DB2 应用程序、存储过程、用户定义的函数和用户出口程序。另外, 在 HP-UX V11 上编译的 DB2 程序可能不能在 HP-UX V10 或更早版本上运行。在 HP-UX V10 上编译和运行的 DB2 程序可以与 HP-UX V11 服务器远程连接。

#### Micro Focus COBOL

用 DB2 版本 2.1.1 或更早版本预编译和用 Micro Focus COBOL 编译的任何现有应用程序都应使用 DB2 的当前版本重新预编译, 再用 Micro Focus COBOL 重新编译。如果未重新预编译用先前版本的 IBM<sup>®</sup> 预编译器构建的这些应用程序, 则发生异常终止时数据库可能会遭到破坏。

**相关概念:**

- 『迁移建议』 ( 《DB2 服务器快速入门》 )
- 第 49 页的『迁移 Java 应用程序、例程和 applet』
- 第 53 页的『在两个版本的 DB2 上运行应用程序』

**相关任务:**

- 『迁移数据库』 ( 《DB2 服务器快速入门》 )
- 『迁移实例 (UNIX)』 ( 《DB2 服务器快速入门》 )
- 『迁移 DB2 UDB (Windows)』 ( 《DB2 服务器快速入门》 )
- 『迁移 DB2 UDB (UNIX)』 ( 《DB2 服务器快速入门》 )

- 第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』
- 第 52 页的『确保应用程序可移植性』

**相关参考:**

- 『Administrative APIs and application migration』 (*Administrative API Reference*)
- 『迁移限制』 (《DB2 服务器快速入门》)
- 『版本 8 与前发行版的不兼容性』 (《管理指南: 计划》)

## 迁移 Java 应用程序、例程和 applet

### SQLJ 应用程序和例程

在 DB2® 版本 8 中, SQLJ 基于一个新的、独立于平台的体系结构, 该体系结构在 DB2 支持的操作系统上大大提高了定制 SQLJ 应用程序和例程的性能及可移植性。已对 DB2 版本 7 SQLJ 支持作出了一些更改。这些更改可能要求对 SQLJ 应用程序和例程作一些修改, 并且将要求重新转换和重新定制要迁移至 DB2 版本 8 的 SQLJ 应用程序和例程。

要将现有 SQLJ 应用程序和例程从版本 7 迁移至版本 8, 必须执行下列步骤:

1. 将所有 VALUES 语句更改为哑选择。例如:

```
#sql [ctxt] hv = {VALUES (DUMMY(1))}
#sql [ctxt] {SELECT DUMMY(1) INTO :hv FROM SYSIBM.SYSDUMMY1}
```

**注:** 在 DB2 版本 8 中, 对于 SQLJ, 不再支持 VALUES 语句和复合 SQL。

2. 除去所有 BLOCK 语句。更改为个别 SQL 语句。如果仅有的两个分块语句是一个可执行 SQL 语句和一个 COMMIT, 则可以在对连接打开自动落实的情况下使用单个语句。可选择调用 SQLJ 批处理 API 替换 BLOCK 语句的使用来达到分块目的。
3. 将版本 8 SQLJ 实用程序与 **sqlj** 命令配合使用来重新转换应用程序和例程, 并使用 **db2sqljcustomize** 命令重新定制这些应用程序和例程。即使没有对源代码进行任何更改也必须执行此步骤, 以便随版本 8 提供的新的跨平台支持可运行这些二进制文件。

### Java™ applet

建议不要使用以前称为“net”驱动程序的类型 3 JDBC 驱动程序。应将 DB2 Java applet 迁移至 DB2 通用 JDBC 驱动程序, 它包含类型 4 连接。要转换类型 3 JDBC applet 以使用新的 DB2 通用 JDBC 驱动程序, 进行下列更改:

1. DB2 通用 JDBC 驱动程序归档为 db2jcc.jar。在与 applet 相关联的 .html 文件中, 将归档从 db2java.zip 更改为 db2jcc.jar。将 db2jcc.jar 复制至 Web 服务器。
2. DB2 通用 JDBC 驱动程序类名为 com.ibm.db2.jcc.DB2Driver。在 applet .java 文件中, 将类型 3 JDBC 驱动程序类名 COM.ibm.db2.jdbc.net.DB2Driver 更改为 DB2 通用 JDBC 驱动程序类名。如果 applet 使用 javax.sql.DataSource 来获取连接, 则可能没有对 JDBC 驱动程序类的引用。
3. 类型 3 和 DB2 通用 JDBC 驱动程序都使用相同格式的数据源 URL: jdbc:db2://server:portnumber/dbname。但是, server、portnumber 和 dbname 这三个部分在两个驱动程序中具有不同的意义。

类型 3 JDBC 驱动程序是一个具有客户机（运行 applet 的浏览器）、JDBC Applet 服务器和 DB2 服务器的三层模型。URL 中的 server 和 portnumber 指的是“JDBC Applet 服务器”。dbname 是在运行“JDBC applet 服务器”的系统上编目的数据库别名。

DB2 通用 JDBC 驱动程序客户机直接连接至 DB2 服务器，因此，server 和 portnumber 就是 DB2 服务器 TCP/IP 侦听器的相应项。dbname 是在 DB2 服务器系统上编目的数据库别名。

如果 applet 使用 DriverManager.getConnection 来连接至 DB2，则使用 DB2 通用 JDBC 驱动程序的新 URL 来更新 .java 文件和 .html 文件（如果需要的话）。

4. 如果 applet 使用的是 COM.ibm.db2.jdbc.DB2DataSource，则必须创建类 com.ibm.jcc.db2.DB2SimpleDataSource 的新 javax.sql.DataSource 对象。必须更新 applet 才可使用此新类。

#### 相关概念:

- 第 48 页的『将应用迁移至 DB2 版本 8』
- 第 104 页的『Java applet 注意事项』

#### 相关任务:

- 第 112 页的『构建 SQLJ 应用程序』
- 第 117 页的『构建 SQLJ 例程』
- 第 110 页的『构建 SQLJ 程序』

## 将应用程序从 32 位环境迁移至 64 位环境

Windows 32 位应用程序可以在 Windows 64 位上按原样运行而不必对 64 位环境作任何更改。在 UNIX 上，在除了 Linux for IA64 和 Linux for zSeries 之外的所有 64 位 DB2 实例中，您可以通过重新绑定现有 32 位本地应用程序并使用正确的库路径设置运行它来迁移这些应用程序。在 HP-UX 上，仅当用 +s 选项链接应用程序时才能执行此操作。如果应用程序不是使用 +s 选项链接的，则必须用 +s 选项重新构建该应用程序，或使用要包括 32 位 DB2 库的嵌入式运行时路径来重新构建该应用程序（如下所示）。

#### 过程:

在 UNIX 上，64 位环境中 32 位应用程序的正确库路径是 lib32。您可能不想将环境变量设置更改为 lib32，因为这样会影响实例环境（32 位和 64 位）中的所有应用程序。要避免发生这种情况，可以使用包装器脚本将环境变量设置为仅用于也从该脚本运行的应用程序。

以下是可以使用的包装器脚本的一个示例:

```
#!/bin/sh
echo <ENV_VAR_SETTING>
export <ENV_VAR_SETTING>
rm -rf $HOME/sqllib/db2dump/* > /dev/null 2>&1
echo
echo Running application...
$1
echo ...Done running application.
```

其中 <ENV\_VAR\_SETTING> 是平台的环境变量设置:

**AIX:** LIBPATH=\$HOME/sqlib/lib32:\$LIBPATH

**HP-UX** (下列其中一项):

SHLIB\_PATH=\$HOME/sqlib/lib32:\$SHLIB\_PATH

LD\_LIBRARY\_PATH=\$HOME/sqlib/lib32:\$LD\_LIBRARY\_PATH

注: 如果应用程序是使用 +s 选项链接的, 则包装器将只在 HP-UX 上起作用。

**Linux:**

LD\_LIBRARY\_PATH=\$HOME/sqlib/lib32:\$LD\_LIBRARY\_PATH

**Solaris:**

LD\_LIBRARY\_PATH=\$HOME/sqlib/lib32:\$LD\_LIBRARY\_PATH

重新绑定应用程序之后, 可以通过在命令行上输入包装器脚本名后跟可执行文件名来运行此包装器程序:

```
<wrapper_script> <executable>
```

如果包装器脚本的库路径与被调用的可执行文件不兼容, 则更改包装器中的环境变量对于调用其它可执行文件(如 C system() 调用)的应用程序可能不起作用。要迁移这些应用程序, 必须重新链接对象文件, 然后重新绑定应用程序。

具有 lib32 的运行时库路径应用来链接对象文件而不是平台的环境变量。样本程序的 C、C++ 和 CLI 构建脚本使用适当的运行时路径来使得新应用程序能方便地移植到 64 位环境。

应使用相同的链接选项来在 64 位环境中链接现有的 32 位应用程序的对象文件(请参阅下面的样本相关链接中的构建脚本)。下列标志可用来将 32 位 db2 库包括在运行时库路径中:

**AIX:** -L\$DB2PATH/lib32

或者, 用户可以使用 -blibpath 链接程序选项来指定完整的运行时库路径。AIX 样本构建脚本使用前一种方法。

**HP-UX:**

-Wl,+b\$DB2PATH/lib32

**Linux:**

-Wl,-rpath,\$DB2PATH/lib32

**Solaris:**

-R\$DB2PATH/lib32

注:

1. 这些命令的每一个命令都假定使用编译器进行链接, 而不是使用 ld 直接链接。
2. 在 Solaris 上, 必须先取消设置 LD\_LIBRARY\_PATH 和 LD\_LIBRARY\_PATH\_32, 再使用运行时路径链接应用程序。如果没有取消设置, 则将使用 LD\_LIBRARY\_PATH 或 LD\_LIBRARY\_PATH\_32 设置而不是使用运行时路径设置。
3. 在 Linux 上, 如果要使用 --enable-new-dtags 链接选项, 则先取消设置 LD\_LIBRARY\_PATH, 再运行 32 位可执行文件。如果没有取消设置, 则将使用 LD\_LIBRARY\_PATH 设置而不是使用运行时路径设置。

**Windows 上的交叉开发**

要在 Windows 32 位环境中开发 64 位应用程序，需确保 LIB 环境变量指向 %DB2PATH%\lib\Win64。缺省情况下，%DB2PATH%\lib (32 位路径) 将被添加至 LIB 路径，因此当您在 32 位环境中交叉开发 64 位应用程序时，必须确保 %DB2PATH%\lib\Win64 路径位于缺省的 32 位路径前面。%DB2PATH%\lib 用来在 32 位环境中开发 32 位应用程序或在 64 位环境中开发 64 位应用程序。

### 更改 Long 数据类型

如果要在 32 位应用程序仍在 32 位服务器上运行的同时迁移该应用程序以在 64 位操作环境中使用，则使用 LONGERROR 预编译选项来准备移植该应用程序。在 32 位环境中将 LONGERROR 设置为 YES，以便预编译器一遇到 Long 类型的主机变量就返回错误。然后遵循以下步骤：

1. 除非 Long 类型是必要的，否则减少使用主机变量的 Long 类型。或者，将使用新的可移植主机变量 sqlint32 或 sqluint32。例如：

```
EXEC SQL BEGIN DECLARE SECTION;
  long y; /* this declaration generates an error on 64 bit */
  sqlint32 x; /* this declaration is acceptable for 64 bit */
EXEC SQL END DECLARE SECTION;
```

2. 针对 64 位服务器上的数据库预编译应用程序。这将为正被移植的应用程序创建新程序包。
3. 以 64 位方式编译应用程序。
4. 将应用程序与新的 64 位 DB2 库链接。
5. 将应用程序绑定至 64 位服务器上的数据库。

注：DB2 不支持在 32 位实例中运行 64 位应用程序。

#### 相关概念：

- 第 31 页的『UNIX 环境变量设置』
- 第 48 页的『将应用迁移至 DB2 版本 8』

#### 相关样本：

- 『bldapp -- Builds AIX C application programs (C)』
- 『bldapp -- Builds HP-UX C applications (C)』
- 『bldapp -- Builds Linux C applications (C)』
- 『bldapp -- Builds Solaris C applications (C)』

## 确保应用程序可移植性

以下是开发应用程序时要注意的地方。它们将帮助使您的应用程序变得可移植。

#### 过程：

- 在 UNIX 上，在应用程序中仅使用缺省库搜索路径 /usr/lib:/lib。在 Windows® 操作系统上，通过使用以下命令确保 LIB 环境变量指向 %DB2PATH%\lib:

```
set LIB=%DB2PATH%\lib;%LIB%
```

还要在缺省路径和您所使用的 DB2 版本之间创建符号链接。确保该链接是链接到您的应用程序所需的最低级别的 DB2。参考适合您平台的《快速入门》一书或『安装』主题以获得有关设置链接的信息。



- 如果您的应用程序需要一个特定版本的 DB2，则使用在应用程序中指定该 DB2 版本的路径。例如，如果 AIX<sup>®</sup> 应用程序需要 DB2 版本 5，则使用 `/usr/lpp/db2_05_00/lib`。一般情况下，不必这样做。
- 当您正在为生产而不是为内部开发构建应用程序时，应用程序中的路径不应指向实例所有者的目录（在 UNIX 上，该目录为 `sql1lib/lib`；在 Windows 操作系统上，该目录为 `sql1lib\lib`）副本。这就使得应用程序非常依赖于特定的用户名和环境。
- 通常，不要使用下列环境变量来改变特定环境中的搜索路径：  
LIBPATH (AIX)、SHLIB\_PATH (HP-UX 32 位)、LD\_LIBRARY\_PATH (HP-UX 64 位、Linux 和 Solaris) 和 LIB (Windows)。这些变量将覆盖在环境中运行的应用程序中指定的搜索路径，所以应用程序可能无法找到它们需要的库或文件。
- 在 DB2 Universal Database™ (DB2 通用数据库) 版本 6、版本 7 和版本 8 中，具有字符串语义的所有字符数组项都具有类型字符型，而不是其它变体（如无符号字符型）。任何用 DB2 通用数据库版本 6、版本 7 或版本 8 编写的应用程序都应遵循此原则。

如果您具有使用无符号 `char` 的 DB2 版本 1 应用程序，则编译器可能会生成警告或错误，这是因为版本 1 应用程序中的无符号 `char` 与版本 6、版本 7 或版本 8 函数原型中的 `char` 之间有类型冲突。如果发生这种情况，则使用编译器选项 `-DSQLOLDCHAR` 来消除该问题。

#### 相关概念:

- 第 31 页的『UNIX 环境变量设置』
- 第 48 页的『将应用迁移至 DB2 版本 8』

#### 相关任务:

- 第 38 页的『设置 Windows 应用程序开发环境』
- 第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』

## 在两个版本的 DB2 上运行应用程序

在 UNIX<sup>®</sup> 平台上，如果您具有来自先前 DB2<sup>®</sup> 发行版的应用程序且想要它们既在先前版本的数据库实例中运行又在同一台机器上的 DB2 版本 8 实例中运行，则可能需要对环境进行一些更改。要确定进行哪些更改，须回答下列问题，然后复查『条件』一节，以了解是否有任何条件适用于您的情况。

使用 AIX<sup>®</sup> 系统来解释提出的观点。相同的概念也适用于其它 UNIX 平台，但是详细信息可能不同，如环境变量和特定的命令。

#### 问题

问题 1: 先前 DB2 版本的应用程序是如何与 DB2 客户机运行时库（例如，AIX 上的 `libdb2.a`）链接的？

要确定可执行文件的嵌入式共享库搜索路径，使用可执行文件驻留的目录（它可能是 `/usr/bin` 或实例目录）中的下列其中一个系统命令：

**AIX**     `dump -H executable_filename`

#### HP-UX

`chatr executable_filename`

**Linux** `objdump -p executable_filename`

**Solaris**

`dump -Lv executable_filename`

其中 `executable_filename` 是应用程序可执行文件的名称。

以下内容是 DB2 版本 7.2 AIX C 版样本应用程序 `dbcat` 的样本转储列表，它取自 DB2 实例的 `samples` 子目录 `/home/dbinst/samples/c:`

```
-----
C 应用程序 dbcat 的“样本转储列表”
-----

dbcat:

***Loader Section***
          Loader Header Information
VERSION#      #SYMtableENT      #RELOCent      LENidSTR
0x00000001    0x0000000f      0x00000015    0x00000047

#IMPfilID     OFFfidSTR      LENstrTBL      OFFstrTBL
0x00000003    0x00000284     0x0000007f    0x000002cb

***Import File Strings***
INDEX  PATH                                BASE                                MEMBER
0      /home/db2inst/sqllib/lib:/usr/lib:/lib
1
2      libc.a                                shr.o
      libdb2.a                               shr.o
-----
```

行 0 (零) 显示可执行文件为查找与它链接的共享库所搜索的目录路径。行 1 和行 2 显示应用程序链接至的共享库。

根据应用程序构建方式的不同，您可能会看到下列不同路径：`/usr/lpp/db2_07_01_0000/lib`、`INSTHOME/sqllib/lib` (其中 `INSTHOME` 是数据库实例所有者的主目录) 或仅 `/usr/lib:/lib` 组合。

问题 2: 在您的系统上，DB2 运行时库是如何配置的？

1 当安装 DB2 版本 1、2、5、6、7 或者 8 中之时，有一个可选的步骤，这个步骤创建从系统缺省共享库路径 `/usr/lib` 到包含 DB2 客户机运行时库的 DB2 安装路径的符号链接。

AIX 上不同 DB2 版本的安装路径如下所示:

- 版本 1  
`/usr/lpp/db2_01_01_0000/lib`
- 版本 2  
`/usr/lpp/db2_02_01/lib`
- 版本 5  
`/usr/lpp/db2_05_00/lib`

### 版本 6.1

`/usr/lpp/db2_06_01/lib`

### 版本 7

`/usr/lpp/db2_07_01/lib`

### 版本 8

`/usr/opt/db2_08_01/lib`

所有这些版本的运行时共享库的名称都为 `libdb2.a`。

在任何时候，这些库中只有一个版本可以是缺省值。DB2 提供了此缺省值，这样当您构建应用程序时，它就不会取决于 DB2 的某一特定版本。

问题 3: 您是否在环境中指定了不同的搜索路径？

可以使用环境变量 `LIBPATH`（在 AIX 上）、`SHLIB_PATH`（在 HP-UX 32 位上）、`SHLIB_PATH` 或 `LD_LIBRARY_PATH`（在 HP-UX 64 位上）或者 `LD_LIBRARY_PATH`（在 Linux 和 Solaris 上）覆盖在应用程序中编码的共享库搜索路径。在 Solaris 上，还可以使用 `LD_LIBRARY_PATH_32`（对于 32 位的应用程序）和 `LD_LIBRARY_PATH_64`（对于 64 位的应用程序）。您可以使用在问题 1 的回答中给出的适用于您平台的适当系统命令来查看库搜索路径。

### 条件

一旦有了上述问题的答案，可能就需要对环境做一些更改。阅读以下列出的条件。如果其中一个条件适合您的情况，则进行必要的更改。

条件 1: 如果版本 7 应用程序从 AIX 缺省共享库路径 `/usr/lib/libdb2.a` 装入共享库，且

- 如果存在从 `/usr/lib/libdb2.a` 至 `/usr/lpp/db2_07_01/lib/libdb2.a` 的符号链接，并且数据库服务器是 DB2 通用数据库版本 8 AIX 版，则执行下列操作之一：

– 更改此符号链接以指向：

`/usr/opt/db2_08_01/lib/libdb2.a`

作为 root 用户，您可以按如下所示使用 “`db2ln`” 命令来更改链接：

`/usr/opt/db2_08_01/cfg/db2ln`

- 将 `LIBPATH` 环境变量设置为指向 `/usr/opt/db2_08_01/lib` 或 `INSTHOME/sql1lib/lib`，其中 `INSTHOME` 是版本 8 DB2 实例所有者的主目录。
- 配置从应用程序（客户机）实例至服务器实例的 TCP/IP 连接。
- 如果存在从 `/usr/lib/libdb2.a` 至 `/usr/opt/db2_08_01/lib/libdb2.a` 的符号链接，且数据库服务器是 DB2 版本 7，则配置从应用程序（客户机）实例至服务器实例的 TCP/IP 连接。

条件 2: 如果版本 7 应用程序从 DB2 版本 7 实例所有者的 `$HOME` 路径（`$HOME/sql1lib/lib/libdb2.a`）装入共享库，且数据库服务器是 DB2 Universal Database™（DB2 通用数据库）版本 8 AIX 版，则执行下列操作之一：

- 将应用程序实例迁移至与该数据库服务器实例相同的版本。
- 将 `LIBPATH` 环境变量设置为指向 `/usr/opt/db2_08_01/lib` 或 `INSTHOME/sql1lib/lib`，其中 `INSTHOME` 是版本 8 实例所有者的主目录。

- 配置从应用程序（客户机）实例至服务器实例的 TCP/IP 连接。

条件 3: 如果版本 7 应用程序从 DB2 版本 7 安装路径 (/usr/lpp/db2\_07\_01/lib/libdb2.a) 装入共享库, 且数据库服务器是 DB2 通用数据库版本 8 AIX 版, 则执行下列操作之一:

- 将 LIBPATH 环境变量设置为指向 /usr/opt/db2\_08\_01/lib 或 INSTHOME/sql1lib/lib, 其中 INSTHOME 是数据库实例所有者的主目录。
- 配置从应用程序（客户机）实例至服务器实例的 TCP/IP 连接。

条件 4: 如果版本 7 应用程序从 DB2 通用数据库版本 8 AIX 版安装路径 (/usr/opt/db2\_08\_01/lib/libdb2.a) 装入共享库, 且数据库服务器是 DB2 版本 7, 则配置从应用程序（客户机）实例至服务器实例的 TCP/IP 连接。

#### 相关概念:

- 第 31 页的『UNIX 环境变量设置』
- 第 5 页的『数据库管理器实例』
- 第 48 页的『将应用迁移至 DB2 版本 8』

#### 相关任务:

- 第 38 页的『设置 Windows 应用程序开发环境』

---

## 下一步

一旦设置好环境, 就可以构建 DB2 应用程序了。下一章讨论样本程序和相关的文件, 包括构建文件。其后的各章使用构建文件和样本来显示如何在编程环境中编译、链接和运行应用程序。根据您的特定应用程序开发需要, 请阅读特定的章节。

## 第 3 章 样本程序和相关文件

样本文件 . . . . .	57	对象链接与嵌入数据库 (OLE DB) 表函数样本 . . . . .	83
按语言 and 应用程序接口分类的样本程序 . . . . .	62	Perl 样本 . . . . .	83
C 样本 . . . . .	62	PHP 样本 . . . . .	84
C++ 样本 . . . . .	65	REXX 样本 . . . . .	85
C# 样本 . . . . .	67	安全性插件样本 . . . . .	87
CLI 样本 . . . . .	69	SQL 过程样本 . . . . .	87
命令行处理器 (CLP) 样本 . . . . .	70	Visual Basic 样本 . . . . .	89
COBOL 样本 . . . . .	71	Visual Basic .NET 样本 . . . . .	91
动态重新配置样本 . . . . .	75	Visual C++ 样本 . . . . .	92
JDBC 样本 . . . . .	75	Windows 管理规范样本 . . . . .	92
SQLJ 样本 . . . . .	78	构建文件、makefile 和错误检查实用程序 . . . . .	93
Java WebSphere 样本 . . . . .	79	构建文件 . . . . .	93
Java 插件样本 . . . . .	80	Makefile . . . . .	95
日志管理用户出口样本 . . . . .	81	错误检查实用程序 . . . . .	98
对象链接与嵌入 (OLE) 样本 . . . . .	82		

本章描述了 DB2 支持的所有平台的编程语言的样本程序和相关文件。它提供了基于 DB2 组件结构的样本设计，并给出了 DB2 样本的列表，且每个样本都有一个描述。它还说明了 DB2 附带的构建文件、makefile 和错误检查实用程序的使用。

### 样本文件

样本程序是随“DB2<sup>®</sup> 应用程序开发 (DB2 AD) 客户机”一起提供的。并非所有样本程序在所有平台上都可用或对于受支持编程语言都可用。可以将样本程序用作模板来创建您自己的应用程序以及了解 DB2 功能的学习工具。

DB2 样本程序“按现状”提供，且不提供任何保证。用户，而不是 IBM<sup>®</sup>，承担着质量、性能和修复任何缺陷的全部风险。

除了样本程序文件之外，DB2 在 `sqllib/samples (UNIX)` 和 `sqllib\samples (Windows)` 下的 `samples` 目录中还提供了其它样本文件。这些文件包括要编译和链接样本程序的构建文件和 makefile、链接至大多数样本程序的错误检查实用程序文件以及帮助应用程序开发的各种脚本文件。例如，提供了脚本来在几个语言子目录中对存储过程和 UDF 进行编目以及取消对它们的编目。每个 `samples` 目录都具有一个描述包含在该目录中的文件的自述文件。

提供了大多数样本程序源文件的 HTML 版本，可以从联机文档存取它们。这些“HTML 格式的样本”链接至文档主题以演示这些主题描述的功能。关键字（如 SQL 语句和 DB2 API）在 HTML 格式的样本中是热链接的，因此用户可以直接转至描述它们的文档。HTML 格式的大多数样本在文件顶部的注释部分中都有一个链接，它指向显示运行已编译样本程序的典型结果的样本输出文件。请注意，在许多情况下，实际输出与机器和平台有关，因此运行同一个程序所接收到的输出可能会有所不同。

下面是一个表，它按平台显示主要受支持编程语言 / API 的样本目录和自述文件。自述文件在联机文档中是热链接的，且这些文件内的样本列表具有指向样本文件源代码的热链接。还可以存取所列样本目录中的样本文件。对于目录路径，使用了 UNIX<sup>®</sup> 式斜杠，如 `samples/c`（在目录只用于 Windows<sup>®</sup> 的地方除外，如 `samples\VB\ADO`）。

表 8. 按平台列示的样本 README 文件

平台 → 语言	AIX®	HP-UX	Linux	Solaris	Windows
C samples/c	README	README	README	README	README
C++ samples/cpp	README	README	README	README	README
C# samples\.\NET\cs	不适用	不适用	不适用	不适用	README
CLI samples/cli	README	README	README	README	README
CLP samples/clp	README	README	README	README	README
IBM COBOL samples/cobol	README	不适用	不适用	不适用	README
Micro Focus COBOL samples/cobol_mf	README	README	README	README	README
JDBC samples/java/jdbc	README	README	README	README	README
SQLJ samples/java/sqlj	README	README	README	README	README
Perl samples/perl	README	README	README	README	README
PHP samples/php	README	README	README	README	README
SQL 过程 samples/sqlproc	README	README	README	README	README
Visual Basic samples\VB\ADO	不适用	不适用	不适用	不适用	ReadMe.txt
Visual Basic .NET samples\.\NET\vb	不适用	不适用	不适用	不适用	README

对于每种受支持的语言以及对于每种语言中嵌入式 SQL 和非嵌入式 SQL 程序，样本程序文件扩展名都不相同。对于一种语言中的程序组，文件扩展名也可能不相同。在下表中对这些不同的样本文件扩展名进行分类：

**按语言分类的样本文件扩展名**

第 59 页的表 9.

**按程序组分类的样本文件扩展名**

第 59 页的表 10.

表 9. 按语言分类的样本文件扩展名

语言	目录	嵌入式 SQL 程序	非嵌入式 SQL 程序
C	samples/c samples/cli (CLI 程序)	.sqc	.c
C++	samples/cpp	.sqc (UNIX) .sqx (Windows)	.C (UNIX) .cxx (Windows)
C#	samples\NET\cs		.cs
COBOL	samples/cobol samples/cobol_mf	.sqb	.cbl
Java™	samples/java/jdbc samples/java/sqlj samples/java/WebSphere samples/java/plugin	.sqlj	.java
REXX	samples/rexx	.cmd	.cmd
Visual Basic	samples\VB\ADO samples\VB\MTS samples\VB\RDO		.bas .frm .vbp
Visual Basic .NET	samples\NET\vb		.vb
Visual C++	samples\VC\ADO		.cpp .dsp .dsw

表 10. 按程序组分类的样本文件扩展名

样本组	目录	文件扩展名
CLP	samples/clp	.db2
OLE	samples\ole\msvb (Visual Basic) samples\ole\msvc (Visual C++)	.bas .vbp (Visual Basic) .cpp (Visual C++)
OLE DB	samples\oledb	.db2
SQL 过程	samples/sqlproc	.db2 (SQL 过程脚本) .c (CLI 客户机应用程序) .sqc (嵌入式 C 客户机应用程序) .java (JDBC 客户机应用程序)
用户出口	samples/c	.ctsm (UNIX 和 Windows) .cdisk (UNIX 和 Windows) .ctape (UNIX) .cxbsa (UNIX)

注:

#### 目录定界符

UNIX 上的目录定界符是一个 /。在 Windows 上，它是一个 \。在这些表中，除非目录只在 Windows 上可用，否则使用 UNIX 定界符。

#### 嵌入式 SQL 程序

需要预编译，但 REXX 嵌入式 SQL 程序除外，因为这种程序的嵌入式 SQL 语句是在程序运行时解释的。

#### IBM COBOL 样本

只在 cobol 子目录中为 AIX 和 Windows 32 位操作系统提供了此样本。

### Micro Focus COBOL 样本

只在 cobol\_mf 子目录中为 AIX、HP-UX、Solaris Operating Environment 和 Windows 32 位操作系统提供了此样本。

### Java 样本

是“Java 数据库连接”（JDBC）applet、应用程序和例程以及嵌入式 SQL Java（SQLJ）applet、应用程序和例程。DB2 控制中心的 WebSphere® 样本和插件示例文件也是这样的样本。提供了所有受支持的 DB2 平台的 Java 样本。

### REXX 样本

只为 AIX 和 Windows 32 位操作系统提供了此样本。

### CLP 样本

是执行 SQL 语句的“命令行处理器”脚本。

### OLE 样本

用于以 Microsoft® Visual Basic 和 Microsoft Visual C++ 编写的“对象链接与嵌入”（OLE），只为 Windows 操作系统提供了此样本。

### Visual Basic 样本

是“ActiveX 数据对象”、“远程数据对象”和“Microsoft 事务处理服务器”样本，只在 Windows 操作系统上提供了此样本。

### Visual C++ 样本

是“ActiveX 数据对象”样本，只在 Windows 操作系统上提供了此样本。

### 用户出口样本

是用于归档和检索数据库日志文件的“日志管理用户出口”程序。这类文件必须用 .c 扩展名重命名，并作为 C 语言程序编译。

样本程序目录在大多数平台上一一般是只读的。在改变或构建样本程序之前，将这些样本程序复制到工作目录中。

## 结构和设计

对用 C、CLI、C++、C#、Java、Perl、PHP、Visual Basic ADO 和 Visual Basic .NET 编写的大多数 DB2 样本进行组织以反映数据库组件基于对象的设计模型。这些样本是以表示不同 DB2 级别的类别来分组的。样本所属级别由样本名开头的两字符前缀指示。对于每个“应用程序编程接口”，样本中并没有表示出所有级别，但是对于作为一个整体的样本，级别按如下方式表示出来：

前缀	DB2 级别
il	安装映像级别
cl	客户机级别
in	实例级别
db	数据库级别
ts	表空间级别
tb	表级别
dt	数据类型级别



各级别显示了一个层次结构。安装映像级别是 DB2 的顶级。在此层次之下，客户机层应用程序可以存取不同实例；一个实例可以具有一个或多个数据库；数据库具有其中存在表（从而保存不同数据类型的数据）的表空间。

此设计不包括所有 DB2 样本。某些样本的目的是演示存取数据的不同方法。这些方法是这些样本的主要目的，因此它们由这些方法按与上面类似的方式表示出来：

**前缀**    编程方法  
**fn**      SQL 函数  
**sp**      存储过程  
**ud**      用户定义的函数

除了这些类别之外，还有一组介绍数据库编程基本概念的教程样本。这些样本使用某些在样本设计中表示的较简单函数，且以三个字符“**tut**”开头。

有一些其它样本未包括在此设计中，如“日志管理用户出口”样本、用 COBOL、Visual C++ 和 REXX 编写的样本、“对象链接与嵌入”（OLE）样本、CLP 脚本以及 SQL 过程。

**注：**Java、C# 和 Visual Basic .NET 程序名的第一个字符（有时还有其它字符）为大写。Java 程序在教程样本名中不使用下划线。Visual Basic ADO 样本有某些字符（但不是第一个字符）为大写。

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

#### 相关概念：

- 第 93 页的『构建文件』
- 第 95 页的『Makefile』
- 第 98 页的『错误检查实用程序』

#### 相关参考：

- 第 62 页的『C 样本』
- 第 69 页的『CLI 样本』
- 第 75 页的『JDBC 样本』
- 第 78 页的『SQLJ 样本』
- 第 87 页的『SQL 过程样本』
- 第 89 页的『Visual Basic 样本』
- 第 92 页的『Visual C++ 样本』
- 第 82 页的『对象链接与嵌入（OLE）样本』
- 第 83 页的『对象链接与嵌入数据库（OLE DB）表函数样本』
- 第 70 页的『命令行处理器（CLP）样本』
- 第 81 页的『日志管理用户出口样本』
- 第 71 页的『COBOL 样本』
- 第 79 页的『Java WebSphere 样本』
- 第 80 页的『Java 插件样本』

- 第 92 页的『Windows 管理规范样本』
- 第 85 页的『REXX 样本』
- 第 75 页的『动态重新配置样本』
- 第 67 页的『C# 样本』
- 第 91 页的『Visual Basic .NET 样本』
- 第 83 页的『Perl 样本』
- 第 84 页的『PHP 样本』
- 第 87 页的『安全性插件样本』

## 按语言和应用程序接口分类的样本程序

### C 样本

UNIX 目录: sqllib/samples/c。Windows 目录: sqllib\samples\c。

文件扩展名: .c (非嵌入式 SQL); .sqc (嵌入式 SQL)

表 11. C 样本程序文件

样本程序名	程序描述
<b>教程样本</b> - 演示基本数据库操作的程序。	
tut_mod.sqc	如何修改表数据。
tut_read.sqc	如何读取表。
tut_use.sqc	如何使用数据库。
<b>客户机级别</b> - 涉及 DB2 客户机级别的样本。	
cli_info.c	如何获取和设置客户机级别信息。
clisnap.c	如何捕获客户机级别快照。
<b>实例级别</b> - 涉及 DB2 实例级别的样本。	
inattach.c	如何连接实例和从实例拆离。
inauth.sqc	如何显示实例级别的权限。
ininfo.c	如何获取和设置实例级别信息。
insnap.c	如何捕获实例级别快照。
instart.c	如何停止和启动当前本地实例。
<b>数据库级别</b> - 涉及 DB2 数据库对象的样本。	
dbauth.sqc	如何授予 / 显示 / 撤销数据库级别的权限。
dbcfg.sqc	如何配置数据库和数据库管理器参数。
dbconn.sqc	如何连接数据库和与数据库断开连接。
dbcreate.c	如何创建和删除数据库。
dbhistfile.sqc	如何读取和更新数据库恢复历史文件条目。
dbinfo.c	如何获取和设置数据库级别的信息。
dbinline.sqc	如何使用直接插入 SQL 过程语言。
dbinspect.sqc	如何使用 DB2 API db2Inspect 来检查体系结构的完整性。
dblogconn.sqc	如何在有数据库连接的情况下异步读取数据库日志文件。
dblognoconn.sqc	如何在没有数据库连接的情况下异步读取数据库日志文件。

表 11. C 样本程序文件 (续)

样本程序名	程序描述
dbmcon.sqc	如何连接多个数据库和与多个数据库断开连接。
dbmcon1.h	dbmcon1.sqc 的头文件。
dbmcon1.sqc	dbmcon.sqc 的支持文件。
dbmcon2.h	dbmcon2.sqc 的头文件。
dbmcon2.sqc	dbmcon.sqc 的支持文件。
dbmigrat.c	如何迁移数据库。
dbpkg.sqc	如何使用程序包。
dbrec.sqc	如何使用 db2GetRecommendations API。
dbrecov.sqc	如何恢复数据库。
dbredirect.sqc	如何执行数据库的重定向复原。
dbrestore.sqc	如何从备份复原数据库。
dbrollfwd.sqc	如何在复原数据库后执行前滚。
dbsample.sqc	如何创建包括“主机”和 AS/400 表和视图的样本数据库。
dbsnap.c	如何捕获数据库级别快照。
dbthrds.sqc	如何在 UNIX 上使用多个上下文 API。
dbthrds.sqc	如何在 Windows 上使用多个上下文 API。
dbuse.sqc	如何使用数据库对象。
<b>表空间级别 - 涉及 DB2 表空间级别的样本。</b>	
tscreate.sqc	如何创建和删除缓冲池和表空间。
tsinfo.sqc	如何获取表空间级别的信息。
<b>表级别 - 涉及 DB2 表对象的样本。</b>	
tbast.sqc	如何将分级表用于更新延迟的“自动总结表”。
tbcompress.sqc	如何创建带 NULL 和缺省值压缩选项的表。
tbconstr.sqc	如何使用表约束。
tbcreate.sqc	如何创建、改变和删除表。
tbident.sqc	如何使用标识列。
tbinfo.sqc	如何获取和设置表级别的信息。
tbintrig.sqc	如何对视图使用“INSTEAD OF”触发器。
tbload.sqc	如何装入到分区数据库中。
tbmerge.sqc	如何使用 MERGE 语句。
tbmod.sqc	如何修改表中的信息。
tbmove.sqc	如何移动表数据。
tbonlineinx.sqc	如何创建和重组表的索引。
tbpriv.sqc	如何授予 / 显示 / 撤销表级别特权。
tbread.sqc	如何读取表中的信息。
tbreorg.sqc	如何重组表。
tbrunstats.sqc	如何对表执行 runstats。
tbsavept.sqc	如何使用外部保存点。
tbse1.sqc	如何从下列三项中选择一项: 插入、更新和删除。
tbse1create.db2	如何为 tbse1 程序创建表。

表 11. C 样本程序文件 (续)

样本程序名	程序描述
tbseldrop.db2	如何删除 tbsel 程序的表。
tbtemp.sqc	如何使用已声明临时表。
tbtrig.sqc	如何对表使用触发器。
tbumqt.sqc	如何使用用户具体查询表 (总结表)。
tbunion.sqc	如何通过 UNION ALL 视图插入。
tbxload.sqc	如何同时从 select 语句中返回数据并将它装入表中。
<b>数据类型级别</b> - 涉及数据类型的样本。	
dtformat.sqc	如何使用装入和导入数据格式扩展名。
dtlob.sqc	如何读写 LOB 数据。
dtudt.sqc	如何创建、使用和删除用户定义的单值类型。
<b>DB2 函数级别</b>	
fnuse.sqc	如何使用 SQL 函数。
<b>存储过程级别</b> - 演示存储过程的样本。	
spcat	spserver 程序的存储过程目录脚本。此脚本调用 spdrop.db2 和 spcreate.db2。
spcreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
spdrop.db2	用于从目录中删除存储过程的 CLP 脚本。
spclient.sqc	用来调用在 spserver.sqc 中声明的服务器例程的客户机程序。
spserver.sqc	在服务器上构建和运行的存储过程例程。
<b>UDF 级别</b> - 演示用户定义的函数的样本。	
udfcli.sqc	调用 udfsrv.c 和 udfsrv.C 中的用户定义的函数的客户机应用程序。
udfsrv.c	udfcli.sqc 调用的用户定义的函数 ScalarUDF。
udfemcli.sqc	调用嵌入式 SQL 用户定义的函数库 udfemsrv 的客户机应用程序。
udfemsrv.sqc	udfemcli 调用的嵌入式 SQL 用户定义的函数库。
<b>其它</b>	
evm.sqc	如何创建文件、管道和表事件监视器并对它们进行语法分析。
utilrecov.c	用于备份、复原和记录文件样本的实用程序。
utilsnap.c	快照监视器样本的实用程序。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

#### 相关概念:

- 第 93 页的『构建文件』
- 第 95 页的『Makefile』
- 第 98 页的『错误检查实用程序』
- 第 57 页的『样本文件』

## C++ 样本

UNIX 目录: sqllib/samples/cpp。 Windows 目录: sqllib\samples\cpp。

UNIX 文件扩展名: .C (非嵌入式 SQL); .sqC (嵌入式 SQL)

Windows 文件扩展名: .cxx (非嵌入式 SQL); .sqx (嵌入式 SQL)

表 12. C++ 样本程序文件

样本程序名	程序描述
<b>教程样本</b> - 演示基本数据库操作的程序。	
tut_mod.sqC	如何修改表数据。
tut_read.sqC	如何读取表。
tut_use.sqC	如何使用数据库。
<b>客户机级别</b> - 涉及 DB2 客户机级别的样本。	
cli_info.C	如何获取和设置客户机级别信息。
clisnap.C	如何捕获客户机级别快照。
<b>实例级别</b> - 涉及 DB2 实例级别的样本。	
inattach.C	如何连接实例和从实例拆离。
inauth.sqC	如何显示实例级别的权限。
ininfo.C	如何获取和设置实例级别信息。
insnap.C	如何捕获实例级别快照。
instart.C	如何停止和启动当前本地实例。
<b>数据库级别</b> - 涉及 DB2 数据库对象的样本。	
dbauth.sqC	如何授予 / 显示 / 撤销数据库级别的权限。
dbcfg.sqC	如何配置数据库和数据库管理器参数。
dbconn.sqC	如何连接数据库和与数据库断开连接。
dbcreate.C	如何创建和删除数据库。
dbhistfile.sqC	如何读取和更新数据库恢复文件条目。
dbinfo.C	如何获取和设置数据库级别的信息。
dbinline.sqC	如何使用直接插入 SQL 过程语言。
dbinspect.sqC	如何使用 DB2 API db2Inspect 来检查体系结构的完整性。
dblogconn.sqC	如何在有数据库连接的情况下异步读取数据库日志文件。
dblognoconn.sqC	如何在没有数据库连接的情况下异步读取数据库日志文件。
dbmcon.sqC	如何连接多个数据库和与多个数据库断开连接。
dbmcon1.h	dbmcon1.sqC 的头文件
dbmcon1.sqC	dbmcon.sqC 的支持文件
dbmcon2.h	dbmcon2.sqC 的头文件
dbmcon2.sqC	dbmcon.sqC 的支持文件
dbmigrat.C	如何迁移数据库。
dbpkg.sqC	如何使用程序包。
dbrec.sqC	如何使用 db2GetRecommendations API。
dbrecov.sqC	如何恢复数据库。
dbredirect.sqC	如何使用重定向复原来恢复数据库。

表 12. C++ 样本程序文件 (续)

样本程序名	程序描述
dbrestore.sql	如何恢复数据库。
dbrollfwd.sql	如何使用前滚恢复来恢复数据库。
dbsample.sql	如何创建包括“主机”和 AS/400 表和视图的样本数据库。
dbsnap.C	如何捕获数据库级别快照。
dbthrd.sqC	如何在 UNIX 上使用多个上下文 API。
dbthrd.sqC	如何在 Windows 上使用多个上下文 API。
dbuse.sql	如何使用数据库对象。
<b>表空间级别 - 涉及 DB2 表空间级别的样本。</b>	
tscree.sql	如何创建和删除缓冲池和表空间。
tsinfo.sql	如何获取表空间级别的信息。
<b>表级别 - 涉及 DB2 表对象的样本。</b>	
tbconstr.sql	如何使用表约束。
tbcreate.sql	如何创建、改变和删除表。
tbident.sql	如何使用标识列。
tbinfo.sql	如何获取和设置表级别的信息。
tbintrig.sql	如何对视图使用“INSTEAD OF”触发器。
tbmerge.sql	如何使用 MERGE 语句。
tbmod.sql	如何修改表中的信息。
tbmove.sql	如何移动表数据。
tbpriv.sql	如何授予 / 显示 / 撤销表级别特权。
tbread.sql	如何读取表中的信息。
tbreorg.sql	如何重组表。
tbsavept.sql	如何使用外部保存点。还演示如何更改列的缺省值。
tbse1.sql	如何从下列三项中选择一项: 插入、更新和删除。
tbse1create.db2	如何为 tbse1 程序创建表。
tbse1drop.db2	如何删除 tbse1 程序的表。
tbtemp.sql	如何使用已声明临时表。
tbtrig.sql	如何对表使用触发器。
tbunion.sql	如何通过 UNION ALL 视图插入。
tbxload.sql	如何同时从 select 语句中返回数据并将它装入表中。
<b>数据类型级别 - 涉及数据类型的样本。</b>	
dtformat.sql	如何使用装入和导入数据格式扩展名。
dtlob.sql	如何读写 LOB 数据。
dtstruct.sql	如何创建、使用和删除结构化类型和类型表的层次结构。
dtudt.sql	如何创建、使用和删除用户定义的单值类型。
<b>DB2 函数级别</b>	
fnuse.sql	如何使用 SQL 函数。
<b>存储过程级别 - 演示存储过程的样本。</b>	
spcat	spserver 程序的存储过程目录脚本。此脚本调用 spdrop.db2 和 spcreate.db2。

表 12. C++ 样本程序文件 (续)

样本程序名	程序描述
spcreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
spdrop.db2	用于从目录中删除存储过程的 CLP 脚本。
spclient.sqC	用来调用在 spserver.sqc 和 spserver.sqC 中声明的服务器例程的客户机程序。
spserver.sqC	在服务器上构建和运行的存储过程例程。
<b>UDF 级别</b> - 演示用户定义的函数的样本。	
udfcli.sqC	调用 udfsrv.c 和 udfsrv.C 中的用户定义的函数的客户机应用程序。
udfsrv.C	udfcli.sqc 和 udfcli.sqC 调用的用户定义的函数 ScalarUDF。
udfemcli.sqC	调用嵌入式 SQL 用户定义的函数库 udfemsrv 的客户机应用程序。
udfemsrv.sqC	udfemcli 调用的嵌入式 SQL 用户定义的函数库。
<b>其它</b>	
evm.sqC	如何创建文件、管道和表事件监视器并对它们进行语法分析。
utilrecov.C	用于备份、复原和记录文件样本的实用程序。
utilsnap.C	快照监视器样本的实用程序。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

**相关概念:**

- 第 93 页的『构建文件』
- 第 95 页的『Makefile』
- 第 98 页的『错误检查实用程序』
- 第 57 页的『样本文件』

## C# 样本

目录: sqllib\samples\ .NET\cs。

表 13. C# .NET 样本程序文件

样本程序名	程序描述
<b>数据库级别</b> - 涉及 DB2 数据库对象的样本。	
DbAuth.cs	如何授予 / 显示 / 撤销数据库级别的权限。
DbDatAdp.cs	如何使用 DB2DataAdapter。
DbDatMap.cs	如何设置 DataTable 和 DataColumn 映射。
DbDsetCn.cs	如何将现有约束添加至数据集。
DbEvent.cs	如何处理 DB2DataAdapter 事件。
DbUse.cs	如何使用数据库对象。
DbValue.cs	如何从数据库中获得单个值。
<b>表级别</b> - 涉及 DB2 表对象的样本。	
TbConstr.cs	如何使用表约束。
TbInfo.cs	如何获取和设置表级别的信息。

表 13. C# .NET 样本程序文件 (续)

样本程序名	程序描述
TbPriv.cs	如何授予 / 显示 / 撤销表级别特权。
TbSel.cs	如何从下列三项中选择一项: 插入、更新和删除。
TbTrig.cs	如何对表使用触发器。
TbUse.cs	如何处理表数据和连接至数据库 / 与数据库断开连接。
<b>数据类型级别</b> - 涉及数据类型的样本。	
DtLob.cs	如何使用 LOB 数据类型。
<b>存储过程</b> - 演示存储过程的样本。	
SpCat.db2	删除在 SpServer.cs 中实现的过程然后创建它们。
SpClient.cs	用来调用 SpServer.cs 中的存储过程的客户机程序。
SpCreate.db2	创建在 SpServer.cs 中实现的外部过程。
SpDrop.db2	删除在 SpCreate.db2 (C#) 中实现的外部过程。
SpReturn.cs	调用存储过程 EMP_DETAILS 并获得它的返回值的客户机应用程序。
SpServer.cs	在 SpCat.db2 中创建的过程的 C# 外部代码实现。
EmpDetails.db2	创建名为 EMP_DETAILS 的存储过程的 CLP 脚本。
<b>用户定义的函数</b> - 演示 UDF 的样本。	
UdfCat.db2	删除在 UdfSrv.cs 中实现的 UDF 然后创建它们。
UdfCli.cs	调用 UdfSrv.cs 中的用户定义的函数的客户机应用程序。
UdfCreate.db2	创建在 UdfSrv.cs 中实现的外部 UDF。
UdfDrop.db2	删除在 udfcreate.db2 (C#) 中创建的外部 UDF。
UdfSrv.cs	UdfCli.cs 调用的用户定义的标量函数。
<b>松散耦合的事务</b>	
empcat.bat	对 C# 客户机程序 SpReturn 的存储过程 EMP_DETAILS 进行编目。
LCTrans.cs	演示松散耦合的事务。
regCOM.bat	注册 C# LCTrans 程序的 COM+ 对象。
RootCOM.cs	此文件用来创建一个库组合件 RootCOM.dll。LCTrans.cs 引用在此文件中定义的类和方法。
SubCOM.cs	此文件用来创建一个库组合件 SubCOM.dll。LCTrans.cs 引用在此文件中定义的类和方法。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

**相关概念:**

- 第 57 页的『样本文件』

**相关任务:**

- 第 240 页的『构建 C# .NET 应用程序』
- 第 246 页的『构建公共语言运行时 (CLR) .NET 例程』



## CLI 样本

UNIX 目录: `sqllib/samples/cli`。Windows 目录: `sqllib\samples\cli`。

表 14. 样本 CLI 程序文件

样本程序名	程序描述
<b>教程样本</b> - 演示基本数据库操作的程序。	
<code>tut_mod.c</code>	如何修改表数据。
<code>tut_read.c</code>	如何读取表。
<code>tut_use.c</code>	如何使用数据库。
<b>安装映像级别</b> - 涉及 DB2 和 CLI 安装映像级别的样本。	
<code>ilinfo.c</code>	如何获取和设置安装级别信息 (如 CLI 驱动程序的版本)。
<b>客户机级别</b> - 涉及 DB2 客户机级别的样本。	
<code>cli_info.c</code>	如何获取和设置客户机级别信息。
<code>clihandl.c</code>	如何分配和释放句柄。
<code>clisqlca.c</code>	如何使用 SQLCA 数据。
<b>实例级别</b> - 涉及 DB2 实例级别的样本。	
<code>ininfo.c</code>	如何获取和设置实例级别信息。
<b>数据库级别</b> - 涉及 DB2 数据库对象的样本。	
<code>dbcongui.c</code>	如何借助“图形用户界面”(GUI)来连接数据库。
<code>dbconn.c</code>	如何连接数据库和与数据库断开连接。
<code>dbinfo.c</code>	如何获取和设置数据库级别的信息。
<code>dbmcon.c</code>	如何连接多个数据库和与多个数据库断开连接。
<code>dbmconx.c</code>	如何使用嵌入式 SQL 连接多个数据库和与多个数据库断开连接。
<code>dbmconx1.h</code>	<code>dbmconx1.sqc</code> 的头文件。
<code>dbmconx1.sqc</code>	<code>dbmconx</code> 程序的嵌入式 SQL 文件。
<code>dbmconx2.h</code>	<code>dbmconx2.sqc</code> 的头文件。
<code>dbmconx2.sqc</code>	<code>dbmconx</code> 程序的嵌入式 SQL 文件。
<code>dbnative.c</code>	如何将包含 ODBC 转义子句的语句转换为数据源特定格式。
<code>dbuse.c</code>	如何使用数据库对象。
<code>dbusemx.sqc</code>	如何将数据库对象与嵌入式 SQL 配合使用。
<code>dbxamon.c</code>	如何显示和回滚不确定事务。
<b>表级别</b> - 涉及 DB2 表对象的样本。	
<code>tbconstr.c</code>	如何使用表约束。
<code>tbcreate.c</code>	如何创建、改变和删除表。
<code>tbinfo.c</code>	如何获取和设置表级别的信息。
<code>tbload.c</code>	使用使用 CLI LOAD 实用程序插入数据。
<code>tbmod.c</code>	如何修改表中的信息。
<code>tbread.c</code>	如何读取表中的信息。
<b>数据类型级别</b> - 涉及数据类型的样本。	
<code>dtinfo.c</code>	如何获取数据类型的信息。
<code>dtlob.c</code>	如何读写 LOB 数据。
<code>dtudt.c</code>	如何创建、使用和删除用户定义的单值类型。

表 14. 样本 CLI 程序文件 (续)

样本程序名	程序描述
<b>存储过程级别</b> - 演示存储过程的样本。	
spcat	spserver 程序的存储过程目录脚本。此脚本调用 spdrop.db2 和 spcreate.db2。
spcreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
spdrop.db2	用于从目录中删除存储过程的 CLP 脚本。
spclient.c	用于调用在 spserver.c 中声明的服务器函数的客户机程序。
spserver.c	在服务器上构建和运行的存储过程函数。
spclires.c	演示多个结果集的 SQLMoreResults 与 SQLNextResults 之间的差别的客户机应用程序。
spcall.c	用于调用任何存储过程的客户机程序。
<b>UDF 级别</b> - 演示用户定义的函数的样本。	
udfcli.c	调用 udfsrv.c 中用户定义的函数的客户机应用程序。
udfsrv.c	udfcli.c 调用的用户定义的函数 ScalarUDF。
<b>公共实用程序文件</b>	
utilcli.c	用在 CLI 样本中的实用程序函数。
utilcli.h	在 CLI 样本中使用的实用程序函数的头文件。

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念：

- 第 93 页的『构建文件』
- 第 95 页的『Makefile』
- 第 98 页的『错误检查实用程序』
- 第 57 页的『样本文件』

## 命令行处理器 (CLP) 样本

UNIX 目录: sqllib/samples/clp。Windows 目录: sqllib\samples\clp。

表 15. 命令行处理器 (CLP) 样本脚本。

样本文件名	文件描述
autocfg.db2	如何根据“性能配置”向导的建议自动配置数据库和数据库管理器配置参数。
const.db2	创建具有 CHECK CONSTRAINT 子句的表。
cte.db2	演示公共表表达式。
flt.db2	演示递归查询。
healthmon.db2	如何使用“健康监视器快照”的表函数。
join.db2	演示表的外连接。
onlineload.db2	如何使用 ALLOW READ ACCESS 选项执行联机装入。
stock.db2	演示触发器的用法。

表 15. 命令行处理器 (CLP) 样本脚本。 (续)

样本文件名	文件描述
testdata.db2	使用 DB2 内置函数, 如 RAND() 和 TRANSLATE(), 用随机生成的测试数据填充表。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念:

- 第 57 页的『样本文件』

相关任务:

- 第 124 页的『从命令行处理器 (CLP) 调用过程』
- 第 123 页的『运行命令行处理器 (CLP) 脚本』

## COBOL 样本

UNIX 目录。IBM COBOL: `sqllib/samples/cobol`; Micro Focus COBOL: `sqllib/samples/cobol_mf`。

Windows 目录。IBM COBOL: `sqllib\samples\cobol`; Micro Focus COBOL: `sqllib\samples\cobol_mf`。

注: 在用于 C、CLI、C++、C#、Java、Perl、PHP、Visual Basic ADO 和 Visual Basic .NET 样本的 DB2 级别设计中未结构化 COBOL 样本。

表 16. 不带嵌入式 SQL 的 COBOL DB2 API 样本程序

样本程序	包括的 API
checkerr.cbl	<ul style="list-style-type: none"> <li>• sqlaintp - 获取错误消息</li> <li>• sqllogstt - 获取 SQLSTATE 消息</li> </ul>
client.cbl	<ul style="list-style-type: none"> <li>• sqleqryc - 查询客户机</li> <li>• sqlesetc - 设置客户机</li> </ul>
d_dbconf.cbl	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqledtin - 拆离</li> <li>• sqlfddb - 获取数据库配置缺省值</li> </ul>
d_dbmcon.cbl	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqledtin - 拆离</li> <li>• sqlfdsys - 获取数据库管理器配置缺省值</li> </ul>
db_udcs.cbl	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqlecrea - 创建数据库</li> <li>• sqledrpd - 删除数据库</li> </ul>

表 16. 不带嵌入式 SQL 的 COBOL DB2 API 样本程序 (续)

样本程序	包括的 API
dbcatt.cb1	<ul style="list-style-type: none"> <li>• sqlcadb - 对数据库进行编目</li> <li>• db2DbDirCloseScan - 关闭数据库目录扫描</li> <li>• db2DbDirGetNextEntry - 获取下一个数据库目录条目</li> <li>• db2DbDirOpenScan - 打开数据库目录扫描</li> <li>• sqluncd - 取消对数据库的编目</li> </ul>
dbcmt.cb1	<ul style="list-style-type: none"> <li>• sqledcgd - 更改数据库注释</li> <li>• db2DbDirCloseScan - 关闭数据库目录扫描</li> <li>• db2DbDirGetNextEntry - 获取下一个数据库目录条目</li> <li>• db2DbDirOpenScan - 打开数据库目录扫描</li> <li>• sqleisig - 安装信号处理程序</li> </ul>
dbconf.cb1	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqlcrea - 创建数据库</li> <li>• sqledrpd - 删除数据库</li> <li>• sqlfrdb - 复位数据库配置</li> <li>• sqlfudb - 更新数据库配置</li> <li>• sqlfxdb - 获取数据库配置</li> </ul>
dbinst.cb1	<ul style="list-style-type: none"> <li>• sqleatcp - 连接和更改密码</li> <li>• sqleatin - 连接</li> <li>• sqledtin - 拆离</li> <li>• sqlegins - 获取实例</li> </ul>
dbmconf.cb1	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqledtin - 拆离</li> <li>• sqlfrsys - 复位数据库管理器配置</li> <li>• sqlfusys - 更新数据库管理器配置</li> <li>• sqlfxsys - 获取数据库管理器配置</li> </ul>
dbsnap.cb1	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqlmonss - 获取快照</li> </ul>
dbstart.cb1	<ul style="list-style-type: none"> <li>• sqlpstart - 启动数据库管理器</li> </ul>
dbstop.cb1	<ul style="list-style-type: none"> <li>• sqlfrce - 强制应用程序</li> <li>• sqlpstp - 停止数据库管理器</li> </ul>
dcscat.cb1	<ul style="list-style-type: none"> <li>• sqlgdad - 对 DCS 数据库进行编目</li> <li>• sqlgdcl - 关闭 DCS 目录扫描</li> <li>• sqlgdel - 取消对 DCS 数据库的编目</li> <li>• sqlgdge - 获取数据库的 DCS 目录条目</li> <li>• sqlgdgt - 获取 DCS 目录条目</li> <li>• sqlgdsc - 打开 DCS 目录扫描</li> </ul>

表 16. 不带嵌入式 SQL 的 COBOL DB2 API 样本程序 (续)

样本程序	包括的 API
ebcdicdb.cbl	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqlecrea - 创建数据库</li> <li>• sqledrpd - 删除数据库</li> </ul>
migrate.cbl	<ul style="list-style-type: none"> <li>• sqlmgdb - 迁移数据库</li> </ul>
monreset.cbl	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqlmrset - 复位监视器</li> </ul>
monsz.cbl	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqlmonss - 获取快照</li> <li>• sqlmonsz - 估计 sqlmonss() 输出缓冲区必需的大小</li> </ul>
nodecat.cbl	<ul style="list-style-type: none"> <li>• sqlctnd - 对节点进行编目</li> <li>• sqlencls - 关闭节点目录扫描</li> <li>• sqlengne - 获取下一个节点目录条目</li> <li>• sqlenops - 打开节点目录扫描</li> <li>• sqluncn - 取消对节点的编目</li> </ul>
restart.cbl	<ul style="list-style-type: none"> <li>• sqlerstd - 重新启动数据库</li> </ul>
setact.cbl	<ul style="list-style-type: none"> <li>• sqlsact - 设置记帐字符串</li> </ul>
sws.cbl	<ul style="list-style-type: none"> <li>• sqleatin - 连接</li> <li>• sqlmon - 获取 / 更新监视器开关</li> </ul>

表 17. COBOL DB2 API 嵌入式 SQL 样本程序

样本程序	包括的 API
dbauth.sqb	<ul style="list-style-type: none"> <li>• sqluadcu - 获得权限</li> </ul>
dbstat.sqb	<ul style="list-style-type: none"> <li>• db2Reorg - 重组表</li> <li>• db2Runstats - 运行统计信息</li> </ul>
expsamp.sqb	<ul style="list-style-type: none"> <li>• db2Export - 导出</li> <li>• sqlimpr - 导入</li> </ul>
impexp.sqb	<ul style="list-style-type: none"> <li>• db2Export - 导出</li> <li>• sqlimpr - 导入</li> </ul>
loadqry.sqb	<ul style="list-style-type: none"> <li>• db2LoadQuery - 装入查询</li> </ul>
rebind.sqb	<ul style="list-style-type: none"> <li>• sqlarbnd - 重新绑定</li> </ul>
tabscont.sqb	<ul style="list-style-type: none"> <li>• sqlbctcq - 关闭表空间容器查询</li> <li>• sqlbftcq - 访存表空间容器查询</li> <li>• sqlbotcq - 打开表空间容器查询</li> <li>• sqlbtcq - 表空间容器查询</li> <li>• sqlfmem - 可用内存</li> </ul>

表 17. COBOL DB2 API 嵌入式 SQL 样本程序 (续)

样本程序	包括的 API
tabspace.sqb	<ul style="list-style-type: none"> <li>• sqlbctsq - 关闭表空间查询</li> <li>• sqlbftpq - 访存表空间查询</li> <li>• sqlbgts - 获取表空间统计信息</li> <li>• sqlbmtsq - 表空间查询</li> <li>• sqlbotsq - 打开表空间查询</li> <li>• sqlbstpq - 单个表空间查询</li> <li>• sqlfmem - 可用内存</li> </ul>
tload.sqb	<ul style="list-style-type: none"> <li>• db2Export - 导出</li> <li>• sqlload - 装入</li> <li>• sqluvqdp - 停顿表的表空间</li> </ul>
tspace.sqb	<ul style="list-style-type: none"> <li>• sqlbctcq - 关闭表空间容器查询</li> <li>• sqlbctsq - 关闭表空间查询</li> <li>• sqlbftcq - 访存表空间容器查询</li> <li>• sqlbftpq - 访存表空间查询</li> <li>• sqlbgts - 获取表空间统计信息</li> <li>• sqlbmtsq - 表空间查询</li> <li>• sqlbotcq - 打开表空间容器查询</li> <li>• sqlbotsq - 打开表空间查询</li> <li>• sqlbstpq - 单个表空间查询</li> <li>• sqlbstsc - 设置表空间容器</li> <li>• sqlbctcq - 表空间容器查询</li> <li>• sqlfmem - 可用内存</li> </ul>

表 18. 不带有任何 DB2 API 的 COBOL 嵌入式 SQL 样本程序

样本程序名	程序描述
advsql.sqb	演示高级 SQL 表达式如 CASE、CAST 和标量 full select 的用法。
cursor.sqb	演示使用静态 SQL 的游标的用法。
delet.sqb	演示从数据库删除项的静态 SQL。
dynamic.sqb	演示使用动态 SQL 的游标的用法。
joinsql.sqb	演示如何使用高级 SQL 连接表达式。
lobeval.sqb	演示 LOB 定位器的用法并延迟对实际 LOB 数据的求值。
lobfile.sqb	演示 LOB 文件句柄的用法。
lobloc.sqb	演示 LOB 定位器的用法。
openftch.sqb	演示如何使用静态 SQL 访存、更新和删除行。
static.sqb	演示如何使用静态 SQL 检索信息。
tabsql.sqb	演示高级 SQL 表表达式的用法。
trigsq1.sqb	演示如何使用高级 SQL 触发器和约束。
updat.sqb	演示如何使用静态 SQL 更新数据库。
varinp.sqb	演示如何使用参数标记作为嵌入式动态 SQL 语句调用的输入变量。

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念：

- 第 93 页的『构建文件』
- 第 95 页的『Makefile』
- 第 98 页的『错误检查实用程序』
- 第 57 页的『样本文件』

## 动态重新配置样本

目录 (AIX 和 Solaris)：sqllib/samples/DLPAR。

表 19. 动态重新配置样本脚本

样本脚本名	文件描述
ibm_db2_slm	这一用于 AIX 的 Korn shell “动态重新配置” 脚本 (DR 脚本) 使得使用随在基于 POWER4 的 pSeries 系统 (如 p690 和 p670) 上运行的 AIX V5.2 提供的 “动态逻辑分区” (DLPAR) 功能很方便。这些功能允许在活动逻辑分区中动态添加和除去资源 (如中央处理器和内存) 而不需要重新引导。可在动态重新配置事务期间调用这一 DR 脚本来确保事件发生而不影响 DB2 的运行。动态修改 DB2 配置会导致该事件。进一步的信息可在 DR 脚本本身中找到。
IBM,DB2	动态重新配置和协调工具使用这一用于 Solaris Operating Environment 的 perl 脚本来提供 DB2 与 “重新配置和协调管理器” (RCM) 之间的接口。这些工具是随在 3800 系列或更新版本的机器上运行的 Solaris 9 或更新版本提供的。正确使用此脚本将确保 DB2 即使在除去硬件资源的情况下也能继续运行。进一步的信息可在脚本本身中找到。

ibm\_db2\_slm 样本在 DB2 AIX 版上的 sqllib/samples/DLPAR 目录中，IBM,DB2 样本在 DB2 Solaris 版上的 sqllib/samples/DLPAR 目录中。

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念：

- 第 57 页的『样本文件』

相关任务：

- 『动态地配置参数』 (《管理指南：性能》)

相关参考：

- 第 8 页的『AIX 支持的开发软件』
- 第 17 页的『Solaris 支持的开发软件』

## JDBC 样本

UNIX 目录：sqllib/samples/java/jdbc。

Windows 目录: sql1lib\samples\java\jdbc。

表 20. JDBC 样本程序文件

样本程序名	程序描述
<b>教程样本</b> - 演示基本数据库操作的程序。	
TutMod.java	如何修改表数据。
TutRead.java	如何读取表。
<b>安装映像级别</b> - 涉及 DB2 安装映像级别的样本。	
IlInfo.java	如何获取和设置安装级别信息。
<b>数据库级别</b> - 涉及 DB2 数据库对象的样本。	
DbAuth.java	如何授予 / 显示 / 撤销数据库级别的权限。
DbConn.java	如何连接数据库和与数据库断开连接。
DbInfo.java	如何获取和设置数据库级别的信息。
DbMCon.java	如何连接多个数据库和与多个数据库断开连接。
DbNative.java	如何将包含 ODBC 转义子句的语句转换为数据源特定格式。
DbRsHold.java	如何在旧的 JDBC 类型 2 和通用 JDBC 驱动程序中使用结果集游标稳定性。要编译此样本, 需要 Java Developer Kit 1.4 或更新版本。要运行此样本, 需要 Java 运行时环境 1.4 或更新版本。
DbSeq.java	如何创建、改变和删除数据库中的序列。
DbUse.java	如何使用数据库对象。
<b>表级别</b> - 涉及 DB2 表对象的样本。	
TbConstr.java	如何使用表约束。
TbCreate.java	如何创建、改变和删除表。
TbGenCol.java	如何使用生成列。
TbIdent.java	如何使用“标识列”。
TbInfo.java	如何获取和设置表级别的信息。
TbInTrig.java	如何对视图使用“INSTEAD OF”触发器。
TbMerge.java	如何使用 MERGE 语句。
TbMod.java	如何修改表中的信息。
TbPriv.java	如何授予 / 显示 / 撤销表级别特权。
TbRead.java	如何读取表中的信息。
TbSel.java	如何从下列三项中选择一项: 插入、更新和删除。
TbTemp.java	如何使用“已声明临时表”。
TbTrig.java	如何对表使用触发器。
TbUnion.java	如何通过 UNION ALL 视图插入。
<b>数据类型级别</b> - 涉及数据类型的样本。	
DtInfo.java	如何获取数据类型的信息。
DtLob.java	如何读写 LOB 数据。
DtUdt.java	如何创建、使用和删除用户定义的单值类型。
<b>applet</b> - 演示 applet 的样本。	
Applt.java	如何使用 applet。
<b>存储过程</b> - 演示存储过程的样本。	



表 20. JDBC 样本程序文件 (续)

样本程序名	程序描述
spscat	spserver 程序的存储过程目录脚本。此脚本调用 SpDrop.db2 和 SpCreate.db2。
SpCreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
SpDrop.db2	用于从目录中删除存储过程的 CLP 脚本。
SpClient.java	用来调用在 SpServer.java 中声明的服务器函数的客户机程序。
SpServer.java	在服务器上构建和运行的存储过程函数。
<b>UDF</b> - 演示用户定义的函数的样本。	
UDFcli.java	调用用户定义的函数库 UDFsrv 的客户机应用程序。
UDFsrv.java	UDFcli.java 调用的用户定义的函数。
udfcat	UDFsrv 程序的 UDF 目录脚本。此脚本调用 UDFDrop.db2 和 UDFCreate.db2。
UDFDrop.db2	从目录中删除 UDF 的 CLP 脚本。
UDFCreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
UDFjcli.java	调用用户定义的函数库 UDFjsrv 的客户机应用程序。
UDFjsrv.java	UDFjcli.java 调用的用户定义的函数。
udfjcat	UDFjsrv 程序的 UDF 目录脚本。此脚本调用 UDFjDrop.db2 和 UDFjCreate.db2。
UDFjDrop.db2	从目录中删除 UDF 的 CLP 脚本。
UDFjCreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
UDFsCreate.db2	如何对包含在 UDFsqlsv.java 中的 UDF 进行编目
UDFsDrop.db2	如何取消对包含在 UDFsqlsv.java 中的 UDF 的编目
UDFsqlcl.java	调用 UDFsqlsv.java 中 UDF
UDFsqlsv.java	UDFsqlcl.java 调用的带有 SQL 语句的用户定义的函数
<b>Java bean</b> - 演示 Java Bean 类的样本。	
CreateEmployee.java	如何创建职员记录。
GeneratePayroll.java	如何按部门生成工资单报告。
其它	
Util.java	JDBC 样本程序的实用程序。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

**相关概念:**

- 第 103 页的『Java 样本程序』
- 第 57 页的『样本文件』

**相关参考:**

- 第 78 页的『SQLJ 样本』
- 第 79 页的『Java WebSphere 样本』
- 第 80 页的『Java 插件样本』

## SQLJ 样本

UNIX 目录: sqllib/samples/java/sqlj。

Windows 目录: sqllib\samples\java\sqlj。

表 21. SQLJ 样本程序文件

样本程序名	程序描述
<b>教程样本</b> - 演示基本数据库操作的程序。	
TutMod.sqlj	如何修改表数据。
TutRead.sqlj	如何读取表。
<b>数据库级别</b> - 涉及 DB2 数据库对象的样本。	
DbAuth.sqlj	如何授予 / 显示 / 撤销数据库级别的权限。
DbConn.sqlj	如何连接数据库和与数据库断开连接。
DbMCon.sqlj	如何连接多个数据库和与多个数据库断开连接。
DbUse.sqlj	如何使用数据库对象。
<b>表级别</b> - 涉及 DB2 表对象的样本。	
TbConstr.sqlj	如何使用表约束。
TbCreate.sqlj	如何创建、改变和删除表。
TbIdent.sqlj	如何使用标识列。
TbInfo.sqlj	如何获取和设置表级别的信息。
TbMod.sqlj	如何修改表中的信息。
TbPriv.sqlj	如何授予 / 显示 / 撤销表级别特权。
TbRead.sqlj	如何读取表中的信息。
TbSel.sqlj	如何从下列三项中选择一项: 插入、更新和删除。
TbTrig.sqlj	如何对表使用触发器。
<b>数据类型级别</b> - 涉及数据类型的样本。	
DtUdt.sqlj	如何创建、使用和删除用户定义的单值类型。
<b>Applet 级别</b> - 演示 applet 的样本。	
Appl1.sqlj	如何使用 applet。
<b>存储过程级别</b> - 演示存储过程的样本。	
spscat	SpServer 程序的存储过程目录脚本。此脚本调用 SpDrop.db2 和 SpCreate.db2。
SpCreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
SpDrop.db2	用于从目录中删除存储过程的 CLP 脚本。
SpClient.sqlj	用来调用在 SpServer.sqlj 中声明的服务器函数的客户机程序。
SpServer.sqlj	在服务器上构建和运行的存储过程函数。
SpIterat.sqlj	SpServer.sqlj 的迭代器类文件。
<b>UDF 级别</b> - 演示用户定义的函数的样本。	
UDFcli.sqlj	调用用户定义的函数库 UDFsrv 的客户机应用程序。
UDFsrv.java	UDFcli 调用的用户定义的函数。
udfcat	UDFsrv 程序的 UDF 目录脚本。此脚本调用 UDFDrop.db2 和 UDFCreate.db2。
UDFDrop.db2	从目录中删除 UDF 的 CLP 脚本。

表 21. SQLJ 样本程序文件 (续)

样本程序名	程序描述
UDFCreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
UDFjcli.sqlj	调用用户定义的函数库 UDFjsrv 的客户机应用程序。
UDFjsrv.java	UDFjcli 调用的用户定义的函数。
udfjcat	UDFjsrv 程序的 UDF 目录脚本。此脚本调用 UDFjDrop.db2 和 UDFjCreate.db2。
UDFjDrop.db2	从目录中删除 UDF 的 CLP 脚本。
UDFjCreate.db2	发出 CREATE PROCEDURE 语句的 CLP 脚本。
<b>Java bean</b> - 演示 Java Bean 类的样本。	
CreateEmployee.sqlj	如何创建职员记录。
GeneratePayroll.sqlj	如何按部门生成工资单报告。
<b>数据源</b> - 演示数据源的样本。	
Batch1Demo.sqlj	SQLJ 批处理 - SQLJ 如何对工作执行批处理。
Batch2Demo.sqlj	SQLJ 批处理 - 使 ExecutionContext 与 BatchContext 关联。
Batch3Demo.sqlj	SQLJ 批处理 - 何时需要隐式执行批处理。
BlobClobDemo.sqlj	如何存取 DB2 表中的 Blob 或 Clob 字段。
createRegisterDS.java	按数据源属性文件指定的那样创建和注册数据源。
CreateDemoSchema.sqlj	此程序为数据源演示程序创建模式。
DbConnDataSource.sqlj	如何将数据源与 DB2 通用 JDBC 驱动程序配合使用来连接至数据库。
DbConnMDatasources.sqlj	如何将“多个数据源”与 DB2 通用 JDBC 驱动程序配合使用来连接至数据库。
ScrollIterDemo.sqlj	如何使用采用 SQLJ 编写的命名的可定位的可滚动迭代器。
<b>其它</b>	
Util.sqlj	SQLJ 样本程序的实用程序。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

**相关概念:**

- 第 103 页的『Java 样本程序』
- 第 57 页的『样本文件』

**相关参考:**

- 第 75 页的『JDBC 样本』
- 第 79 页的『Java WebSphere 样本』
- 第 80 页的『Java 插件样本』

## Java WebSphere 样本

UNIX 目录: `sqllib/samples/java/WebSphere`。

Windows 目录: sqllib\samples\java\websphere。

表 22. Java WebSphere 样本文件

样本程序名	程序描述
AccessEmployee.ear	此“企业归档”(.EAR)文件由四个模块组成, 这些模块包含 32 个不同的 .class、.JSP 和 .HTML 文件。此 EAR 文件(很容易使用 IBM WebSphere Application Server 来部署它)演示 Java 客户机可以如何与“企业 Java Bean”(EJB)进行交互以存取存储在 DB2 中的数据。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

#### 相关概念:

- 第 103 页的『Java 样本程序』
- 第 57 页的『样本文件』

#### 相关参考:

- 第 75 页的『JDBC 样本』
- 第 78 页的『SQLJ 样本』
- 第 80 页的『Java 插件样本』

## Java 插件样本

UNIX 目录: sqllib/samples/java/plugin。

Windows 目录: sqllib\samples\java\plugin。

表 23. Java 控制中心插件样本文件

样本程序名	程序描述
Example1.java	如何将一个新工具栏按钮添加至控制中心工具栏。
Example2.java	如何将新菜单操作添加至“控制中心数据库”对象。
Example3.java	如何在“控制中心”树中的“数据库对象”下添加新对象。
Example3Child.java	如何在“控制中心”树中的“数据库对象”下添加插件对象。
Example3Folder.java	如何在“控制中心”树中的“数据库对象”下添加新对象。

#### 相关概念:

- 『控制中心的插件体系结构简介』(《管理指南: 实现》)
- 『编译和运行示例插件』(《管理指南: 实现》)
- 第 103 页的『Java 样本程序』
- 第 57 页的『样本文件』
- 『编写插件作为控制中心扩展』(《管理指南: 实现》)

#### 相关任务:

- 『创建用来添加工具栏按钮的插件』(《管理指南: 实现》)
- 『设置插件树对象的属性』(《管理指南: 实现》)

#### 相关参考:

- 第 75 页的『JDBC 样本』
- 第 78 页的『SQLJ 样本』
- 第 79 页的『Java WebSphere 样本』

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

## 日志管理用户出口样本

UNIX 目录: `sqllib/samples/c`。Windows 目录: `sqllib\samples\c`。

注：在下表中列示的每个源文件的顶部给出了用于编译“日志管理用户出口”程序的指示信息。

表 24. 日志管理用户出口样本程序文件。

样本文件名	文件描述
db2uext2.ctsm	这是一个样本“用户出口”，它利用 Tivoli Storage Manager (TSM) API 来归档和检索数据库日志文件。该样本提供调用的审计跟踪（有关每个选项的日志存储在单独的文件中），包括时间戳记和接收到的参数。它还提供发生错误时调用的跟踪（包括时间戳记和错误隔离字符串）以确定问题所在。可禁用这些选项。该文件必须重命名为 <code>db2uext2.c</code> ，并作为 C 程序编译。在 UNIX 和 Windows 操作系统上可用。 注：必须使用 <code>xlC_r</code> 或 <code>xlC_r</code> （而不是 <code>xlC</code> 或 <code>xlC</code> ）编译器调用来构建在 AIX 上使用“TSM API 客户机”的应用程序（即使应用程序是单线程的也是如此）。这将确保库是线程安全的。如果您有一个用非线性安全库编译过的应用程序，则可以应用修订测试 IC21925E 或者联系应用程序供应商。该修订测试在 <a href="http://index.storsys.ibm.com">index.storsys.ibm.com</a> 匿名 FTP 服务器上可用的。
db2uext2.cdisk	这是一个样本“用户出口”，它利用交付时所在的特定平台的系统复制命令。该程序归档和检索数据库日志文件，并提供调用的审计跟踪（每个选项的日志存储在单独的文件中），包括时间戳记和接收到的参数。它还提供发生错误时调用的错误跟踪，包括时间戳记和用于确定问题的错误隔离字符串。可禁用这些选项。该文件必须重命名为 <code>db2uext2.c</code> ，并作为 C 程序编译。在 UNIX 和 Windows 操作系统上可用。
db2uext2.ctape	这是一个样本“用户出口”，它利用交付时所在的特定 UNIX 平台的系统磁带命令。该程序归档和检索数据库日志文件。对系统磁带命令的所有限制也适用于此用户出口。该样本提供调用的审计跟踪（有关每个选项的日志存储在单独的文件中），包括时间戳记和接收到的参数。它还提供发生错误时调用的错误跟踪，包括时间戳记和用于确定问题的错误隔离字符串。可禁用这些选项。该文件必须重命名为 <code>db2uext2.c</code> ，并作为 C 程序编译。仅在 UNIX 平台上可用。
db2uext2.cxbsa	这是一个样本“用户出口”，它利用 XBSA API 来“归档和检索”数据库日志文件。该样本提供调用的审计跟踪（有关每个选项的日志存储在单独的文件中），包括时间戳记和接收到的参数。它还提供发生错误时调用的错误跟踪，包括时间戳记和用于确定问题的错误隔离字符串。可禁用这些选项。该文件必须重命名为 <code>db2uext2.c</code> ，并作为 C 程序编译。仅在 UNIX 平台上可用。

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念:

- 『通过日志归档管理日志文件』（《数据恢复及高可用性指南与参考》）
- 第 57 页的『样本文件』

相关参考:

- 『Tivoli Storage Manager』（《数据恢复及高可用性指南与参考》）

## 对象链接与嵌入（OLE）样本

目录。Visual Basic: sql1ib\samples\ole\msvb; Visual C++: sql1ib\samples\ole\msvc。

表 25. “对象链接与嵌入”（OLE）样本程序

样本程序名	程序描述
sales	演示对 Microsoft Excel 销售电子表格的上卷查询（用 Visual Basic 实现）。
names	查询 Lotus Notes 地址簿（用 Visual Basic 实现）。
inbox	通过 OLE/Messaging 查询 Microsoft Exchange 收件箱的电子邮件消息（用 Visual Basic 实现）。
invoice	一种 OLE 自动化用户定义的函数，它将 Microsoft Word 票据文档作为电子邮件附件发送（用 Visual Basic 实现）。
bcounter	一个 OLE 自动化用户定义的函数，使用实例变量演示暂存区（用 Visual Basic 实现）。
ccount	计数器 OLE 自动化用户定义的函数（用 Visual C++ 实现）。
salsrv	一个 OLE 自动化存储过程，它计算 sample 数据库中 STAFF 表的平均薪水（用 Visual Basic 实现）。
salcltvc	一个 Visual C++ DB2 CLI 样本，它调用 Visual Basic 存储过程 salsrv。
salcltvb	一个 Visual Basic DB2 CLI 样本，它调用 Visual Basic 存储过程 salsrv。
salsvado	一个在 32 位 Visual Basic 和 ADO 中实现的 OLE 自动化存储过程，它通过计算新建的 STAFF2 表中的中间工资来演示输出参数，还通过从表中检索数据来演示结果集。
salclado	一个 Visual Basic 客户机，它调用 Visual Basic 存储过程 salsvado。
testcli	一个 OLE 自动化嵌入式 SQL 客户机应用程序，它调用存储过程 tstsrv（用 Visual C++ 实现）。
tstsrv	一个 OLE 自动化存储过程，演示在客户机和存储过程之间传递各种类型（用 Visual Basic 实现）。

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念:

- 第 57 页的『样本文件』

相关参考:

- 第 83 页的『对象链接与嵌入数据库（OLE DB）表函数样本』

## 对象链接与嵌入数据库（OLE DB）表函数样本

目录: sql1lib\samples\oledb。

表 26. “对象链接与嵌入数据库”（OLE DB）表函数

样本程序名	程序描述
inora.db2	INTERSOLV Oracle8 OLE DB 提供程序
jet.db2	Microsoft.Jet.OLEDB.3.51 提供程序
jetsrv.db2	具有 Microsoft.Jet.OLEDB.4.0 提供程序的联合数据库功能。
mapi.db2	用于 MAPI 的 INTERSOLV Connect OLE DB
msdaora.db2	用于 Oracle 的 Microsoft OLE DB 提供程序
msdasql.db2	用于 ODBC 驱动程序的 Microsoft OLE DB 提供程序
msidxs.db2	Microsoft OLE DB 索引服务器提供程序
notes.db2	用于 Notes 的 INTERSOLV Connect OLE DB
samprov.db2	Microsoft OLE DB 样本提供程序
sqloledb.db2	用于 SQL Server 的 Microsoft OLE DB 提供程序

相关概念:

- 第 57 页的『样本文件』

相关参考:

- 第 82 页的『对象链接与嵌入（OLE）样本』

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

## Perl 样本

UNIX 目录: sql1lib/samples/perl。

Windows 目录: sql1lib\samples\perl。

表 27. Perl 样本程序文件

样本程序名	程序描述
数据库级别 - 涉及 DB2 数据库对象的样本。	
dbauth.pl	如何授予 / 显示 / 撤销数据库级别的权限。
dbuse.pl	如何使用数据库。
表级别 - 涉及 DB2 表对象的样本。	
tbconstr.pl	如何创建、使用和删除约束。
tbinfo.pl	如何在表级别获取有关表的信息。
tbpriv.pl	如何授予、显示和撤销对表的特权。
tbset.pl	如何从下列三项中选择一项: 插入、更新和删除。
tbsetcreate.pl	如何为 tbset 程序创建表。
tbsetdrop.pl	如何删除 tbset 程序的表。
tbtrig.pl	如何对表使用触发器。
tbuse.pl	如何执行基本数据库操作和连接至数据库 / 与数据库断开连接。

表 27. Perl 样本程序文件 (续)

样本程序名	程序描述
数据类型级别 - 涉及数据类型的样本。	
dtlob.pl	如何使用 LOB 数据类型。
存储过程 - 演示存储过程的样本。	
spclient.pl	包含调用存储过程的十个函数的客户机程序。
其它样本文件	
DB2SampUtil.pm	定义诸如命令行自变量检查之类的公共函数。还定义用于预编译和执行 SQL 语句以及在发生错误时回滚的函数。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念:

- 『Programming Considerations for Perl』 (*Application Development Guide: Programming Client Applications*)
- 第 57 页的『样本文件』

相关任务:

- 第 133 页的『构建 Perl 应用程序』

## PHP 样本

UNIX 目录: `sqllib/samples/php`。

Windows 目录: `sqllib\samples\php`。

表 28. PHP 样本程序文件

样本程序名	程序描述
数据库级别 - 涉及 DB2 数据库对象的样本。	
dbauth.php	如何授予 / 显示 / 撤销数据库级别的权限。
dbuse.php	如何使用数据库。
表级别 - 涉及 DB2 表对象的样本。	
tbconstr.php	如何创建、使用和删除约束。
tbinfo.php	如何在表级别获取有关表的信息。
tbpriv.php	如何授予、显示和撤销对表的特权。
tbsetl.php	如何从下列三项中选择一项: 插入、更新和删除。
tbsetlcreate.db2	如何为 tbsetl 程序创建表。
tbsetldrop.db2	如何删除 tbsetl 程序的表。
tbtrig.php	如何对表使用触发器。
tbuse.php	如何执行基本数据库操作和连接至数据库 / 与数据库断开连接。
数据类型级别 - 涉及数据类型的样本。	
dtlob.php	如何使用 LOB 数据类型。
用户定义的函数 - 演示用户定义的函数的样本。	
udfcli.php	调用各种类型的用户定义的函数的客户机程序。



表 28. PHP 样本程序文件 (续)

样本程序名	程序描述
其它样本文件	
util_funcs.php	定义诸如命令行自变量检查之类的公共函数。还定义用于预编译和执行 SQL 语句以及在发生错误时回滚的函数。

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念：

- 第 57 页的『样本文件』

相关任务：

- 第 135 页的『构建 PHP 应用程序』

## REXX 样本

AIX 目录: `sqlllib/samples/rexx`。Windows 目录: `sqlllib\samples\rexx`。

表 29. REXX 样本程序文件。

样本文件名	文件描述
blobfile.cmd	演示“二进制大对象”(BLOB)处理。
chgisl.cmd	演示 CHANGE ISOLATION LEVEL API。
client.cmd	演示 SET CLIENT 和 QUERY CLIENT API。
d_dbconf.cmd	演示 API: GET DATABASE CONFIGURATION DEFAULTS
d_dbmcon.cmd	演示 API: GET DATABASE MANAGER CONFIGURATION DEFAULTS
db_udcs.cmd	演示 CREATE DATABASE 和 DROP DATABASE API 以模拟 DB2 MVS/ESA 版 CCSID 500 (国际化的 EBCDIC) 整理顺序的整理行为
dbauth.cmd	演示 GET AUTHORIZATIONS API
dbcacat.cmd	演示下列 API: CATALOG DATABASE CLOSE DATABASE DIRECTORY SCAN GET NEXT DATABASE DIRECTORY ENTRY OPEN DATABASE DIRECTORY SCAN UNCATALOG DATABASE
dbcmt.cmd	演示下列 API: CHANGE DATABASE COMMENT GET ERROR MESSAGE INSTALL SIGNAL HANDLER
dbconf.cmd	演示下列 API: CREATE DATABASE DROP DATABASE GET DATABASE CONFIGURATION RESET DATABASE CONFIGURATION UPDATE DATABASE CONFIGURATION
dbinst.cmd	演示下列 API: ATTACH TO INSTANCE DETACH FROM INSTANCE GET INSTANCE

表 29. REXX 样本程序文件。(续)

样本文件名	文件描述
dbmconf.cmd	演示下列 API: GET DATABASE MANAGER CONFIGURATION RESET DATABASE MANAGER CONFIGURATION UPDATE DATABASE MANAGER CONFIGURATION
dbstart.cmd	演示 START DATABASE MANAGER API
dbstat.cmd	演示下列 API: REORGANIZE TABLE RUN STATISTICS
dbstop.cmd	演示下列 API: FORCE USERS STOP DATABASE MANAGER
dcscat.cmd	演示下列 API: ADD DCS DIRECTORY ENTRY CLOSE DCS DIRECTORY SCAN GET DCS DIRECTORY ENTRY FOR DATABASE GET DCS DIRECTORY ENTRIES OPEN DCS DIRECTORY SCAN UNCATALOG DCS DIRECTORY ENTRY
dynamic.cmd	演示使用动态 SQL 的“CURSOR”的用法
ebcdicdb.cmd	演示 CREATE DATABASE 和 DROP DATABASE API 以模拟 DB2 MVS/ESA 版 CCSID 037 (EBCDIC 美国英语) 整理顺序的整理行为
impexp.cmd	演示 EXPORT 和 IMPORT API
lobeval.cmd	演示延迟数据库中 LOB 的求值
lobfile.cmd	演示 LOB 文件句柄的用法
lobloc.cmd	演示 LOB 定位器的用法
lobval.cmd	演示 LOB 的用法
migrate.cmd	演示 MIGRATE DATABASE API
nodecat.cmd	演示下列 API: CATALOG NODE CLOSE NODE DIRECTORY SCAN GET NEXT NODE DIRECTORY ENTRY OPEN NODE DIRECTORY SCAN UNCATALOG NODE
quitab.cmd	演示 API: QUIESCE TABLESPACES FOR TABLE
rechist.cmd	演示下列 API: CLOSE RECOVERY HISTORY FILE SCAN GET NEXT RECOVERY HISTORY FILE ENTRY OPEN RECOVER HISTORY FILE SCAN PRUNE RECOVERY HISTORY FILE ENTRY UPDATE RECOVERY HISTORY FILE ENTRY
restart.cmd	演示 RESTART DATABASE API
sqlecsrx.cmd	整理顺序的示例
updat.cmd	使用动态 SQL 来更新数据库

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念:

- 第 57 页的『样本文件』

相关任务:

- 第 184 页的『在 AIX 上构建 REXX 应用程序』
- 第 276 页的『在 Windows 上构建 Object REXX 应用程序』

## 安全性插件样本

UNIX 目录: `sqllib/samples/security/plugins`。

Windows 目录: `sqllib\samples\security\plugins`。

表 30. 安全性插件样本程序文件

样本程序名	程序描述
<code>combined.c</code>	组合的用户标识/密码认证和组查找样本。
<code>group_file.c</code>	基于文件的简单组管理插件样本。
<code>gssapi_simple.c</code>	基本 GSS-API 认证插件样本 (客户机和服务器)。
<code>IBMkrb5.c</code>	用于 UNIX 的 IBM 提供的 Kerberos 安全性插件的源代码。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念:

- 第 57 页的『样本文件』
- 『Security plug-ins』 (*Application Development Guide: Programming Client Applications*)

## SQL 过程样本

UNIX 目录: `sqllib/samples/sqlproc`。

Windows 目录: `sqllib\samples\sqlproc`。

表 31. SQL 过程样本程序文件

样本程序名	程序描述
<code>basecase.db2</code>	UPDATE_SALARY 过程提高 “sample” 数据库的 “staff” 表中由 “empno” IN 参数标识的职员工资。该过程根据使用 “rating” IN 参数的 CASE 语句确定提高的数量。
<code>basecase.sqc</code>	调用 UPDATE_SALARY 过程。
<code>baseif.db2</code>	UPDATE_SALARY_IF 过程提高 “sample” 数据库的 “staff” 表中由 “empno” IN 参数标识的职员工资。该过程根据使用 “rating” IN 参数的 IF 语句确定提高的数量。
<code>baseif.sqc</code>	调用 UPDATE_SALARY_IF 过程。
<code>dynamic.db2</code>	CREATE_DEPT_TABLE 过程将使用动态 DDL 来创建新表。该表的名称基于该过程的 IN 参数值。
<code>dynamic.sqc</code>	调用 CREATE_DEPT_TABLE 过程。

表 31. SQL 过程样本程序文件 (续)

样本程序名	程序描述
iterate.db2	ITERATOR 过程将使用 FETCH 循环检索 “department” 表的数据。如果 “deptno” 列的值不为 “D11”，则将修改的数据插入 “department” 表中。如果 “deptno” 列的值为 “D11”，则 ITERATE 语句将控制流传送回 LOOP 语句的开头。
iterate.sqc	调用 ITERATOR 过程。
leave.db2	LEAVE_LOOP 过程统计在一个 LOOP 语句中执行的 FETCH 操作数，然后 “not_found” 条件处理程序调用 LEAVE 语句。LEAVE 语句使控制流退出循环，并结束该存储过程。
leave.sqc	调用 LEAVE_LOOP 过程。
loop.db2	LOOP_UNTIL_SPACE 过程统计在一个 LOOP 语句中执行的 FETCH 操作数，直到游标检索到 “midinit” 列含有空格 (‘ ’) 值的一行。该循环语句使控制流退出循环，并结束该存储过程。
loop.sqc	调用 LOOP_UNTIL_SPACE 过程。
nestcase.db2	BUMP_SALARY 过程使用嵌套 CASE 语句来提高 “sample” 数据库的 “staff” 表中由 dept IN 参数指示的部门的职员工资。
nestcase.sqc	调用 BUMP_SALARY 过程。
nestif.db2	BUMP_SALARY_IF 过程使用嵌套 IF 语句来提高 “sample” 数据库的 “staff” 表中由 dept IN 参数指示的部门的职员工资。
nestif.sqc	调用 BUMP_SALARY_IF 过程。
nestedsp.db2	OUT_AVERAGE、OUT_MEDIAN 和 MAX_SALARY 过程返回样本数据库的 “staff” 表的平均值、中等值和最大值。
nestedspdrop.db2	删除用 nestedsp.db2 脚本创建的 OUT_AVERAGE、OUT_MEDIAN 和 MAX_SALARY SQL 过程。
NestedSP.java	调用 OUT_AVERAGE 过程。
repeat.db2	REPEAT_STMT 过程统计在一个重复语句中执行的 FETCH 操作数，直到游标再也检索不到行。条件处理程序使控制流退出重复循环，并结束该存储过程。
repeat.sqc	调用 REPEAT_STMT 过程。
rsultset.c	调用 MEDIAN_RESULT_SET 过程，显示平均工资，然后显示由 SQL 过程生成的结果集。此客户机是用 CLI API 编写的，它可接受结果集。
rsultset.db2	MEDIAN_RESULT_SET 过程从 “sample” 数据库的 “staff” 表中获取由 “dept” IN 参数指示的部门的职员的平均工资。将该平均值赋予工资 OUT 参数，并返回给 “rsultset” 客户机。该过程然后打开一个 WITH RETURN 游标，以返回工资值大于平均值的职员的结果集。该过程将结果集返回给客户机。
spserver.db2	此 CLP 脚本中的 SQL 过程演示基本的错误处理、嵌套存储过程调用，然后将结果集返回给客户机应用程序或发出调用的应用程序。可在 CLI samples 目录中使用 “spcall” 应用程序调用这些过程。还可使用位于 C 和 CPP samples 目录中的 “spclient” 应用程序调用不返回结果集的过程。
tbfn.db2	创建在 tbfnuse 样本中使用的表和表函数。在运行了 tbfnuse 脚本之后，将回滚所有更改并删除在此文件中创建的表和函数。
tbfnuse.db2	演示在 tbfnuse 样本中创建的表函数的使用。在此脚本结束时，会回滚语句并且会删除在 tbfn.db2 中创建的表和函数。

表 31. SQL 过程样本程序文件 (续)

样本程序名	程序描述
tbssel.sqc	如何从下列三项中选择一项: 插入、更新和删除。此样本调用从 tbsselcreate.db2 创建的 SQL 过程 BUY_COMPANY。BUY_COMPANY 包含从数据更改语句使用 SELECT 的用法示例。
tbsselcreate.db2	如何创建在 tbssel 程序中使用的表和过程。
tbsseldrop.db2	如何删除在 tbssel 程序中使用的表和过程。
whiles.db2	DEPT_MEDIAN 过程从 “sample” 数据库的 “staff” 表中获取由 “dept” IN 参数指示的部门的职员平均工资。将该平均值赋予工资 OUT 参数, 并返回给 “whiles” 客户机。然后 whiles 客户机打印平均工资。
whiles.sqc	调用 DEPT_MEDIAN 过程。

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

相关概念:

- 第 57 页的『样本文件』

相关任务:

- 第 127 页的『创建 SQL 过程』

## Visual Basic 样本

目录。ActiveX 数据对象: sqllib\samples\VB\ADO; Microsoft Transaction Server: sqllib\samples\VB\MTS; 远程数据对象: sqllib\samples\VB\RDO。

表 32. Visual Basic ActiveX 数据对象样本程序文件

样本程序名	程序描述
<b>客户机级别</b>	
cliExeSQL.bas	如何执行 SQL 语句。
cli_Info.bas	如何获取 / 设置客户机级别信息。
<b>数据库级别</b>	
dbConn.bas	如何连接数据库和与数据库断开连接。
dbInfo.bas	如何获取和设置数据库级别的信息。
dbCommit.bas	如何动态控制数据库级别的自动落实。
<b>数据类型级别</b>	
dtHier.bas	如何检索分层数据。
dtLob.bas	如何获取 LOB 数据。
<b>存储过程</b>	
spCall.bas	如何调用存储过程。
<b>用户定义的函数</b>	
udfUse.bas	如何创建和使用 UDT 和 UDF。

表 33. Visual Basic Microsoft Transaction Server 样本程序文件

样本程序名	程序描述
db2com.vbp	<p>此 Visual Basic 项目演示如何使用“Microsoft 事务处理服务器”更新数据库。它创建一个由客户机程序 db2mts.vbp 使用的服务器 DLL，并有四个类模块：</p> <ul style="list-style-type: none"> <li>• UpdateNumberColumn.cls</li> <li>• UpdateRow.cls</li> <li>• UpdateStringColumn.cls</li> <li>• VerifyUpdate.cls</li> </ul> <p>对于此程序，在 sample 数据库中创建了一个临时表 DB2MTS。</p>
db2mts.vbp	<p>这是一个客户机程序的 Visual Basic 项目，它使用“Microsoft 事务处理服务器”调用根据 db2com.vbp 创建的服务器 DLL。</p>
LCTransTest.vbp	<p>此 Visual Basic 项目演示 DB2 数据库上松散耦合的事务。它创建客户机程序 main.vbp 使用的服务器 DLL，并具有一个类模块 TestClass.cls。在样本数据库中创建了一个临时表 LCTEST。</p>
main.vbp	<p>这是使用 Microsoft Transaction Server 调用通过 LCTransTest.vbp 创建的服务器 DLL 的客户机程序的 Visual Basic 项目。</p>

表 34. Visual Basic 远程数据对象样本程序文件

样本程序名	程序描述
Bank.vbp	<p>一个 RDO 程序，它创建和维护银行支行的数据，能够对客户帐户执行事务处理。该程序可以使用用户指定的任何数据库，因为它包含 DDL，可以创建应用程序存储数据所需的表。</p>

有关最新的样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

**相关概念：**

- 第 57 页的『样本文件』

**相关任务：**

- 第 233 页的『使用 Visual Basic 构建 ADO 应用程序』
- 第 238 页的『使用 Visual Basic 构建 RDO 应用程序』
- 第 236 页的『使用 Visual Basic 构建松散耦合事务』
- 第 237 页的『Visual Basic 松散耦合事务项目故障诊断』

**相关参考：**

- 第 92 页的『Windows 管理规范样本』

有关最新的样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

## Visual Basic .NET 样本

目录: sql1lib\samples\.NET\vb。

表 35. 样本 Visual Basic .NET 程序文件

样本程序名	程序描述
<b>数据库级别 - 涉及 DB2 数据库对象的样本。</b>	
DbAuth.vb	如何授予 / 显示 / 撤销数据库级别的权限。
DbDatAdp.vb	如何使用 DB2DataAdapter。
DbDatMap.vb	如何设置 DataTable 和 DataColumn 映射。
DbDsetCn.vb	如何将现有约束添加至数据集。
DbEvent.vb	如何处理 DB2DataAdapter 事件。
DbUse.vb	如何使用数据库对象。
DbValue.vb	如何从数据库中获得单个值。
<b>表级别 - 涉及 DB2 表对象的样本。</b>	
TbConstr.vb	如何使用表约束。
TbInfo.vb	如何获取和设置表级别的信息。
TbPriv.vb	如何授予 / 显示 / 撤销表级别特权。
TbSel.vb	如何从下列三项中选择一项: 插入、更新和删除。
TbTrig.vb	如何对表使用触发器。
TbUse.vb	如何处理表数据和连接至数据库 / 与数据库断开连接。
<b>数据类型级别 - 涉及数据类型的样本。</b>	
DtLob.vb	如何使用 LOB 数据类型。
<b>存储过程 - 演示存储过程的样本。</b>	
SpCat.db2	删除在 SpServer.vb 中实现的过程然后创建它们。
SpClient.vb	用来调用 SpServer.vb 中的存储过程的客户机程序。
SpCreate.db2	创建在 SpServer.vb 中实现的外部过程。
SpDrop.db2	删除在 SpCreate.db2 中创建的外部过程。
SpReturn.vb	调用存储过程 EMP_DETAILS 并获得它的返回值的客户机应用程序。
SpServer.vb	在 SpCat.db2 中创建的过程的 Visual Basic .NET 外部代码实现。
EmpDetails.db2	创建名为 EMP_DETAILS 的存储过程的 CLP 脚本。
<b>用户定义的函数 - 演示 UDF 的样本。</b>	
UdfCat.db2	删除在 UdfSrv.vb 中实现的外部 UDF 然后创建它们。
UdfCli.vb	调用 UdfSrv.vb 中的用户定义的函数的客户机应用程序。
UdfCreate.db2	创建在 UdfSrv.vb 中实现的外部 UDF。
UdfDrop.db2	在 udfcreate.db2 中创建的外部 UDF。
UdfSrv.vb	UdfCli 调用的用户定义的标量函数。
<b>松散耦合的事务</b>	
empcat.bat	对客户机程序 SpReturn 的存储过程 EMP_DETAILS 进行编目。
LCTrans.vb	演示松散耦合的事务。
regCOM.bat	注册 LCTrans 程序的 COM+ 对象。
RootCOM.vb	此文件用来创建一个库组合件 RootCOM.dll。LCTrans.vb 引用在此文件中定义的类和方法。

表 35. 样本 Visual Basic .NET 程序文件 (续)

样本程序名	程序描述
SubCOM.vb	此文件用来创建一个库组合件 SubCOM.dll。LCTrans.vb 引用在此文件中定义的类和方法。

**相关概念:**

- 第 57 页的『样本文件』

**相关任务:**

- 第 243 页的『构建 Visual Basic .NET 应用程序』
- 第 246 页的『构建公共语言运行时 (CLR) .NET 例程』

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

## Visual C++ 样本

目录: sql1lib\samples\VC\ADO。

表 36. Visual C++ 样本程序文件

样本程序名	程序描述
BLOBAccess.dsw	此样本演示如何用 Microsoft Visual C++ 存取 ADO/Blob。它类似于 Visual Basic 样本 Blob.vbp。该 BLOB 样本有两个主要功能: 1. 从样本数据库读取 BLOB, 并将它显示到屏幕上。 2. 从文件读取 BLOB, 并将它插入到数据库中。(导入)
VarChar.dsp	一个 Visual C++ 程序, 它使用 ADO 将 VarChar 数据作为文本字段存取。它提供一个图形用户界面, 允许用户查看和更新 sample 数据库的 ORG 表中的数据。

**相关概念:**

- 第 253 页的『使用 Visual C++ 的对象链接与嵌入 (OLE) 自动化』
- 第 57 页的『样本文件』

**相关任务:**

- 第 251 页的『使用 Visual C++ 构建 ADO 应用程序』

有关最新样本更新, 请访问 DB2 应用程序开发样本 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

## Windows 管理规范样本

目录: sql1lib\samples\wmi。

表 37. Windows 管理规范样本程序文件

样本文件名	文件描述
backupdb.vbs	如何备份数据库
createdb.vbs	如何创建和删除数据库。



表 37. Windows 管理规范样本程序文件 (续)

样本文件名	文件描述
listsrvr.vbs	如何枚举服务器实例和启动 / 停止 DB2 实例。
perfmon.mof	perfmon.vbs 的 MOF 文件。
perfmon.vbs	如何获取 DB2 性能计数器。注意：必须首先运行“mofcomp perfmon.mof”。
regvar.mof	regvar.vbs 的 MOF 文件。
regvar.vbs	如何获取 DB2 注册表变量。注意：必须首先运行“mofcomp regvar.mof”。
restoredb.vbs	如何复原数据库。
rollfwddb.vbs	如何前滚数据库。
updatedbcfg.vbs	如何获取和更新数据库配置。
updatedbmcfg.vbs	如何获取和更新数据库管理器配置。

**相关概念:**

- 第 57 页的『样本文件』
- 第 233 页的『Windows 管理规范 (WMI)』

**相关参考:**

- 第 89 页的『Visual Basic 样本』

有关最新样本更新，请访问 DB2 应用程序开发样本 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

## 构建文件、makefile 和错误检查实用程序

### 构建文件

用来演示构建样本程序的文件在 UNIX® 上称为脚本文件，在 Windows® 上称为批处理文件。我们将它们统称为构建文件。它们包含受支持平台编译器的建议使用的编译和链接命令。

DB2® 在支持的平台上提供每一种语言的构建文件，平台上提供有这些构建文件所构建的程序类型，它们位于与每种语言的样本程序相同的目录中。下表列示了用于构建不同程序类型的不同构建文件类型。这些构建文件（除非另有指示）用于所有受支持平台上的受支持语言。构建文件在 Windows 上具有 .bat（批处理）的扩展名，但不包括在表中。对于 UNIX 平台，没有扩展名。

表 38. DB2 构建文件

构建文件	构建的程序类型
bldapp	应用程序
bldrtn	例程（存储过程和 UDF）
bldsqlj	Java™ SQLJ 应用程序
bldsqljs	Java SQLJ 例程（存储过程和 UDF）
bldmc	C/C++ 多连接应用程序
bldmt	C/C++ 多线程应用程序

表 38. DB2 构建文件 (续)

构建文件	构建的程序类型
bldcli	sqlproc samples 子目录中 SQL 过程的 CLI 客户机应用程序。

注: bldcli 文件与 samples/cli 目录中的 bldapp 文件相同。给予它另一个名称, 原因是嵌入式 C bldapp 文件还包括在 samples/sqlproc 目录中。

下表按平台和编程语言列示了构建文件, 并列示了这些文件所在的目录。在联机文档中, 构建文件名被热链接至 HTML 格式的源文件。用户还可以访问适当的样本目录中的文本文件。

表 39. 按语言和平台列示的构建文件

平台 → 语言	AIX®	HP-UX	Linux	Solaris	Windows
C samples/c	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.bat bldmt.bat bldmc.bat
C++ samples/cpp	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.bat bldmt.bat bldmc.bat
CLI samples/cli	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp.bat bldrtn.bat bldmc.bat
SQLJ samples/java/sqlj	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj.bat bldsqljs.bat
IBM® COBOL samples/cobol	bldapp bldrtn	不适用	不适用	不适用	bldapp.bat bldrtn.bat
Micro Focus COBOL samples/cobol_mf	bldapp bldrtn	bldapp bldrtn	bldapp bldrtn	bldapp bldrtn	bldapp.bat bldrtn.bat
C# samples\.\NET\cs	不适用	不适用	不适用	不适用	bldapp.bat bldrtn.bat
Visual Basic .NET samples\.\NET\vb	不适用	不适用	不适用	不适用	bldapp.bat bldrtn.bat

构建文件在文档中用于构建应用程序和例程, 原因是它们非常清晰地演示了 DB2 建议用于受支持编译器的编译和链接选项。除此之外, 还有很多其它可用的编译和链接选项, 用户可以随意试用它们。参阅您的编译器文档, 以了解提供的全部编译和链接选项。除构建样本程序外, 开发者还可以用构建文件构建自己的程序。这些样本程序可用作模板, 用户可修改这些模板来更好地进行编程开发。

为方便起见, 构建文件被设计为可以构建具有编译器允许的任何文件名的源文件。这不同于 makefile, 后者的程序名以硬编码的方式写入文件中。各 makefile 将存取构建文件以编译和链接它们创建的程序。构建文件使用 \$1 变量 (在 UNIX 上) 或 %1 变量 (在 Windows 操作系统上) 来在内部替换程序名。这些变量名的可增量的数字将替换可能需要的其它自变量。

这些构建文件允许进行快速而容易的实验，因为每一种构建文件适合于特定种类的程序构建（例如，独立应用程序、例程（存储过程和 UDF））或者更专门的程序类型（例如，多连接或多线程程序）。凡是受编译器支持的特定类型的程序都提供了对应类型的构建文件。

每次构建程序时，即使没有修改源文件，也会自动覆盖构建文件生成的对象文件和可执行文件。使用 `makefile` 时，情况则不是这样。这意味着开发者可重新构建现有的程序，而不必删除先前的对象文件和可执行文件或修改源文件。

构建文件包含样本数据库的缺省设置。如果用户正在存取另一个数据库，则该用户提供另一个参数来覆盖缺省值即可。如果他们始终使用该数据库，可将数据库名称以硬编码的方式写入构建文件来替换 `sample`。

对于嵌入式 SQL 程序，除了在 Windows 上使用 IBM COBOL 预编译器的情况之外，构建文件可以调用另一个文件 `embprep`，该文件包含嵌入式 SQL 程序的预编译和绑定步骤。这些步骤可能需要用户标识和密码的可选参数，这取决于嵌入式 SQL 程序的构建环境。

除了 SQLJ 之外，如果开发者正在数据库所在的服务器实例上构建程序，则用户标识和密码对于两者来说是公共的，所以不必再提供该标识和密码。另一方面，如果开发者在另一个实例中，比如在远程存取服务器数据库的客户机上，则必须提供这两个参数。

SQLJ 构建文件需要用于 `db2sqljcustomize` 定制程序的用户标识和密码（即使是存取本地数据库）。这遵循 DB2 通用 JDBC 驱动程序的约定。

最后，开发者为了自身的方便，可修改构建文件。除了可更改构建文件中的数据库名称之外（以上说明过），开发者还可轻易地将其它参数以硬编码的方式写入到该文件中、更改编译和链接选项或更改缺省的 DB2 实例路径。构建文件简单、明了和独特的特性使您可以轻松地修改它们，以满足自己的需要。

#### 相关概念:

- 第 95 页的『[Makefile](#)』
- 第 98 页的『[错误检查实用程序](#)』
- 第 57 页的『[样本文件](#)』

#### 相关参考:

- 第 8 页的『[AIX 支持的开发软件](#)』
- 第 10 页的『[HP-UX 支持的开发软件](#)』
- 第 12 页的『[Linux 支持的开发软件](#)』
- 第 17 页的『[Solaris 支持的开发软件](#)』
- 第 19 页的『[Windows 支持的开发软件](#)』

## Makefile

受支持编译器的每个 `samples` 目录包括用于构建目录内大多数所提供样本程序的 `makefile`。`makefile` 将调用构建文件来编译和链接每个程序。`makefile` 的语法及其命令的输出在一些重要的方面与构建文件不同。但是，通过将 `makefile` 用作构建文件的“前端”，用户就能够利用到 `makefile` 简单而功能强大的命令：

**make <program\_name>**

编译和链接指定的程序。

**make all**

编译和链接 makefile 中列出的所有程序。

**make clean**

删除 makefile 中列出的所有程序的中间文件，如对象文件。

**make cleanall**

删除 makefile 中列出的所有程序的所有中间文件和可执行文件。

Java™ 不经常使用 makefile，而 make 可执行文件不是随 Java Developer Kit 一起交付的。但是，DB2® 提供了 makefile 作为 Java 样本的选项，以使用户在想要利用 make 命令时可以使用。要使用 Java makefile，必须具有可用的 make 可执行文件，它通常与另一种语言编译器配合使用。

以下是按 DB2 为主编程语言 / API 提供的平台列示的 makefile，并且还列示了这些文件所在的样本目录。这些文件在联机文档中是热链接的，且它们构建的样本程序是在它们内部链接的。还可以在样本目录中存取这些文件。

表 40. 按平台列示的样本 makefile

平台 → 语言	AIX®	HP-UX	Linux	Solaris	Windows®
C samples/c	makefile	makefile	makefile	makefile	makefile
C++ samples/cpp	makefile	makefile	makefile	makefile	makefile
CLI samples/cli	makefile	makefile	makefile	makefile	makefile
JDBC samples/java/jdbc	makefile	makefile	makefile	makefile	makefile
SQLJ samples/java/sqlj	makefile	makefile	makefile	makefile	makefile
IBM® COBOL samples/cobol	makefile	不适用	不适用	不适用	makefile
Micro Focus COBOL samples/cobol_mf	makefile	makefile	makefile	makefile	makefile
SQL 过程 samples/sqlproc	makefile	makefile	makefile	makefile	makefile
C# samples\.\NET\cs	不适用	不适用	不适用	不适用	makefile
Visual Basic .NET samples\.\NET\vb	不适用	不适用	不适用	不适用	makefile

与构建文件不同，makefile 将不覆盖它当中列示的程序的现有中间文件和可执行文件。如果部分文件已有可执行文件，使用 make all 命令可以更快地为其它文件创建可执行文件，因为 make all 将忽略已有可执行文件的那些文件。但这也说明了 make clean 和 make cleanall 命令的用处，可使用这两个命令删除那些不需要的现有对象文件和可执行文件。

makefile 可用于程序开发。因为它们需要就在文件自身内对程序名进行硬编码，所以您可能会觉得 makefile 使用起来没有构建文件那么方便，但是如果想要利用到 make 命令的强大功能和便利性，则不妨可以考虑一下使用这一方式。

makefile 按以变量表示的若干个客户机和服务器程序类别来组织它们调用的程序（请参阅 makefile 以了解详细信息）。如果正在将程序添加至 makefile，则应确保添加的程序是以正确的变量存取的。例如，可在任何客户机上运行的程序（本地（对于服务器）或远程）被置于 client\_run 变量之下。

还需要在 cleanall 变量下指定程序名以确保可由 make cleanall 命令删除生成的可执行文件。另外，如果它是一个嵌入式 SQL 程序，则在 clean 变量下指定作为预编译结果创建的非嵌入式 SQL 文件，以便 make clean 命令（以及调用它的 make cleanall 命令）将删除非嵌入式 SQL 文件。

另外，还需要使用正确的语法指定新文件来调用适当的构建文件以编译和链接程序。

要了解新文件需要在何处添加至其中一个样本 makefile，以下是嵌入式 SQL 程序 dbauth 在 AIX C makefile 中所在的所有位置：

```
*****
#
#           2f - make client_run
*****

client_run : \
    cli_info clisnap \
    dbauth dbconn dbcreate dbinfo dbmcon \
    . . .
*****
#
#           2g - make clean
*****

clean :
    $(ERASE) *.o
    $(ERASE) *.DEL *.TXT *.MSG
    $(ERASE) dbauth.c dbcfg.c dbconn.c dbmcon.c dbmcon1.c dbmcon2.c
    . . .
*****
#
#           2h - make cleanall
*****

cleanall : \
    clean
    $(ERASE) *.bnd
    $(ERASE) cli_info clisnap
    $(ERASE) dbauth dbcfg dbconn dbcreate dbinfo dbmcon dbmcon1 dbmcon2
    . . .
*****
#
#           3b - regular samples, embedded SQL
*****

dbauth :
    $(BLDAPP) dbauth $(ALIAS) $(UID) $(PWD)
```

上面最后一行中程序名后面的三个变量 ALIAS、UID 和 PWD 分别表示数据库的数据库别名、用户标识和密码。这些变量被传递到构建文件中（在这种情况下，bldapp 构建文件由 BLDAPP 变量表示）。如果程序使用嵌入式 SQL，则 ALIAS、UID 和 PWD 依次被传递至 embprep 预编译和绑定脚本（构建文件将调用该脚本）。在使用 makefile 之前，可能需要更改这些变量的值。缺省情况下，ALIAS 设置为 sample 数据库，而 UID 和 PWD 则未设置任何值。

UID 和 PWD 是可选参数，如果用户已在服务器数据库所在的实例中工作，则不必设置这些参数。然而，如果情况不是这样（例如，如果用户正在从客户机远程连接至服务器），则用户需要修改 `makefile` 以将正确的值赋予 UID 和 PWD 变量，而后才可存取数据库。

对于多连接程序，C、CLI 和 C++ `makefile` 还具有另一个数据库别名 ALIAS2，缺省情况下它设置为 `sample2` 数据库。相应的用户标识和密码变量 UID2 和 PWD2 未设置任何值。如 UID 和 PWD 变量一样，它们并不需要值（如果在本地存取了另一个数据库的话）。

`makefile` 还定义了 ERASE 变量以在调用 `make clean` 和 `make cleanall` 命令时删除文件。在 UNIX<sup>®</sup> 上，该变量设置为 `rm -f`；而在 Windows 上，它设置为 `del`。

#### 相关概念:

- 第 93 页的『构建文件』
- 第 98 页的『错误检查实用程序』
- 第 57 页的『样本文件』

#### 相关参考:

- 第 8 页的『AIX 支持的开发软件』
- 第 10 页的『HP-UX 支持的开发软件』
- 第 12 页的『Linux 支持的开发软件』
- 第 17 页的『Solaris 支持的开发软件』
- 第 19 页的『Windows 支持的开发软件』

## 错误检查实用程序

“DB2<sup>®</sup> AD 客户机”提供了几个实用程序文件。这些文件具有错误检查和打印错误消息的功能。在 `samples` 目录中为每一种语言都提供了实用程序文件。当用于某个应用程序时，错误检查实用程序文件会提供有用的错误消息，从而使调试 DB2 程序更容易。大多数错误检查实用程序使用 DB2 API `GET SQLSTATE MESSAGE (sqllogstt)` 和 `GETERROR MESSAGE (sqlaintp)` 以获取与在程序执行过程中遇到的问题相关的相应 SQLSTATE 和 SQLCA 信息。DB2 CLI 实用程序文件 `utilcli.c` 不使用这些 DB2 API，而使用等效的 DB2 CLI 语句。使用所有错误检查实用程序打印描述性错误消息，以便开发者快速了解问题。

某些 DB2 程序（例如：例程（存储过程和用户定义的函数））不需要使用实用程序。对于 Java<sup>™</sup>，这些实用程序也不是必要的，因为发生异常时会抛出 `SQLException` 对象。

以下是 DB2 支持的用于不同编程语言的编译器所用的错误检查实用程序文件:

表 41. 按语言列示的错误检查实用程序文件

语言	非嵌入式 SQL 源文件	非嵌入式 SQL 头文件	嵌入式 SQL 源文件	嵌入式 SQL 头文件
C samples/c	utilapi.c	utilapi.h	utilemb.sqc	utilemb.h
C++ samples/cpp	utilapi.C	utilapi.h	utilemb.sqC	utilemb.h
CLI samples/cli	utilcli.c	utilcli.h	不适用	不适用
IBM® COBOL samples/cobol	checkerr.cbl	不适用	不适用	不适用
Micro Focus COBOL samples/cobol_mf	checkerr.cbl	不适用	不适用	不适用

为了使用这些实用程序函数，首先必须编译该实用程序文件，然后在创建目标程序的可执行文件期间链接它的对象文件。由 samples 目录中的 makefile 和构建文件为那些需要错误检查实用程序的程序完成此操作。

以下示例演示在 DB2 程序中如何使用错误检查实用程序。utilemb.h 头文件定义代替函数 SqlInfoPrint() 和 TransRollback() 的 EMB\_SQL\_CHECK 宏:

```

/* macro for embedded SQL checking */
#define EMB_SQL_CHECK( MSG_STR )      \
SqlInfoPrint(MSG_STR, &sqlca, __LINE__, __FILE__);    \
if (sqlca.sqlcode < 0)                \
{                                       \
    TransRollback();                  \
    return 1;                          \
}

```

SqlInfoPrint() 检查 SQLCODE 标志。并打印出与此标志指示的特定错误相关的任何可用信息。还指出该错误发生在源代码中的何处。TransRollback() 允许实用程序文件安全回滚发生错误的事务。此函数需要调用嵌入式 SQL 语句来连接数据库并执行回滚。下面是一个示例，说明 C 程序 dbuse 如何通过使用宏并对 SqlInfoPrint() 函数的 MSG\_STR 参数提供值 "Delete with host variables -- Execute" 来调用实用程序函数:

```

EXEC SQL DELETE FROM org
      WHERE deptnumb = :hostVar1 AND
            division = :hostVar2;
      EMB_SQL_CHECK("Delete with host variables -- Execute");

```

EMB\_SQL\_CHECK 宏将确保: 如果 DELETE 语句失败, 将会安全回滚事务, 并打印出一条相应的错误消息。

我们鼓励开发者在创建他们自己的 DB2 程序时使用并扩展这些错误检查实用程序。

#### 相关概念:

- 第 93 页的『构建文件』
- 第 95 页的『Makefile』
- 第 57 页的『样本文件』





---

## 第 2 部分 构建和运行独立于平台的应用程序



---

## 第 4 章 Java

Java 样本程序 . . . . .	103	SQLJ 应用程序和 applet 的 UNIX 构建脚本 . . . . .	113
Java applet 注意事项 . . . . .	104	UNIX 的 SQLJ 应用程序和 applet 选项. . . . .	115
JDBC . . . . .	106	SQLJ 应用程序和 applet 的 Windows 批处理文件 . . . . .	115
构建 JDBC applet . . . . .	106	Windows 的 SQLJ 应用程序和 applet 选项. . . . .	117
构建 JDBC 应用程序. . . . .	107	构建 SQLJ 例程 . . . . .	117
构建 JDBC 例程 . . . . .	108	SQLJ 例程的 UNIX 构建脚本 . . . . .	119
SQLJ . . . . .	110	UNIX 的 SQLJ 例程选项 . . . . .	120
构建 SQLJ 程序 . . . . .	110	SQLJ 例程的 Windows 批处理文件 . . . . .	120
构建 SQLJ applet . . . . .	111	Windows 的 SQLJ 例程选项 . . . . .	122
构建 SQLJ 应用程序 . . . . .	112		

本章提供了构建 Java applet 和应用程序的详细信息。有关最新的 DB2 Java 应用程序开发更新内容，请访问以下 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad/v8/java>

---

### Java 样本程序

DB2<sup>®</sup> 提供了一些样本程序来演示构建和运行独占使用动态 SQL 的 JDBC 程序和使用静态 SQL 的 SQLJ 程序。在 java samples 目录下存在用于 JDBC 和 SQLJ samples 的独立目录。以下是 UNIX<sup>®</sup> 和 Windows<sup>®</sup> 操作系统上的 Java<sup>™</sup> samples 目录结构：

- 在 UNIX 上：

- sqllib/samples/java**

- 包含在所有子目录中 Java 样本程序的 README 文件。

- sqllib/samples/java/jdbc**

- 包含 JDBC 样本程序文件。

- sqllib/samples/java/sqlj**

- 包含 SQLJ 样本程序。

- sqllib/samples/java/WebSphere**

- 包含 WebSphere 样本程序。

- sqllib/samples/java/plugin**

- 包含 DB2 控制中心的插件示例文件。

- sqllib/samples/java/plugin/doc**

- 包含插件接口的 javadoc 文件。

- 在 Windows 上：

- sqllib\samples\java**

- 包含在所有子目录中 Java 样本程序的 README 文件。

- sqllib\samples\java\jdbc**

- 包含 JDBC 样本程序。

- sqllib\samples\java\sqlj**

- 包含 SQLJ 样本程序。

### **sqllib\samples\java\WebSphere**

包含 WebSphere 样本程序。

### **sqllib\samples\java\plugin**

包含 DB2 控制中心的插件示例文件。

### **sqllib\samples\java\plugin\doc**

包含插件接口的 javadoc 文件。

SQLJ samples 目录包含构建文件（在 UNIX 上为脚本，而在 Windows 上为批处理文件）来构建嵌入式 SQL Java 程序。JDBC 目录不包含构建文件，因为在命令行上使用 javac 构建 JDBC 程序很简单，并不需要构建文件。

JDBC 和 SQLJ samples 目录也都包含可选的 makefile。Makefile 并与 Java 配合使用的情况并不多见，因而 Java Development Kit (JDK) 不与 make 可执行文件一起交付。DB2 提供了 Java 样本 makefile，以便用户在想要利用这些文件提供的便利时用得上。每个 Java makefile 将在 JDBC 或 SQLJ samples 目录中构建所有提供的样本程序。可以使用将与另一个语言编译器配合使用的 make 程序（例如，gnumake）。

提供了两个 SQLJ 构建文件：UNIX 上的 blsqlj 或 Windows 上的 blsqlj.bat（它们构建 SQLJ applet 和应用程序）以及 UNIX 上的 blsqljs 或 Windows 上的 blsqljs.bat（它们将构建 SQLJ 例程（存储过程和用户定义的函数））。

#### **相关任务:**

- 第 26 页的『设置 Java 环境』
- 第 106 页的『构建 JDBC applet』
- 第 107 页的『构建 JDBC 应用程序』
- 第 108 页的『构建 JDBC 例程』
- 第 111 页的『构建 SQLJ applet』
- 第 112 页的『构建 SQLJ 应用程序』
- 第 117 页的『构建 SQLJ 例程』
- 第 110 页的『构建 SQLJ 程序』

#### **相关参考:**

- 第 75 页的『JDBC 样本』
- 第 78 页的『SQLJ 样本』
- 第 79 页的『Java WebSphere 样本』
- 第 80 页的『Java 插件样本』

---

## **Java applet 注意事项**

可以通过使用 Java™ applet 来存取 DB2® 数据库。当使用它们时，请注意以下几点：

1. 如果正在使用现在建议不使用的类型 3 驱动程序（也称为“net”驱动程序），Java applet 使用的 db2java.zip 文件必须与 JDBC applet 服务器处于相同的修订包级别。正常情况下，db2java.zip 文件是从运行 JDBC applet 服务器的“Web 服务器”上装入的。这样可确保正确的匹配。但是，如果您的配置中 Java applet 从不同位置装

入 db2java.zip 文件，将会产生不匹配现象。在连接时，两个文件之间的“修订包”级别必须严格匹配。如果检测到不匹配现象，将会拒绝连接，并且客户机会接收到以下异常中的一个：

- 如果 db2java.zip 是 DB2 版本 7 修订包 2 或更高版本：

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0621E  Unsupported JDBC server configuration.
```

- 如果 db2java.zip 的级别早于 DB2 版本 7 修订包 2：

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0601E  Invalid statement handle or statement is closed.
SQLSTATE=S1000
```

如果出现了不匹配现象，JDBC applet 服务器会将以下消息之一记录在 jdbcerr.log 文件中：

- 如果 JDBC applet 服务器是 DB2 版本 7 修订包 2 或更高版本：

```
jdbcFSQLConnect: JDBC Applet Server and client (db2java.zip)
versions do not match. Unable to proceed with connection., einfo= -111
```

- 如果 JDBC applet 服务器的级别早于 DB2 版本 7 修订包 2：

```
jdbcServiceConnection(): Invalid Request Received., einfo= 0
```

**注：**不应将 sqllib\samples\java (Windows) 或 sqllib/samples/java (UNIX) 中的 db2JDBCVersion.java 样本文件与 DB2 版本 8 配合使用。可以将此程序与 DB2 版本 7 配合使用来检查当前正在使用哪个版本的 DB2 JDBC 驱动程序以及是否已为它正确设置了 JDBC 环境。

强烈建议用户将其 applet 迁移至 DB2 通用 JDBC 驱动程序。

2. 对于由几个 Java 类组成的一个较大的 JDBC 或 SQLJ applet，可以选择将它的所有类打包到一个 JAR 文件中。对于 SQLJ applet，还必须将它的序列化概要文件与它的类打包在一起。如果您选择这样做，则要将 JAR 文件添加到“applet”标记中的 archive 参数中。有关详细信息，请参阅 Java Developer Kit V1.3 文档。

对于 SQLJ applet，某些浏览器尚不支持从与 applet 相关联的资源文件中装入序列化对象。例如，当尝试在这些浏览器中装入提供的样本 applet Applet 时，将会出现以下错误消息：

```
java.lang.ClassNotFoundException: Applet_SJProfile0
```

一个变通方法是，使用一个将序列化概要文件转换为以 Java 类格式存储的概要文件的实用程序。该实用程序是一个 Java 类，称为 sqlj.runtime.profile.util.SerProfileToClass。它将序列化概要文件资源文件作为输入，并生成一个包含该概要文件的 Java 类作为输出。可使用下列其中一个命令转换概要文件：

```
profconv Applet_SJProfile0.ser
```

或

```
java sqlj.runtime.profile.util.SerProfileToClass Applet_SJProfile0.ser
```

结果，创建类 `Applt_SJProfile0.class`。用 `.class` 格式的概要文件替换 applet 使用的 `.ser` 格式的所有概要文件，这样就应当可以解决问题了。

3. 可以将文件 `db2java.zip` 和 / 或 `db2jcc.jar` 放置在可从 Web 站点装入的几个 applet 共享的目录中。`db2java.zip` 用于使用 JDBC 类型 3 驱动程序的 applet; `db2jcc.jar` 用于使用 DB2 通用 JDBC 驱动程序的 applet 或任何 SQLJ applet。在 Windows<sup>®</sup> 操作系统上，这些文件位于 `sqllib\java` 目录中，而在 UNIX<sup>®</sup> 上则位于 `sqllib/java` 目录中。您可能需要将 `codebase` 参数添加到 HTML 文件的“applet”标记中，以标识该目录。有关详细信息，请参阅 `Java Developer Kit V1.3` 文档。
4. 自 DB2 版本 5.2 起，已将信号处理添加到 JDBC applet 服务器（侦听器）`db2jd`，使其功能更强大。这样，就不能使用 `CTRL-C` 命令来杀死 `db2jd`。因此，终止侦听器的唯一方法是通过使用 `kill -9`（对于 Unix）或“任务管理器”（对于 Windows）来停止进程。
5. 有关在 Web 服务器上（特别在 Domino<sup>™</sup> Go Webserver 上）运行 DB2 Java applet 的信息，请访问：

<http://www.ibm.com/software/data/db2/db2lotus/gojava.htm>

#### 相关任务:

- 第 26 页的『设置 Java 环境』
- 第 106 页的『构建 JDBC applet』
- 第 111 页的『构建 SQLJ applet』

---

## JDBC

### 构建 JDBC applet

`Applt` 演示一个访问 DB2 数据库的动态 SQL Java applet。

#### 过程:

可以使用现在建议不使用的类型 3 驱动程序（也称为“net”驱动程序），也可以使用随 DB2 Java Enablement 安装的通用 JDBC 驱动程序。关于与这两种驱动程序连接的章节将在后面提供。强烈建议将 applet 迁移至通用 JDBC 驱动程序。

要通过在命令行输入的命令构建和运行 JDBC applet `Applt`，应确保在您的 DB2 机器（服务器或客户机）上安装了 Web 服务器且它在该机器上运行，或通过您的客户机工作目录中输入以下命令来使用 Java Development Kit 附带的 applet 查看器：

```
appletviewer Applt.html
```

#### 与类型 3（“net”）驱动程序连接

要与类型 3 驱动程序连接，首先要根据 `Applt.html` 文件中的指示信息修改该文件。然后，在 `Applt.html` 中指定的 TCP/IP 端口上启动 JDBC applet 服务器。例如，如果在 `Applt.html` 中指定了 `param name=port value='6789'`，应输入：

```
db2jstrt 6789
```

请确定连接字符串中的 JDBC 端口号为建议的缺省值“6789”。只有在确定该号码不会与其它端口号冲突的情况下才能更改号码。不要使用数据库端口号“50000”。

## 与通用 JDBC 驱动程序连接

要与通用 JDBC 驱动程序连接，请根据文件中的指示信息修改 `Applt.html` 文件。对于 TCP/IP 端口号，请使用数据库端口号“50000”。

### 构建 applet

1. 使用以下命令编译 `Applt.java` 来生成文件 `Applt.class`:

```
javac Applt.java
```

2. 确保您的 Web 浏览器可存取工作目录。否则，将 `Applt.class` 和 `Applt.html` 复制到可存取的目录中。
3. 如果正在使用类型 3 驱动程序，则将 Windows 上的 `sql1lib\java\db2java.zip` 或 UNIX 上的 `sql1lib/java/db2java.zip` 复制到 `Applt.class` 和 `Applt.html` 所在的目录中。

如果正在使用通用 JDBC 驱动程序，则将 Windows 上的 `sql1lib\java\db2jcc.jar` 或 UNIX 上的 `sql1lib/java/db2jcc.jar` 复制到 `Applt.class` 和 `Applt.html` 所在的目录中。

4. 在您的客户机上，启动 Web 浏览器（它必须支持 Java 1.3）并装入 `Applt.html`。

也可使用 `Java makefile` 构建此程序。

#### 相关概念:

- 第 104 页的『Java applet 注意事项』

#### 相关任务:

- 第 107 页的『构建 JDBC 应用程序』
- 第 108 页的『构建 JDBC 例程』
- 第 111 页的『构建 SQLJ applet』

#### 相关参考:

- 第 75 页的『JDBC 样本』

#### 相关样本:

- 『`Applt.java` -- A Java applet that use JDBC applet driver to access a database (JDBC)』

## 构建 JDBC 应用程序

`DbInfo` 演示一个存取 DB2 数据库的动态 SQL Java 应用程序。

#### 过程:

要通过在命令行输入命令来构建和运行此应用程序:

1. 使用以下命令编译 `DbInfo.java` 来生成文件 `DbInfo.class`:

```
javac DbInfo.java
```

2. 使用如下命令，对该应用程序运行 `java` 解释器:

```
java DbInfo
```

也可使用 `Java makefile` 构建此程序。

**注:** 如果正在 Unix 上的 64 位 DB2 实例中运行 Java 应用程序 (但 Java Developer Kit 为 32 位), 则您必须在运行应用程序之前更改 DB2 库路径。例如, 在 AIX 上:

- 如果使用 bash 或 Korn shell:

```
export LIBPATH=$HOME/sql1lib/lib32
```
- 如果使用 C shell:

```
setenv LIBPATH $HOME/sql1lib/lib32
```

#### 相关任务:

- 第 106 页的『构建 JDBC applet』
- 第 108 页的『构建 JDBC 例程』
- 第 112 页的『构建 SQLJ 应用程序』

#### 相关参考:

- 第 75 页的『JDBC 样本』

#### 相关样本:

- 『DbInfo.java -- How to get/set info in a database (JDBC)』

## 构建 JDBC 例程

DB2 提供了演示 JDBC 例程 (存储过程和用户定义的函数) 的样本程序, 在 UNIX 上, 这些样本程序在 `samples/java/jdbc` 目录中, 而在 Windows 上, 则在 `samples\java\jdbc` 目录中。例程是在服务器上编译和存储的。当客户机应用程序调用存储过程时, 它们存取服务器数据库并将信息返回至客户机应用程序。

#### 过程:

下列示例显示如何构建具有下列内容的例程:

- 存储过程
- 不带有 SQL 语句的用户定义的函数
- 带有 SQL 语句的用户定义的函数

#### 存储过程

SpServer 演示动态 SQL PARAMETER STYLE JAVA 存储过程。

要使用命令行在服务器上构建和运行此程序:

1. 使用以下命令编译 SpServer.java 来生成文件 SpServer.class:

```
javac SpServer.java
```

2. 将 SpServer.class 复制到 Windows 操作系统上的 `sql1lib/function` 目录中或 UNIX 上的 `sql1lib/function` 目录中。
3. 接着, 通过在服务器上运行 `spcat` 脚本对例程进行编目。输入:

```
spcat
```

此脚本连接至样本数据库, 如果先前通过调用 SpDrop.db2 对这些例程进行了编目, 则取消对它们的编目, 然后通过调用 SpCreate.db2 对它们进行编目, 最后与数据库断开连接。还可以分别运行 SpDrop.db2 和 SpCreate.db2 脚本。



- 接着，停止数据库然后重新启动它以允许识别新类文件。如有必要，将类文件的文件方式设置为“读”，以便可由受防护用户读取它。
- 编译并运行 SpClient 客户机应用程序以存取存储过程类。

### 不带有 SQL 语句的用户定义的函数

UDFsrv 是一个不包含 SQL 语句的用户定义的函数库。DB2 提供了 JDBC 客户机应用程序 UDFcli 和 SQLJ 客户机应用程序 UDFcli，它们可以存取 UDFsrv 库。

要使用命令行在服务器上构建和运行 UDF 程序：

- 使用如下命令编译 UDFsrv.java 来生成文件 UDFsrv.class：

```
javac UDFsrv.java
```

- 将 UDFsrv.class 复制到 Windows 操作系统上的 sqllib\function 目录中或 UNIX 上的 sqllib/function 目录中。
- 要存取 UDFsrv 库，可使用 JDBC 或 SQLJ 客户机应用程序。客户机程序的两种版本都包含 CREATE FUNCTION SQL 语句（可使用它来向数据库注册包含在 UDFsrv 中的 UDF），并且还包含使用 UDF 的 SQL 语句（一旦注册了这些 UDF 的话）。

### 带有 SQL 语句的用户定义的函数

UDFsqlsv 是一个包含 SQL 语句的用户定义的函数库。DB2 提供了 JDBC 客户机应用程序 UDFsqlcl 来存取 UDFsqlsv 库。

要使用命令行在服务器上构建和运行 UDF 程序：

- 使用以下命令编译 UDFsqlsv.java 来生成文件 UDFsqlsv.class：

```
javac UDFsqlsv.java
```

- 将 UDFsqlsv.class 复制到 Windows 操作系统上的 sqllib\function 目录中或 UNIX 上的 sqllib/function 目录中。
- 要存取 UDFsqlsv 库，使用客户机程序 UDFsqlcl，它包含您用来向数据库注册包含在 UDFsqlsv 中的 UDF 的 CREATE FUNCTION SQL 语句。客户机程序还包含使用 UDF 的 SQL 语句（一旦注册了这些 UDF 的话）。

还可以使用 Java makefile 来构建上述程序。

### 相关任务：

- 第 106 页的『构建 JDBC applet』
- 第 107 页的『构建 JDBC 应用程序』
- 第 117 页的『构建 SQLJ 例程』

### 相关参考：

- 第 75 页的『JDBC 样本』

### 相关样本：

- 『spcat -- To catalog SQLj stored procedures on UNIX』
- 『SpClient.java -- Call a variety of types of stored procedures from SpServer.java (JDBC)』
- 『SpCreate.db2 -- How to catalog the stored procedures contained in SpServer.java』
- 『SpDrop.db2 -- How to uncatalog the stored procedures contained in SpServer.java』

- 『 SpServer.java -- Provide a variety of types of stored procedures to be called from (JDBC) 』
- 『 UDFcli.java -- Call the UDFs in UDFsrv.java (JDBC) 』
- 『 UDFCreate.db2 -- How to catalog the Java UDFs contained in UDFsrv.java 』
- 『 UDFDrop.db2 -- How to uncatalog the Java UDFs contained in UDFsrv.java 』
- 『 UDFsCreate.db2 -- How to catalog the UDFs contained in UDFsqlsv.java 』
- 『 UDFsDrop.db2 -- How to uncatalog the UDFs contained in UDFsqlsv.java 』
- 『 UDFsqlcl.java -- Call the UDFs in UDFsqlsv.java (JDBC) 』
- 『 UDFsqlsv.java -- Provide UDFs to be called by UDFsqlcl.java (JDBC) 』
- 『 UDFsrv.java -- Provide UDFs to be called by UDFcli.java (JDBC) 』

---

## SQLJ

### 构建 SQLJ 程序

DB2 提供构建文件以进行构建 SQLJ 样本程序。对于 applet 和应用程序，您可以在 UNIX 上使用 bldsqlj 脚本或在 Windows 上使用 bldsqlj.bat 批处理文件。对于例程（存储过程和用户定义的函数），可以使用 UNIX 上的 bldsqljs 脚本或 Windows 上的 bldsqljs.bat 批处理文件。

随 DB2 一起交付的 SQLJ 转换程序调用 Java 编译器将已转换的 .java 文件编译为 .class 文件。因此，构建文件使用 **sqlj** 命令来执行这两个操作。

注:

1. 在 DB2 的先前版本中，db2profcc 命令使用了以下格式为 -url=jdbc:db2:dbname 的 URL，其中 dbname 是本地已编目数据库别名。新的 db2sqljcustomize 命令遵循 DB2 通用 JDBC 驱动程序的约定: -url jdbc:db2://hostname:portnumber/dbname，其中 hostname 是 DB2 服务器的名称，portnumber 是 DB2 服务器的 TCP/IP 侦听器端口号，而 dbname 是 DB2 服务器上编目的数据库别名。这意味着必须为 TCP/IP 连接配置“DB2 服务器”。
2. 必须用 DB2 版本 8 sqlj 命令重新转换已用先前版本的 sqlj 命令转换的 SQLJ 程序并用 db2sqljcustomize 命令定制它们。
3. DB2 SQLJ 概要文件打印程序 db2sqljprint，它以纯文本格式打印 DB2 概要文件的内容。

过程:

要构建不同类型的 DB2 SQLJ 程序，请参阅下列主题:

- 第 111 页的『构建 SQLJ applet』
- 第 112 页的『构建 SQLJ 应用程序』
- 第 117 页的『构建 SQLJ 例程』

相关概念:

- 第 103 页的『Java 样本程序』
- 第 104 页的『Java applet 注意事项』

相关任务:

- 第 111 页的『构建 SQLJ applet』
- 第 112 页的『构建 SQLJ 应用程序』
- 第 117 页的『构建 SQLJ 例程』

**相关参考:**

- 『db2sqljcustomize - DB2 SQLJ Profile Customizer Command』 ( *Command Reference* )
- 『db2sqljprint - DB2 SQLJ Profile Printer Command』 ( *Command Reference* )
- 第 78 页的『SQLJ 样本』
- 『sqlj - DB2 SQLJ Translator Command』 ( *Command Reference* )

## 构建 SQLJ applet

以下步骤显示如何构建 Applet 样本，它将演示访问 DB2 数据库的 SQLJ applet。这些步骤使用构建文件 bldsqlj (UNIX) 或 bldsqlj.bat (Windows)，它包含构建 SQLJ applet 或应用程序的命令。

该构建文件最多使用 6 个参数：在 UNIX 上为 \$1、\$2、\$3、\$4、\$5 和 \$6，而在 Windows 上为 %1、%2、%3、%4、%5 和 %6。第一个参数指定您的程序的名称；第二个参数指定数据库实例的用户标识；第三个参数指定密码；第四个参数指定服务器名；第五个参数指定端口号；而第六个参数指定数据库名称。对于除第一个参数（程序名）之外的所有参数，都可以使用缺省值。有关使用缺省参数值的详细信息，请参阅构建文件。

**过程:**

| 可以使用现在建议不使用的类型 3 驱动程序（也称为“net”驱动程序），也可以使用  
| 随 DB2 Java Enablement 安装的通用 JDBC 驱动程序。关于与这两种驱动程序连接的  
| 章节将在后面提供。强烈建议将 applet 迁移至通用 JDBC 驱动程序。

要运行此 applet，应确保在您的 DB2 机器（服务器或客户机）上安装了 Web 服务器且它在该机器上运行，或通过您的客户机工作目录中输入以下命令来使用 Java Development Kit 附带的 applet 查看器：

```
appletviewer Applet.html
```

**与类型 3 (“net”) 驱动程序连接**

要与类型 3 驱动程序连接，首先要根据 Applet.html 文件中的指示信息修改该文件。然后，在 Applet.html 中指定的 TCP/IP 端口上启动 JDBC applet 服务器。例如，如果在 Applet.html 中指定了 param name=port value='6789'，应输入：

```
db2jstrt 6789
```

请确定连接字符串中的 JDBC 端口号为建议的缺省值“6789”。只有在确定该号码不会与其它端口号冲突的情况下才能更改号码。不要使用数据库端口号“50000”。

| **与通用 JDBC 驱动程序连接**

| 要与通用 JDBC 驱动程序连接，请根据文件中的指示信息修改 Applet.html 文件。对于  
| TCP/IP 端口号，应使用数据库端口号“50000”。

**构建 applet**

1. 使用如下命令构建该 applet:

```
bldsqlj Applt <userid> <password> <server_name> <port_number> <db_name>
```

其中所有参数（除了程序名外）都可以具有缺省值，如构建文件所述的那样。

2. 应确保您的 Web 浏览器或您的 applet 查看器（如果正在使用它）可存取您的工作目录。如果不可存取您的目录，则将下列文件复制到可存取的目录中：

```
Applt.html                      Applt.class
Applt_Cursor1.class             Applt_Cursor2.class
Applt_SJProfileKeys.class       Applt_SJProfile0.ser
```

3. 如果使用类型 3 驱动程序，则将 Windows 上的 sqllib\java\db2jcc.jar 和 sqllib\java\db2java.zip 或 UNIX 上的 sqllib/java/db2jcc.jar 和 sqllib/java/db2java.zip 复制到 Applt.class 和 Applt.html 所在的目录中。

如果正在使用通用 JDBC 驱动程序，则将 Windows 上的 sqllib\java\db2jcc.jar 或 UNIX 上的 sqllib/java/db2jcc.jar 复制到 Applt.class 和 Applt.html 所在的目录中。

4. 在客户机上，启动 Web 浏览器（它必须支持 Java Developer Kit 1.3）或 applet 查看器，并装入 Applt.html。

也可使用 Java makefile 构建此程序。

#### 相关概念:

- 第 104 页的『Java applet 注意事项』

#### 相关任务:

- 第 106 页的『构建 JDBC applet』
- 第 112 页的『构建 SQLJ 应用程序』
- 第 117 页的『构建 SQLJ 例程』

#### 相关参考:

- 第 115 页的『UNIX 的 SQLJ 应用程序和 applet 选项』
- 第 117 页的『Windows 的 SQLJ 应用程序和 applet 选项』
- 第 78 页的『SQLJ 样本』

#### 相关样本:

- 『Applt.sqlj -- An SQLJ applet that uses a JDBC applet driver to access a database (SQLj)』
- 『bldsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows』
- 『bldsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX』

## 构建 SQLJ 应用程序

下列步骤显示如何构建 TbMod 样本，它将演示存取 DB2 数据库的 SQLJ 应用程序。这些步骤使用构建文件 bldsqlj（UNIX）或 bldsqlj.bat（Windows），它包含构建 SQLJ applet 或应用程序的命令。

该构建文件最多使用 6 个参数：在 UNIX 上为 \$1、\$2、\$3、\$4、\$5 和 \$6，而在 Windows 上为 %1、%2、%3、%4、%5 和 %6。第一个参数指定您的程序的名称。第二个参数指定数据库实例的用户标识，第三个参数指定密码。第四个参数指定服务器

名。第五个参数指定端口号。而第六个参数指定数据库名称。对于除第一个参数（程序名）之外的所有参数，都可以使用缺省值。有关使用缺省参数值的详细信息，请参阅构建文件。

#### 过程:

要使用构建文件 `bldsqlj` (UNIX) 或 `bldsqlj.bat` (Windows) 构建 `TbMod`，请输入以下命令:

```
bldsqlj TbMod <userid> <password> <server_name> <port_number> <db_name>
```

其中，所有参数（除了程序名外）都可以具有缺省值，如构建文件所述的那样。

使用如下命令，对该应用程序运行 Java 解释器:

```
java TbMod
```

也可使用 `Java makefile` 构建此程序。

**注:** 如果正在 Unix 上的 64 位 DB2 实例中运行 Java 应用程序（但 Java Developer Kit 为 32 位），则您必须在运行应用程序之前更改 DB2 库路径。例如，在 AIX 上:

- 如果使用 `bash` 或 `Korn shell`:

```
export LIBPATH=$HOME/sql1lib/lib32
```

- 如果使用 `C shell`:

```
setenv LIBPATH $HOME/sql1lib/lib32
```

#### 相关任务:

- 第 107 页的『构建 JDBC 应用程序』
- 第 111 页的『构建 SQLJ applet』
- 第 117 页的『构建 SQLJ 例程』

#### 相关参考:

- 第 115 页的『UNIX 的 SQLJ 应用程序和 applet 选项』
- 第 117 页的『Windows 的 SQLJ 应用程序和 applet 选项』
- 第 78 页的『SQLJ 样本』

#### 相关样本:

- 『`bldsqlj.bat` -- Builds a Java embedded SQL (SQLJ) application or applet on Windows』
- 『`bldsqlj` -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX』
- 『`TbMod.sqlj` -- How to modify table data (SQLj)』

## SQLJ 应用程序和 applet 的 UNIX 构建脚本

```
|  
| #!/bin/sh  
| # SCRIPT: bldsqlj  
| # Builds Java embedded SQL (SQLJ) applications and applets on UNIX  
| # Usage: bldsqlj prog_name (requires hardcoding user ID and password)  
| #       bldsqlj prog_name userid password  
| #       bldsqlj prog_name userid password server_name  
| #       bldsqlj prog_name userid password server_name port_number  
| #       bldsqlj prog_name userid password server_name port_number db_name  
| #  
| # Defaults:
```

```

#         userid      = $USER variable requires updating if used
#         password    = $PSWD variable requires updating if used
#         server_name = $SERVER variable set to local hostname
#         port_number = $PORTNUM variable set to 50000
#         db_name     = $DB variable set to "sample"

# To hardcode user ID (USER) and password (PSWD)
# Replace "NULL" with the correct values in quotes
USER="NULL"
PSWD="NULL"
# You can replace the defaults for each of the following
# with a new value. Note that the PORTNUM number cannot
# be one already used by another process.
SERVER=`hostname`
PORTNUM=50000
DB="sample"

# Translate and compile the SQLJ source file
# and bind the package to the database.
if ( ( [ $# -eq 1 ] && [ $USER != "NULL" ] && [ $PSWD != "NULL" ] ) || \
    ( [ $# -ge 3 ] && [ $# -le 6 ] ) )
then
    # Remove .sqlj extension
    progname=${1%.sqlj}

    sqlj "${progname}.sqlj"

    if [ $# -eq 1 ]
    then
        db2sqljcustomize -url jdbc:db2://$SERVER:$PORTNUM/$DB \
            -user $USER -password $PSWD "${progname}_SJProfile0"
    elif [ $# -eq 3 ]
    then
        db2sqljcustomize -url jdbc:db2://$SERVER:$PORTNUM/$DB -user $2 -password $3 \
            "${progname}_SJProfile0"
    elif [ $# -eq 4 ]
    then
        db2sqljcustomize -url jdbc:db2://$4:$5/$DB -user $2 -password $3 \
            "${progname}_SJProfile0"
    elif [ $# -eq 5 ]
    then
        db2sqljcustomize -url jdbc:db2://$4:$5/$6 -user $2 -password $3 \
            "${progname}_SJProfile0"
    else
        db2sqljcustomize -url jdbc:db2://$4:$5/$6 -user $2 -password $3 \
            "${progname}_SJProfile0"
    fi
else
    echo 'Usage: bldsqlj prog_name (requires hardcoding user ID and password)'
    echo '      bldsqlj prog_name userid password'
    echo '      bldsqlj prog_name userid password server_name'
    echo '      bldsqlj prog_name userid password server_name port_number'
    echo '      bldsqlj prog_name userid password server_name port_number db_name'
    echo ''
    echo '      Defaults:'
    echo '          userid      = '$USER
    echo '          password    = '$PSWD
    echo '          server_name = '$SERVER
    echo '          port_number = '$PORTNUM
    echo '          db_name     = '$DB
fi

```

## UNIX 的 SQLJ 应用程序和 applet 选项

下表包含在 UNIX 上的 bldsqlj 构建脚本中使用的 SQLJ 转换程序和定制程序选项。这些是 DB2 建议您用来在 UNIX 平台上构建 SQLJ 应用程序和 applet 的选项。

bldsqlj 的转换程序和定制程序选项	
<b>sqlj</b>	SQLJ 转换程序（还用来编译程序）。
<b>"\${progname}.sqlj"</b>	SQLJ 源文件。如果输入文件名包含扩展名，则 progname=\${1%.sqlj} 除去扩展名，因此当再次添加扩展名时，它不会重复。
<b>db2sqljcustomize</b>	DB2 Java 版概要文件定制程序。
<b>-url</b>	指定用于建立数据库连接的 JDBC URL，例如， jdbc:db2://servername:50000/sample。
<b>-user</b>	指定用户标识。
<b>-password</b>	指定密码。
<b>"\${progname}_SJProfile0"</b>	指定该程序的序列化概要文件。

### 相关任务:

- 第 111 页的『构建 SQLJ applet』
- 第 112 页的『构建 SQLJ 应用程序』

### 相关参考:

- 第 120 页的『UNIX 的 SQLJ 例程选项』

### 相关样本:

- 『bldsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX』

## SQLJ 应用程序和 applet 的 Windows 批处理文件

```
@echo off
rem BATCH FILE: bldsqlj.bat
rem Builds a Java embedded SQL (SQLJ) application or applet on Windows

rem To add defaults for user ID (USER) and password (PSWD)
rem Uncomment the following and add the appropriate values
rem set USR=
rem set PSWD=

rem You can replace the defaults for each of the following
rem with a new value. Note that the PORTNUM number cannot be
rem one already used by another process.
set SERVER=%COMPUTERNAME%
set PORTNUM=50000
set DB=sample

goto start
:usage
echo Usage: bldsqlj prog_name (requires hardcoding user ID and password)
echo          bldsqlj prog_name userid password
echo          bldsqlj prog_name userid password server_name
```

```

echo          blsqlj prog_name userid password server_name port_number
echo          blsqlj prog_name userid password server_name port_number db_name
echo.
echo          Defaults:
echo          userid          = %USR%
echo          password        = %PSWD%
echo          server_name     = %SERVER%
echo          port_number     = %PORTNUM%
echo          db_name         = %DB%
goto exit

:start
rem Translate and compile the SQLJ source file
rem and bind the package to the database.
if "%1" == "" goto usage
if "%2" == "" goto case1
if "%3" == "" goto usage
if "%4" == "" goto case3
if "%5" == "" goto case4
if "%6" == "" goto case5
if "%7" == "" goto case6
goto usage

:case1
if "%USR%" == "" goto usage
if "%PSWD%" == "" goto usage
if "%SERVER%" == "" goto nohostname
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%SERVER%:%PORTNUM%/%DB% -user %USR%
                -password %PSWD% %1_SJProfile0
goto continue

:case3
if "%SERVER%" == "" goto nohostname
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%SERVER%:%PORTNUM%/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case4
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%PORTNUM%/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case5
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%5/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case6
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%5/%6 -user %2
                -password %3 %1_SJProfile0
goto continue

:continue
goto exit

:nohostname
echo Server name (hostname) could not be determined.
echo.
goto usage

:exit
@echo on

```



## Windows 的 SQLJ 应用程序和 applet 选项

下表包含在 Windows 操作系统上的 `bldsqlj.bat` 批处理文件中使用的 SQLJ 转换程序和定制程序选项。这些是 DB2 建议您用来在 Windows 上构建 SQLJ 应用程序和 applet 的选项。

bldsqlj.bat 的转换程序和定制程序选项	
<b>sqlj</b>	SQLJ 转换程序（还用来编译程序）。
<b>%1.sqlj</b>	SQLJ 源文件。
<b>db2sqljcustomize</b>	DB2 Java 版概要文件定制程序。
<b>-url</b>	指定用于建立数据库连接的 JDBC URL，例如， jdbc:db2://servername:50000/sample。
<b>-user</b>	指定用户标识。
<b>-password</b>	指定密码。
<b>%1_SJProfile0</b>	指定该程序的序列化概要文件。

### 相关任务:

- 第 111 页的『构建 SQLJ applet』
- 第 112 页的『构建 SQLJ 应用程序』

### 相关参考:

- 第 122 页的『Windows 的 SQLJ 例程选项』

### 相关样本:

- 『bldsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows』

## 构建 SQLJ 例程

DB2 提供了演示 SQLJ 例程（存储过程和用户定义的函数）的样本程序，在 UNIX 上，这些样本程序在 `samples/java/sqlj` 目录中，而在 Windows 上，则在 `samples\java\sqlj` 目录中。例程是在服务器上编译和存储的。当客户机应用程序调用存储过程时，它们存取服务器数据库并将信息返回至客户机应用程序。

在同一目录中，DB2 还提供了构建文件 `bldsqljs`（UNIX）或 `bldsqljs.bat`（Windows），它包含构建例程的命令。

该构建文件最多使用 6 个参数：在 UNIX 上为 \$1、\$2、\$3、\$4、\$5 和 \$6，而在 Windows 上为 %1、%2、%3、%4、%5 和 %6。第一个参数指定您的程序的名称。第二个参数指定数据库实例的用户标识，第三个参数指定密码。第四个参数指定服务器名。第五个参数指定端口号。而第六个参数指定数据库名称。对于除第一个参数（程序名）之外的所有参数，都可以使用缺省值。有关使用缺省参数值的详细信息，请参阅构建文件。

## 过程:

以下示例显示如何使用存储过程构建类文件。

SpServer 演示 PARAMETER STYLE JAVA 存储过程，它使用 JDBC 应用程序驱动程序来存取 DB2 数据库。

要使用构建文件 bldsqljs (UNIX) 或 bldsqljs.bat (Windows) 来构建此存储过程类:

1. 输入以下命令:

```
bldsqljs SpServer <userid> <password> <server_name> \  
<port_number> <db_name>
```

其中，所有参数（除了程序名外）都可以具有缺省值，如构建文件所述的那样。

2. 接着，通过在服务器上运行 spcat 脚本对例程进行编目。输入:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 SpDrop.db2 对这些例程进行了编目，则取消对它们的编目，然后通过调用 SpCreate.db2 对它们进行编目，最后与数据库断开连接。还可以分别运行 SpDrop.db2 和 SpCreate.db2 脚本。

3. 接着，停止数据库然后重新启动它以允许识别新类文件。如有必要，将类文件的文件方式设置为“读”，以便可由受防护用户读取它。
4. 构建并运行 SpClient 客户机应用程序以调用存储过程。可以使用应用程序构建文件 bldsqlj (UNIX) 或 bldsqlj.bat (Windows) 来构建 SpClient。

还可以使用 Java makefile 来构建上述程序。

## 相关任务:

- 第 108 页的『构建 JDBC 例程』
- 第 111 页的『构建 SQLJ applet』
- 第 112 页的『构建 SQLJ 应用程序』

## 相关参考:

- 第 120 页的『UNIX 的 SQLJ 例程选项』
- 第 122 页的『Windows 的 SQLJ 例程选项』
- 第 78 页的『SQLJ 样本』

## 相关样本:

- 『bldsqljs.bat -- Builds a Java embedded SQL (SQLJ) stored procedure on Windows』
- 『bldsqljs -- Builds Java embedded SQL (SQLJ) stored procedures on UNIX』
- 『spcat -- To catalog SQLj stored procedures on UNIX』
- 『SpClient.sqlj -- Call a variety of types of stored procedures from SpServer.sqlj (SQLj)』
- 『SpCreate.db2 -- How to catalog the stored procedures contained in SpServer.sqlj』
- 『SpDrop.db2 -- How to uncatalog the stored procedures contained in SpServer.sqlj』
- 『SpIterat.sqlj -- Iterator class file for SpServer.sqlj (SQLj)』
- 『SpServer.sqlj -- Provide a variety of types of stored procedures to be called from (SQLj)』

## SQLJ 例程的 UNIX 构建脚本

```
#!/bin/sh
# SCRIPT: bldsqljs
# Builds Java embedded SQL (SQLJ) stored procedures on UNIX
# Usage: bldsqljs prog_name (requires hardcoding user ID and password)
#        bldsqljs prog_name userid password
#        bldsqljs prog_name userid password server_name
#        bldsqljs prog_name userid password server_name port_number
#        bldsqljs prog_name userid password server_name port_number db_name
#
# Defaults:
#   userid       = $USER variable requires updating if used
#   password     = $PSWD variable requires updating if used
#   server_name  = $SERVER variable set to local hostname
#   port_number  = $PORTNUM variable set to 50000
#   db_name      = $DB variable set to "sample"

# To hardcode user ID (USER) and password (PSWD)
# Replace "NULL" with the correct values in quotes
USER="NULL"
PSWD="NULL"
# You can replace the defaults for each of the following
# with a new value. Note that the PORTNUM number cannot
# be one already used by another process.
SERVER=`hostname`
PORTNUM=50000
DB="sample"

# Translate and compile the SQLJ source file
# and bind the package to the database.
if (( [ $# -eq 1 ] && [ $USER != "NULL" ] && [ $PSWD != "NULL" ] ) \
    || ( [ $# -ge 3 ] && [ $# -le 6 ] ))
then
    # Remove .sqlj extension
    progname=${1%.sqlj}

    sqlj "${progname}.sqlj"

    if [ $# -eq 1 ]
    then
        db2sqljcustomize -url jdbc:db2://$SERVER:$PORTNUM/$DB \
            -user $USER -password $PSWD "${progname}_SJProfile0"
    elif [ $# -eq 3 ]
    then
        db2sqljcustomize -url jdbc:db2://$SERVER:$PORTNUM/$DB -user $2 \
            -password $3 "${progname}_SJProfile0"
    elif [ $# -eq 4 ]
    then
        db2sqljcustomize -url jdbc:db2://$4:$5/$DB -user $2 -password $3 \
            "${progname}_SJProfile0"
    elif [ $# -eq 5 ]
    then
        db2sqljcustomize -url jdbc:db2://$4:$5/$DB -user $2 -password $3 \
            "${progname}_SJProfile0"
    else
        db2sqljcustomize -url jdbc:db2://$4:$5/$6 -user $2 -password $3 \
            "${progname}_SJProfile0"
    fi

    # Copy the *.class and *.ser files to the 'function' directory.
    rm -f "$DB2PATH/function/${progname}*.class"
    rm -f "$DB2PATH/function/${progname}*.ser"
    cp "${progname}*.class" "$DB2PATH/function"
    cp "${progname}*.ser" "$DB2PATH/function"

else
```

```

|         echo 'Usage: bldsqljs prog_name (requires hardcoding user ID and password)'
|         echo '         bldsqljs prog_name userid password'
|         echo '         bldsqljs prog_name userid password server_name'
|         echo '         bldsqljs prog_name userid password server_name port_number'
|         echo '         bldsqljs prog_name userid password server_name port_number db_name'
|         echo ''
|         echo '         Defaults:'
|         echo '             userid      = '$USER
|         echo '             password     = '$PSWD
|         echo '             server_name  = '$SERVER
|         echo '             port_number  = '$PORTNUM
|         echo '             db_name     = '$DB
|     fi

```

## UNIX 的 SQLJ 例程选项

下表包含在 UNIX 上的 bldsqljs 构建脚本中使用的 SQLJ 转换程序和定制程序选项。这些是 DB2 建议您用来在 UNIX 平台上构建 SQLJ 例程（存储过程和用户定义的函数）的选项。

bldsqljs 的转换程序和定制程序选项	
<b>sqlj</b>	SQLJ 转换程序（还用来编译程序）。
<b>"\${progname}.sqlj"</b>	SQLJ 源文件。如果输入文件名包含扩展名，则 progname=\${1%.sqlj} 除去扩展名，因此当再次添加扩展名时，它不会重复。
<b>db2sqljcustomize</b>	DB2 Java 版概要文件定制程序。
<b>-url</b>	指定用于建立数据库连接的 JDBC URL，例如， jdbc:db2://servername:50000/sample。
<b>-user</b>	指定用户标识。
<b>-password</b>	指定密码。
<b>"\${progname}_SJProfile0"</b>	指定该程序的序列化概要文件。

### 相关任务:

- 第 117 页的『构建 SQLJ 例程』

### 相关参考:

- 第 115 页的『UNIX 的 SQLJ 应用程序和 applet 选项』

### 相关样本:

- 『bldsqljs -- Builds Java embedded SQL (SQLJ) stored procedures on UNIX』

## SQLJ 例程的 Windows 批处理文件

```

| @echo off
| rem BATCH FILE: bldsqljs.bat
| rem Builds a Java embedded SQL (SQLJ) stored procedure on Windows
|
| rem To add defaults for user ID (USR) and password (PSWD)
| rem Uncomment the following and add the appropriate values
| rem set USR=

```

```

rem set PSWD=
rem You can replace the defaults for each of the following
rem with a new value. Note that the PORTNUM number cannot be
rem one already used by another process.
set SERVER=%COMPUTERNAME%
set PORTNUM=50000
set DB=sample

goto start
:usage
echo Usage: bldsqljs prog_name (requires hardcoding user ID and password)
echo          bldsqljs prog_name userid password
echo          bldsqljs prog_name userid password server_name
echo          bldsqljs prog_name userid password server_name port_number
echo          bldsqljs prog_name userid password server_name port_number db_name
echo.
echo          Defaults:
echo          userid      = %USR%
echo          password    = %PSWD%
echo          server_name = %SERVER%
echo          port_number = %PORTNUM%
echo          db_name     = %DB%
goto exit

:start
rem Translate and compile the SQLJ source file
rem and bind the package to the database.
if "%DB2PATH%" == "" goto nodb2cmd
if "%1" == "" goto usage
if "%2" == "" goto case1
if "%3" == "" goto usage
if "%4" == "" goto case3
if "%5" == "" goto case4
if "%6" == "" goto case5
if "%7" == "" goto case6
goto usage

:case1
if "%USR%" == "" goto usage
if "%PSWD%" == "" goto usage
if "%SERVER%" == "" goto nohostname
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%SERVER%:%PORTNUM%/%DB% -user %USR%
                -password %PSWD% %1_SJProfile0
goto continue

:case3
if "%SERVER%" == "" goto nohostname
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%SERVER%:%PORTNUM%/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case4
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%PORTNUM%/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case5
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%5/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case6
sqlj %1.sqlj

```

```

db2sqljcustomize -url jdbc:db2://%4:%5/%6 -user %2
                -password %3 %1_SJProfile0
goto continue

:continue
rem Copy the *.class and *.ser files to the 'function' directory.
copy %1*.class %DB2PATH%\function\
copy %1*.ser %DB2PATH%\function\
goto exit

:nodb2cmd
echo DB2 command line environment not initialized. Please run db2cmd and try again.
goto exit

:nohostname
echo Server name (hostname) could not be determined.
echo.
goto usage

:exit
@echo on

```

## Windows 的 SQLJ 例程选项

下表包含在 Windows 操作系统上的 bldsqljs.bat 批处理文件中使用的 SQLJ 转换程序和定制程序选项。这些是 DB2 建议您用来构建 SQLJ 例程（存储过程和用户定义的函数）的选项。

bldsqljs.bat 的转换程序和定制程序选项			
<b>sqlj</b>	SQLJ 转换程序（还用来编译程序）。		
<b>%1.sqlj</b>	SQLJ 源文件。		
<b>db2sqljcustomize</b>	DB2 Java 版概要文件定制程序。		
<b>-url</b>	指定用于建立数据库连接的	JDBC	URL，例如，
	jdbc:db2://servername:50000/sample.		
<b>-user</b>	指定用户标识。		
<b>-password</b>	指定密码。		
<b>%1_SJProfile0</b>	指定该程序的序列化概要文件。		

### 相关任务:

- 第 117 页的『构建 SQLJ 例程』

### 相关参考:

- 第 117 页的『Windows 的 SQLJ 应用程序和 applet 选项』

### 相关样本:

- 『bldsqljs.bat -- Builds a Java embedded SQL (SQLJ) stored procedure on Windows』

---

## 第 5 章 命令行处理器

运行命令行处理器 (CLP) 脚本 . . . . . 123      从命令行处理器 (CLP) 调用过程 . . . . . 124

本章提供了通过运行 CLP 脚本和通过使用 CALL 语句在命令行上调用 DB2 存储过程以使用命令行处理器来存取 DB2 数据库的详细信息。

SQL 过程也编入 CLP 脚本中。有关创建 SQL 过程的信息在第 127 页的第 6 章, 『SQL 过程』中。

有关最新的 DB2 应用程序开发情况的更新, 请访问以下 Web 页面:

<http://www.ibm.com/software/data/db2/udb/ad>

---

### 运行命令行处理器 (CLP) 脚本

CLP 可从 DB2 实例的命令行直接访问。它可以通过在命令行上输入 “db2” 来以交互方式运行。它也可以通过在关键字 “db2” 后面输入 DB2 命令或包含 DB2 命令的 DB2 脚本来以非交互方式运行。此处使用的示例将使用非交互方式。

CLP 脚本直接对 SQL 语句进行编码不是将它们嵌入主语言中, 因此用户只需知道 SQL 就能编写 CLP 脚本。对于 CLP 脚本, 在运行脚本之前无需编译和链接源文件。

DB2 提供了对 sample 数据库创建和运行 SQL 语句的 CLP 脚本。在 UNIX 上, 这些脚本位于 `sqllib/samples/clp` 目录中, 而在 Windows 上, 这些脚本位于 `sqllib\samples\clp` 目录中。目录中的 README 文件描述了这些程序以及如何运行它们。

过程:

CLP 脚本 `cte.db2` 定义了两个通过 SELECT 语句访问的公共表表达式: `PAYLEVEL` 和 `PAYBYED`。

要运行此脚本:

1. 首先连接至样本数据库:

```
db2 connect to sample
```

2. 接着输入以下命令:

```
db2 -vf cte.db2 -t
```

“v” 不是必需的冗余标志, 但是使用它可提供详细的输出信息, 因此建议使用它。

缺省情况下结果显示在屏幕上。它给出 SQL 公共表表达式以及它的输出:

```
WITH PAYLEVEL AS (SELECT EMPNO, YEAR(HIREDATE) AS HIREYEAR, EDLEVEL,  
SALARY+BONUS+COMM AS TOTAL_PAY FROM EMPLOYEE WHERE EDLEVEL > 16 ),  
PAYBYED (EDUC_LEVEL, YEAR_OF_HIRE, AVG_TOTAL_PAY) AS (SELECT EDLEVEL,  
HIREYEAR, AVG(TOTAL_PAY) FROM PAYLEVEL GROUP BY EDLEVEL, HIREYEAR )  
SELECT EMPNO, EDLEVEL, YEAR_OF_HIRE, TOTAL_PAY, AVG_TOTAL_PAY FROM  
PAYLEVEL, PAYBYED WHERE EDLEVEL=EDUC_LEVEL AND HIREYEAR = YEAR_OF_HIRE  
AND TOTAL_PAY < AVG_TOTAL_PAY
```

```

EMPNO  EDLEVEL YEAR_OF_HIRE TOTAL_PAY    AVG_TOTAL_PAY
-----
000210      17      1979      20132.00      25896.500000000000000000000000

```

1 record(s) selected.

#### 相关概念:

- 『Command Line Processor (CLP)』 (*Command Reference*)

#### 相关任务:

- 第 124 页的 『从命令行处理器 (CLP) 调用过程』

#### 相关参考:

- 第 70 页的 『命令行处理器 (CLP) 样本』
- 『db2 - Command Line Processor Invocation Command』 (*Command Reference*)

#### 相关样本:

- 『cte.db2 -- How to create a COMMON TABLE EXPRESSION 』

---

## 从命令行处理器 (CLP) 调用过程

可以通过从 DB2 命令行处理器界面使用 CALL 语句调用存储过程。必须在 DB2 系统目录表中定义被调用的存储过程。

#### 过程:

要调用存储过程，首先连接至数据库:

```
db2 connect to sample user userid using password
```

其中 *userid* 和 *password* 是 sample 数据库所在的实例的用户标识和密码。

要使用 CALL 语句，输入存储过程名称加上任何 IN 或 INOUT 参数值，并输入 “?” 作为每个 OUT 参数值的占位符。

存储过程的参数是在程序源文件中的存储过程的 CREATE PROCEDURE 语句中给定的。

#### SQL 过程示例

有关创建 SQL 过程的信息，请参阅创建 SQL 过程。

在 whiles.db2 文件中，DEPT\_MEDIAN 过程特征符的 CREATE PROCEDURE 语句如下所示:

```
CREATE PROCEDURE DEPT_MEDIAN
(IN deptNumber SMALLINT, OUT medianSalary DOUBLE)
```

要调用此过程，可使用 CALL 语句，必须在该语句中指定过程名称和适当的参数自变量，在此示例中，参数自变量是 IN 参数的值和作为 OUT 参数的值的问号 “?”。此过程的 SELECT 语句使用 STAFF 表的 DEPT 列上的 deptNumber 值，因此要获取有意义的输出，IN 参数必须是 DEPT 列中的有效值；例如，值 “51”：

```
db2 call dept_median (51, ?)
```

**注：**在 UNIX 平台上，圆括号对于命令 shell 具有特殊意义，因此必须将 “\” 字符置于其前面或将其用引号引起来，如下所示:



```
db2 "call dept_median (51, ?)"
```

如果您正在使用命令行处理器的交互方式，则不要使用引号。  
在运行以上命令之后，应会出现以下结果：

输出参数的值

```
-----  
参数名: MEDIANSALARY  
参数值: +1.76545000000000E+004  
  
返回状态 = 0
```

### C 存储过程示例

还可以通过使用命令行处理器调用由受支持的主语言创建的存储过程。在 UNIX 上的 `samples/c` 目录中和 Windows 上的 `samples\c` 目录中，DB2 提供了用于创建存储过程的文件。spserver 共享库包含许多可从源文件 `spserver.sqc` 创建的存储过程。`spcreate.db2` 文件对存储过程进行编目。

在 `spcreate.db2` 文件中，MAIN\_EXAMPLE 过程的 CREATE PROCEDURE 语句开头如下：

```
CREATE PROCEDURE MAIN_EXAMPLE (IN job CHAR(8),  
                                OUT salary DOUBLE,  
                                OUT errorcode INTEGER)
```

要调用此存储过程，需要对 IN 参数 job 指定一个 CHAR 值并对每个 OUT 参数指定一个问号“?”。此过程的 SELECT 语句使用 EMPLOYEE 表的 JOB 列上的 job 值，因此要获取有意义的输出，IN 参数必须是 JOB 列中的有效值。调用该存储过程的 C 样本程序 `spclient` 使用 'DESIGNER' 作为 JOB 值。不用进行任何修改我们就可以使用它，如下所示：

```
db2 "call MAIN_EXAMPLE ('DESIGNER', ?, ?)"
```

在运行以上命令之后，应会出现以下结果：

输出参数的值

```
-----  
参数名      : SALARY  
参数值      : +2.37312500000000E+004  
  
参数名      : ERRORCODE  
参数值      : 0  
  
返回状态 = 0
```

ERRORCODE 为零表示运行结果是成功的。

注意，与 `spclient` 程序相比，`spclient` 用十进制对结果进行了格式化，使其更容易查看：

```
CALL stored procedure named MAIN_EXAMPLE  
Stored procedure returned successfully  
Average salary for job DESIGNER = 23731.25
```

#### 相关任务:

- 第 127 页的『创建 SQL 过程』
- 第 128 页的『使用客户机应用程序调用 SQL 过程』
- 『Calling procedures from triggers or SQL routines』 (*Application Development Guide: Programming Server Applications*)

- 『Calling procedures from applications or external routines』 (*Application Development Guide: Programming Server Applications*)

相关样本:

- 『spclient.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc (C)』
- 『spserver.sqc -- Definition of various types of stored procedures (C)』
- 『whiles.db2 -- To create the DEPT\_MEDIAN SQL procedure 』
- 『whiles.sqc -- To call the DEPT\_MEDIAN SQL procedure 』

---

## 第 6 章 SQL 过程

创建 SQL 过程 . . . . .	127	备份和复原在 DB2 8.2 之前创建的 SQL 过程 . . .	130
使用客户机应用程序调用 SQL 过程 . . . . .	128	重新绑定 SQL 过程 . . . . .	131
为 SQL 过程定制预编译和绑定选项 . . . . .	129		

---

### 创建 SQL 过程

UNIX 上的 `sqllib/samples/sqlproc` 目录和 Windows 上的 `sqllib\samples\sqlproc` 目录中的“DB2 命令行处理器”脚本（它们以 `.db2` 扩展名结尾），执行 `CREATE PROCEDURE` 语句可以在服务器上创建存储过程。每个 CLP 脚本都具有同名的相应客户机应用程序文件，其扩展名表示其语言和应用程序接口：`.sqc`（表示 C 嵌入式 SQL）、`.c`（表示 DB2 CLI）或 `.java`（表示 JDBC）。

#### 注:

1. 从 DB2 版本 8.2 开始，创建 SQL 过程就不要求在服务器上有 C 或 C++ 编译器了，因此不需要安装 C 或 C++ 编译器。当创建了 SQL 过程之后，它的过程语句会被转换为存储在数据库目录中的本机表示法，就如对其它 SQL 语句执行的转换一样。当 SQL 过程被调用时，就会从目录中装入此表示法，而 DB2 引擎就会执行该过程。
2. `CALL` 在 DB2 版本 8 中是 SQL 语句。它表示您不能再创建任何顺序的过程。编译器检查编译时调用的过程是否存在，并在未找到该过程时显示 `SQLCODE -440` 的错误。

#### 过程:

运行 `CREATE PROCEDURE` CLP 脚本之前，使用以下命令连接样本数据库:

```
db2 connect to sample user userid using password
```

其中 `userid` 和 `password` 是 `sample` 数据库所在的实例的用户标识和密码。

要执行 `rsultset.db2` 脚本文件中包含的 `CREATE PROCEDURE` 语句，请输入以下命令:

```
db2 -td@ -vf rsultset.db2
```

现在，SQL 过程就可供调用了。

#### 相关任务:

- 第 129 页的『为 SQL 过程定制预编译和绑定选项』
- 第 131 页的『重新绑定 SQL 过程』

#### 相关样本:

- 『`rsultset.db2` -- To register and create the `MEDIAN_RESULT_SET` SQL procedure』

## 使用客户机应用程序调用 SQL 过程

如在第 127 页的『创建 SQL 过程』中所述创建了 SQL 过程之后，可以通过构建并运行客户机应用程序来调用该 SQL 过程。DB2 在 `sql1lib/samples/sqlproc` (UNIX) 和 `sql1lib\samples\sqlproc` (Windows) 中提供了样本客户机程序。存在 DB2 CLI、C 嵌入式 SQL 以及 JDBC 的客户机源文件。DB2 在 UNIX 目录中提供了命令行处理器脚本，在 Windows 目录中提供了批处理文件，来调用 SQL 过程。

### 过程:

根据正在使用的应用程序接口，可以通过遵循下列示例来构建和运行调用 SQL 过程的样本客户机程序:

#### DB2 CLI

要根据源文件 `rsultset.c` 构建 DB2 CLI 客户机应用程序 `rsultset`，请输入:

```
bldcli rsultset
```

此命令在 UNIX 上创建可执行文件 `rsultset`，在 Windows 上创建可执行文件 `rsultset.exe`。

要调用 SQL 过程，通过输入可执行文件名、要连接至的数据库名称以及数据库实例的用户标识和密码运行样本客户机应用程序:

```
rsultset database userid password
```

#### C 嵌入式 SQL

要从源文件 `basecase.sqc` 构建嵌入式 SQL 客户机应用程序 `basecase`，请输入脚本文件名、可执行文件名、正在连接的数据库以及数据库实例的用户标识和密码:

```
bldapp basecase database userid password
```

结果是 UNIX 上的可执行文件 `basecase` 和 Windows 上的可执行文件 `basecase.exe`。

要调用 SQL 过程，通过输入以下内容运行样本客户机应用程序:

```
basecase database userid password
```

**JDBC** 要从源文件 `NestedSP.java` 构建 JDBC 客户机应用程序 `NestedSP`，编译源文件:

```
javac NestedSP.java
```

产生类文件 `NestedSP.class`。

要调用 SQL 过程，对应用程序运行 Java 解释器:

```
java NestedSP database userid password
```

### 相关任务:

- 第 127 页的『创建 SQL 过程』
- 第 124 页的『从命令行处理器 (CLP) 调用过程』
- 第 131 页的『重新绑定 SQL 过程』

### 相关样本:

- 『basecase.sqc -- To call the UPDATE\_SALARY SQL procedure』
- 『NestedSP.java -- Client Application for Nested Stored Procedure』
- 『resultset.c -- To call the MEDIAN\_RESULT\_SET SQL procedure』

---

## 为 SQL 过程定制预编译和绑定选项

过程:

可通过使用以下命令设置实例范围的 DB2 注册表变量 DB2\_SQLROUTINE\_PREOPTS 来为 SQL 过程来定制预编译和绑定选项:

```
db2set DB2_SQLROUTINE_PREOPTS=<options>
```

仅允许指定下列选项:

```
BLOCKING {UNAMBIG | ALL | NO}
DATETIME {DEF | USA | EUR | ISO | JIS | LOC}
DEGREE {1 | degree-of-parallelism | ANY}
DYNAMICRULES {BIND | RUN}
EXPLAIN {NO | YES | ALL}
EXPLSNAP {NO | YES | ALL}
FEDERATED {NO | YES}
INSERT {DEF | BUF}
ISOLATION {CS |RR |UR |RS |NC}
QUERYOPT optimization-level
VALIDATE {RUN | BIND}
```

对于 SET\_ROUTINE\_OPTS 存储过程, 可在过程级别更改这些选项。在当前会话中为创建 SQL 过程而设置的选项值可使用 GET\_ROUTINE\_OPTS 函数获取。

示例。

在此示例中使用的 SQL 过程将在 CLP 脚本 (已在下面提供) 中定义。这些脚本不在 sqlproc 样本目录中, 但是您可以通过将 CREATE 过程语句剪切并粘贴到您自己的文件中轻松地创建这些文件。

样本使用一个名为 “expenses” 的表, 您可以在样本数据库中按如下所示创建该表:

```
db2 connect to sample
db2 CREATE TABLE expenses(amount DOUBLE, date DATE)
db2 connect reset
```

首先, 我们指定使用 ISO 格式的日期作为实例范围的设置:

```
db2set DB2_SQLROUTINE_PREOPTS="DATETIME ISO"
db2stop
db2start
```

停止 DB2 然后重新启动它以使更改生效。

然后, 连接至数据库:

```
db2 connect to sample
```

第一个过程是在 CLP 脚本 maxamount.db2 中按如下所示定义的:

```
CREATE PROCEDURE maxamount(OUT maxamnt DOUBLE)
BEGIN
  SELECT max(amount) INTO maxamnt FROM expenses;
END @
```

将使用选项 DATETIME ISO 和 ISOLATION UR 创建它:

```
db2 "CALL SET_ROUTINE_OPTS(GET_ROUTINE_OPTS() || ' ISOLATION UR')"  
db2 -td@ -vf maxamount.db2
```

下一个过程是在 CLP 脚本 fullamount.db2 中按如下所示定义的:

```
CREATE PROCEDURE fullamount(OUT fullamnt DOUBLE)  
BEGIN  
  SELECT sum(amount) INTO fullamnt FROM expenses;  
END @
```

将使用选项 ISOLATION CS 创建它 (注意, 在此示例中不使用实例范围的 DATETIME ISO 设置):

```
CALL SET_ROUTINE_OPTS('ISOLATION CS')  
db2 -td@ -vf fullamount.db2
```

示例中的最后一个过程是在 CLP 脚本 perday.db2 中按如下所示定义的:

```
CREATE PROCEDURE perday()  
BEGIN  
  DECLARE cur1 CURSOR WITH RETURN FOR  
  SELECT date, sum(amount)  
  FROM expenses  
  GROUP BY date;  
  
  OPEN cur1;  
END @
```

最后的 SET\_ROUTINE\_OPTS 调用使用 NULL 值作为自变量。这会复原在 DB2\_SQLROUTINE\_PREPOPTS 注册表中指定的全局设置, 因此, 最后一个过程将使用选项 DATETIME ISO 创建:

```
CALL SET_ROUTINE_OPTS(NULL)  
db2 -td@ -vf perday.db2
```

**相关任务:**

- 第 130 页的『备份和复原在 DB2 8.2 之前创建的 SQL 过程』
- 第 127 页的『创建 SQL 过程』
- 第 124 页的『从命令行处理器 (CLP) 调用过程』
- 第 128 页的『使用客户机应用程序调用 SQL 过程』
- 第 131 页的『重新绑定 SQL 过程』

**相关参考:**

- 『PRECOMPILE Command』 (*Command Reference*)

---

## 备份和复原在 DB2 8.2 之前创建的 SQL 过程

在 DB2 版本 8.2 中, SQL 过程完全是在数据库中进行管理的, 就类似于触发器和视图, 在这个意义上, SQL 过程是一流的数据库对象。因此, 在备份和复原期间, 它们不需要特殊对待。在版本 8.2 之前创建的过程却需要特殊对待, 如下面说明的那样, 因为它们涉及在文件系统中创建 DLL (动态链接库)。当在 V8.2 之前的 DB2 版本中创建 SQL 过程时, 生成的共享动态链接库 (DLL) 与源文本、程序包和相关文件一起保存在数据库目录中。因此, 执行数据库备份时将保存所有这些信息。

**过程:**

在数据库恢复期间，将会除去文件系统中属于所恢复数据库的所有 SQL 过程可执行文件。如果索引创建配置参数 `indexrec` 设置为 `RESTART`，那么所有 SQL 过程可执行文件将会从目录表中抽取出来并且在下次连接时放回文件系统。否则，在第一次执行 SQL 过程时，将抽取 SQL 可执行文件。

可执行文件将会放回到以下目录中：

**UNIX** `$HOME/sqllib/function/routine/sqlproc/<database_name>`

#### Windows

`sqllib\function\routine\sqlproc\<database_name>`

其中 `<database_name>` 代表数据库，可以使用该数据库来创建 SQL 过程。

如果在复原操作后第一次尝试连接数据库时返回：

SQL2048N 存取对象“SQL PROCEDURE FILES”时出现错误。  
原因码：“7”。

则使用 `db2stop` 停止 DB2，并使用 `db2start` 重新启动它。

#### 相关任务：

- 第 129 页的『为 SQL 过程定制预编译和绑定选项』
- 第 127 页的『创建 SQL 过程』
- 第 124 页的『从命令行处理器（CLP）调用过程』
- 第 128 页的『使用客户机应用程序调用 SQL 过程』
- 第 131 页的『重新绑定 SQL 过程』

---

## 重新绑定 SQL 过程

#### 过程：

要重新绑定与 SQL 过程相对应的程序包，调用 `SYSPROC.REBIND_ROUTINE_PACKAGE` 内置存储过程。

例如，如果数据库中存在一个名为 `MYSHEMA.MYPROC` 的 SQL 过程，则可以从“命令行处理器”（CLP）通过发出以下命令重新绑定其程序包：

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE('P', 'MYSHEMA.MYPROC', 'CONSERVATIVE')
```

其中 `'P'` 指示 `'MYSHEMA.MYPROC'` 是一个过程名称。第一个参数的值 `'SP'` 指示 `'MYSHEMA.MYPROC'` 是一个特定过程名称。`'CONSERVATIVE'` 指示应该应用保守重新绑定语义。请参阅下面相关链接中的 `REBIND` 命令以获取有关保守重新绑定的更多详细信息。

#### 相关任务：

- 第 129 页的『为 SQL 过程定制预编译和绑定选项』
- 第 130 页的『备份和复原在 DB2 8.2 之前创建的 SQL 过程』
- 第 127 页的『创建 SQL 过程』
- 第 124 页的『从命令行处理器（CLP）调用过程』
- 第 128 页的『使用客户机应用程序调用 SQL 过程』

相关参考:

- 『REBIND Command』 (*Command Reference*)



---

## 第 7 章 Perl

构建 Perl 应用程序 . . . . . 133

本章提供有关构建 Perl 程序以存取 DB2 数据库的详细信息。

有关最新的 DB2 应用程序开发情况的更新，请访问以下 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad>

---

### 构建 Perl 应用程序

DB2 支持用 Perl 编写的客户机应用程序进行数据库存取。在交付印刷时，“Perl 数据库接口”（Perl DBI）版本 0.93 或更新版本的 DB2 UDB 驱动程序（DBD::DB2）发行版 0.76 可用于 AIX、HP-UX、Linux、Solaris 和 Windows。有关如何获取最新驱动程序的信息，请访问：

<http://www.ibm.com/software/data/db2/perl>

DB2 提供了一些 Perl 样本程序，这些程序在 UNIX 上位于 `sqllib/samples/perl` 目录中，而在 Windows 上位于 `sqllib\samples\perl` 目录中。

#### 在 UNIX 上设置 64 位环境

在 UNIX 上，如果您未在使用 64 位版本的 Perl，则应遵循在第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』中建议在 64 位环境中运行 32 位应用程序的过程，除非不需要重新绑定应用程序。如果正在使用包装器程序，则应将 `perl $1` 替换为 `$1`。

**注：**在以下示例中，如果是在 UNIX 64 位环境中使用包装器程序而不是 `perl <program_name>`，则替换 `<wrapper_name> <program_name>`。

#### 过程：

要在命令行上对 DB2 Perl 程序运行 perl 解释器，请输入解释器名和程序名（包括扩展名）：

- 如果是在服务器上本地连接：

```
perl dbauth.pl
```

- 如果是从远程客户机连接：

```
perl dbauth.pl sample <userid> <password>
```

某些程序需要支持文件才能运行。tbsel 样本程序需要 `tbselcreate.db2` CLP 脚本创建的一些表。如果这些表已存在，则 `tbselinit` 脚本（UNIX）或 `tbselinit.bat` 批处理文件（Windows）首先将调用 `tbseldrop.db2` 来删除它们，然后调用 `tbselcreate.db2` 来创建它们。因此，要运行程序，应输入下列命令：

- 如果是在服务器上本地连接：

```
tbselinit  
perl tbsel.pl
```

- 如果是从远程客户机连接：

```
|         tbselinit  
|         perl tbsel.pl sample <userid> <password>
```

| **注:** 对于远程客户机, 需要修改 tbselinit 或 tbselinit.bat 文件中的连接语句以对用户标识和密码进行硬编码: db2 connect to sample user <userid> using <password>

### | 调用例程

| DB2 客户机应用程序可以存取用受支持的主语言或 SQL 过程创建的例程 (存储过程和用户定义的函数)。例如, 如果数据库中存在 SQL 过程 spserver 共享库, 则样本程序 spclient.pl 可以存取该库。

| **注:** 要构建主语言例程, 必须在服务器上设置相应的编译器。SQL 过程不需要编译器。共享库只能在服务器上构建, 不能从远程客户机构建。

| 要演示调用 SQL 过程, 转至服务器上的 samples/sqlproc 目录 (UNIX) 或 samples\sqlproc 目录 (Windows), 并运行下列命令以在 spserver 库中创建 SQL 过程并对它们进行编目:

```
|         db2 connect to sample  
|         db2 -td@ -vf spserver.db2
```

| 接着, 回到 perl 样本目录 (此目录可能在远程客户机上), 并对客户机程序运行 Perl 解释器以存取 spserver 共享库:

- | • 如果是在服务器上本地连接, 则输入以下命令:

```
|         perl spclient
```

- | • 如果是从远程客户机连接, 则输入以下命令:

```
|         perl spclient sample <userid> <password>
```

### | 相关概念:

- | • 『Programming Considerations for Perl』 (*Application Development Guide: Programming Client Applications*)
- | • 『Perl DBI』 (*Application Development Guide: Programming Client Applications*)

### | 相关任务:

- | • 第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』

### | 相关参考:

- | • 第 83 页的『Perl 样本』

---

## 第 8 章 PHP

构建 PHP 应用程序 . . . . . 135

本章提供有关构建 PHP 程序以存取 DB2 数据库的详细信息。

有关最新的 DB2 应用程序开发情况的更新，请访问以下 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad>

---

### 构建 PHP 应用程序

DB2 支持用 PHP 编写的客户机应用程序进行数据库存取。PHP 是一种服务器端、内嵌 HTML 和交叉平台的脚本语言。它支持使用“统一 ODBC”存取方法来存取 DB2，在这种方法中，用户级 PHP 使用 ODCB 调用和 DB2 通信。与标准 ODBC 不同，使用“统一 ODBC”方法时，通信直接到达 DB2 CLI 层，并不通过 ODBC 层。要获取更多关于通过 DB2 使用 PHP 的信息，请搜索 DB2 支持站点。

<http://www.ibm.com/software/data/db2/udb/winos2unix/support>

在交付印刷时，最新版本是 PHP 4.3.4。可从以下站点下载 PHP 的最新版本：

<http://www.php.net>

下载 tar 文件之后，使用选项 `--with-ibm-db2=<DIR>` 编译该文件，其中 `<DIR>` 由平台路径确定：

- 在 UNIX 上，选项应是：`--with-ibm-db2=$HOME/sqllib`
- 在 Windows 上，选项应是：`--with-ibm-db2=%DB2PATH%`

在解取 `php-4.3.4.tar` 文件（或 PHP 更新版本的文件，如果可用的话）之后，`php-4.3.4` 目录将包含具有新 PHP 安装的缺省设置的 `php.ini-dist` 文件。必须将该文件复制到安装路径并将文件名更改为 `php.ini`。下列命令采用了典型安装路径：

- 在 UNIX 上：

```
cd ../php-4.x.y
cp php.ini-dist /usr/local/lib/php.ini
```

- 在 Windows 上：

```
cd ..\php-4.x.y
copy php.ini-dist C:\Windows\php.ini
```

注：在安装期间可通过使用以下选项指定其它路径：

```
--with-config-file-path=<path>
```

如果是在 Unix 上使用 RPM 进行安装，则 `php.ini` 的缺省路径是 `/etc`。对于在 Linux 上 RPM，在主目录下面创建一个新文件 `.odbc.ini` 并包括以下内容来为 DB2 配置 ODBC：

```
[ODBC Data Sources]
Sample           = <description>

[Sample]
Driver           = $HOME/sqllib/lib/libdb2.so
Description     = <description>
```

```
| Host          = localhost
| UserName     = <user>
| Password    = <password>
| Database    = sample
```

如果是在 Windows 上使用 InstallShield 进行安装，则添加 IBM DB2 ODBC 驱动程序并通过使用管理工具中的“数据源”（ODBC）为样本数据库配置它。

有关更多信息，请参阅 INSTALL 文件。

可以编辑 php.ini 文件以设置 PHP 选项。DB2 建议设置以下选项以运行 DB2 样本程序：

```
| track_errors = 0n
| register_globals = 0n
| register_argc_argv = 0n
| max_execution_time = 60
| odbc.defaultlrl = 100000
```

注：

1. track\_errors = 0n: 允许跟踪错误。
2. register\_globals = 0n: 如果 register\_globals = Off，则应使用 \$\_SERVER['argc'] 和 \$\_SERVER['argv'][0] 而不是 \$argc 和 \$argv。
3. register\_argc\_argv = 0n: 允许命令行自变量。
4. max\_execution\_time = 60: 有些脚本运行时间可能要长一些，因此需要更改缺省值。
5. odbc.defaultlrl = 100000: 这是以字节计的，大小足以容纳样本中的 BLOB 数据。

DB2 提供了一些 PHP 样本程序，这些程序在 UNIX 上位于 sqllib/samples/php 目录中，而在 Windows 上则位于 sqllib\samples\php 文件夹中。

### 在 UNIX 上设置 64 位环境

在 UNIX 上，如果您未在使用 64 位版本的 PHP，则应遵循在第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』中建议用来在 64 位环境中运行 32 位应用程序所用的过程，除非不需要重新绑定应用程序。如果正在使用包装器程序，则应将 php \$1 替换为 \$1。

注：以下示例中，如果是在 UNIX 64 位环境中使用包装器程序而不是 php <program\_name>，则替换 <wrapper\_name> <program\_name>。

过程：

要在命令行上对 DB2 PHP 源文件运行 php 解释器，请输入解释器名和源文件名（包括扩展名）：

- 如果是在服务器上本地连接：

```
php dbauth.php
```

- 如果是从远程客户机连接：

```
php dbauth.php sample <userid> <password>
```

某些程序需要支持文件才能运行。tbse1 样本程序需要 tbse1create.db2 CLP 脚本创建的一些表。如果这些表已存在，则 Unix 上的 tbse1init 脚本或 Windows 上的 tbse1init.bat 批处理文件首先调用 tbse1drop.db2 来删除它们，然后调用 tbse1create.db2 来创建它们。因此，要运行程序，应输入下列命令：

- 如果是在服务器上本地连接:

```
tbselinit  
php tbsel.php
```

- 如果是从远程客户机连接:

```
tbselinit  
php tbsel.php sample <userid> <password>
```

**注:** 对于远程客户机, 需要修改 `tbselinit` 或 `tbselinit.bat` 文件中的连接语句以对  
用户标识和密码进行硬编码: `db2 connect to sample user <userid> using  
<password>`

### 调用用户定义的函数

DB2 客户机应用程序可以存取用受支持的主语言创建的用户定义的函数。例如, 如果数  
据库中存在 C 用户定义的函数 `udfsrv` 共享库, 则样本程序 `udfcli.php` 可以存取该  
库。

**注:** 要构建主语言用户定义的函数共享库, 必须在服务器上设置适当的编译器。共享  
库只能在服务器上构建, 不能从远程客户机构建。PHP 不支持客户机程序调用存储  
过程。

假定在服务器上设置了 C 编译器, 可通过转至服务器上的 `samples/c` 目录 (UNIX)  
或 `samples\c` 目录 (Windows) 来演示调用用户定义的函数, 并运行以下命令来在数据  
库中创建 `udfsrv` 库:

```
bldrtn udfsrv
```

接着, 回到 `php` 样本目录 (此目录可能在远程客户机上), 并对客户机程序运行 `php` 解  
释器以存取 `udfsrv` 共享库:

- 如果是在服务器上本地连接, 则输入以下命令:

```
php udfcli.php
```

- 如果是从远程客户机连接, 则输入以下命令:

```
php udfcli.php sample <userid> <password>
```

### 相关任务:

- 第 50 页的『将应用程序从 32 位环境迁移至 64 位环境』

### 相关参考:

- 第 84 页的『PHP 样本』



---

## 第 3 部分 构建和运行特定于平台的应用程序





---

## 第 9 章 UNIX

构建 UNIX C 应用程序 . . . . .	141	构建 UNIX C++ 多连接应用程序 . . . . .	149
构建 UNIX C 多连接应用程序 . . . . .	142	构建 UNIX C++ 例程 . . . . .	151
构建 UNIX C 例程 . . . . .	144	构建 UNIX Micro Focus COBOL 应用程序 . . . . .	154
构建 UNIX C++ 应用程序 . . . . .	147	构建 UNIX Micro Focus COBOL 例程 . . . . .	155

本章描述为受支持的 UNIX 操作系统构建应用程序和例程的公共步骤。特定于平台的详细信息（如编译器选项）在本章之后的各平台章节中提供。

有关构建 C/C++ 多线程应用程序的信息包含在各平台章节中，因为每个操作系统具有对这些程序的特定构建需求。

---

### 构建 UNIX C 应用程序

DB2 提供了用于编译和链接 C 嵌入式 SQL 和 DB2 API 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 `sqllib/samples/c` 目录中。

构建文件 `bldapp` 包含构建 DB2 应用程序的命令。

第一个参数 `$1` 指定源文件名。这是唯一的必需参数，且是不包含嵌入式 SQL 的 DB2 API 程序所需的唯一参数。构建嵌入式 SQL 程序需要连接数据库，因此还需要提供三个可选参数：第二个参数 `$2` 指定您想连接的数据库的名称；第三个参数 `$3` 指定数据库的用户标识；第四个参数 `$4` 指定密码。

对于嵌入式 SQL 程序，`bldapp` 将这些参数传送给预编译和绑定脚本文件 `embprep`。如果未提供数据库名称，则使用缺省 `sample` 数据库。仅当构建程序所在的实例不同于数据库所在的实例时，才需要用户标识和密码参数。

#### 过程:

下列示例显示如何构建和运行 DB2 API 和嵌入式 SQL 应用程序。

要从源文件 `cli_info.c` 构建 DB2 API 非嵌入式 SQL 样本程序 `cli_info`，请输入：

```
bldapp cli_info
```

得到一个可执行文件 `cli_info`。

要运行可执行文件，请输入可执行文件名：

```
cli_info
```

#### 构建和运行嵌入式 SQL 应用程序

从源文件 `tbmod.sqc` 构建嵌入式 SQL 应用程序 `tbmod` 有三种方法：

1. 如果与同一实例中的样本数据库连接，则输入：

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接，还需要输入该数据库名称：

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接，还需要输入该数据库实例的用户标识和密码：

```
bldapp tbmod database userid password
```

得到一个可执行文件 `tbmod`。

有三种方法运行此嵌入式 SQL 应用程序：

1. 如果存取同一实例中的 `sample` 数据库，只须输入可执行文件名：

```
tbmod
```

2. 如果存取同一实例中的另一个数据库，则输入可执行文件名和数据库名称：

```
tbmod database
```

3. 如果存取另一个实例中的数据库，则输入可执行文件名、数据库名称以及该数据库实例的用户标识和密码：

```
tbmod database userid password
```

相关概念：

- 第 93 页的『构建文件』

相关任务：

- 第 144 页的『构建 UNIX C 例程』

相关参考：

- 第 162 页的『AIX C 应用程序编译和链接选项』
- 第 188 页的『HP-UX C 应用程序编译和链接选项』
- 第 204 页的『Linux C 应用程序编译和链接选项』
- 第 218 页的『Solaris C 应用程序编译和链接选项』

相关样本：

- 『`bldapp -- Builds AIX C application programs (C)`』
- 『`bldapp -- Builds HP-UX C applications (C)`』
- 『`bldapp -- Builds Linux C applications (C)`』
- 『`bldapp -- Builds Solaris C applications (C)`』
- 『`cli_info.c -- Set and get information at the client level (C)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)`』
- 『`tbmod.sqc -- How to modify table data (C)`』

---

## 构建 UNIX C 多连接应用程序

DB2 提供了用于编译和链接 C 嵌入式 SQL 和 DB2 API 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 `sqllib/samples/c` 目录中。

构建文件 `bldmc` 包含用于构建 DB2 多连接程序的命令，需要两个数据库。编译和链接选项与在 `bldapp` 中所用的选项相同。

第一个参数 `$1` 指定源文件名。第二个参数 `$2` 指定要连接至的第一个数据库的名称。第三个参数 `$3` 指定要连接至的第二个数据库。这些参数都是必需参数。

| **注:** makefile 对数据库名称的缺省值 “sample” 和 “sample2” (分别对应于 \$2 和 \$3)  
| 进行硬编码, 因此, 如果要使用 makefile 并接受这些缺省值, 则只需指定程序名  
| (\$1 参数)。如果要使用 bldmc 脚本, 则必须指定所有三个参数。

| 可选参数对于本地连接不是必需的, 但对于从远程客户机至服务器的连接则是必需  
| 的。可选参数是: \$4 和 \$5 (分别用于指定第一个数据库的用户标识和密码) 以及 \$6 和  
| \$7 (分别用于指定第二个数据库的用户标识和密码)。

### | 过程:

| 对于多连接样本程序 dbmcon, 需要两个数据库。如果尚未创建 sample 数据库, 则可以  
| 通过在命令行上输入 db2samp1 来创建它。第二个数据库 (此处名为 sample2) 可使用  
| 下列其中一个命令创建:

| 如果是本地创建数据库:

```
| db2 create db sample2
```

| 如果是远程创建数据库:

```
| db2 attach to <node_name>  
| db2 create db sample2  
| db2 detach  
| db2 catalog db sample2 as sample2 at node <node_name>
```

| 其中 <node\_name> 是数据库驻留的节点。

| 多连接还要求 TCP/IP 侦听器正在运行。为了确保它正在运行, 执行下列操作:

- | 1. 按如下所示将环境变量 DB2COMM 设置为 TCP/IP:

```
| db2set DB2COMM=TCPIP
```

- | 2. 按 services 文件中指定的那样, 使用 TCP/IP 服务名称来更新数据库管理器配置文  
| 件:

```
| db2 update dbm cfg using SVCENAME <TCP/IP service name>
```

| 每个实例都在 services 文件中列示了 TCP/IP 服务名称。如果找不到该服务名称或  
| 没有更改 services 文件的文件许可权, 请向系统管理员咨询。

- | 3. 停止数据库管理器然后重新启动它以使这些更改生效:

```
| db2stop  
| db2start
```

| dbmcon 程序由五个文件组成:

#### | **dbmcon.sqc**

| 用来连接至两个数据库的主源文件。

#### | **dbmcon1.sqc**

| 用来创建绑定至第一个数据库的程序包的源文件。

#### | **dbmcon1.h**

| 包括在 dbmcon.sqc 中的 dbmcon1.sqc 的头文件, 用来访问用于创建和删除要  
| 绑定至第一个数据库的表的 SQL 语句。

#### | **dbmcon2.sqc**

| 用来创建绑定至第二个数据库的程序包的源文件。

## dbmcon2.h

包括在 dbmcon.sqc 中的 dbmcon2.sqc 的头文件，用来访问用于创建和删除要绑定至第二个数据库的表的 SQL 语句。

要构建多连接样本程序 dbmcon，请输入：

```
bldmc dbmcon sample sample2
```

得到一个可执行文件 dbmcon。

要运行可执行文件，请输入可执行文件名：

```
dbmcon
```

程序将演示向两个数据库的一阶段落实。

### 相关概念：

- 第 93 页的『构建文件』

### 相关任务：

- 第 141 页的『构建 UNIX C 应用程序』

### 相关参考：

- 『svcname - TCP/IP 服务名称配置参数』（《管理指南：性能》）
- 第 162 页的『AIX C 应用程序编译和链接选项』
- 第 188 页的『HP-UX C 应用程序编译和链接选项』
- 第 204 页的『Linux C 应用程序编译和链接选项』
- 第 218 页的『Solaris C 应用程序编译和链接选项』

### 相关样本：

- 『bldmc -- Builds AIX C multi-connection applications (C)』
- 『bldmc -- Builds HP-UX C multi-connection applications (C)』
- 『bldmc -- Builds Linux C multi-connection applications (C)』
- 『bldmc -- Builds Solaris C multi-connection applications (C)』
- 『dbmcon.sqc -- How to use multiple databases (C)』
- 『dbmcon1.h -- Function declarations for the source file, dbmcon1.sqc (C)』
- 『dbmcon1.sqc -- Functions used in the multiple databases program dbmcon.sqc (C)』
- 『dbmcon2.h -- Function declarations for the source file, dbmcon2.sqc (C)』
- 『dbmcon2.sqc -- Functions used in the multiple databases program dbmcon.sqc (C)』

---

## 构建 UNIX C 例程

DB2 提供了用于编译和链接 C 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 sqllib/samples/c 目录中。

脚本 bldrtn 包含构建例程（存储过程和用户定义的函数）的命令。该脚本将例程编译成可由数据库管理器装入且由客户机应用程序调用的共享库。

第一个参数 \$1 指定源文件名。第二个参数 \$2 指定要连接的数据库名称。

数据库参数是可选的。如果未提供数据库名称，则该程序使用缺省的 `sample` 数据库。并且，由于必须在数据库所在的那个实例中构建存储过程，所以没有任何有关用户标识和密码的参数。

#### 过程:

下列示例显示如何构建具有下列内容的例程共享库:

- 存储过程
- 非嵌入式 SQL 用户定义的函数 (UDF)
- 嵌入式 SQL 用户定义的函数 (UDF)

#### 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至 `sample` 数据库，则输入构建脚本名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还需要输入该数据库名称:

```
bldrtn spserver database
```

脚本将共享库复制到服务器的路径 `sqllib/function` 中。

2. 接着，通过在服务器上运行 `spcat` 脚本对例程进行编目:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 对这些例程进行了编目，则取消对它们的编目，然后通过调用 `spcreate.db2` 对它们进行编目，最后与数据库断开连接。还可以分别调用 `spdrop.db2` 和 `spcreate.db2` 脚本。

3. 然后，如果这不是第一次构建存储过程，则停止数据库然后重新启动它以确保重组新版本的共享库。为此，可在命令行上输入 `db2stop` 后跟 `db2start`。

一旦构建了共享库 `spserver`，就可构建存取该共享库的客户机应用程序 `spclient`。

可以通过使用脚本 `bldapp` 来构建 `spclient`。

要调用共享库中的存储过程，可输入以下命令运行样本客户机应用程序:

```
spclient database userid password
```

其中,

#### **database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

#### **password**

是用户标识的有效密码。

客户机应用程序存取共享库 `spserver`，并在服务器数据库上执行大量存储过程函数。然后将输出返回给该客户机应用程序。

## 非嵌入式 SQL UDF 共享库

要从源文件 `udfsrv.c` 构建用户定义的函数程序 `udfsrv`，请输入构建脚本名和程序名：

```
bldrtn udfsrv
```

该脚本将 UDF 复制到 `sqllib/function` 目录中。

一旦构建了 `udfsrv`，就可构建调用它的客户机应用程序 `udfcli`。提供了此程序的 DB2 CLI 和嵌入式 SQL 版本。可以从 `sqllib/samples/cli` 中的源文件 `udfcli.c` 通过使用脚本 `bldapp` 来构建 DB2 CLI `udfcli` 客户机程序。

可以从 `sqllib/samples/c` 中的源文件 `udfcli.sqc` 通过使用脚本 `bldapp` 来构建嵌入式 SQL `udfcli` 客户机程序。

要调用共享库中的 UDF，通过输入以下命令来运行客户机应用程序：

```
udfcli database userid password
```

其中，

### **database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

### **password**

是用户标识的有效密码。

客户机应用程序将存取共享库 `udfsrv`，并在服务器数据库上执行用户定义的函数。然后将输出返回给该客户机应用程序。

## 嵌入式 SQL UDF 共享库

要从源文件 `udfemsrv.sqc` 构建嵌入式 SQL 用户定义的函数程序 `udfemsrv`，如果连接至 `sample` 数据库，则输入构建脚本名和程序名：

```
bldrtn udfemsrv
```

如果连接至另一个数据库，还需要输入该数据库名称：

```
bldrtn udfemsrv database
```

该脚本将 UDF 复制到 `sqllib/function` 目录中。

一旦构建了 `udfemsrv`，就可构建调用它的客户机应用程序 `udfemcli`。可以从 `sqllib/samples/c` 中的源文件 `udfemcli.sqc` 通过使用脚本 `bldapp` 来构建 `udfemcli` 客户机程序。

要调用共享库中的 UDF，通过输入以下命令来运行客户机应用程序：

```
udfemcli database userid password
```

其中，

### **database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

### **password**

是用户标识的有效密码。

客户机应用程序将存取共享库 `udfemsrv`，并在服务器数据库上执行用户定义的函数。然后将输出返回给该客户机应用程序。

### 相关概念:

- 第 93 页的『构建文件』

### 相关任务:

- 第 141 页的『构建 UNIX C 应用程序』

### 相关参考:

- 第 164 页的『AIX C 例程编译和链接选项』
- 第 190 页的『HP-UX C 例程编译和链接选项』
- 第 206 页的『Linux C 例程编译和链接选项』
- 第 220 页的『Solaris C 例程编译和链接选项』

### 相关样本:

- 『`bldrtn -- Builds AIX C routines (stored procedures and UDFs) (C)`』
- 『`bldrtn -- Builds HP-UX C routines (stored procedures and UDFs) (C)`』
- 『`bldrtn -- Builds Linux C routines (stored procedures or UDFs) (C)`』
- 『`bldrtn -- Builds Solaris C routines (stored procedures or UDFs) (C)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)`』
- 『`spclient.sql -- Call various stored procedures (C)`』
- 『`spserver.sql -- Definition of various types of stored procedures (C)`』
- 『`udfcli.sql -- Call a variety of types of user-defined functions (C)`』
- 『`udfemcli.sql -- Call a variety of types of embedded SQL user-defined functions. (C)`』
- 『`udfemsrv.sql -- Call a variety of types of embedded SQL user-defined functions. (C)`』
- 『`udfsrv.c -- Defines a variety of types of user-defined functions (C)`』

---

## 构建 UNIX C++ 应用程序

DB2 提供了用于编译和链接 C++ 嵌入式 SQL 和 DB2 API 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 `sqllib/samples/cpp` 目录中。

构建文件 `bldapp` 包含构建 DB2 API 和嵌入式 SQL 应用程序的命令。

第一个参数 `$1` 指定源文件名。这是唯一的必需参数，且是不包含嵌入式 SQL 的 DB2 API 程序所需的唯一参数。构建嵌入式 SQL 程序需要连接数据库，因此还需要提供三

个可选参数: 第二个参数 \$2 指定您想连接的数据库的名称; 第三个参数 \$3 指定数据库的用户标识; 第四个参数 \$4 指定密码。

对于嵌入式 SQL 程序, bldapp 将这些参数传送给预编译和绑定脚本文件 embprep。如果未提供数据库名称, 则使用缺省 sample 数据库。仅当构建程序所在的实例不同于数据库所在的实例时, 才需要用户标识和密码参数。

#### 过程:

下列示例显示如何构建和运行 DB2 API 和嵌入式 SQL 应用程序。

要从源文件 cli\_info.C 构建非嵌入式 SQL 样本程序 cli\_info, 请输入:

```
bldapp cli_info
```

得到一个可执行文件 cli\_info。可以输入如下命令, 对 sample 数据库运行该可执行文件:

```
cli_info
```

#### 构建和运行嵌入式 SQL 应用程序

从源文件 tbmod.sqlC 构建嵌入式 SQL 应用程序 tbmod 有三种方法:

1. 如果与同一实例中的样本数据库连接, 则输入:

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接, 还需要输入该数据库名称:

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接, 还需要输入该数据库实例的用户标识和密码:

```
bldapp tbmod database userid password
```

得到一个可执行文件 tbmod。

有三种方法运行此嵌入式 SQL 应用程序:

1. 如果存取同一实例中的 sample 数据库, 只须输入可执行文件名:

```
tbmod
```

2. 如果存取同一实例中的另一个数据库, 则输入可执行文件名和数据库名称:

```
tbmod database
```

3. 如果存取另一个实例中的数据库, 则输入可执行文件名、数据库名称以及该数据库实例的用户标识和密码:

```
tbmod database userid password
```

#### 相关概念:

- 第 93 页的『构建文件』

#### 相关任务:

- 第 151 页的『构建 UNIX C++ 例程』

#### 相关参考:

- 第 166 页的『AIX C++ 应用程序编译和链接选项』
- 第 193 页的『HP-UX C++ 应用程序编译和链接选项』
- 第 209 页的『Linux C++ 应用程序编译和链接选项』



- 第 223 页的『Solaris C++ 应用程序编译和链接选项』

相关样本:

- 『bldapp -- Builds AIX C++ applications (C++)』
- 『bldapp -- Builds HP-UX C++ applications (C++)』
- 『bldapp -- Builds Linux C++ applications (C++)』
- 『bldapp -- Builds Solaris C++ applications (C++)』
- 『cli\_info.C -- Set and get information at the client level (C++)』
- 『tbmod.sqC -- How to modify table data (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## 构建 UNIX C++ 多连接应用程序

DB2 提供了用于编译和链接 C++ 嵌入式 SQL 和 DB2 API 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序都位于 `sqllib/samples/cpp` 目录中。

构建文件 `bldmc` 包含用于构建 DB2 多连接程序的命令，需要两个数据库。编译和链接选项与在 `bldapp` 中的用法相同。

第一个参数 `$1` 指定源文件名。第二个参数 `$2` 指定要连接至的第一个数据库的名称。第三个参数 `$3` 指定要连接至的第二个数据库。这些参数都是必需参数。

**注:** `makefile` 对数据库名称的“`sample`”和“`sample2`”的缺省值（分别为 `$2` 和 `$3`）进行硬编码，因此，如果要使用 `makefile` 并接受这些缺省值，则只需指定程序名（`$1` 参数）。如果要使用 `bldmc` 脚本，则必须指定所有三个参数。

可选参数对于本地连接不是必需的，但对于从远程客户机至服务器的连接则是必需的。可选参数是：`$4` 和 `$5`（分别用于指定第一个数据库的用户标识和密码）以及 `$6` 和 `$7`（分别用于指定第二个数据库的用户标识和密码）。

**过程:**

对于多连接样本程序 `dbmcon`，需要两个数据库。如果尚未创建 `sample` 数据库，则可以通过在命令行上输入 `db2samp1` 来创建它。第二个数据库（此处名为 `sample2`）可使用下列其中一个命令创建:

如果是本地创建数据库:

```
db2 create db sample2
```

如果是远程创建数据库:

```
db2 attach to <node_name>
db2 create db sample2
db2 detach
db2 catalog db sample2 as sample2 at node <node_name>
```

其中 `<node_name>` 是数据库驻留的节点。

多连接还要求 TCP/IP 侦听器正在运行。为了确保它正在运行，执行下列操作:

1. 按如下所示将环境变量 `DB2COMM` 设置为 TCP/IP:

```
db2set DB2COMM=TCP/IP
```

- 按 `services` 文件中指定的那样，使用 TCP/IP 服务名称来更新数据库管理器配置文件：

```
db2 update dbm cfg using SVCENAME <TCP/IP service name>
```

每个实例都在 `services` 文件中列示了 TCP/IP 服务名称。如果找不到该服务名称或没有更改 `services` 文件的文件许可权，请向系统管理员咨询。

- 停止数据库管理器然后重新启动它以使这些更改生效：

```
db2stop  
db2start
```

`dbmcon` 程序由五个文件组成：

#### **dbmcon.sqC**

用来连接至两个数据库的主源文件。

#### **dbmcon1.sqC**

用来创建绑定至第一个数据库的程序包的源文件。

#### **dbmcon1.h**

包括在 `dbmcon.sqC` 中的 `dbmcon1.sqC` 的头文件，用来访问用于创建和删除要绑定至第一个数据库的表的 SQL 语句。

#### **dbmcon2.sqC**

用来创建绑定至第二个数据库的程序包的源文件。

#### **dbmcon2.h**

包括在 `dbmcon.sqC` 中的 `dbmcon2.sqC` 的头文件，用来访问用于创建和删除要绑定至第二个数据库的表的 SQL 语句。

要构建多连接样本程序 `dbmcon`，请输入：

```
bldmc dbmcon sample sample2
```

得到一个可执行文件 `dbmcon`。

要运行可执行文件，请输入可执行文件名：

```
dbmcon
```

程序将演示向两个数据库的一阶段落实。

#### **相关概念：**

- 第 93 页的『构建文件』

#### **相关任务：**

- 第 147 页的『构建 UNIX C++ 应用程序』

#### **相关参考：**

- 『`svcename` - TCP/IP 服务名称配置参数』（《管理指南：性能》）
- 第 166 页的『AIX C++ 应用程序编译和链接选项』
- 第 193 页的『HP-UX C++ 应用程序编译和链接选项』
- 第 209 页的『Linux C++ 应用程序编译和链接选项』
- 第 223 页的『Solaris C++ 应用程序编译和链接选项』

相关样本:

- 『 bldmc -- Builds AIX C++ multi-connection applications (C++) 』
- 『 bldmc -- Builds HP-UX C++ multi-connection applications (C++) 』
- 『 bldmc -- Builds Linux C++ multi-connection applications (C++) 』
- 『 bldmc -- Builds Solaris C++ multi-connection applications (C++) 』
- 『 dbmcon.sqlC -- How to use multiple databases (C++) 』
- 『 dbmcon1.h -- Class declaration for the source file, dbmcon1.sqlC (C++) 』
- 『 dbmcon1.sqlC -- Functions used in the multiple databases program dbmcon.sqlC (C++) 』
- 『 dbmcon2.h -- Class declaration for the source file, dbmcon2.sqlC (C++) 』
- 『 dbmcon2.sqlC -- Functions used in the multiple databases program dbmcon.sqlC (C++) 』

---

## 构建 UNIX C++ 例程

DB2 提供了用于编译和链接 C++ 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 `sqllib/samples/cpp` 目录中。

脚本文件 `bldrtn` 包含构建例程的命令。该脚本文件将例程编译成可由数据库管理器装入且由客户机应用程序调用的共享库。

第一个参数 `$1` 指定源文件名。第二个参数 `$2` 指定要连接的数据库名称。

数据库参数是可选的。如果未提供数据库名称，则该程序使用缺省的 `sample` 数据库。并且，由于必须在数据库所在的那个实例中构建存储过程，所以没有任何有关用户标识和密码的参数。

过程:

下列示例显示如何构建具有下列内容的例程共享库:

- 存储过程
- 非嵌入式 SQL 用户定义的函数 (UDF)
- 嵌入式 SQL 用户定义的函数 (UDF)

### 存储过程共享库

要从源文件 `spserver.sqlC` 构建样本程序 `spserver`:

1. 如果连接至 `sample` 数据库，则输入构建脚本名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还需要输入该数据库名称:

```
bldrtn spserver database
```

此脚本文件将共享库复制到服务器的 `sqllib/function` 路径中。

2. 接着，通过在服务器上运行 `spcat` 脚本对例程进行编目:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 对这些例程进行了编目，则取消对它们的编目，然后通过调用 `spcreate.db2` 对它们进行编目，最后与数据库断开连接。还可以分别调用 `spdrop.db2` 和 `spcreate.db2` 脚本。

3. 然后，如果这不是第一次构建存储过程，则停止数据库然后重新启动它以确保重组新版本的共享库。为此，可在命令行上输入 `db2stop` 后跟 `db2start`。

一旦构建了共享库 `spserver`，就可构建存取该共享库的客户机应用程序 `spclient`。可使用脚本文件 `bldapp` 构建 `spclient`。

要调用共享库中的存储过程，可输入以下命令运行样本客户机应用程序：

```
spclient database userid password
```

其中，

**database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

**password**

是用户标识的有效密码。

客户机应用程序存取共享库 `spserver`，并在服务器数据库上执行大量存储过程函数。然后将输出返回给该客户机应用程序。

### 非嵌入式 SQL UDF 共享库

要从源文件 `udfsrv.C` 构建用户定义的函数程序 `udfsrv`，请输入构建脚本名和程序名：

```
bldrtn udfsrv
```

此脚本文件将 UDF 复制到 `sqllib/function` 目录中。

如有必要，对 UDF 设置文件方式，以便数据库管理器可以存取它。

一旦构建了 `udfsrv`，就可构建调用它的客户机应用程序 `udfcli`。可以从源文件 `udfcli.sqC` 通过使用脚本文件 `bldapp` 来构建 `udfcli`。

要调用共享库中的 UDF，通过输入以下命令来运行客户机应用程序：

```
udfcli database userid password
```

其中，

**database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

**password**

是用户标识的有效密码。

客户机应用程序将存取共享库 `udfsrv`，并在服务器数据库上执行用户定义的函数。然后将输出返回给该客户机应用程序。

## 嵌入式 SQL UDF 共享库

要从源文件 `udfemsrv.sqC` 构建嵌入式 SQL 用户定义的函数程序 `udfemsrv`，如果连接至 `sample` 数据库，则输入构建脚本名和程序名：

```
bldrtn udfemsrv
```

如果连接至另一个数据库，还需要输入该数据库名称：

```
bldrtn udfemsrv database
```

此脚本文件将 UDF 复制到 `sqllib/function` 目录中。

一旦构建了 `udfemsrv`，就可构建调用它的客户机应用程序 `udfemcli`。可以从源文件 `udfemcli.sqC` 通过使用脚本文件 `bldapp` 来构建 `udfemcli`。

要调用共享库中的 UDF，通过输入以下命令来运行客户机应用程序：

```
udfemcli database userid password
```

其中，

### **database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

### **password**

是用户标识的有效密码。

客户机应用程序将存取共享库 `udfemsrv`，并在服务器数据库上执行用户定义的函数。然后将输出返回给该客户机应用程序。

### 相关概念：

- 第 93 页的『构建文件』

### 相关任务：

- 第 147 页的『构建 UNIX C++ 应用程序』

### 相关参考：

- 第 168 页的『AIX C++ 例程编译和链接选项』
- 第 196 页的『HP-UX C++ 例程编译和链接选项』
- 第 210 页的『Linux C++ 例程编译和链接选项』
- 第 225 页的『Solaris C++ 例程编译和链接选项』

### 相关样本：

- 『`bldrtn -- Builds AIX C++ routines (stored procedures and UDFs) (C++)`』
- 『`bldrtn -- Builds HP-UX C++ routines (stored procedures and UDFs) (C++)`』
- 『`bldrtn -- Builds Linux C++ routines (stored procedures and UDFs) (C++)`』
- 『`bldrtn -- Builds Solaris C++ routines (stored procedures or UDFs) (C++)`』
- 『`spclient.sqC -- Call various stored procedures (C++)`』
- 『`spserver.sqC -- Definition of various types of stored procedures (C++)`』

- 『udfcli.sqlC -- Call a variety of types of user-defined functions (C++)』
- 『udfemcli.sqlC -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『udfemsrv.sqlC -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『udfsrv.C -- Defines a variety of types of user-defined functions (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## 构建 UNIX Micro Focus COBOL 应用程序

DB2 提供了用于编译和链接 Micro Focus COBOL 嵌入式 SQL 和 DB2 API 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 `sqllib/samples/cobol_mf` 目录中。

构建文件 `bldapp` 包含构建 DB2 应用程序的命令。

第一个参数 `$1` 指定源文件名。这是不包含嵌入式 SQL 的程序所需的唯一参数。构建嵌入式 SQL 程序需要连接数据库，因此还需要提供三个可选参数：第二个参数 `$2` 指定您想连接的数据库的名称；第三个参数 `$3` 指定数据库的用户标识；第四个参数 `$4` 指定密码。

对于嵌入式 SQL 程序，`bldapp` 将这些参数传送给预编译和绑定脚本文件 `embprep`。如果未提供数据库名称，则使用缺省 `sample` 数据库。仅当构建程序所在的实例不同于数据库所在的实例时，才需要用户标识和密码参数。

### 过程:

要从源文件 `client.cbl` 构建非嵌入式 SQL 样本程序 `client`，请输入：

```
bldapp client
```

得到一个可执行文件 `client`。可以输入如下命令，对 `sample` 数据库运行该可执行文件：

```
client
```

### 构建和运行嵌入式 SQL 应用程序

从源文件 `updat.sqb` 构建嵌入式 SQL 应用程序 `updat` 有三种方法：

1. 如果与同一实例中的样本数据库连接，则输入：

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接，还需要输入该数据库名称：

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接，还需要输入该数据库实例的用户标识和密码：

```
bldapp updat database userid password
```

得到一个可执行文件 `updat`。

有三种方法运行此嵌入式 SQL 应用程序：

1. 如果存取同一实例中的 `sample` 数据库，只须输入可执行文件名：

```
updat
```

2. 如果存取同一实例中的另一个数据库，则输入可执行文件名和数据库名称：

```
updat database
```

3. 如果存取另一个实例中的数据库，则输入可执行文件名、数据库名称以及该数据库实例的用户标识和密码：

```
updat database userid password
```

#### 相关任务：

- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』

#### 相关参考：

- 第 181 页的『AIX Micro Focus COBOL 应用程序编译和链接选项』
- 第 199 页的『HP-UX Micro Focus COBOL 应用程序编译和链接选项』
- 第 228 页的『Solaris Micro Focus COBOL 应用程序编译和链接选项』
- 第 214 页的『Linux Micro Focus COBOL 应用程序编译和链接选项』

#### 相关样本：

- 『`bldapp` -- Builds AIX Micro Focus COBOL applications』
- 『`bldapp` -- Builds HP-UX Micro Focus COBOL applications』
- 『`bldapp` -- Builds Linux Micro Focus COBOL applications』
- 『`bldapp` -- Builds Solaris Micro Focus COBOL applications』
- 『`client.cbl` -- How to set and query a client (MF COBOL)』
- 『`updat.sqb` -- How to update, delete and insert table data (MF COBOL)』
- 『`embprep` -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## 构建 UNIX Micro Focus COBOL 例程

DB2 提供了用于编译和链接 Micro Focus COBOL 嵌入式 SQL 和 DB2 API 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 `sqllib/samples/cobol_mf` 目录中。

脚本 `bldrtn` 包含构建例程（存储过程）的命令。该脚本将例程源文件编译成可由客户机应用程序调用的共享库。

第一个参数 `$1` 指定源文件名。该脚本将源文件名用作共享库名。第二个参数 `$2` 指定要连接的数据库名称。由于必须在数据库所在的那个实例中构建共享库，所以没有任何有关用户标识和密码的参数。

只有第一个参数即源文件名是必需的。数据库名称是可选的。如果未提供数据库名称，则该程序使用缺省的 `sample` 数据库。

#### 特定于 Solaris 的设置

在 Solaris 上构建 Micro Focus 例程之前，运行以下命令：

```
db2stop
db2set DB2LIBPATH=$LD_LIBRARY_PATH
db2set DB2ENVLIST="COBDIR LD_LIBRARY_PATH"
db2set
db2start
```

确保使用命令 `db2stop` 停止数据库。发出最后一个 `db2set` 命令检查设置：确保 `DB2LIBPATH` 和 `DB2ENVLIST` 设置正确。

#### 过程:

要从源文件 `outsrv.sqb` 构建样本程序 `outsrv`，如果连接样本数据库，则输入：

```
bldrtn outsrv
```

如果连接另一个数据库，还需要输入该数据库名称：

```
bldrtn outsrv database
```

此脚本文件将共享库复制到服务器的 `sqllib/function` 路径中。

一旦构建了存储过程 `outsrv`，就可构建调用它的客户机应用程序 `outcli`。可使用 `bldapp` 脚本文件构建 `outcli`。

要调用存储过程，可输入如下命令运行样本客户机应用程序：

```
outcli database userid password
```

其中，

#### **database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

#### **password**

是用户标识的有效密码。

客户机应用程序存取共享库 `outsrv`，并在服务器数据库上执行同名的存储过程函数。然后将输出返回给该客户机应用程序。

#### 相关任务:

- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』

#### 相关参考:

- 第 183 页的『AIX Micro Focus COBOL 例程编译和链接选项』
- 第 200 页的『HP-UX Micro Focus COBOL 例程编译和链接选项』
- 第 230 页的『Solaris Micro Focus COBOL 例程编译和链接选项』
- 第 215 页的『Linux Micro Focus COBOL 例程编译和链接选项』

#### 相关样本:

- 『`bldrtn -- Builds AIX Micro Focus COBOL routines (stored procedures)`』
- 『`bldrtn -- Builds HP-UX Micro Focus COBOL routines (stored procedures)`』
- 『`bldrtn -- Builds Linux Micro Focus COBOL routines (stored procedures)`』
- 『`bldrtn -- Builds Solaris Micro Focus COBOL routines (stored procedures)`』



- | • 『outcli.sqb -- Call stored procedures using the SQLDA structure (MF COBOL)』
- | • 『outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (MF
- | COBOL)』
- | • 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL
- | programs (C)』



## 第 10 章 AIX

重要注意事项	159	使用配置文件构建 C++ 嵌入式 SQL 应用程序	171
例程的 AIX 导出文件	159	使用配置文件构建 C++ 存储过程	172
AIX 例程和 CREATE 语句	160	使用配置文件构建 C++ 用户定义的函数	173
替换 AIX 共享库	161	IBM COBOL Set AIX 版	174
在 AIX 上安装 COBOL 的注意事项	161	在 AIX 上配置 IBM COBOL 编译器	174
IBM C	161	在 AIX 上构建 IBM COBOL 应用程序	175
C 应用程序的构建脚本	161	IBM COBOL 应用程序的构建脚本	176
AIX C 应用程序编译和链接选项	162	AIX IBM COBOL 应用程序编译和链接选项	177
C 例程的构建脚本	163	在 AIX 上构建 IBM COBOL 例程	178
AIX C 例程编译和链接选项	164	IBM COBOL 例程的构建脚本	179
在 AIX 上构建 C 多线程应用程序	165	AIX IBM COBOL 例程编译和链接选项	179
VisualAge C++	165	Micro Focus COBOL	180
C++ 应用程序的构建脚本	166	在 AIX 上配置 Micro Focus COBOL 编译器	180
AIX C++ 应用程序编译和链接选项	166	Micro Focus COBOL 应用程序的构建脚本	181
C++ 例程的构建脚本	167	AIX Micro Focus COBOL 应用程序编译和链接选项	181
AIX C++ 例程编译和链接选项	168	项	181
在 AIX 上构建 C++ 多线程应用程序	169	Micro Focus COBOL 例程的构建脚本	182
VisualAge C++ 配置文件	170	AIX Micro Focus COBOL 例程编译和链接选项	183
使用配置文件构建 VisualAge C++ 程序	170	REXX	184
使用配置文件构建 C++ DB2 API 应用程序	171	在 AIX 上构建 REXX 应用程序	184

本章提供在 AIX 上构建应用程序的详细信息。有关 AIX 的最新 DB2 应用程序开发情况的更新，请访问以下 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad>

### 重要注意事项

本节提供了在各种受支持编译器上构建 DB2 应用程序的特定于 AIX 的信息。

### 例程的 AIX 导出文件

外部例程是在服务器上编译的且在服务器上的共享库中存储和执行。这些共享库是在编译例程时创建的。

在 AIX<sup>®</sup> 上，您必须提供指定库中的哪些全局函数可从外部调用的导出文件，或者可以使用编译器创建具有“全部导出”行为的 AIX 共享对象和库，如下例所示：

```
x1C -qmkshrobj -q32 -g dlsources.C -o libtestit.so
```

DB2<sup>®</sup> 样本使用导出文件。此文件必须包括库中所有例程的名称。其它 UNIX<sup>®</sup> 平台则是导出共享库中所有全局函数。以下是 AIX 导出文件的一个示例：

```
#! spserver export file  
outlanguage
```

导出文件 spserver.exp 将列示存储过程 outlanguage。链接程序使用 spserver.exp 来创建包含 outlanguage 存储过程的共享库 spserver。

AIX 链接程序文档具有有关导出文件的其它信息。

相关概念:

- 第 160 页的『AIX 例程和 CREATE 语句』

相关任务:

- 第 161 页的『替换 AIX 共享库』

## AIX 例程和 CREATE 语句

下面说明编译和链接例程与在 CREATE 语句的 EXTERNAL NAME 子句中提供的信息之间的关系。

当编译和链接程序时，可以通过使用以 -bE: 选项指定的导出文件来标识外部函数。

假定库 myrtns 包含三个例程: modify、remove 和 add。通过将 modify 置为在链接步骤中链接的导出文件中的第一个入口来将它标识为缺省入口点。通过也将 remove 和 add 函数包括在导出文件中来将它们指示为其它可导出函数。

在链接步骤中，指定:

```
-bE:myrtns.exp
```

这将标识导出文件 myrtns.exp。

该导出文件看起来类似于:

```
modify  
remove  
add
```

最后，例程的 EXTERNAL NAME 子句（它们是使用 modify、remove 和 add 函数来实现的）是按如下方式编码的:

```
EXTERNAL NAME '/u/mydir/routines/myrtns!modify'
```

和

```
EXTERNAL NAME '/u/mydir/routines/myrtns!remove'
```

和

```
EXTERNAL NAME '/u/mydir/routines/myrtns!add'
```

注: 使用的缺省路径将为 sqllib/function。这意味着，如果 EXTERNAL NAME 子句是按如下所示指定的:

```
EXTERNAL NAME 'myrtns!modify'
```

则 DB2® 将会尝试从 sqllib/function 装入 myrtns。

相关概念:

- 第 159 页的『例程的 AIX 导出文件』

相关任务:

- 第 161 页的『替换 AIX 共享库』

## 替换 AIX 共享库

### 过程:

构建共享库之后，通常将它复制到 DB2 将从其存取该库的目录中。当尝试替换例程共享库时，应该运行 `/usr/sbin/slibclean` 来刷新 AIX 共享库高速缓存，或从目标目录中除去该库而后再将该库从源目录复制到目标目录中。否则，因为 AIX 会保留所引用库的高速缓存且不允许覆盖该库，从而可能导致复制操作失败。

### 相关概念:

- 第 159 页的『例程的 AIX 导出文件』
- 第 160 页的『AIX 例程和 CREATE 语句』

## 在 AIX 上安装 COBOL 的注意事项

由于 AIX<sup>®</sup> 装入例程和分辨这些例程内库引用的方式，对应如何安装 COBOL 有一定的要求。这些要求是 COBOL 程序在运行时装入共享库（例程）时要考虑的一个因素。

当装入一个例程时，也必须装入它引用的一系列库。当 AIX 搜索只能由您的程序间接引用的库时，它必须使用已编译到该库中的路径，该库在语言提供程序（IBM COBOL 或 Micro Focus COBOL）构建该路径时会引用该路径。此路径可能与编译器的安装路径完全不同。如果在该系列中找不到该库，则例程装入将失败，您将接收到 SQLCODE -444。

要确保不发生这种情况，在您期望之处安装编译器，然后为所有语言库创建从安装目录到目录 `/usr/lib`（当需要装入库时，几乎总是要搜索的目录）的符号链接。可以将这些库链接到 `sqllib/function`（例程目录），但这只对一个数据库实例起作用；而 `/usr/lib` 则对机器上的每个数据库实例都起作用。

### 相关任务:

- 第 30 页的『设置 UNIX 应用程序开发环境』
- 第 174 页的『在 AIX 上配置 IBM COBOL 编译器』
- 第 180 页的『在 AIX 上配置 Micro Focus COBOL 编译器』

---

## IBM C

DB2 CLI 应用程序和例程的构建信息在 *CLI Guide and Reference* 中。

有关在受支持的 UNIX 操作系统上构建 C 应用程序的信息，请参阅第 141 页的『构建 UNIX C 应用程序』。有关在受支持的 UNIX 操作系统上构建 C 例程的信息，请参阅第 144 页的『构建 UNIX C 例程』。

## C 应用程序的构建脚本

```
|  
| #! /bin/sh  
| # SCRIPT: bldapp  
| # Builds AIX C application programs  
| # Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ] ]  
|  
| # Set DB2PATH to where DB2 will be accessed.  
| # The default is the standard instance path.
```

```

DB2PATH=$HOME/sql1lib

# Set lib32 for 32-bit programs, lib for 64-bit,
# and set extra compile flag for 64-bit programs.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB=lib32
    EXTRA_CFLAG=
else
    LIB=lib
    EXTRA_CFLAG=-q64
fi

# If an embedded SQL program, precompile and bind it.
# Note: some .sqc files contain no SQL but link in
# utilemb.sqc, so if you get this warning, ignore it:
# SQL0053W No SQL statements were found in the program.
if [ -f $1".sqc" ]
then
./embprep $1 $2 $3 $4
    # Compile the utilemb.c error-checking utility.
    xlc $EXTRA_CFLAG -I$DB2PATH/include -c utilemb.c
else
    # Compile the utilapi.c error-checking utility.
    xlc $EXTRA_CFLAG -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
xlc $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
    # Link the program with utilemb.o
    xlc $EXTRA_CFLAG -o $1 $1.o utilemb.o -ldb2 -L$DB2PATH/$LIB
else
    # Link the program with utilapi.o
    xlc $EXTRA_CFLAG -o $1 $1.o utilapi.o -ldb2 -L$DB2PATH/$LIB
fi

```

## AIX C 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 AIX IBM C 编译器来构建 C 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

bldapp 的编译和链接选项	
<b>编译选项:</b>	
<b>xlc</b>	IBM C 编译器。
<b>\$EXTRA_CFLAG</b>	对于启用了 64 位支持的实例，包含“-q64”；否则，它不包含值。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。例如: \$HOME/sql1lib/include。
<b>-c</b>	只执行编译；不链接。编译和链接是两个独立的步骤。

## bldapp 的编译和链接选项

### 链接选项:

**xlc** 使用编译器作为链接程序的前端。

### **\$EXTRA\_CFLAG**

对于启用了 64 位支持的实例，包含 “-q64”；否则，它不包含值。

**-o \$1** 指定可执行程序。

**\$1.o** 指定该程序的对象文件。

### **utilemb.o**

如果是嵌入式 SQL 程序，应包括嵌入式 SQL 实用程序对象文件以便检查错误。

### **utilapi.o**

如果不是嵌入式 SQL 程序，应包括 DB2 API 实用程序对象文件以便检查错误。

**-ldb2** 链接 DB2 库。

### **-L\$DB2PATH/\$LIB**

指定 DB2 运行时共享库的位置。例如：\$HOME/sqllib/\$LIB。如果不指定 -L 选项，则编译器将采用以下路径：/usr/lib:/lib。

有关其它编译器选项，参考编译器文档。

### 相关任务:

- 第 141 页的『构建 UNIX C 应用程序』

### 相关参考:

- 第 164 页的『AIX C 例程编译和链接选项』

### 相关样本:

- 『bldapp -- Builds AIX C application programs (C)』

## C 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds AIX C routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set lib32 for 32-bit programs, lib for 64-bit,
# and set extra compile flag for 64-bit programs.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB=lib32
    EXTRA_CFLAG=
else
    LIB=lib
    EXTRA_CFLAG=-q64
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi
```

```

# Compile the program.
xlc_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

# Link the program using the export file $1.exp,
xlc_r $EXTRA_CFLAG -qmkshrobj -o $1 $1.o -ldb2 -L$DB2PATH/$LIB -bE:$1.exp

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

## AIX C 例程编译和链接选项

下列内容是 DB2 建议用于使用 AIX IBM C 编译器来构建 C 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
<b>编译选项:</b>	
<b>xlc_r</b>	使用 IBM C 编译器的多线程版本，该版本是必需的，因为例程可能会在其它例程所在的进程中运行（THREADSAFE）或在引擎本身当中运行（NOT FENCED）。
<b>\$EXTRA_CFLAG</b>	对于启用了 64 位支持的实例，包含“-q64”；否则，它不包含值。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。例如：\$HOME/sqllib/include。
<b>-c</b>	只执行编译；不链接。编译和链接是两个独立的步骤。
<b>链接选项:</b>	
<b>xlc_r</b>	将编译器的多线程版本用作链接程序的前端。
<b>\$EXTRA_CFLAG</b>	对于启用了 64 位支持的实例，包含“-q64”；否则，它不包含值。
<b>-qmkshrobj</b>	创建共享库。
<b>-o \$1</b>	指定输出文件名。
<b>\$1.o</b>	指定对象文件。
<b>-ldb2</b>	链接 DB2 库。
<b>-L\$DB2PATH/\$LIB</b>	指定 DB2 运行时共享库的位置。例如：\$HOME/sqllib/\$LIB。如果不指定 -L 选项，则编译器将采用以下路径：/usr/lib:/lib。
<b>-bE:\$1.exp</b>	指定导出文件。该导出文件包含一个例程列表。
有关其它编译器选项，参考编译器文档。	

### 相关任务:

- 第 144 页的『构建 UNIX C 例程』

### 相关参考:

- 第 162 页的『AIX C 应用程序编译和链接选项』



相关样本:

- 『bldrtn -- Builds AIX C routines (stored procedures and UDFs) (C)』

## 在 AIX 上构建 C 多线程应用程序

需要使用 `xlc_r` 编译器而不是 `xlc` 编译器（对于 C++，则需要使用 `x1C_r` 编译器而不是 `x1C` 编译器）来编译和链接 AIX 上的 C 多线程应用程序。`_r` 版本为多线程编译设置适当的预处理器定义，并向链接程序提供适当的线程库名称。

可从编译器文档中获得关于使用多线程编译器前端的编译器和链接标志设置的其它信息。

```
| sqllib/samples/c 中的脚本文件 bldmt 包含构建嵌入式 SQL 多线程程序所需的命令。  
| 除 xlc_r 编译器外，在没有链接任何实用程序文件的情况下，编译和链接选项与用于嵌  
| 入式 SQL 脚本文件 bldapp 的编译和链接选项相同。
```

过程:

要从源文件 `dbthrds.sqc` 构建多线程样本程序 `dbthrds`，请输入:

```
bldmt dbthrds
```

得到一个可执行文件 `dbthrds`。要对 `sample` 数据库运行该可执行文件，请输入该可执行文件名称:

```
dbthrds
```

相关概念:

- 第 93 页的『构建文件』

相关任务:

- 第 141 页的『构建 UNIX C 应用程序』

相关参考:

- 第 162 页的『AIX C 应用程序编译和链接选项』
- 第 62 页的『C 样本』

相关样本:

- 『bldmt -- Builds AIX C multi-threaded applications (C)』
- 『dbthrds.sqc -- How to use multiple context APIs on UNIX (C)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## VisualAge C++

有关在受支持的 UNIX 操作系统上构建 C++ 应用程序的信息，请参阅第 147 页的『构建 UNIX C++ 应用程序』。有关在受支持的 UNIX 操作系统上构建 C++ 例程的信息，请参阅第 151 页的『构建 UNIX C++ 例程』。

## C++ 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds AIX C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set lib32 for 32-bit programs, lib for 64-bit,
# and set extra compile flag for 64-bit programs.
bitwidth=~LANG=C db2level | awk '/bits/{print $5}'
if [ $bitwidth = "\"32\"" ]; then
    LIB=lib32
    EXTRA_CFLAG=
else
    LIB=lib
    EXTRA_CFLAG=-q64
fi

# If an embedded SQL program, precompile and bind it.
# Note: some .sqc files contain no SQL but link in
# utilemb.sqc, so if you get this warning, ignore it:
# SQL0053W No SQL statements were found in the program.
if [ -f $1".sqc" ]
then
./embprep $1 $2 $3 $4
# Compile the utilemb.C error-checking utility.
xlc $EXTRA_CFLAG -qstaticinline -I$DB2PATH/include -c utilemb.C
else
# Compile the utilapi.C error-checking utility.
xlc $EXTRA_CFLAG -qstaticinline -I$DB2PATH/include -c utilapi.C
fi

# Compile the program.
xlc $EXTRA_CFLAG -qstaticinline -I$DB2PATH/include -c $1.C

if [ -f $1".sqc" ]
then
# Link the program with utilemb.o
xlc $EXTRA_CFLAG -o $1 $1.o utilemb.o -ldb2 -L$DB2PATH/$LIB
else
# Link the program with utilapi.o
xlc $EXTRA_CFLAG -o $1 $1.o utilapi.o -ldb2 -L$DB2PATH/$LIB
fi
```

## AIX C++ 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 AIX IBM VisualAge C++ 编译器来构建 C++ 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

<b>bldapp</b> 的编译和链接选项	
<b>编译选项:</b>	
<b>x1C</b>	VisualAge C++ 编译器。
<b>EXTRA_CFLAG</b>	
	对于启用了 64 位支持的实例，包含 “-q64”；否则，它不包含值。
<b>-qstaticinline</b>	
	为非内联的内联函数提供内部链接，因此在链接时即使函数存在于多个对象文件中也不会出现警告。
<b>-I\$DB2PATH/include</b>	
	指定 DB2 包含文件的位置。例如: \$HOME/sql1lib/include。
<b>-c</b>	
	只执行编译；不链接。编译和链接是两个独立的步骤。
<b>链接选项:</b>	
<b>x1C</b>	使用编译器作为链接程序的前端。
<b>EXTRA_CFLAG</b>	
	对于启用了 64 位支持的实例，包含 “-q64”；否则，它不包含值。
<b>-o \$1</b>	指定可执行程序。
<b>\$1.o</b>	指定该程序的对象文件。
<b>utilapi.o</b>	
	包括用于非嵌入式 SQL 程序的 API 实用程序对象文件。
<b>utilemb.o</b>	
	包括用于嵌入式 SQL 程序的嵌入式 SQL 实用程序对象文件。
<b>-ldb2</b>	链接 DB2 库。
<b>-L\$DB2PATH/\$LIB</b>	
	指定 DB2 运行时共享库的位置。例如: \$HOME/sql1lib/\$LIB。如果不指定 -L 选项，则编译器假定如下路径: /usr/lib:/lib。
有关其它编译器选项，参考编译器文档。	

#### 相关任务:

- 第 147 页的『构建 UNIX C++ 应用程序』
- 第 171 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 171 页的『使用配置文件构建 C++ DB2 API 应用程序』

#### 相关参考:

- 第 168 页的『AIX C++ 例程编译和链接选项』

#### 相关样本:

- 『bldapp -- Builds AIX C++ applications (C++)』

## C++ 例程的构建脚本

```

#!/bin/sh
# SCRIPT: bldrtn
# Builds AIX C++ routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.

```

```

# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set lib32 for 32-bit programs, lib for 64-bit,
# and set extra compile flag for 64-bit programs.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB=lib32
    EXTRA_CFLAG=
else
    LIB=lib
    EXTRA_CFLAG=-q64
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compile the program.
x1C_r $EXTRA_CFLAG -qstaticinline -I$DB2PATH/include -c $1.C

# Link using export file $1.exp, creating shared library $1
x1C_r $EXTRA_CFLAG -qmksrobj -o $1 $1.o -L$DB2PATH/$LIB -ldb2 -bE:$1.exp

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

## AIX C++ 例程编译和链接选项

下列内容是 DB2 建议用于使用 AIX VisualAge C++ 编译器来构建 C++ 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>x1C_r</b>	IBM VisualAge C++ 编译器的多线程版本，该版本是必需的，因为例程可能会在其它例程所在的进程中运行（THREADSAFE）或在引擎本身当中运行（NOT FENCED）。
<b>\$EXTRA_CFLAG</b>	对于启用了 64 位支持的实例，包含“-q64”；否则，它不包含值。
<b>-qstaticinline</b>	为非内联的内联函数提供内部链接，因此在链接时即使函数存在于多个对象文件中也不会出现警告。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。例如：\$HOME/sqllib/include。
<b>-c</b>	只执行编译；不链接。编译和链接是两个独立的步骤。

## bldrtn 的编译和链接选项

### 链接选项:

**x1C\_r** 将编译器的多线程版本用作链接程序的前端。

### **\$EXTRA\_CFLAG**

对于启用了 64 位支持的实例，包含 “-q64”；否则，它不包含值。

### **-qmksprobj**

创建共享库。

**-o \$1** 指定输出为共享库文件。

**\$1.o** 指定该程序的对象文件。

### **-L\$DB2PATH/\$LIB**

指定 DB2 运行时共享库的位置。例如：\$HOME/sql1lib/\$LIB。如果不指定 -L 选项，则编译器将采用以下路径：/usr/lib:/lib。

**-ldb2** 链接 DB2 库。

### **-bE:\$1.exp**

指定导出文件。该导出文件包含一个例程列表。

有关其它编译器选项，参考编译器文档。

### 相关任务:

- 第 144 页的『构建 UNIX C 例程』
- 第 172 页的『使用配置文件构建 C++ 存储过程』
- 第 173 页的『使用配置文件构建 C++ 用户定义的函数』

### 相关参考:

- 第 166 页的『AIX C++ 应用程序编译和链接选项』

### 相关样本:

- 『bldrtn -- Builds AIX C++ routines (stored procedures and UDFs) (C++)』

## 在 AIX 上构建 C++ 多线程应用程序

需要使用 x1C\_r 编译器而不是 x1C 编译器（对于 C，则需要使用 x1c\_r 编译器而不是 x1c 编译器）来编译和链接 AIX 上的 C++ 多线程应用程序。\_r 版本为多线程编译设置适当的预处理器定义，并向链接程序提供适当的线程库名称。

可从编译器文档中获得关于使用多线程编译器前端的编译器和链接标志设置的其它信息。

脚本 bldmt 包含构建多线程应用程序的命令。除了以上讨论的 x1C\_r 编译器之外，在没有链接任何实用程序文件的情况下，其它编译和链接选项与在嵌入式 SQL 脚本文件 bldapp 中所用的选项相同。

### 过程:

要从源文件 dbthrds.sqC 构建多线程样本程序 dbthrds，请输入：

```
bldmt dbthrds
```

得到一个可执行文件 dbthrds。要对 sample 数据库运行该可执行文件，请输入该可执行文件名称：

```
dbthrds
```

相关概念：

- 第 93 页的『构建文件』

相关任务：

- 第 147 页的『构建 UNIX C++ 应用程序』

相关参考：

- 第 166 页的『AIX C++ 应用程序编译和链接选项』
- 第 62 页的『C 样本』

相关样本：

- 『bldmt -- Builds AIX C++ multi-threaded applications (C++)』
- 『dbthrds.sqC -- How to use multiple context APIs on UNIX (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## VisualAge C++ 配置文件

注：CLI 应用程序和例程的构建信息在 *CLI Guide and Reference* 中。

### 使用配置文件构建 VisualAge C++ 程序

VisualAge C++ 版本 5.0 具有增量编译器和批处理方式编译器。当批处理方式编译器使用 makefile 和构建文件时，增量编译器就使用配置文件。请参阅 VisualAge C++ 版本 5.0 附带提供的文档以了解有关此情况的更多信息。

DB2 为可用 VisualAge C++ 编译器构建的不同类型的 DB2 程序提供配置文件。

过程：

要使用 DB2 配置文件，首先将一个环境变量设置为要编译的程序名。接着用 VisualAge C++ 提供的命令编译该程序。以下是一些主题，它们描述可以如何使用 DB2 提供的配置文件来编译不同类型的程序：

- 使用配置文件构建 C++ 嵌入式 SQL 应用程序
- 使用配置文件构建 C++ DB2 API 应用程序
- 使用配置文件构建 C++ 存储过程
- 使用配置文件构建 C++ 用户定义的函数

相关任务：

- 第 171 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 171 页的『使用配置文件构建 C++ DB2 API 应用程序』
- 第 172 页的『使用配置文件构建 C++ 存储过程』
- 第 173 页的『使用配置文件构建 C++ 用户定义的函数』
- 第 141 页的『构建 UNIX C 应用程序』

- 第 144 页的『构建 UNIX C 例程』
- 第 147 页的『构建 UNIX C++ 应用程序』
- 第 151 页的『构建 UNIX C++ 例程』

## 使用配置文件构建 C++ DB2 API 应用程序

sqlllib/samples/c 和 sqlllib/samples/cpp 中的配置文件 api.icc 允许您在 AIX 上使用 C 或 C++ 来构建 DB2 API 程序。

过程:

要使用配置文件来从源文件 cli\_info.c 构建 DB2 API 样本程序 cli\_info, 执行以下操作:

1. 输入以下命令, 将 API 环境变量设置为该程序名:

- 对于 bash 或 Korn shell:

```
export API=cli_info
```

- 对于 C shell:

```
setenv API cli_info
```

2. 如果在工作目录中有 api.ics 文件, 该文件是用 api.icc 文件构建另一个程序生成的, 可使用以下命令删除 api.ics 文件:

```
rm api.ics
```

不必删除对将要再次构建的同一程序生成的现有 api.ics 文件。

3. 输入以下命令, 编译样本程序:

```
vacbld api.icc
```

**注:** vacbld 命令是由 VisualAge C++ 提供的。

得到一个可执行文件 cli\_info。可输入可执行文件名来运行该程序:

```
cli_info
```

相关任务:

- 第 171 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 172 页的『使用配置文件构建 C++ 存储过程』
- 第 173 页的『使用配置文件构建 C++ 用户定义的函数』

## 使用配置文件构建 C++ 嵌入式 SQL 应用程序

sqlllib/samples/c 和 sqlllib/samples/cpp 中的配置文件 emb.icc 允许您在 AIX 上使用 C 和 C++ 来构建 DB2 嵌入式 SQL 应用程序。

过程:

要使用配置文件来从源文件 tmod.sqc 构建嵌入式 SQL 应用程序 tmod, 执行以下操作:

1. 输入以下命令, 将 EMB 环境变量设置为该程序名:

- 对于 bash 或 Korn shell:

```
export EMB=tbmod
```

- 对于 C shell:

```
setenv EMB tbmod
```

2. 如果在工作目录中有 emb.ics 文件, 该文件是用 emb.icc 文件构建另一个程序生成的, 可使用以下命令删除 emb.ics 文件:

```
rm emb.ics
```

不必删除对将要再次构建的同一程序生成的现有 emb.ics 文件。

3. 输入以下命令, 编译样本程序:

```
vacbld emb.icc
```

**注:** vacbld 命令是由 VisualAge C++ 提供的。

得到一个可执行文件 tbmod。可输入可执行文件名来运行该程序:

```
tbmod
```

**相关任务:**

- 第 171 页的『使用配置文件构建 C++ DB2 API 应用程序』
- 第 172 页的『使用配置文件构建 C++ 存储过程』
- 第 173 页的『使用配置文件构建 C++ 用户定义的函数』

## 使用配置文件构建 C++ 存储过程

sqllib/samples/c 和 sqllib/samples/cpp 中的配置文件 stp.icc 允许您在 AIX 上使用 C 和 C++ 来构建 DB2 嵌入式 SQL 存储过程。

**过程:**

要使用配置文件来从源文件 spserver.sqc 构建嵌入式 SQL 存储过程共享库 spserver, 执行以下操作:

1. 输入以下命令, 将 STP 环境变量设置为程序名:

- 对于 bash 或 Korn shell:

```
export STP=spserver
```

- 对于 C shell:

```
setenv STP spserver
```

2. 如果在工作目录中有 stp.ics 文件, 该文件是用 stp.icc 文件构建另一个程序生成的, 可使用以下命令删除 stp.ics 文件:

```
rm stp.ics
```

不必删除对将要再次构建的同一程序生成的现有 stp.ics 文件。

3. 输入以下命令, 编译样本程序:

```
vacbld stp.icc
```

**注:** vacbld 命令是由 VisualAge C++ 提供的。

存储过程共享库被复制到服务器的 sqllib/function 路径中。

接着, 通过在服务器上运行 spcat 脚本对共享库中的存储过程进行编目:



spcat

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 对这些存储过程进行了编目，则取消对它们的编目，然后通过调用 `spcreate.db2` 对它们进行编目，最后与数据库断开连接。还可以分别调用 `spdrop.db2` 和 `spcreate.db2` 脚本。

最后停止数据库然后重新启动它以便可识别新的共享库。必要时，对共享库设置文件方式，以便 DB2 实例可存取它。

一旦构建了存储过程共享库 `spserver`，就可以构建调用库中存储过程的客户机应用程序 `spclient`。可使用配置文件 `emb.icc` 构建 `spclient`。

要调用存储过程，通过输入下列命令来运行样本客户机应用程序：

```
spclient database userid password
```

其中，

**database**

是要连接的数据库的名称。 该名称可以是 `sample`、其远程别名或其它名称。

**userid** 是有效的用户标识。

**password**

是有效的密码。

客户机应用程序存取共享库 `spserver`，并在服务器数据库上执行大量存储过程函数。然后将输出返回给该客户机应用程序。

**相关任务：**

- 第 171 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 171 页的『使用配置文件构建 C++ DB2 API 应用程序』
- 第 173 页的『使用配置文件构建 C++ 用户定义的函数』

## 使用配置文件构建 C++ 用户定义的函数

`sqllib/samples/c` 和 `sqllib/samples/cpp` 中的配置文件 `udf.icc` 允许您在 AIX 上使用 C 和 C++ 构建用户定义的函数。

**过程：**

要从源文件 `udfsrv.c` 使用配置文件构建用户定义的函数程序 `udfsrv`，执行以下操作：

1. 输入以下命令，将 UDF 环境变量设置为程序名：

- 对于 `bash` 或 `Korn shell`：

```
export UDF=udfsrv
```

- 对于 `C shell`：

```
setenv UDF udfsrv
```

2. 如果在工作目录中有 `udf.ics` 文件，该文件是用 `udf.icc` 文件构建另一个程序生成的，可使用以下命令删除 `udf.ics` 文件：

```
rm udf.ics
```

不必删除对将要再次构建的同一程序生成的现有 `udf.ics` 文件。

3. 输入以下命令，编译样本程序：

```
vacblld udf.icc
```

注：`vacblld` 命令是由 VisualAge C++ 提供的。

UDF 库被复制到服务器的 `sqllib/function` 路径中。

必要时，对用户定义的函数设置文件方式，以便 DB2 实例可运行它。

一旦构建了 `udfsrv`，就可构建调用它的客户机应用程序 `udfcli`。提供了此程序的 DB2 CLI 和嵌入式 SQL 版本。

可以从 `sqllib/samples/cli` 中的源文件 `udfcli.c` 通过使用配置文件 `cli.icc` 来构建 DB2 CLI `udfcli` 程序。

可以从 `sqllib/samples/c` 中的源文件 `udfcli.sqc` 通过使用配置文件 `emb.icc` 来构建嵌入式 SQL `udfcli` 程序。

要调用 UDF，可输入可执行文件名运行样本调用应用程序：

```
udfcli
```

调用应用程序从 `udfsrv` 库中调用 `ScalarUDF` 函数。

相关任务：

- 第 171 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 171 页的『使用配置文件构建 C++ DB2 API 应用程序』
- 第 172 页的『使用配置文件构建 C++ 存储过程』

---

## IBM COBOL Set AIX 版

### 在 AIX 上配置 IBM COBOL 编译器

如果开发包含嵌入式 SQL 和 DB2 API 调用的应用程序，且正在使用 IBM COBOL Set AIX 版编译器，则需要执行下列步骤。

过程：

- 当使用命令行处理器命令 `db2 prep` 预编译应用程序时，要使用 `target ibmcob` 选项。
- 在源文件中不要使用制表符。
- 在源文件的首行可使用 `PROCESS` 和 `CBL` 关键字，用以设置编译选项。
- 如果您的应用程序只包含嵌入式 SQL，而不包含 DB2 API 调用，则不需要使用 `pgmname(mixed)` 编译选项。如果使用 DB2 API 调用，则必须使用 `pgmname(mixed)` 编译选项。
- 如果使用的是 IBM COBOL Set AIX 版编译器的“System/390 主机数据类型支持”功能部件，则您的应用程序的 DB2 包含文件位于如下目录中：

```
$HOME/sqllib/include/cobol_i
```

如果要使用提供的脚本文件构建 DB2 样本程序，必须将在脚本文件中指定的包含文件路径更改为指向 `cobol_i` 目录而不是 `cobol_a` 目录。

如果未使用 IBM COBOL Set AIX 版编译器的“System/390 主机数据类型支持”功能部件或使用的是此编译器的较早版本，则您的应用程序的 DB2 包含文件位于如下目录中：

```
$HOME/sqlllib/include/cobol_a
```

按如下所示指定包括 `.cbl` 扩展名的 COPY 文件名：

```
COPY "sql.cbl".
```

#### 相关概念：

- 第 161 页的『在 AIX 上安装 COBOL 的注意事项』

#### 相关任务：

- 第 30 页的『设置 UNIX 应用程序开发环境』
- 第 175 页的『在 AIX 上构建 IBM COBOL 应用程序』
- 第 178 页的『在 AIX 上构建 IBM COBOL 例程』

#### 相关参考：

- 第 177 页的『AIX IBM COBOL 应用程序编译和链接选项』
- 第 179 页的『AIX IBM COBOL 例程编译和链接选项』

## 在 AIX 上构建 IBM COBOL 应用程序

DB2 提供了用于编译和链接 IBM COBOL 嵌入式 SQL 和 DB2 API 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 `sqlllib/samples/cobol` 目录中。

构建文件 `bldapp` 包含构建 DB2 应用程序的命令。

第一个参数 `$1` 指定源文件名。这是不包含嵌入式 SQL 的程序所需的唯一参数。构建嵌入式 SQL 程序需要连接数据库，因此还需要提供三个可选参数：第二个参数 `$2` 指定您想连接的数据库的名称；第三个参数 `$3` 指定数据库的用户标识；第四个参数 `$4` 指定密码。

对于嵌入式 SQL 程序，`bldapp` 将这些参数传送给预编译和绑定脚本文件 `embprep`。如果未提供数据库名称，则使用缺省 `sample` 数据库。仅当构建程序所在的实例不同于数据库所在的实例时，才需要用户标识和密码参数。

#### 过程：

要从源文件 `client.cbl` 构建非嵌入式 SQL 样本程序 `client`，请输入：

```
bldapp client
```

得到一个可执行文件 `client`。可以输入如下命令，对 `sample` 数据库运行该可执行文件：

```
client
```

#### 构建和运行嵌入式 SQL 应用程序

从源文件 `updat.sqb` 构建嵌入式 SQL 应用程序 `updat` 有三种方法：

1. 如果与同一实例中的样本数据库连接，则输入：  
`bldapp updat`
2. 如果与同一实例中的另一个数据库连接，还需要输入该数据库名称：  
`bldapp updat database`
3. 如果与另一个实例中的数据库连接，还需要输入该数据库实例的用户标识和密码：  
`bldapp updat database userid password`

得到一个可执行文件 `updat`。

有三种方法运行此嵌入式 SQL 应用程序：

1. 如果存取同一实例中的 `sample` 数据库，只须输入可执行文件名：  
`updat`
2. 如果存取同一实例中的另一个数据库，则输入可执行文件名和数据库名称：  
`updat database`
3. 如果存取另一个实例中的数据库，则输入可执行文件名、数据库名称以及该数据库实例的用户标识和密码：  
`updat database userid password`

相关概念：

- 第 93 页的『构建文件』

相关任务：

- 第 178 页的『在 AIX 上构建 IBM COBOL 例程』

相关参考：

- 第 177 页的『AIX IBM COBOL 应用程序编译和链接选项』
- 第 71 页的『COBOL 样本』

相关样本：

- 『`bldapp -- Builds AIX COBOL applications`』
- 『`client.cbl -- How to set and query a client (IBM COBOL)`』
- 『`embprep -- To prep and bind a COBOL embedded SQL sample on AIX`』
- 『`updat.sqb -- How to update, delete and insert table data (IBM COBOL)`』

## IBM COBOL 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds AIX COBOL applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
./embprep $1 $2 $3 $4
fi

# Compile the checkerr.cbl error checking utility.
cob2 -qpgmname\(mixed\) -qlib -I$DB2PATH/include/cobol_a \
```

```

-c checkerr.cbl

# Compile the program.
cob2 -qpgmname\mixed\ -qlib -I$DB2PATH/include/cobol_a \
-c $1.cbl

# Link the program.
cob2 -o $1 $1.o checkerr.o -L$DB2PATH/lib -ldb2

```

## AIX IBM COBOL 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 IBM AIX COBOL Set 编译器来构建 COBOL 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

<b>bldapp 的编译和链接选项</b>	
<b>编译选项:</b>	
<b>cob2</b>	IBM COBOL Set 编译器。
<b>-qpgmname\mixed\</b>	指示编译器可用大小写混合的名称调用 (CALL) 库入口点。
<b>-qlib</b>	指示编译器处理 COPY 语句。
<b>-I\$DB2PATH/include/cobol_a</b>	指定 DB2 包含文件的位置。例如: \$HOME/sql1lib/include/cobol_a。
<b>-c</b>	只执行编译; 不链接。编译和链接是两个独立的步骤。
<b>链接选项:</b>	
<b>cob2</b>	使用编译器作为链接程序的前端。
<b>-o \$1</b>	指定可执行程序。
<b>\$1.o</b>	指定该程序的对象文件。
<b>checkerr.o</b>	包括该实用程序的对象文件以便检查错误。
<b>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。例如: \$HOME/sql1lib/lib。如果不指定 -L 选项, 则编译器将采用以下路径: /usr/lib:/lib。
<b>-ldb2</b>	链接数据库管理器库。
有关其它编译器选项, 参考编译器文档。	

### 相关任务:

- 第 175 页的『在 AIX 上构建 IBM COBOL 应用程序』

### 相关参考:

- 第 179 页的『AIX IBM COBOL 例程编译和链接选项』

### 相关样本:

- 『bldapp -- Builds AIX COBOL applications』

## 在 AIX 上构建 IBM COBOL 例程

DB2 提供了用于编译和链接 COBOL 嵌入式 SQL 和 DB2 API 程序的构建脚本。这些脚本和可以使用这些文件构建的样本程序一同位于 `sqllib/samples/cobol` 目录中。

`sqllib/samples/cobol` 中的脚本 `bldrtn` 包含构建例程（存储过程）的命令。该脚本将例程编译成可由客户机应用程序调用的共享库。

第一个参数 `$1` 指定源文件名。第二个参数 `$2` 指定要连接的数据库名称。由于必须在数据库所在的那个实例中构建共享库，所以没有任何有关用户标识和密码的参数。

只有第一个参数即源文件名是必需的。该脚本将源文件名 `$1` 用作共享库名。数据库名称是可选的。如果未提供数据库名称，则该程序使用缺省的 `sample` 数据库。

### 过程:

要从源文件 `outsrv.sqb` 构建样本程序 `outsrv`，连接至样本数据库而后输入:

```
bldrtn outsrv
```

如果连接另一个数据库，还需要输入该数据库名称:

```
bldrtn outsrv database
```

此脚本文件将共享库复制到服务器的 `sqllib/function` 路径中。

一旦构建了例程共享库 `outsrv`，就可以构建调用库中例程的客户机应用程序 `outcli`。可使用 `bldapp` 脚本文件构建 `outcli`。

要调用例程，通过输入下列命令来运行样本客户机应用程序:

```
outcli database userid password
```

其中，

#### **database**

是要连接的数据库的名称。该名称可以是 `sample`、其远程别名或其它名称。

**userid** 是有效的用户标识。

#### **password**

是用户标识的有效密码。

客户机应用程序将存取共享库 `outsrv`，并在服务器数据库上执行同名的例程，然后将输出返回给客户机应用程序。

### 相关概念:

- 第 93 页的『构建文件』

### 相关任务:

- 第 175 页的『在 AIX 上构建 IBM COBOL 应用程序』

### 相关参考:

- 第 179 页的『AIX IBM COBOL 例程编译和链接选项』
- 第 71 页的『COBOL 样本』

### 相关样本:

- 『 bldrtn -- Builds AIX COBOL routines (stored procedures) 』
- 『 embprep -- To prep and bind a COBOL embedded SQL sample on AIX 』
- 『 outcli.sqb -- Call stored procedures using the SQLDA structure (IBM COBOL) 』
- 『 outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (IBM COBOL) 』

## IBM COBOL 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds AIX COBOL routines (stored procedures)
# Usage: bldrtn <program_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Precompile and bind the program.
./embprep $1 $2

# Compile the checkerr.cbl error checking utility.
cob2 -qpgmname\mixed\ -qlib -I$DB2PATH/include/cobol_a \
    -c checkerr.cbl

# Compile the program.
cob2 -qpgmname\mixed\ -qlib -c -I$DB2PATH/include/cobol_a $1.cbl

# Link the program creating shared library $1 with export file $1.exp
cob2 -o $1 $1.o checkerr.o -bnoentry -bE:$1.exp \
    -L$DB2PATH/lib -ldb2

# Copy the shared library to the sqllib/function directory of the DB2 instance.
# This assumes the user has write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

## AIX IBM COBOL 例程编译和链接选项

下列内容是 DB2 建议用于使用 AIX IBM COBOL Set 编译器来构建 COBOL 例程（存储过程）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>cob2</b>	IBM COBOL Set 编译器。
<b>-qpgmname\mixed\</b>	指示编译器可用大小写混合的名称调用 (CALL) 库入口点。
<b>-qlib</b>	指示编译器处理 COPY 语句。
<b>-c</b>	只执行编译；不链接。编译和链接是两个独立的步骤。
<b>-I\$DB2PATH/include/cobol_a</b>	指定 DB2 包含文件的位置。例如: \$HOME/sqllib/include/cobol_a。

## bldrtn 的编译和链接选项

### 链接选项:

**cob2** 使用编译器来链接编辑。

**-o \$1** 指定输出为共享库文件。

**\$1.o** 指定存储过程的对象文件。

### checkerr.o

包括该实用程序的对象文件以便检查错误。

### -bnoentry

不要指定共享库的缺省入口点。

### -bE:\$1.exp

指定导出文件。该导出文件包含存储过程的列表。

### -L\$DB2PATH/lib

指定 DB2 运行时共享库的位置。例如: \$HOME/sql1lib/lib。如果不指定 -L 选项, 则编译器将采用以下路径: /usr/lib:/lib。

**-ldb2** 链接数据库管理器库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 178 页的『在 AIX 上构建 IBM COBOL 例程』

### 相关参考:

- 第 177 页的『AIX IBM COBOL 应用程序编译和链接选项』

### 相关样本:

- 『bldrtn -- Builds AIX COBOL routines (stored procedures)』

## Micro Focus COBOL

有关在受支持的 UNIX 操作系统上构建 Micro Focus COBOL 应用程序的信息, 请参阅第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』。有关在受支持的 UNIX 操作系统上构建 Micro Focus COBOL 例程的信息, 请参阅第 155 页的『构建 UNIX Micro Focus COBOL 例程』。

## 在 AIX 上配置 Micro Focus COBOL 编译器

如果使用 Micro Focus COBOL 编译器开发包含嵌入式 SQL 和 DB2 API 调用的应用程序, 则执行以下操作。

### 过程:

- 当使用命令行处理器命令 `db2 prep` 预编译应用程序时, 使用 `target mfcob` 选项。
- 在 Micro Focus COBOL 环境变量 `COBCPY` 中必须包括 DB2 COBOL COPY 文件目录。COBCPY 环境变量指定 COPY 文件的位置。Micro Focus COBOL 的 DB2 COPY 文件驻留在该数据库实例目录下的 `sql1lib/include/cobol_mf` 中。

要包括此目录, 请输入:



- 在 bash 或 Korn shell 上:
 

```
export COBCPY=$COBCPY:$HOME/sqlllib/include/cobol_mf
```
- 在 C shell 上:
 

```
setenv COBCPY $COBCPY:$HOME/sqlllib/include/cobol_mf
```

注: 您可能想要在 .profile 或 .login 文件中设置 COBCPY。

#### 相关概念:

- 第 161 页的『在 AIX 上安装 COBOL 的注意事项』

#### 相关任务:

- 第 30 页的『设置 UNIX 应用程序开发环境』
- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』
- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』

#### 相关参考:

- 第 181 页的『AIX Micro Focus COBOL 应用程序编译和链接选项』
- 第 183 页的『AIX Micro Focus COBOL 例程编译和链接选项』

## Micro Focus COBOL 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds AIX Micro Focus COBOL applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqlllib

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
./embprep $1 $2 $3 $4
fi

# Compile the checkerr.cbl error checking utility.
cob -c -x checkerr.cbl

# Compile the program.
cob -c -x $1.cbl

# Link the program.
cob -x -o $1 $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf
```

## AIX Micro Focus COBOL 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 AIX Micro Focus COBOL 编译器来构建 COBOL 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

<b>bldapp</b> 的编译和链接选项	
<b>编译选项:</b>	
<b>cob</b>	COBOL 编译器。
<b>-c</b>	只执行编译; 不链接。
<b>-x</b>	当与 <b>-c</b> 配合使用时, 产生对象文件。
<b>链接选项:</b>	
<b>cob</b>	使用编译器作为链接程序的前端。
<b>-x</b>	产生可执行程序。
<b>-o \$1</b>	指定可执行程序。
<b>\$1.o</b>	指定该程序的对象文件。
<b>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。例如: \$HOME/sqllib/lib。如果不指定 <b>-L</b> 选项, 则编译器将采用以下路径: /usr/lib:/lib。
<b>-ldb2</b>	链接 DB2 库。
<b>-ldb2gmf</b>	与 Micro Focus COBOL 的 DB2 异常处理程序库链接。
有关其它编译器选项, 参考编译器文档。	

#### 相关任务:

- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』

#### 相关参考:

- 第 183 页的『AIX Micro Focus COBOL 例程编译和链接选项』

#### 相关样本:

- 『bldapp -- Builds AIX Micro Focus COBOL applications』

## Micro Focus COBOL 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds AIX Micro Focus COBOL routines (stored procedures)
# Usage: bldrtn <program_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Precompile and bind the program.
./embprep $1 $2

# Compile the program.
cob -c -x $1.cbl

# Link the program.
cob -x -o $1 $1.o -Q -bnoentry \
    -Q -bI:$DB2PATH/lib/db2g.imp -L$DB2PATH/lib -ldb2 -ldb2gmf
```

```
# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

## AIX Micro Focus COBOL 例程编译和链接选项

下列内容是 DB2 建议用于使用 AIX Micro Focus COBOL 编译器来构建 COBOL 例程（存储过程）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
<b>编译选项:</b>	
<b>cob</b>	COBOL 编译器。
<b>-c</b>	只执行编译；不链接。编译和链接是两个独立的步骤。
<b>-x</b>	当与 <b>-c</b> 选项配合使用时，编译对象模块。
<b>链接选项:</b>	
<b>cob</b>	将编译器用作链接程序的前端。
<b>-x</b>	产生共享库。
<b>-o \$1</b>	指定可执行程序。
<b>\$1.o</b>	指定该程序的对象文件。
<b>-Q -bnoentry</b>	不要指定共享库的缺省入口点。
<b>-Q -bI:\$DB2PATH/lib/db2g.imp</b>	提供至 DB2 应用程序库的入口点的列表。
<b>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。例如: \$HOME/sqllib/lib。如果不指定 <b>-L</b> 选项，则编译器将采用以下路径: /usr/lib:/lib。
<b>-ldb2</b>	链接 DB2 库。
<b>-ldb2gmf</b>	与 Micro Focus COBOL 的 DB2 异常处理程序库链接。
有关其它编译器选项，参考编译器文档。	

### 相关任务:

- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』

### 相关参考:

- 第 181 页的『AIX Micro Focus COBOL 应用程序编译和链接选项』

### 相关样本:

- 『bldrtn -- Builds AIX Micro Focus COBOL routines (stored procedures)』

## 在 AIX 上构建 REXX 应用程序

以下内容显示了如何在 AIX 上构建 REXX 应用程序。DB2 AIX 版支持经典 REXX 以及 Object REXX。Object REXX 是 REXX 语言的面向对象版本。面向对象扩充已添加到经典 REXX 中，但其现有的函数和指令未更改。Object REXX 解释器是其先前版本的增强版本，它还支持：

- 类、对象和方法
- 消息传递和多态性
- 单继承性和多重继承性

Object REXX 与经典 REXX 完全兼容。在本节中，只要用到 REXX，都指所有 REXX 版本（包括 Object REXX）。

不要预编译或绑定 REXX 程序。

### 过程:

要在 AIX 上运行 DB2 REXX/SQL 程序，必须设置 LIBPATH 环境变量以包括 DB2 安装目录下的 lib。

对于 bash 或 Korn shell，请输入：

```
export LIBPATH=$LIBPATH:/lib:/usr/lib:/usr/opt/db2_08_01/lib
```

对于 C shell，请输入：

```
setenv LIBPATH $LIBPATH:/lib:/usr/lib:/usr/opt/db2_08_01/lib
```

在 AIX 上，您的应用程序文件可有任何文件扩展名。可以使用下列两种方法中的任一种来运行您的应用程序：

1. 在 shell 命令提示符下，输入 `rexx name`，其中 *name* 是 REXX 程序的名称（包括扩展名，如果存在这样的扩展名的话）。
2. 如果 REXX 程序的首行包含“特殊符号”（#!），且标识出 REXX/6000 解释器所在的目录，则可在 shell 命令提示符下输入 REXX 程序名，以运行它。例如，如果 REXX/6000 解释器文件在 `/usr/bin` 目录中，则将以下行包括在内作为 REXX 程序的第一行：

```
#!/usr/bin/rexx
```

然后，在 shell 命令提示符下输入如下命令，使该程序成为可执行程序：

```
chmod +x name
```

在 shell 命令提示符下输入 REXX 程序的文件名，以运行它。

REXX 样本程序在目录 `sqllib/samples/rexx` 中。要运行样本 REXX 程序 `updat.cmd`，请输入：

```
updat.cmd
```

### 相关任务:

- 第 30 页的『设置 UNIX 应用程序开发环境』

相关参考:

- 第 85 页的『REXX 样本』



## 第 11 章 HP-UX

HP-UX C . . . . .	187	HP-UX C++ 例程编译和链接选项 . . . . .	196
C 应用程序的构建脚本 . . . . .	187	在 HP-UX 上构建 C++ 多线程应用程序 . . . . .	197
HP-UX C 应用程序编译和链接选项 . . . . .	188	Micro Focus COBOL . . . . .	198
C 例程的构建脚本 . . . . .	189	在 HP-UX 上构建 Micro Focus COBOL 编译器 . . . . .	198
HP-UX C 例程编译和链接选项 . . . . .	190	Micro Focus COBOL 应用程序的构建脚本 . . . . .	199
在 HP-UX 上构建 C 多线程应用程序 . . . . .	191	HP-UX Micro Focus COBOL 应用程序编译和链接选项 . . . . .	199
HP-UX C++ . . . . .	192	Micro Focus COBOL 例程的构建脚本 . . . . .	200
C++ 应用程序的构建脚本 . . . . .	192	HP-UX Micro Focus COBOL 例程编译和链接选项 . . . . .	200
HP-UX C++ 应用程序编译和链接选项 . . . . .	193		
C++ 例程的构建脚本 . . . . .	195		

本章提供在 HP-UX 上构建 DB2 应用程序的详细信息。有关 HP-UX 的最新 DB2 应用程序开发情况的更新，请访问 DB2 应用程序开发 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad>

### HP-UX C

DB2 CLI 应用程序和例程的构建信息在 *CLI Guide and Reference* 中。

有关在受支持的 UNIX 操作系统上构建 C 应用程序的信息，请参阅第 141 页的『构建 UNIX C 应用程序』。有关在受支持的 UNIX 操作系统上构建 C 例程的信息，请参阅第 144 页的『构建 UNIX C 例程』。

#### C 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds HP-UX C applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

# Determine the HP platform and set correct compile/link options
hpplat=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $hpplat = "ia64" ]; then
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DD64"
        LIB="lib"
    else
        EXTRA_CFLAG="+DD32"
        LIB="lib32"
    fi
else
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DA2.0W"
        LIB="lib"
    else
        EXTRA_CFLAG=
        LIB="lib32"
    fi
fi
```

```

# The runtime path is recommended for all applications.
# If you need to use SHLIB_PATH or LD_LIBRARY_PATH, unset
# the RUNTIME variable by commenting out the following line.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,+b$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
# Note: some .sqc files contain no SQL but link in
# utilemb.sqc, so if you get this warning, ignore it:
# SQL0053W No SQL statements were found in the program.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.c error-checking utility.
    cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c utilemb.c
else
    # Compile the utilapi.c error-checking utility.
    cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
    # Link the program with utilemb.o.
    cc $EXTRA_CFLAG -o $1 $1.o utilemb.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
else
    # Link the program with utilapi.o.
    cc $EXTRA_CFLAG -o $1 $1.o utilapi.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
fi

```

## HP-UX C 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 HP-UX C 编译器构建 C 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

bldapp 的编译和链接选项	
<b>编译选项:</b>	
<b>cc</b>	C 编译器。
<b>\$EXTRA_CFLAG</b>	如果 HP-UX 平台是 IA64 并且启用了 64 位支持，则此标志包含值 <b>+DD64</b> ；如果启用了 32 位支持，则它包含值 <b>+DD32</b> 。如果 HP-UX 平台是 PA-RISC 并且启用了 64 位支持，则它包含值 <b>+DA2.0W</b> 。对于 PA-RISC 平台上的 32 位支持，此标志不包含值。 <b>+DD64</b> 必须用来为 HP-UX on IA64 生成 64 位代码。 <b>+DD32</b> 必须用来为 HP-UX on IA64 生成 32 位代码。 <b>+DA2.0W</b> 必须用来为 HP-UX on PA-RISC 生成 64 位代码。
<b>-Ae</b>	启用 HP ANSI 扩展方式。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。
<b>-c</b>	只执行编译；不链接。编译和链接是两个独立的步骤。



## bldapp 的编译和链接选项

### 链接选项:

**cc** 将编译器用作链接程序的前端。

### **\$EXTRA\_CFLAG**

如果 HP-UX 平台是 IA64 并且启用了 64 位支持, 则此标志包含值 **+DD64**; 如果启用了 32 位支持, 则它包含值 **+DD32**。如果 HP-UX 平台是 PA-RISC 并且启用了 64 位支持, 则它包含值 **+DA2.0W**。对于 PA-RISC 平台上的 32 位支持, 此标志不包含值。

**+DD64** 必须用来为 HP-UX on IA64 生成 64 位代码。

**+DD32** 必须用来为 HP-UX on IA64 生成 32 位代码。

**+DA2.0W**

必须用来为 HP-UX on PA-RISC 生成 64 位代码。

**-o \$1** 指定可执行文件。

**\$1.o** 指定该程序的对象文件。

### **utilemb.o**

如果是嵌入式 SQL 程序, 应包括嵌入式 SQL 实用程序对象文件以便检查错误。

### **utilapi.o**

如果不是嵌入式 SQL 程序, 应包括 DB2 API 实用程序对象文件以便检查错误。

### **\$EXTRA\_LFLAG**

指定运行时路径。如果设置了它, 则对于 32 位, 它包含值 **-Wl,+b\$HOME/sqllib/lib32**, 对于 64 位, 它包含值 **-Wl,+b\$HOME/sqllib/lib**。如果未设置它, 则它不包含值。

### **-L\$DB2PATH/\$LIB**

指定 DB2 运行时共享库的位置。对于 32 位: **\$HOME/sqllib/lib32**; 对于 64 位: **\$HOME/sqllib/lib**。

**-ldb2** 链接 DB2 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 141 页的『构建 UNIX C 应用程序』

### 相关样本:

- 『bldapp -- Builds HP-UX C applications (C)』

## C 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds HP-UX C routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Determine the HP platform and set correct compile/link options
hpplat=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $hpplat = "ia64" ]; then
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DD64"
        LIB="_lib"
    else
        EXTRA_CFLAG="+DD32"
        LIB="_lib32"
    fi
else
    EXTRA_CFLAG="+DD32"
    LIB="_lib32"
fi
```

```

fi
else
    if [ $bitwidth = "\64\" ]; then
        EXTRA_CFLAG="+DA2.0W"
        LIB="lib"
    else
        EXTRA_CFLAG=
        LIB="lib32"
    fi
fi

# The runtime path is recommended for all applications.
# If you need to use SHLIB_PATH or LD_LIBRARY_PATH, unset
# the RUNTIME variable by commenting out the following line.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="+b$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2
fi

# Compile the program.
cc $EXTRA_CFLAG +u1 +z -Ae -I$DB2PATH/include \
-D_POSIX_C_SOURCE=199506L -c $1.c

# Link the program to create a shared library
ld -b -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2 -lpthread

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

## HP-UX C 例程编译和链接选项

下列内容是 DB2 建议用于使用 HP-UX C 编译器构建 C 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

## bldrtn 的编译和链接选项

### 编译选项:

**cc** C 编译器。

### **\$EXTRA\_CFLAG**

如果 HP-UX 平台是 IA64 并且启用了 64 位支持, 则此标志包含值 **+DD64**; 如果启用了 32 位支持, 则它包含值 **+DD32**。如果 HP-UX 平台是 PA-RISC 并且启用了 64 位支持, 则它包含值 **+DA2.0W**。对于 PA-RISC 平台上的 32 位支持, 此标志不包含值。

**+DD64** 必须用来为 HP-UX on IA64 生成 64 位代码。

**+DD32** 必须用来为 HP-UX on IA64 生成 32 位代码。

**+DA2.0W**

必须用来为 HP-UX on PA-RISC 生成 64 位代码。

**+u1** 允许存取未对齐的数据。仅在应用程序使用未对齐的数据时使用。

**+z** 生成与位置无关的代码。

**-Ae** 启用 HP ANSI 扩展方式。

### **-I\$DB2PATH/include**

指定 DB2 包含文件的位置。例如: `-I$DB2PATH/include`。

### **-D\_POSIX\_C\_SOURCE=199506L**

确保定义 `_REENTRANT` 的 POSIX 线程库选项, 该选项是必需的, 因为例程可能会在其它例程所在的进程中运行 (`THREADSAFE`) 或在引擎本身当中运行 (`NOT FENCED`)。

**-c** 只执行编译; 不链接。编译和链接是两个独立的步骤。

### 链接选项:

**ld** 使用链接程序进行链接。

**-b** 创建一个共享库而不是通常的可执行文件。

**-o \$1** 指定输出为共享库文件。

**\$1.o** 指定该程序的对象文件。

### **\$EXTRA\_LFLAG**

指定运行时路径。如果设置了它, 则对于 32 位, 它包含值 `+b$HOME/sql1lib/lib32`, 对于 64 位, 它包含值 `+b$HOME/sql1lib/lib`。如果未设置它, 则它不包含值。

### **-L\$DB2PATH/\$LIB**

指定 DB2 运行时共享库的位置。对于 32 位: `$HOME/sql1lib/lib32`; 对于 64 位: `$HOME/sql1lib/lib`。

**-ldb2** 链接 DB2 库。

### **-lpthread**

与 POSIX 线程库进行链接。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 144 页的『构建 UNIX C 例程』

### 相关样本:

- 『`bldrtn -- Builds HP-UX C routines (stored procedures and UDFs) (C)`』

## 在 HP-UX 上构建 C 多线程应用程序

HP-UX 提供 POSIX 线程库和 DCE 线程库。DB2 只支持使用 POSIX 线程库的多线程应用程序。

在 HP-UX 上构建多线程应用程序需要定义 `_REENTRANT`，以便编译它们。HP-UX 文档建议使用 `-D_POSIX_C_SOURCE=199506L` 来编译。这也将确保定义了 `_REENTRANT`。应用程序还需要使用 `-lpthread` 来链接。

脚本文件 `bldmt` 包含构建多线程应用程序的命令。除了上面指定的选项之外，其它编译和链接选项与在嵌入式 SQL 脚本文件 `bldapp` 中所用的选项相同。

#### 过程:

要从源文件 `dbthrrds.sqc` 构建样本程序 `dbthrrds`，请输入：

```
bldmt dbthrrds
```

得到一个可执行文件 `dbthrrds`。要对 `sample` 数据库运行该可执行文件，请输入该可执行文件名称：

```
dbthrrds
```

#### 相关概念:

- 第 93 页的『构建文件』

#### 相关参考:

- 第 62 页的『C 样本』
- 第 188 页的『HP-UX C 应用程序编译和链接选项』

#### 相关样本:

- 『`bldmt -- Builds HP-UX C multi-threaded applications (C)`』
- 『`dbthrrds.sqc -- How to use multiple context APIs on UNIX (C)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)`』

---

## HP-UX C++

有关在受支持的 UNIX 操作系统上构建 C++ 应用程序的信息，请参阅第 147 页的『构建 UNIX C++ 应用程序』。有关在受支持的 UNIX 操作系统上构建 C++ 例程的信息，请参阅第 151 页的『构建 UNIX C++ 例程』。

### C++ 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds HP-UX C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Determine the HP platform and set correct compile/link options
hpplat=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $hpplat = "ia64" ]; then
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DD64 -AA"
        LIB="lib"
    else
        EXTRA_CFLAG="+DD32 -AA"
    fi
fi
```

```

        LIB="lib32"
    fi
else
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DA2.0W"
        LIB="lib"
    else
        EXTRA_CFLAG=
        LIB="lib32"
    fi
fi

# The runtime path is recommended for all applications.
# If you need to use SHLIB_PATH or LD_LIBRARY_PATH, unset
# the RUNTIME variable by commenting out the following line.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,+b$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
# Note: some .sqc files contain no SQL but link in
# utilemb.sqc, so if you get this warning, ignore it:
# SQL0053W No SQL statements were found in the program.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.C error-checking utility.
    aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c utilemb.C
else
    # Compile the utilapi.C error-checking utility.
    aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c utilapi.C
fi

# Compile the program.
aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c $1.C

if [ -f $1".sqc" ]
then
    # Link the program with utilemb.o.
    aCC $EXTRA_CFLAG -o $1 $1.o utilemb.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
else
    # Link the program with utilapi.o.
    aCC $EXTRA_CFLAG -o $1 $1.o utilapi.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
fi

```

## HP-UX C++ 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 HP-UX C++ 编译器构建 C++ 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

## bldapp 的编译和链接选项

### 编译选项:

**aCC** HP aC++ 编译器。

### **\$EXTRA\_CFLAG**

如果 HP-UX 平台是 IA64 并且启用了 64 位支持, 则此标志包含值 **+DD64 -AA**; 如果启用了 32 位支持, 则它包含值 **+DD32 -AA**。如果 HP-UX 平台是 PA-RISC 并且启用了 64 位支持, 则它包含值 **+DA2.0W**。对于 PA-RISC 平台上的 32 位支持, 此标志不包含值。

**+DD64** 必须用来为 HP-UX on IA64 生成 64 位代码。

**+DD32** 必须用来为 HP-UX on IA64 生成 32 位代码。

**-AA** 在 IA64 上允许使用 ANSI C++ 标准功能部件 (如名称空间 std 和 C++ 标准库)。

### **+DA2.0W**

必须用来为 HP-UX on PA-RISC 生成 64 位代码。

**-ext** 允许各种 C++ 扩展名, 包括 “long long” 支持。

### **-I\$DB2PATH/include**

指定 DB2 包含文件的位置。例如: \$HOME/sqlllib/include

**-c** 只执行编译; 不链接。编译和链接是两个独立的步骤。

### 链接选项:

**aCC** 将 HP aC++ 编译器用作链接程序的前端。

### **\$EXTRA\_CFLAG**

如果 HP-UX 平台是 IA64 并且启用了 64 位支持, 则此标志包含值 **+DD64 -AA**; 如果启用了 32 位支持, 则它包含值 **+DD32 -AA**。如果 HP-UX 平台是 PA-RISC 并且启用了 64 位支持, 则它包含值 **+DA2.0W**。对于 PA-RISC 平台上的 32 位支持, 此标志不包含值。

**+DD64** 必须用来为 HP-UX on IA64 生成 64 位代码。

**+DD32** 必须用来为 HP-UX on IA64 生成 32 位代码。

**-AA** 在 IA64 上允许使用 ANSI C++ 标准功能部件 (如名称空间 std 和 C++ 标准库)。

### **+DA2.0W**

必须用来为 HP-UX on PA-RISC 生成 64 位代码。

**-o \$1** 指定可执行文件。

**\$1.o** 指定该程序的对象文件。

### **utilemb.o**

如果是嵌入式 SQL 程序, 应包括嵌入式 SQL 实用程序对象文件以便检查错误。

### **utilapi.o**

如果不是嵌入式 SQL 程序, 应包括 DB2 API 实用程序对象文件以便检查错误。

### **\$EXTRA\_LFLAG**

指定运行时路径。如果设置了它, 则对于 32 位, 它包含值 **-Wl,+b\$HOME/sqlllib/lib32**, 对于 64 位, 它包含值 **-Wl,+b\$HOME/sqlllib/lib**。如果未设置它, 则它不包含值。

### **-L\$DB2PATH/\$LIB**

指定 DB2 运行时共享库的位置。对于 32 位: \$HOME/sqlllib/lib32; 对于 64 位: \$HOME/sqlllib/lib。

**-ldb2** 链接 DB2 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 147 页的『构建 UNIX C++ 应用程序』

相关样本:

- 『 bldapp -- Builds HP-UX C++ applications (C++) 』

## C++ 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds HP-UX C++ routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Determine the HP platform and set correct compile/link options
hpplat=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $hpplat = "ia64" ]; then
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DD64 -AA"
        LIB="lib"
    else
        EXTRA_CFLAG="+DD32 -AA"
        LIB="lib32"
    fi
else
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DA2.0W"
        LIB="lib"
    else
        EXTRA_CFLAG=
        LIB="lib32"
    fi
fi

# The runtime path is recommended for all applications.
# If you need to use SHLIB_PATH or LD_LIBRARY_PATH, unset
# the RUNTIME variable by commenting out the following line.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,+b$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqC" ]
then
    ./embprep $1 $2
fi

# Compile the program. First ensure it is coded with extern "C".
aCC $EXTRA_CFLAG +u1 +z -ext -mt -I$DB2PATH/include -c $1.C

# Link the program to create a shared library.
aCC $EXTRA_CFLAG -mt -b -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

## HP-UX C++ 例程编译和链接选项

下列内容是 DB2 建议用于使用 HP-UX C++ 编译器构建 C++ 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>aCC</b>	HP aC++ 编译器。
<b>\$EXTRA_CFLAG</b>	
	如果 HP-UX 平台是 IA64 并且启用了 64 位支持，则此标志包含值 <b>+DD64 -AA</b> ；如果启用了 32 位支持，则它包含值 <b>+DD32 -AA</b> 。如果 HP-UX 平台是 PA-RISC 并且启用了 64 位支持，则它包含值 <b>+DA2.0W</b> 。对于 PA-RISC 平台上的 32 位支持，此标志不包含值。
<b>+DD64</b>	必须用来为 HP-UX on IA64 生成 64 位代码。
<b>+DD32</b>	必须用来为 HP-UX on IA64 生成 32 位代码。
<b>-AA</b>	在 IA64 上允许使用 ANSI C++ 标准功能部件（如名称空间 std 和 C++ 标准库）。
<b>+DA2.0W</b>	必须用来为 HP-UX on PA-RISC 生成 64 位代码。
<b>+u1</b>	允许存取未对齐的数据。
<b>+z</b>	生成与位置无关的代码。
<b>-ext</b>	允许各种 C++ 扩展名（包括“long long”支持）。
<b>-mt</b>	允许 HP aC++ 编译器的线程支持，该选项是必需的，因为例程可能会在其它例程所在的进程中运行（THREADSAFE）或在引擎本身当中运行（NOT FENCED）。
<b>-I\$DB2PATH/include</b>	
	指定 DB2 包含文件的位置。例如：\$DB2PATH/include。
<b>-c</b>	只执行编译；不链接。编译和链接是两个独立的步骤。



## bldrtn 的编译和链接选项

### 链接选项:

**aCC** 将 HP aC++ 编译器用作链接程序的前端。

### **\$EXTRA\_CFLAG**

如果 HP-UX 平台是 IA64 并且启用了 64 位支持, 则此标志包含值 **+DD64 -AA**; 如果启用了 32 位支持, 则它包含值 **+DD32 -AA**。如果 HP-UX 平台是 PA-RISC 并且启用了 64 位支持, 则它包含值 **+DA2.0W**。对于 PA-RISC 平台上的 32 位支持, 此标志不包含值。

**+DD64** 必须用来为 HP-UX on IA64 生成 64 位代码。

**+DD32** 必须用来为 HP-UX on IA64 生成 32 位代码。

**-AA** 在 IA64 上允许使用 ANSI C++ 标准功能部件 (如名称空间 std 和 C++ 标准库)。

### **+DA2.0W**

必须用来为 HP-UX on PA-RISC 生成 64 位代码。

**-mt** 允许 HP aC++ 编译器的线程支持, 该选项是必需的, 因为例程可能会在其它例程所在的进程中运行 (THREADSAFE) 或在引擎本身当中运行 (NOT FENCED)。

**-b** 创建一个共享库而不是通常的可执行文件。

**-o \$1** 指定可执行文件。

**\$1.o** 指定该程序的对象文件。

### **\$EXTRA\_LFLAG**

指定运行时路径。如果设置了它, 则对于 32 位, 它包含值 **-Wl,+b\$HOME/sql1lib/lib32**, 对于 64 位, 它包含值 **-Wl,+b\$HOME/sql1lib/lib**。如果未设置它, 则它不包含值。

### **-L\$DB2PATH/\$LIB**

指定 DB2 运行时共享库的位置。对于 32 位: **\$HOME/sql1lib/lib32**; 对于 64 位: **\$HOME/sql1lib/lib**。

**-ldb2** 链接 DB2 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 151 页的『构建 UNIX C++ 例程』

### 相关样本:

- 『bldrtn -- Builds HP-UX C++ routines (stored procedures and UDFs) (C++)』

## 在 HP-UX 上构建 C++ 多线程应用程序

HP-UX 提供 POSIX 线程库和 DCE 线程库。HP-UX 上的 DB2 只支持使用 POSIX 线程库的多线程应用程序。

对于 HP-UX C++ 编译器, 在编译和链接步骤中, **-mt** 都必须用于多线程应用程序。

脚本 **bldmt** 包含构建多线程应用程序的命令。除了上面指定的选项之外, 其它编译和链接选项与在嵌入式 SQL 脚本文件 **bldapp** 中所用的选项相同。

### 过程:

要从源文件 **dbthrds.sqC** 构建样本程序 **dbthrds**, 请输入:

```
bldmt dbthrds
```

得到一个可执行文件 dbthdrs。要对 sample 数据库运行该可执行文件，请输入该可执行文件名称：

```
dbthdrs
```

**相关概念：**

- 第 93 页的『构建文件』

**相关参考：**

- 第 62 页的『C 样本』
- 第 193 页的『HP-UX C++ 应用程序编译和链接选项』

**相关样本：**

- 『bldmt -- Builds HP-UX C++ multi-threaded applications (C++)』
- 『dbthdrs.sqlC -- How to use multiple context APIs on UNIX (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## Micro Focus COBOL

有关在受支持的 UNIX 操作系统上构建 Micro Focus COBOL 应用程序的信息，请参阅第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』。有关在受支持的 UNIX 操作系统上构建 Micro Focus COBOL 例程的信息，请参阅第 155 页的『构建 UNIX Micro Focus COBOL 例程』。

### 在 HP-UX 上构建 Micro Focus COBOL 编译器

如果开发包含嵌入式 SQL 和 DB2 API 调用的应用程序，且正在使用 Micro Focus COBOL 编译器，则应注意以下几点。

**过程：**

- 当使用命令行处理器命令 db2 prep 预编译应用程序时，使用 target mfcob 选项。
- 在 Micro Focus COBOL 环境变量 COBCPY 中必须包括 DB2 COBOL COPY 文件目录。COBCPY 环境变量指定 COPY 文件的位置。Micro Focus COBOL 的 DB2 COPY 文件驻留在该数据库实例目录下的 sqllib/include/cobol\_mf 中。

要包括该目录，

– 在 bash 或 Korn shell 上，请输入：

```
export COBCPY=${COBCPY}:${HOME}/sqllib/include/cobol_mf
```

– 在 C shell 上，请输入：

```
setenv COBCPY ${COBCPY}:${HOME}/sqllib/include/cobol_mf
```

**注：**您可能想要在 .profile 或 .login 文件中设置 COBCPY。

**相关任务：**

- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』
- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』

**相关参考：**

- 第 199 页的『HP-UX Micro Focus COBOL 应用程序编译和链接选项』

- 第 200 页的『HP-UX Micro Focus COBOL 例程编译和链接选项』

## Micro Focus COBOL 应用程序的构建脚本

```

#!/bin/sh
# SCRIPT: bldapp
# Builds HP-UX Micro Focus COBOL applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
./embprep $1 $2 $3 $4
fi

# Compile the checkerr.cbl error checking utility.
cob -cx checkerr.cbl

# Compile the program.
cob -cx $1.cbl

# Link the program.
cob -x $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf

```

## HP-UX Micro Focus COBOL 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Micro Focus COBOL 编译器在 HP-UX 上构建 COBOL 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

<b>bldapp 的编译和链接选项</b>	
<b>编译选项:</b>	
<b>cob</b>	Micro Focus COBOL 编译器。
<b>-cx</b>	编译为对象模块。
<b>链接选项:</b>	
<b>cob</b>	使用编译器作为链接程序的前端。
<b>-x</b>	指定可执行程序。
<b>\$1.o</b>	包括该程序的对象文件。
<b>checkerr.o</b>	包括该实用程序的对象文件以便检查错误。
<b>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。
<b>-ldb2</b>	链接 DB2 库。
<b>-ldb2gmf</b>	与 Micro Focus COBOL 的 DB2 异常处理程序库链接。
有关其它编译器选项，参考编译器文档。	

相关任务:

- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』

相关样本:

- 『bldapp -- Builds HP-UX Micro Focus COBOL applications』

## Micro Focus COBOL 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds HP-UX Micro Focus COBOL routines (stored procedures)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
embprep $1 $2
fi

# Compile the program.
cob +z -cx $1.cbl

# Link the program.
ld -b -o $1 $1.o -L$DB2PATH/lib -ldb2 -ldb2gmf \
-L$COBDIR/coblib -lcobol -lcrtn

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

## HP-UX Micro Focus COBOL 例程编译和链接选项

下列内容是 DB2 建议用于使用 Micro Focus COBOL 编译器在 HP-UX 上构建 COBOL 例程（存储过程）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>cob</b>	COBOL 编译器。
<b>+z</b>	生成与位置无关的代码。
<b>-cx</b>	编译为对象模块。

## bldrtn 的编译和链接选项

### 链接选项:

- ld** 使用链接程序进行链接。
- b** 创建一个共享库而不是通常的可执行文件。
- o \$1** 指定可执行文件。
- \$1.o** 包括该程序的对象文件。
- L\$DB2PATH/lib**  
指定 DB2 运行时共享库的位置。
- ldb2** 与 DB2 共享库链接。
- ldb2gmf**  
与 Micro Focus COBOL 的 DB2 异常处理程序库链接。
- L\$COBDIR/coblib**  
指定 COBOL 运行时库的位置。
- lcobol**  
链接 COBOL 库。
- lcrtn**  
链接 crtn 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』

### 相关样本:

- 『bldrtn -- Builds HP-UX Micro Focus COBOL routines (stored procedures)』



## 第 12 章 Linux

Linux C . . . . .	203	Linux C++ 例程编译和链接选项 . . . . .	210
C 应用程序的构建脚本 . . . . .	203	在 Linux 上构建 C++ 多线程应用程序 . . . . .	211
Linux C 应用程序编译和链接选项 . . . . .	204	Micro Focus COBOL . . . . .	212
C 例程的构建脚本 . . . . .	205	在 Linux 上配置 Micro Focus COBOL 编译器 . . . . .	212
Linux C 例程编译和链接选项 . . . . .	206	Micro Focus COBOL 应用程序的构建脚本 . . . . .	214
在 Linux 上构建 C 多线程应用程序 . . . . .	207	Linux Micro Focus COBOL 应用程序编译和链接选项 . . . . .	214
Linux C++ . . . . .	208	Micro Focus COBOL 例程的构建脚本 . . . . .	215
C++ 应用程序的构建脚本 . . . . .	208	Linux Micro Focus COBOL 例程编译和链接选项 . . . . .	215
Linux C++ 应用程序编译和链接选项 . . . . .	209		
C++ 例程的构建脚本 . . . . .	210		

本章提供在 Linux 上构建应用程序的详细信息。有关 Linux 的最新 DB2 应用程序开发情况的更新信息，请访问以下的 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad>

### Linux C

DB2 CLI 应用程序和例程的构建信息在 *CLI Guide and Reference* 中。

有关在受支持的 UNIX 操作系统上构建 C 应用程序的信息，请参阅第 141 页的『构建 UNIX C 应用程序』。有关在受支持的 UNIX 操作系统上构建 C 例程的信息，请参阅第 144 页的『构建 UNIX C 例程』。

#### C 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Linux C applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# Determine if we are running with 32-bit, and
# if we are running with 32-bit on Linux AMD64
LIB="lib"
EXTRA_C_FLAGS=""
HARDWAREPLAT=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\32\" ]; then
    LIB="lib32"
    if [ "$HARDWAREPLAT" = "x86_64" ]; then
        EXTRA_C_FLAGS="-m32"
    fi
fi

# The runtime path is recommended for all applications.
# If you need to use LD_LIBRARY_PATH, unset the RUNTIME
# variable by commenting out the following line.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath,$DB2PATH/$LIB"
```

```

else
  EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
# Note: some .sql files contain no SQL but link in
# utilemb.sql, so if you get this warning, ignore it:
# SQL0053W No SQL statements were found in the program.
if [ -f $1".sql" ]
then
  ./embprep $1 $2 $3 $4
  # Compile the utilemb.c error-checking utility.
  gcc $EXTRA_C_FLAGS -I$DB2PATH/include -c utilemb.c
else
  # Compile the utilapi.c error-checking utility.
  gcc $EXTRA_C_FLAGS -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
gcc $EXTRA_C_FLAGS -I$DB2PATH/include -c $1.c

if [ -f $1".sql" ]
then
  # Link the program with utilemb.o.
  gcc $EXTRA_C_FLAGS -o $1 $1.o utilemb.o $EXTRA_LFLAG \
    -L$DB2PATH/$LIB -ldb2
else
  # Link the program with utilapi.o.
  gcc $EXTRA_C_FLAGS -o $1 $1.o utilapi.o $EXTRA_LFLAG \
    -L$DB2PATH/$LIB -ldb2
fi

```

## Linux C 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Linux C 编译器构建 C 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

bldapp 的编译和链接选项	
编译选项:	
<b>gcc</b>	GNU/Linux C 编译器。
<b>\$EXTRA_C_FLAGS</b>	对于 AMD64 上的 32 位支持，包含“-m32”；否则不包含值。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。
<b>-c</b>	只执行编译；不链接。此脚本文件有单独的编译和链接步骤。



## bldapp 的编译和链接选项

### 链接选项:

**gcc** 将编译器用作链接程序的前端。

### **\$EXTRA\_C\_FLAGS**

对于 AMD64 上的 32 位支持, 包含 “-m32”; 否则不包含值。

**-o \$1** 指定可执行文件。

**\$1.o** 指定对象文件。

### **utilemb.o**

如果是嵌入式 SQL 程序, 应包括嵌入式 SQL 实用程序对象文件以便检查错误。

### **utilapi.o**

如果不是嵌入式 SQL 程序, 应包括 DB2 API 实用程序对象文件以便检查错误。

### **\$EXTRA\_LFLAG**

如果未注释 `'RUNTIME=true'`, 则对于 32 位, 它包含值 “-Wl,-rpath,\$DB2PATH/lib32”, 对于 64 位, 它包含值 “-Wl,-rpath,\$DB2PATH/lib”。否则, 它不包含值。

### **-L\$DB2PATH/\$LIB**

指定链接时 DB2 静态库和共享库的位置。例如, 对于 32 位: `$HOME/sql1lib/lib32`, 对于 64 位: `$HOME/sql1lib/lib`。如果不指定 `-L` 选项, 则假定为 `/usr/lib/lib`。

**-ldb2** 链接 DB2 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 141 页的『构建 UNIX C 应用程序』

### 相关样本:

- 『bldapp -- Builds Linux C applications (C)』

## C 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Linux C routines (stored procedures or UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# Determine if we are running with 32-bit, and
# if we are running with 32-bit on Linux AMD64
LIB="lib"
EXTRA_C_FLAGS=""
HARDWAREPLAT=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB="lib32"
    if [ "$HARDWAREPLAT" = "x86_64" ]; then
        EXTRA_C_FLAGS="-m32"
    fi
fi

# Set the runtime path.
```

```

EXTRA_LFLAG="-Wl,-rpath,$DB2PATH/$LIB"

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compile the program.
gcc $EXTRA_C_FLAGS -fpic -I$DB2PATH/include -c $1.c -D_REENTRANT

# Link the program and create a shared library
gcc $EXTRA_C_FLAGS -shared -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2 -lpthread

# Copy the shared library to the function subdirectory.
# The user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

## Linux C 例程编译和链接选项

下列内容是 DB2 建议用于使用 Linux C 编译器构建 C 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>gcc</b>	GNU/Linux C 编译器。
<b>\$EXTRA_C_FLAGS</b>	对于 AMD64 上的 32 位支持，包含 “-m32”；否则不包含值。
<b>-fpic</b>	生成与位置无关的代码。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。
<b>-c</b>	只执行编译；不链接。此脚本文件有单独的编译和链接步骤。
<b>-D_REENTRANT</b>	将定义 _REENTRANT，该选项是必需的，因为例程可能会在其它例程所在的进程中运行（THREADSAFE）或在引擎本身当中运行（NOT FENCED）。

## bldrtn 的编译和链接选项

### 链接选项:

**gcc** 将编译器用作链接程序的前端。

### **\$EXTRA\_CFLAG**

对于 AMD64 上的 32 位支持, 包含 “-m32”; 否则不包含值。

### **-shared**

生成共享库。

**-o \$1** 指定可执行文件。

**\$1.o** 包括该程序的对象文件。

### **\$EXTRA\_LFLAG**

指定运行时 DB2 共享库的位置。对于 32 位, 它包含值 “-Wl,-rpath,\$DB2PATH/lib32”。

对于 64 位, 它包含值 “-Wl,-rpath,\$DB2PATH/lib”。

### **-L\$DB2PATH/\$LIB**

指定链接时 DB2 静态库和共享库的位置。例如, 对于 32 位: \$HOME/sql1lib/lib32, 对于 64 位: \$HOME/sql1lib/lib。如果不指定 -L 选项, 则假定为 /usr/lib/lib。

**-ldb2** 链接 DB2 库。

### **-pthread**

与 POSIX 线程库进行链接。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 144 页的『构建 UNIX C 例程』

### 相关样本:

- 『bldrtn -- Builds Linux C routines (stored procedures or UDFs) (C)』

## 在 Linux 上构建 C 多线程应用程序

使用 Linux C 编写的多线程应用程序需用 `-D_REENTRANT` 编译, 并用 `-pthread` 链接。

脚本 `bldmt` 包含构建多线程应用程序的命令。除了上面指定的选项之外, 其它编译和链接选项与在嵌入式 SQL 脚本文件 `bldapp` 中所用的选项相同。

### 过程:

要从源文件 `dbthrs.sqc` 构建样本程序 `dbthrs`, 请输入:

```
bldmt dbthrs
```

得到一个可执行文件 `dbthrs`。要对 `sample` 数据库运行该可执行文件, 请输入:

```
dbthrs
```

### 相关概念:

- 第 93 页的『构建文件』

### 相关参考:

- 第 62 页的『C 样本』

- 第 204 页的『Linux C 应用程序编译和链接选项』

相关样本:

- 『bldmt -- Builds Linux C multi-threaded applications (C)』
- 『dbthrs.sqc -- How to use multiple context APIs on UNIX (C)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## Linux C++

有关在受支持的 UNIX 操作系统上构建 C++ 应用程序的信息, 请参阅第 147 页的『构建 UNIX C++ 应用程序』。有关在受支持的 UNIX 操作系统上构建 C++ 例程的信息, 请参阅第 151 页的『构建 UNIX C++ 例程』。

### C++ 应用程序的构建脚本

```

#!/bin/sh
# SCRIPT: bldapp
# Builds Linux C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Determine if we are running with 32-bit, and
# if we are running with 32-bit on Linux AMD64
LIB="lib"
EXTRA_C_FLAGS=""
HARDWAREPLAT=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB="lib32"
    if [ "$HARDWAREPLAT" = "x86_64" ]; then
        EXTRA_C_FLAGS="-m32"
    fi
fi

# The runtime path is recommended for all applications.
# If you need to use LD_LIBRARY_PATH, unset the RUNTIME
# variable by commenting out the following line.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath,$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
# Note: some .sqc files contain no SQL but link in
# utilemb.sqc, so if you get this warning, ignore it:
# SQL0053W No SQL statements were found in the program.
if [ -f $1.sqc ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.C error-checking utility.
    g++ $EXTRA_C_FLAGS -I$DB2PATH/include -c utilemb.C
else
    # Compile the utilapi.C error-checking utility.
    g++ $EXTRA_C_FLAGS -I$DB2PATH/include -c utilapi.C

```

```

fi

# Compile the program.
g++ $EXTRA_C_FLAGS -I$DB2PATH/include -c $1.C

if [ -f $1".sqC" ]
then
  # Link the program with utilemb.o
  g++ $EXTRA_C_FLAGS -o $1 $1.o utilemb.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
else
  # Link the program with utilapi.o
  g++ $EXTRA_C_FLAGS -o $1 $1.o utilapi.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
fi

```

## Linux C++ 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Linux C++ 编译器构建 C++ 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示那样。

bldapp 的编译和链接选项	
<b>编译选项:</b>	
<b>g++</b>	GNU/Linux C++ 编译器。
<b>\$EXTRA_C_FLAGS</b>	对于 AMD64 上的 32 位支持，包含“-m32”；否则不包含值。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。
<b>-c</b>	只执行编译；不链接。此脚本文件有单独的编译和链接步骤。
<b>链接选项:</b>	
<b>g++</b>	将编译器用作链接程序的前端。
<b>\$EXTRA_C_FLAGS</b>	对于 AMD64 上的 32 位支持，包含“-m32”；否则不包含值。
<b>-o \$1</b>	指定可执行文件。
<b>\$1.o</b>	包括该程序的对象文件。
<b>utilemb.o</b>	如果是嵌入式 SQL 程序，应包括嵌入式 SQL 实用程序对象文件以便检查错误。
<b>utilapi.o</b>	如果不是嵌入式 SQL 程序，应包括 DB2 API 实用程序对象文件以便检查错误。
<b>\$EXTRA_LFLAG</b>	如果未注释 <code>'RUNTIME=true'</code> ，则对于 32 位，它包含值“-Wl,-rpath,\$DB2PATH/lib32”，对于 64 位，它包含值“-Wl,-rpath,\$DB2PATH/lib”。否则，它不包含值。
<b>-L\$DB2PATH/\$LIB</b>	指定链接时 DB2 静态库和共享库的位置。例如，对于 32 位：\$HOME/sql1lib/lib32，对于 64 位：\$HOME/sql1lib/lib。如果不指定 -L 选项，则假定为 /usr/lib:/lib。
<b>-ldb2</b>	链接 DB2 库。
有关其它编译器选项，参考编译器文档。	

相关任务:

- 第 147 页的『构建 UNIX C++ 应用程序』

相关样本:

- 『bldapp -- Builds Linux C++ applications (C++)』

## C++ 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Linux C++ routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# Determine if we are running with 32-bit, and
# if we are running with 32-bit on Linux AMD64
LIB="lib"
EXTRA_C_FLAGS=""
HARDWAREPLAT=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB="lib32"
    if [ "$HARDWAREPLAT" = "x86_64" ]; then
        EXTRA_C_FLAGS="-m32"
    fi
fi

# Set the runtime path.
EXTRA_LFLAG="-Wl,-rpath,$DB2PATH/$LIB"

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compile the program.
g++ $EXTRA_C_FLAGS -fpic -I$DB2PATH/include -c $1.C -D_REENTRANT

# Link the program and create a shared library.
g++ $EXTRA_C_FLAGS -shared -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2 -lpthread

# Copy the shared library to the function subdirectory.
# The user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

## Linux C++ 例程编译和链接选项

下列内容是 DB2 建议用于使用 Linux C++ 编译器构建 C++ 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
<b>编译选项:</b>	
<b>g++</b>	GNU/Linux C++ 编译器。
<b>\$EXTRA_C_FLAGS</b>	对于 AMD64 上的 32 位支持, 包含 “-m32”; 否则不包含值。
<b>-fpic</b>	生成与位置无关的代码。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。
<b>-c</b>	只执行编译; 不链接。此脚本文件有单独的编译和链接步骤。
<b>-D_REENTRANT</b>	将定义 <code>_REENTRANT</code> , 该选项是必需的, 因为例程可能会在其它例程所在的进程中运行 (THREADSAFE) 或在引擎本身当中运行 (NOT FENCED)。
<b>链接选项:</b>	
<b>g++</b>	将编译器用作链接程序的前端。
<b>\$EXTRA_C_FLAGS</b>	对于 AMD64 上的 32 位支持, 包含 “-m32”; 否则不包含值。
<b>-shared</b>	生成共享库。
<b>-o \$1</b>	指定可执行文件。
<b>\$1.o</b>	包括该程序的对象文件。
<b>\$EXTRA_LFLAG</b>	指定运行时 DB2 共享库的位置。对于 32 位, 它包含值 “-Wl,-rpath,\$DB2PATH/lib32”。对于 64 位, 它包含值 “-Wl,-rpath,\$DB2PATH/lib”。
<b>-\$DB2PATH/\$LIB</b>	指定链接时 DB2 静态库和共享库的位置。例如, 对于 32 位: <code>\$HOME/sql1lib/lib32</code> , 对于 64 位: <code>\$HOME/sql1lib/lib</code> 。如果不指定 <code>-L</code> 选项, 则假定为 <code>/usr/lib:/lib</code> 。
<b>-ldb2</b>	链接 DB2 库。
<b>-lpthread</b>	与 POSIX 线程库进行链接。
有关其它编译器选项, 参考编译器文档。	

#### 相关任务:

- 第 151 页的『构建 UNIX C++ 例程』

#### 相关样本:

- 『bldrtn -- Builds Linux C++ routines (stored procedures and UDFs) (C++)』

## 在 Linux 上构建 C++ 多线程应用程序

使用 Linux C++ 编写的多线程应用程序需要用 `-D_REENTRANT` 编译, 而用 `-lpthread` 链接。

脚本 `bldmt` 包含构建嵌入式 SQL 多线程程序的命令。除了上面指定的选项之外, 其它编译和链接选项与在嵌入式 SQL 脚本文件 `bldapp` 中所用的选项相同。

过程:

要从源文件 dbthrds.sqC 构建样本程序 dbthrds, 请输入:

```
bldmt dbthrds
```

得到一个可执行文件 dbthrds。要对 sample 数据库运行该可执行文件, 请输入:

```
dbthrds
```

相关概念:

- 第 93 页的『构建文件』

相关参考:

- 第 62 页的『C 样本』
- 第 209 页的『Linux C++ 应用程序编译和链接选项』

相关样本:

- 『bldmt -- Builds Linux C++ multi-threaded applications (C++)』
- 『dbthrds.sqC -- How to use multiple context APIs on UNIX (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## Micro Focus COBOL

在 Linux 上, Micro Focus COBOL 只在下列体系结构上受支持:

- **Linux on Intel x86 (32 位)**
- **Linux on s/390**

有关在受支持的 UNIX 操作系统上构建 Micro Focus COBOL 应用程序的信息, 请参阅第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』。有关在受支持的 UNIX 操作系统上构建 Micro Focus COBOL 例程的信息, 请参阅第 155 页的『构建 UNIX Micro Focus COBOL 例程』。

## 在 Linux 上配置 Micro Focus COBOL 编译器

过程:

要运行 Micro Focus COBOL 例程, Linux 运行时链接程序必须能够存取某些 COBOL 共享库并且 DB2 必须能够装入这些库。因为进行此装入操作的程序是以 setuid 特权运行的, 所以它将仅查找 /usr/lib 中的从属库。

为 COBOL 共享库创建指向 /usr/lib 的符号链接。必须以 root 用户的身份完成此操作。最简单的方法是将所有 COBOL 库文件从 \$COBDIR/lib 链接至 /usr/lib:

```
ln -s $COBDIR/lib/libcob* /usr/lib
```

其中 \$COBDIR 是 Micro Focus COBOL 的安装位置, 通常是 /opt/lib/mfcobol。

以下是链接每个文件的命令 (假定 Micro Focus COBOL 安装在 /opt/lib/mfcobol 中):



```

| ln -s /opt/lib/mfcobol/lib/libcobrts.so /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobrts_t.so /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobrts.so.2 /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobrts_t.so.2 /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobcrtn.so /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobcrtn.so.2 /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobmisc.so /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobmisc_t.so /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobmisc.so.2 /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobmisc_t.so.2 /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobscreen.so /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobscreen.so.2 /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobtrace.so /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobtrace_t.so /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobtrace.so.2 /usr/lib
| ln -s /opt/lib/mfcobol/lib/libcobtrace_t.so.2 /usr/lib

```

需对每个 DB2 实例执行下列操作:

- 当使用命令行处理器命令 `db2 prep` 预编译应用程序时, 使用 `target mfcob` 选项。
- 在 Micro Focus COBOL 环境变量 `COBCPY` 中必须包括 DB2 COBOL COPY 文件目录。 `COBCPY` 环境变量指定 COPY 文件的位置。 Micro Focus COBOL 的 DB2 COPY 文件驻留在该数据库实例目录下的 `sqllib/include/cobol_mf` 中。

要包括此目录, 请输入:

- 在 bash 或 Korn shell 上:

```
export COBCPY=$HOME/sqllib/include/cobol_mf:$COBDIR/cpylib
```

- 在 C shell 上:

```
setenv COBCPY $HOME/sqllib/include/cobol_mf:$COBDIR/cpylib
```

- 更新环境变量:

- 在 bash 或 Korn shell 上:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/sqllib/lib:$COBDIR/lib
```

- 在 C shell 上:

```
setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:$HOME/sqllib/lib:$COBDIR/lib
```

- 设置 DB2 环境列表:

```
db2set DB2ENVLIST="COBDIR LD_LIBRARY_PATH"
```

注: 有时您可能想要在 `.bashrc`、`.kshrc` (取决于正在使用的 shell)、`.bash_profile`、`.profile` (取决于正在使用的 shell) 或 `.login` 中设置 `COBCPY`、`COBDIR` 和 `LD_LIBRARY_PATH`。

相关任务:

- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』
- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』

相关参考:

- 第 214 页的『Linux Micro Focus COBOL 应用程序编译和链接选项』
- 第 215 页的『Linux Micro Focus COBOL 例程编译和链接选项』

## Micro Focus COBOL 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Linux Micro Focus COBOL applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
  ./embprep $1 $2 $3 $4
fi

# Compile the checkerr.cbl error checking utility.
cob -cx checkerr.cbl

# Compile the program.
cob -cx $1.cbl

# Link the program.
cob -x -o $1 $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf
```

## Linux Micro Focus COBOL 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Micro Focus COBOL 编译器在 Linux 上构建 COBOL 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

bldapp 的编译和链接选项	
<b>编译选项:</b>	
<b>cob</b>	Micro Focus COBOL 编译器。
<b>-cx</b>	编译为对象模块。
<b>链接选项:</b>	
<b>cob</b>	使用编译器作为链接程序的前端。
<b>-x</b>	指定可执行程序。
<b>-o \$1</b>	包括可执行文件。
<b>\$1.o</b>	包括该程序的对象文件。
<b>checkerr.o</b>	包括该实用程序的对象文件以便检查错误。
<b>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。
<b>-ldb2</b>	链接 DB2 库。
<b>-ldb2gmf</b>	与 Micro Focus COBOL 的 DB2 异常处理程序库链接。
有关其它编译器选项，参阅编译器文档。	

### 相关任务:

- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』

- 第 212 页的『在 Linux 上配置 Micro Focus COBOL 编译器』

相关样本:

- 『bldapp -- Builds Linux Micro Focus COBOL applications』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

## Micro Focus COBOL 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Linux Micro Focus COBOL routines (stored procedures)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Precompile and bind the program.
./embprep $1 $2

# Compile the program.
cob -cx $1.cb1

# Link the program.
cob -x -o $1 $1.o -Q -G -L$DB2PATH/lib -ldb2 -ldb2gmf

# Copy the shared library to the sql1ib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

## Linux Micro Focus COBOL 例程编译和链接选项

下列内容是 DB2 建议用于使用 Micro Focus COBOL 编译器在 Linux 上构建 COBOL 例程（存储过程）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>cob</b>	COBOL 编译器。
<b>-cx</b>	编译为对象模块。

## bldrtn 的编译和链接选项

### 链接选项:

- cob** 使用编译器作为链接程序的前端。
- x** 指定可执行程序。
- o \$1** 指定可执行文件。
- \$1.o** 包括该程序的对象文件。
- Q -G** 生成共享库。
- L\$DB2PATH/1ib**  
指定 DB2 运行时共享库的位置。
- ldb2** 链接 DB2 库。
- ldb2gmf**  
与 Micro Focus COBOL 的 DB2 异常处理程序库链接。

有关其它编译器选项, 参阅编译器文档。

### 相关任务:

- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』
- 第 212 页的『在 Linux 上配置 Micro Focus COBOL 编译器』

### 相关样本:

- 『bldrtn -- Builds Linux Micro Focus COBOL routines (stored procedures)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## 第 13 章 Solaris

Solaris C . . . . .	217	Solaris C++ 例程编译和链接选项 . . . . .	225
C 应用程序的构建脚本 . . . . .	217	在 Solaris 上构建 C++ 多线程应用程序 . . . . .	226
Solaris C 应用程序编译和链接选项 . . . . .	218	Micro Focus COBOL . . . . .	227
C 例程的构建脚本 . . . . .	219	在 Solaris 上配置 Micro Focus COBOL 编译器 . . . . .	227
Solaris C 例程编译和链接选项 . . . . .	220	Micro Focus COBOL 应用程序的构建脚本 . . . . .	228
在 Solaris 上构建 C 多线程应用程序 . . . . .	221	Solaris Micro Focus COBOL 应用程序编译和链接选项 . . . . .	228
Solaris C++ . . . . .	222	Micro Focus COBOL 例程的构建脚本 . . . . .	229
C++ 应用程序的构建脚本 . . . . .	222	Solaris Micro Focus COBOL 例程编译和链接选项 . . . . .	230
Solaris C++ 应用程序编译和链接选项 . . . . .	223		
C++ 例程的构建脚本 . . . . .	224		

本章提供在 Solaris Operating Environment 中构建应用程序的详细信息。有关 Solaris 的最新 DB2 应用程序开发情况的更新信息，请访问以下的 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad>

---

## Solaris C

DB2 CLI 应用程序和例程的构建信息在 *CLI Guide and Reference* 中。

有关在受支持的 UNIX 操作系统上构建 C 应用程序的信息，请参阅第 141 页的『构建 UNIX C 应用程序』。有关在受支持的 UNIX 操作系统上构建 C 例程的信息，请参阅第 144 页的『构建 UNIX C 例程』。

### C 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Solaris C applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

# Set compile and link flags for 32-bit and 64-bit programs.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"64\"" ];
then
  CFLAG_ARCH=v9
  LIB=lib
else
  CFLAG_ARCH=v8plusa
  LIB=lib32
fi

# Set the runtime path.
# LD_LIBRARY_PATH will be followed instead of the runtime path unless
# you unset LD_LIBRARY_PATH first to allow the runtime path to be used.
EXTRA_LFLAG="-R$DB2PATH/$LIB"

# If an embedded SQL program, precompile and bind it.
# Note: some .sqc files contain no SQL but link in
# utilemb.sqc, so if you get this warning, ignore it:
# SQL0053W No SQL statements were found in the program.
if [ -f $1".sqc" ]
```

```

then
  ./embprep $1 $2 $3 $4
  # Compile the utilemb.c error-checking utility.
  cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilemb.c
else
  # Compile the utilapi.c error-checking utility.
  cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
  # Link the program with utilemb.o
  cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilemb.o \
    -L$DB2PATH/$LIB $EXTRA_LFLAG -ldb2
else
  # Link the program with utilapi.o
  cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilapi.o \
    -L$DB2PATH/$LIB $EXTRA_LFLAG -ldb2
fi

```

## Solaris C 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Forte C 编译器构建 C 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

<b>bldapp 的编译和链接选项</b>	
<b>编译选项:</b>	
<b>cc</b>	C 编译器。
<b>-xarch=\$CFLAG_ARCH</b>	此选项确保编译器在使用 libdb2.so 进行链接时将生成有效的可执行文件。 \$CFLAG_ARCH 的值设置为 “v8plusa”（对于 32 位）或 “v9”（对于 64 位）。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。例如: \$HOME/sqllib/include
<b>-c</b>	只执行编译；不链接。此脚本具有独立的编译和链接步骤。

## bldapp 的编译和链接选项

### 链接选项:

**cc** 将编译器用作链接程序的前端。

### **-xarch=\$CFLAG\_ARCH**

此选项确保编译器在使用 `libdb2.so` 进行链接时将生成有效的可执行文件。  
`$CFLAG_ARCH` 的值设置为 “v8plusa” (对于 32 位) 或 “v9” (对于 64 位)。

**-mt** 多线程支持中的链接。它是使用 `libdb2` 进行链接所需要的。

**注:** 如果使用了 POSIX 线程, 则 DB2 应用程序还必须使用 `-lpthread` 来进行链接, 而不论它们是否为线程应用程序。

**-o \$1** 指定可执行文件。

**\$1.o** 包括该程序的对象文件。

### **utilemb.o**

如果是嵌入式 SQL 程序, 应包括嵌入式 SQL 实用程序对象文件以便检查错误。

### **utilapi.o**

如果不是嵌入式 SQL 程序, 应包括 DB2 API 实用程序对象文件以便检查错误。

### **-L\$DB2PATH/\$LIB**

指定链接时 DB2 静态库和共享库的位置。例如, 对于 32 位: `$HOME/sql1lib/lib32`,  
对于 64 位: `$HOME/sql1lib/lib`。

### **\$EXTRA\_LFLAG**

指定运行时 DB2 共享库的位置。对于 32 位, 它包含值 “`-R$DB2PATH/lib32`”, 对于  
64 位, 它包含值 “`-R$DB2PATH/lib`”。

**-ldb2** 链接 DB2 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 141 页的『构建 UNIX C 应用程序』

### 相关样本:

- 『`bldapp -- Builds Solaris C applications (C)`』

## C 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Solaris C routines (stored procedures or UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# Set compile and link flags for 32-bit and 64-bit programs.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"64\"" ];
then
    CFLAG_ARCH=v9
    LIB=lib
else
    CFLAG_ARCH=v8plusa
    LIB=lib32
```

```

fi

# Set the runtime path.
# LD_LIBRARY_PATH will be followed instead of the runtime path unless
# you unset LD_LIBRARY_PATH first to allow the runtime path to be used.
EXTRA_LFLAG="-R$DB2PATH/$LIB"

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compile the program.
cc -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -Kpic \
-I$DB2PATH/include -c $1.c

# Link the program and create a shared library
cc -xarch=$CFLAG_ARCH -mt -G -o $1 $1.o -L$DB2PATH/$LIB \
$EXTRA_LFLAG -Tdb2

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

## Solaris C 例程编译和链接选项

下列内容是 DB2 建议用于使用 Forte C 编译器构建 C 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
<b>编译选项:</b>	
<b>cc</b>	C 编译器。
<b>-xarch=\$CFLAG_ARCH</b>	此选项确保编译器在使用 libdb2.so 进行链接时将生成有效的可执行文件。 \$CFLAG_ARCH 的值设置为 “v8plusa”（对于 32 位）或 “v9”（对于 64 位）。
<b>-mt</b>	允许多线程支持，该选项是必需的，因为例程可能会在其它例程所在的进程中运行（THREADSAFE）或在引擎本身当中运行（NOT FENCED）。
<b>-DUSE_UI_THREADS</b>	允许使用 Sun 公司的 “UNIX International” 线程 API。
<b>-Kpic</b>	为共享库生成与位置无关的代码。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。
<b>-c</b>	只执行编译；不链接。此脚本文件有单独的编译和链接步骤。



## bldrtn 的编译和链接选项

### 链接选项:

**cc** 将编译器用作链接程序的前端。

### **-xarch=\$CFLAG\_ARCH**

此选项确保编译器在使用 `libdb2.so` 进行链接时将生成有效的可执行文件。  
`$CFLAG_ARCH` 的值设置为 “v8plusa” (对于 32 位) 或 “v9” (对于 64 位)。

**-mt** 这是必需的, 因为 DB2 库是使用 `-mt` 进行链接的。

**-G** 生成共享库。

**-o \$1** 指定可执行文件。

**\$1.o** 包括该程序的对象文件。

### **-L\$DB2PATH/\$LIB**

指定链接时 DB2 静态库和共享库的位置。例如, 对于 32 位: `$HOME/sql1lib/lib32`,  
对于 64 位: `$HOME/sql1lib/lib`。

### **\$EXTRA\_LFLAG**

指定运行时 DB2 共享库的位置。对于 32 位, 它包含值 “`-R$DB2PATH/lib32`”, 对于  
64 位, 它包含值 “`-R$DB2PATH/lib`”。

**-ldb2** 链接 DB2 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 144 页的『构建 UNIX C 例程』

### 相关样本:

- 『`bldrtn -- Builds Solaris C routines (stored procedures or UDFs) (C)`』

## 在 Solaris 上构建 C 多线程应用程序

DB2 支持使用 SUN 和 POSIX 线程库的多线程应用程序。缺省值为 Sun 线程。在 Solaris 上使用 Forte C 的多线程应用程序需要使用 `-mt` 来编译和链接。它将 `-D_REENTRANT` 传送至预处理器, 并将 `-lthread` 传送至链接程序。您还需要指定编译定义 `-DUSE_UI_THREADS` 以使用 Sun 公司的 “Unix International” 线程 API。

**注:** 如果想要使用 POSIX 线程, 则必须添加编译器选项 `-D_POSIX_PTHREAD_SEMANTICS` (它允许函数 (如 `getpwnam_r()`) 的 POSIX 变体), 且还需添加链接选项 `-lpthread`。如果要使用提供的 `blgmt` 脚本, 则还必须删除 `-DUSE_UI_THREADS` 定义。

脚本 `blgmt` 包含构建多线程应用程序的命令。除了上面指定的选项之外, 其它编译和链接选项与在嵌入式 SQL 脚本文件 `blgdapp` 中所用的选项相同。

### 过程:

要从源文件 `dbthdrs.sqc` 构建样本程序 `dbthdrs`, 请输入:

```
blgmt dbthdrs
```

得到一个可执行文件 `dbthdrs`。要对 `sample` 数据库运行该可执行文件, 请输入:

dbthrds

注: 对于具有相当数量的连接的多线程程序, 除了其缺省值之外, 可能还必须设置内核参数 `semsys:seminfo_semume` 和 `shmsys:shminfo_shmseg`。请参阅下面有关 `db2osconf` 实用程序的相关链接以获取有关为这些参数设置的值的建议。

相关概念:

- 第 93 页的『构建文件』

相关参考:

- 第 62 页的『C 样本』
- 第 218 页的『Solaris C 应用程序编译和链接选项』
- 『db2osconf - Utility for Kernel Parameter Values Command』 (*Command Reference*)

相关样本:

- 『bldmt -- Builds Solaris C multi-threaded applications (C)』
- 『dbthrds.sqc -- How to use multiple context APIs on UNIX (C)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## Solaris C++

有关在受支持的 UNIX 操作系统上构建 C++ 应用程序的信息, 请参阅第 147 页的『构建 UNIX C++ 应用程序』。有关在受支持的 UNIX 操作系统上构建 C++ 例程的信息, 请参阅第 151 页的『构建 UNIX C++ 例程』。

### C++ 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Solaris C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set compile and link flags for 32-bit and 64-bit programs.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"64\"" ];
then
    CFLAG_ARCH=v9
    LIB=lib
else
    CFLAG_ARCH=v8plusa
    LIB=lib32
fi

# Set the runtime path.
# LD_LIBRARY_PATH will be followed instead of the runtime path unless
# you unset LD_LIBRARY_PATH first to allow the runtime path to be used.
EXTRA_LFLAG="-R$DB2PATH/$LIB"

# If an embedded SQL program, precompile and bind it.
# Note: some .sqc files contain no SQL but link in
# utilemb.sqc, so if you get this warning, ignore it:
# SQL0053W No SQL statements were found in the program.
```

```

if [ -f $1".sqC" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.C error-checking utility.
    CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilemb.C
else
    # Compile the utilapi.C error-checking utility.
    CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilapi.C
fi

# Compile the program.
CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c $1.C

if [ -f $1".sqC" ]
then
    # Link the program with utilemb.o
    CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilemb.o \
        -L$DB2PATH/$LIB $EXTRA_LFLAG -ldb2
else
    # Link the program with utilapi.o
    CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilapi.o \
        -L$DB2PATH/$LIB $EXTRA_LFLAG -ldb2
fi

```

## Solaris C++ 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Forte C++ 编译器构建 C++ 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

<b>bldapp 的编译和链接选项</b>	
<b>编译选项:</b>	
<b>CC</b>	C++ 编译器。
<b>-xarch=\$CFLAG_ARCH</b>	此选项确保编译器在使用 libdb2.so 进行链接时将生成有效的可执行文件。 \$CFLAG_ARCH 的值设置为 “v8plusa”（对于 32 位）或 “v9”（对于 64 位）。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。例如: \$HOME/sql1lib/include
<b>-c</b>	只执行编译；不链接。此脚本具有独立的编译和链接步骤。

## bldapp 的编译和链接选项

### 链接选项:

**CC** 将编译器用作链接程序的前端。

### **-xarch=\$CFLAG\_ARCH**

此选项确保编译器在使用 `libdb2.so` 进行链接时将生成有效的可执行文件。  
`$CFLAG_ARCH` 的值设置为 “v8plusa” (对于 32 位) 或 “v9” (对于 64 位)。

**-mt** 多线程支持中的链接。它是使用 `libdb2` 进行链接所需要的。

**注:** 如果使用了 POSIX 线程, 则 DB2 应用程序还必须使用 `-lpthread` 来进行链接, 而不论它们是否为线程应用程序。

**-o \$1** 指定可执行文件。

**\$1.o** 包括该程序的对象文件。

### **utilemb.o**

如果是嵌入式 SQL 程序, 应包括嵌入式 SQL 实用程序对象文件以便检查错误。

### **utilapi.o**

如果不是嵌入式 SQL 程序, 应包括 DB2 API 实用程序对象文件以便检查错误。

### **-L\$DB2PATH/\$LIB**

指定链接时 DB2 静态库和共享库的位置。例如, 对于 32 位: `$HOME/sqllib/lib32`, 对于 64 位: `$HOME/sqllib/lib`。

### **\$EXTRA\_LFLAG**

指定运行时 DB2 共享库的位置。对于 32 位, 它包含值 “`-R$DB2PATH/lib32`”, 对于 64 位, 它包含值 “`-R$DB2PATH/lib`”。

**-ldb2** 链接 DB2 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 147 页的『构建 UNIX C++ 应用程序』

### 相关样本:

- 『`bldapp -- Builds Solaris C++ applications (C++)`』

## C++ 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Solaris C++ routines (stored procedures or UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set compile and link flags for 32-bit and 64-bit programs.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"64\"" ];
then
    CFLAG_ARCH=v9
    LIB=lib
else
    CFLAG_ARCH=v8plusa
    LIB=lib32
```

```

fi

# Set the runtime path.
# LD_LIBRARY_PATH will be followed instead of the runtime path unless
# you unset LD_LIBRARY_PATH first to allow the runtime path to be used.
EXTRA_LFLAG="-R$DB2PATH/$LIB"

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compile the program.
CC -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -Kpic \
-I$DB2PATH/include -c $1.C

# Link the program and create a shared library
CC -xarch=$CFLAG_ARCH -mt -G -o $1 $1.o -L$DB2PATH/$LIB \
$EXTRA_LFLAG -Tdb2

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

## Solaris C++ 例程编译和链接选项

下列内容是 DB2 建议用于使用 Forte C++ 编译器构建 C++ 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
<b>编译选项:</b>	
<b>CC</b>	C++ 编译器。
<b>-xarch=\$CFLAG_ARCH</b>	此选项确保编译器在使用 libdb2.so 进行链接时将生成有效的可执行文件。 \$CFLAG_ARCH 的值设置为 “v8plusa”（对于 32 位）或 “v9”（对于 64 位）。
<b>-mt</b>	允许多线程支持，该选项是必需的，因为例程可能会在其它例程所在的进程中运行（THREADSAFE）或在引擎本身当中运行（NOT FENCED）。
<b>-DUSE_UI_THREADS</b>	允许使用 Sun 公司的 “UNIX International” 线程 API。
<b>-Kpic</b>	为共享库生成与位置无关的代码。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。
<b>-c</b>	只执行编译；不链接。此脚本具有独立的编译和链接步骤。

## bldrtn 的编译和链接选项

### 链接选项:

**CC** 将编译器用作链接程序的前端。

### **-xarch=\$CFLAG\_ARCH**

此选项确保编译器在使用 `libdb2.so` 进行链接时将生成有效的可执行文件。  
`$CFLAG_ARCH` 的值设置为 “v8plusa” (对于 32 位) 或 “v9” (对于 64 位)。

**-mt** 这是必需的, 因为 DB2 库是使用 `-mt` 进行链接的。

**-G** 生成共享库。

**-o \$1** 指定可执行文件。

**\$1.o** 包括该程序的对象文件。

### **-L\$DB2PATH/\$LIB**

指定链接时 DB2 静态库和共享库的位置。例如, 对于 32 位: `$HOME/sql1lib/lib32`,  
对于 64 位: `$HOME/sql1lib/lib`。

### **\$EXTRA\_LFLAG**

指定运行时 DB2 共享库的位置。对于 32 位, 它包含值 “`-R$DB2PATH/lib32`”, 对于  
64 位, 它包含值 “`-R$DB2PATH/lib`”。

**-ldb2** 链接 DB2 库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 151 页的『构建 UNIX C++ 例程』

### 相关样本:

- 『`bldrtn -- Builds Solaris C++ routines (stored procedures or UDFs) (C++)`』

## 在 Solaris 上构建 C++ 多线程应用程序

DB2 支持使用 SUN 和 POSIX 线程库的多线程应用程序。缺省值为 Sun 线程。在 Solaris 上使用 Forte C++ 的多线程应用程序需要使用 `-mt` 来编译和链接。它将 `-D_REENTRANT` 传送至预处理器, 并将 `-lthread` 传送至链接程序。您还需要指定编译定义 `-DUSE_UI_THREADS` 以使用 Sun 公司的 “Unix International” 线程 API。

**注:** 如果想要使用 POSIX 线程, 则必须添加编译器选项 `-D_POSIX_PTHREAD_SEMANTICS` (它允许函数 (如 `getpwnam_r()`) 的 POSIX 变体), 且还需添加链接选项 `-lpthread`。如果要使用提供的 `bldmt` 脚本, 则还必须删除 `-DUSE_UI_THREADS` 定义。

脚本 `bldmt` 包含构建多线程应用程序的命令。除了上面指定的选项之外, 其它编译和链接选项与在嵌入式 SQL 脚本文件 `bldapp` 中所用的选项相同。

### 过程:

要从源文件 `dbthrds.sqC` 构建样本程序 `dbthrds`, 请输入:

```
bldmt dbthrds
```

得到一个可执行文件 `dbthrds`。要对 `sample` 数据库运行该可执行文件, 请输入:

dbthrds

注：对于具有相当数量的连接的多线程程序，除了其缺省值之外，可能还必须设置内核参数 `semsys:seminfo_semume` 和 `shmsys:shminfo_shmseg`。请参阅下面有关 `db2osconf` 实用程序的相关链接以获取有关为这些参数设置的值的建议。

相关概念：

- 第 93 页的『构建文件』

相关参考：

- 第 62 页的『C 样本』
- 第 223 页的『Solaris C++ 应用程序编译和链接选项』
- 『db2osconf - Utility for Kernel Parameter Values Command』 (*Command Reference*)

相关样本：

- 『bldmt -- Builds Solaris C++ multi-threaded applications (C++)』
- 『dbthrds.sqC -- How to use multiple context APIs on UNIX (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)』

---

## Micro Focus COBOL

有关在受支持的 UNIX 操作系统上构建 Micro Focus COBOL 应用程序的信息，请参阅第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』。有关在受支持的 UNIX 操作系统上构建 Micro Focus COBOL 例程的信息，请参阅第 155 页的『构建 UNIX Micro Focus COBOL 例程』。

### 在 Solaris 上配置 Micro Focus COBOL 编译器

如果开发包含嵌入式 SQL 和 DB2 API 调用的应用程序，且正在使用 Micro Focus COBOL 编译器，则必须注意以下几点。

过程：

- 当使用命令行处理器命令 `db2 prep` 预编译应用程序时，使用 `target mfcob` 选项。
- 在 Micro Focus COBOL 环境变量 `COBCPY` 中必须包括 DB2 COBOL COPY 文件目录。`COBCPY` 环境变量指定 COPY 文件的位置。Micro Focus COBOL 的 DB2 COPY 文件驻留在该数据库实例目录下的 `sqllib/include/cobol_mf` 中。

要包括此目录，请输入：

– 在 bash 或 Korn shell 上：

```
export COBCPY=$COBCPY:$HOME/sqllib/include/cobol_mf
```

– 在 C shell 上：

```
setenv COBCPY $COBCPY:$HOME/sqllib/include/cobol_mf
```

注：应在 `.profile` 文件中设置 `COBCPY`。

相关任务：

- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』
- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』

### 相关参考:

- 第 228 页的『Solaris Micro Focus COBOL 应用程序编译和链接选项』
- 第 230 页的『Solaris Micro Focus COBOL 例程编译和链接选项』

## Micro Focus COBOL 应用程序的构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Solaris Micro Focus COBOL applications
# Usage: bldapp [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
  ./embprep $1 $2 $3 $4
fi

# Compile the checkerr.cbl error-checking utility.
cob -cx checkerr.cbl

# Compile the program.
cob -cx $1.cbl

# Link the program.
cob -x $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf
```

## Solaris Micro Focus COBOL 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Micro Focus COBOL 编译器在 Solaris 上构建 COBOL 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp 构建脚本中所演示的那样。

bldapp 的编译和链接选项	
编译选项:	
<b>cob</b>	Micro Focus COBOL 编译器。
<b>-cx</b>	编译为对象模块。



## bldapp 的编译和链接选项

### 链接选项:

- cob** 将编译器用作链接程序的前端。
- x** 指定可执行程序。
- \$1.o** 包括该程序的对象文件。
- checkerr.o**  
包括该实用程序的对象文件以便检查错误。
- L\$DB2PATH/lib**  
指定链接时 DB2 静态库和共享库的位置。例如: \$HOME/sqllib/lib。
- ldb2** 链接 DB2 库。
- ldb2gmf**  
链接 Micro Focus COBOL 的 DB2 异常处理程序库。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 154 页的『构建 UNIX Micro Focus COBOL 应用程序』

### 相关样本:

- 『bldapp -- Builds Solaris Micro Focus COBOL applications』

## Micro Focus COBOL 例程的构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Solaris Micro Focus COBOL routines (stored procedures)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
./embprep $1 $2
fi

# Compile the program.
cob -cx $1.cbl

# Link the program.
cob -yo $1 $1.o -L$DB2PATH/lib -ldb2 -ldb2gmf

# Copy the shared library to the sqllib/function subdirectory.
# The user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

## Solaris Micro Focus COBOL 例程编译和链接选项

下列内容是 DB2 建议用于使用 Micro Focus COBOL 编译器在 Solaris 上构建 COBOL 例程（存储过程）的编译和链接选项，如在 bldrtn 构建脚本中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>cob</b>	COBOL 编译器。
<b>-cx</b>	编译为对象模块。
链接选项:	
<b>cob</b>	将编译器用作链接程序的前端。
<b>-y</b>	创建独立的共享库。
<b>-o \$1</b>	指定可执行程序。
<b>\$1.o</b>	指定该程序的对象文件。
<b>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。例如: \$HOME/sql1lib/lib。如果不指定 -L 选项, 则编译器假定如下路径: /usr/lib:/lib。
<b>-ldb2</b>	链接 DB2 库。
<b>-ldb2gmf</b>	与 Micro Focus COBOL 的 DB2 异常处理程序库链接。
有关其它编译器选项, 参考编译器文档。	

### 相关任务:

- 第 155 页的『构建 UNIX Micro Focus COBOL 例程』

### 相关样本:

- 『bldrtn -- Builds Solaris Micro Focus COBOL routines (stored procedures)』

## 第 14 章 Windows

WCHARTYPE CONVERT 预编译选项 . . . . .	231	Windows C/C++ 应用程序编译和链接选项 . . . . .	256
对象链接与嵌入数据库 (OLE DB) 表函数 . . . . .	232	在 Windows 上构建 C/C++ 例程 . . . . .	257
Windows 管理规范 (WMI) . . . . .	233	C/C++ 例程的批处理文件 . . . . .	260
Microsoft Visual Basic . . . . .	233	Windows C/C++ 例程编译和链接选项 . . . . .	261
使用 Visual Basic 构建 ADO 应用程序 . . . . .	233	在 Windows 上构建 C/C++ 多连接应用程序 . . . . .	262
使用 Visual Basic 构建松散耦合事务 . . . . .	236	IBM VisualAge COBOL . . . . .	264
Visual Basic 松散耦合事务项目故障诊断 . . . . .	237	在 Windows 上配置 IBM COBOL 编译器 . . . . .	264
使用 Visual Basic 构建 RDO 应用程序 . . . . .	238	在 Windows 上构建 IBM COBOL 应用程序 . . . . .	264
使用 Visual Basic 的对象链接与嵌入 (OLE) 自 动化 . . . . .	240	IBM COBOL 应用程序的批处理文件 . . . . .	266
.NET . . . . .	240	Windows IBM COBOL 应用程序编译和链接选项 . . . . .	267
构建 C# .NET 应用程序 . . . . .	240	在 Windows 上构建 IBM COBOL 例程 . . . . .	268
C# .NET 应用程序的批处理文件 . . . . .	241	IBM COBOL 例程的批处理文件 . . . . .	269
C# .NET 应用程序编译和链接选项 . . . . .	242	Windows IBM COBOL 例程编译和链接选项 . . . . .	270
构建 Visual Basic .NET 应用程序 . . . . .	243	Micro Focus COBOL . . . . .	271
Visual Basic .NET 应用程序的批处理文件 . . . . .	244	在 Windows 上配置 Micro Focus COBOL 编译 器 . . . . .	271
Visual Basic .NET 应用程序编译和链接选项 . . . . .	245	在 Windows 上构建 Micro Focus COBOL 应用 程序 . . . . .	272
构建公共语言运行时 (CLR) .NET 例程 . . . . .	246	Micro Focus COBOL 应用程序的批处理文件 . . . . .	273
C# .NET 例程的批处理文件 . . . . .	249	Windows Micro Focus COBOL 应用程序编译和 链接选项 . . . . .	273
Visual Basic .NET 例程的批处理文件 . . . . .	250	在 Windows 上构建 Micro Focus COBOL 例程 . . . . .	274
CLR .NET 例程编译和链接选项 . . . . .	250	Micro Focus COBOL 例程的批处理文件 . . . . .	275
Microsoft Visual C++ . . . . .	251	Windows Micro Focus COBOL 例程编译和链接 选项 . . . . .	276
使用 Visual C++ 构建 ADO 应用程序 . . . . .	251	Object REXX . . . . .	276
使用 Visual C++ 的对象链接与嵌入 (OLE) 自动 化 . . . . .	253	在 Windows 上构建 Object REXX 应用程序 . . . . .	276
在 Windows 上构建 C/C++ 应用程序 . . . . .	254		
C/C++ 应用程序的批处理文件 . . . . .	256		

本章提供了在 Windows 操作系统上构建应用程序的详细信息。有关 Windows 的最新 DB2 应用程序开发情况的更新信息，请访问以下的 Web 页面：

<http://www.ibm.com/software/data/db2/udb/ad>

### WCHARTYPE CONVERT 预编译选项

WCHARTYPE 预编译选项确定是采用多字节格式还是使用 `wchar_t` 数据类型的宽位字符格式处理图形数据。

对于 DB2® Windows® 版操作系统，使用 Microsoft® Visual C++ 编译器编译的应用程序支持 WCHARTYPE CONVERT 选项。但是，如果您的应用程序将数据插入一个 DB2 数据库，而数据所用的代码页与该数据库所用的代码页不同，则在使用此编译器时不要选择 CONVERT 选项。在这种情况下 DB2 通常会执行代码页转换；但 Microsoft C 运行时环境不处理特定双字节字符的字符替代。这可能导致运行时转换错误。

WCHARTYPE 的缺省选项为 NOCONVERT。通过使用 NOCONVERT 选项，应用程序和数据库管理器之间就不会发生隐式字符转换。图形主机变量中的数据作为未改变的“双字节字符集” (DBCS) 字符发送到数据库管理器并从数据库管理器接收。

如果要将图形数据由宽位字符格式转换到多字节格式，则使用 `wcstombs()` 函数。例如：

```
wchar_t widechar[200];
wchar_t mb[200];
wcstombs((char *)mb,widechar,200);

EXEC SQL INSERTINTO TABLENAME VALUES(:mb);
```

类似地，可使用 `mbstowcs()` 函数从多字节格式转换到宽位字符格式。

如果您的应用程序与 C 运行时库静态绑定，则不要从应用程序发出 `setlocale()` 调用，因为这可能导致 C 运行时转换错误。如果应用程序与 C 运行时库动态绑定，则可使用 `setlocale()`。对于例程（存储过程和用户定义的函数），情况也是如此。

相关概念：

- 『`wchar_t` and `sqldbcchar` Data Types in C and C++』 (*Application Development Guide: Programming Client Applications*)
- 『`WCHARTYPE` Precompiler Option in C and C++』 (*Application Development Guide: Programming Client Applications*)
- 『Graphic host variables in C/C++ routines』 (*Application Development Guide: Programming Server Applications*)

相关参考：

- 『`PRECOMPILE` Command』 (*Command Reference*)

---

## 对象链接与嵌入数据库 (OLE DB) 表函数

DB2® 支持 OLE DB 表函数。对于这些函数，除创建 `CREATE FUNCTION DDL` 外不需要进行应用程序构建。DB2 在 `sqllib\samples\oledb` 目录中提供了 OLE DB 表函数样本文件。它们是“命令行处理器” (CLP) 文件。可用下列步骤构建它们：

1. `db2 connect to database_name`
2. `db2 -t -v -f file_name.db2`
3. `db2 terminate`

其中 `database_name` 是要连接的数据库，而 `file_name` 是 CLP 文件的名称，扩展名为 `.db2`。

必须在“DB2 命令”窗口中完成这些命令。

相关概念：

- 『OLE DB Table Functions』 (*Application Development Guide: Programming Client Applications*)
- 『OLE DB user-defined table functions』 (*Application Development Guide: Programming Server Applications*)

相关参考：

- 第 83 页的『对象链接与嵌入数据库 (OLE DB) 表函数样本』

---

## Windows 管理规范 (WMI)

Windows<sup>®</sup> 管理规范 (WMI) 是 Microsoft<sup>®</sup> 的 Windows 管理服务的一个关键组件。WMI 提供了配置、状态以及应用程序和系统各操作方面的一致且充分描述了模型。

DB2<sup>®</sup> WMI 提供程序允许 WMI 应用程序监视 DB2 服务器服务、枚举和创建数据库、配置操作设置以及执行数据库备份、复原和前滚操作。

DB2 为位于 `sqllib\samples\wmi` 目录中的“Visual Basic 脚本语言”提供了 WMI 样本文件。在运行样本程序之前，应确保通过运行下列命令注册了“DB2 WMI 提供程序”：

```
mofcomp %DB2PATH%\bin\db2wmi.mof
regsvr32 %DB2PATH%\bin\db2wmi.dll
```

其中 `%DB2PATH%` 是 DB2 的安装路径。

使用 `cscript` 命令来运行“Visual Basic 脚本”样本。例如，要运行 `listsvr` 样本脚本，请输入：

```
cscript listsvr.vbs
```

相关概念：

- 『Windows 管理规范 (WMI) 简介』 (《管理指南: 实现》)
- 『DB2 通用数据库与 Windows 管理规范集成』 (《管理指南: 实现》)

相关参考：

- 第 92 页的『Windows 管理规范样本』

---

## Microsoft Visual Basic

### 使用 Visual Basic 构建 ADO 应用程序

“ActiveX 数据对象” (ADO) 用来编写一个应用程序，以便通过 OLE DB 提供程序存取和处理数据库服务器中的数据。ADO 的主要优点是高速、易用、内存开销较低以及占用磁盘较少。

Visual Basic ADO 样本程序位于 `sqllib\samples\VB\ADO` 目录中。

注：要运行 DB2 ADO 样本，建议使用下列组件的这些版本或更新版本：

1. Visual Basic 6.0 专业版
2. Microsoft Data Access 2.7 SDK (可选择地使用 DB2 版本 8 安装)
3. Visual Basic 服务包 5，网址为：  
<http://msdn.microsoft.com/vstudio/sp/vs6sp5/vbfixes.asp>
4. 最新 Visual Studio 服务包，网址为：<http://msdn.microsoft.com/vstudio/>

过程：

可以使用两个与 ODBC 兼容的提供程序之一：

- IBM OLE DB provider for DB2
- Microsoft OLE DB provider for ODBC

## 使用 IBM OLE DB provider for DB2

Windows 操作系统上的 DB2 版本 8.2 客户机将可选择安装 IBMDADB2 (即 IBM OLE DB 2.0-compliant provider for DB2)。该提供程序为希望存取 DB2 数据库中数据的客户提供接口。IBM OLE DB provider for DB2 支持下列 ADO 应用程序类型:

- Microsoft Active Server Pages (ASP)
- Microsoft Visual Studio C++ 和 Visual Basic 应用程序
- Microsoft Visual Interdev

有关这些类型的应用程序的详细信息, 请参考 ADO 文档。

要使用 IBM OLEDB provider for DB2 访问 DB2 服务器, Visual Basic 应用程序应按如下所示在 ADO 连接字符串中指定 PROVIDER 关键字:

```
Dim c1 As ADODB.Connection
Dim c1str As String
c1str = "Provider=ibmdadb2; DSN=db2alias; UID=userid; PWD=password"
c1.Open c1str
...
```

其中 db2alias 是在 DB2 数据库目录中编目的 DB2 数据库的别名。

注: 当使用 IBM OLE DB provider for DB2 时, 不需要对数据源执行 ODBC 编目步骤。当使用 OLE DB provider for ODBC 时, 此步骤是必需的。

## 使用 Microsoft OLE DB provider for ODBC

要通过 Microsoft OLE DB 提供程序和 Visual Basic 来使用 ADO, 需要建立对 ADO 类型库的引用。执行下列各项操作:

1. 从“项目”菜单选择“引用”
2. 选择“Microsoft ActiveX 数据对象 <version\_number> Library”的复选框
3. 单击“确定”。

其中 <version\_number> 是 ADO 库的当前版本。

一旦完成此操作, 则可通过“VBA 对象浏览器”和“IDE 编辑器”存取 ADO 对象、方法和属性。

建立连接:

```
Dim db As Connection
Set db = New Connection
```

设置本地游标库提供的客户端游标:

```
db.CursorLocation = adUseClient
```

并设置提供程序以便 ADO 将使用 Microsoft ODBC 驱动程序。

## 使用 ADO 存取样本数据库

一个完整的 Visual Basic 程序包括表单和其它图形元素, 您需要在 Visual Basic 环境中查看它。以下是在用上述的 IBM OLE DB 提供程序或 Microsoft OLE DB 提供程序连接数据库之后, 存取 DB2 sample 数据库的、作为程序一部分的 Visual Basic 命令。

不指定用户标识或密码打开 sample 数据库; 也就是使用当前用户:

```
db.Open "SAMPLE"
```

创建一个记录集:

```
Set adoPrimaryRS = New Recordset
```

使用 select 语句填充该记录集:

```
adoPrimaryRS.Open "select EMPNO, LASTNAME, FIRSTNAME, MIDINIT, EDLEVEL, JOB  
from EMPLOYEE Order by EMPNO", db
```

此时, 程序员可使用 ADO 方法存取数据, 如移动到下一个记录集:

```
adoPrimaryRS.MoveNext
```

删除记录集中的当前记录:

```
adoPrimaryRS.Delete
```

程序员还可执行下列操作来存取个别字段:

```
Dim Text1 as String  
Text1 = adoPrimaryRS!LASTNAME
```

相关概念:

- 『 Purpose of the IBM OLE DB Provider for DB2 』 ( *Application Development Guide: Programming Client Applications* )
- 『 Application Types Supported by the IBM OLE DB Provider for DB2 』 ( *Application Development Guide: Programming Client Applications* )
- 『 Connections to Data Sources with Visual Basic ADO Applications 』 ( *Application Development Guide: Programming Client Applications* )
- 『 OLE DB Services Automatically Enabled by IBM OLE DB Provider 』 ( *Application Development Guide: Programming Client Applications* )
- 『 Large Object Manipulation with the IBM OLE DB Provider 』 ( *Application Development Guide: Programming Client Applications* )
- 『 MTS and COM+ Distributed Transaction Support and the IBM OLE DB Provider 』 ( *Application Development Guide: Programming Client Applications* )
- 『 IBM OLE DB Provider Restrictions 』 ( *Application Development Guide: Programming Client Applications* )
- 『 ActiveX Data Objects and Remote Data Objects 』 ( *Application Development Guide: Programming Client Applications* )

相关参考:

- 『 IBM OLE DB Provider Support for OLE DB Components and Interfaces 』 ( *Application Development Guide: Programming Client Applications* )
- 『 IBM OLE DB Provider support for OLE DB properties 』 ( *Application Development Guide: Programming Client Applications* )
- 『 IBM OLE DB Provider Support for ADO Methods and Properties 』 ( *Application Development Guide: Programming Client Applications* )
- 第 89 页的 『 Visual Basic 样本 』

## 使用 Visual Basic 构建松散耦合事务

XA 提供了两种方法，控件的应用程序线程可使用这两种方法来参与单个 XA 全局事务：紧密耦合和松散耦合。样本项目 LCTransTest 演示了 XA 松散耦合事务。样本文件位于 sql1lib\samples\VB\MTS 目录中。

过程:

要构建和运行松散耦合事务样本，遵循下列步骤:

### 1. 构建 LCTransTest.vbp 项目

- a. 通过双击“LCTransTest.vbp”项目打开它。
- b. 如果接收到错误消息：“无法设置版本兼容的组件 X:\...\LCTransTest.dll”，则单击“确定”继续。
- c. 编译该项目。转至“文件”->“建立 LCTransTest.dll”，然后单击“确定”。
- d. 要修正版本不兼容问题，右键单击位于右上面板中的“LCTransTest (LCTransTest.vbp)”项目。然后选择“LCTransTest 属性”。在“LCTransTest - 项目属性”窗口上，单击“组件”选项卡。在“版本兼容性”部分中，选择“二进制文件兼容性”。
- e. 保存项目。转至“文件”->“保存项目”。
- f. 关闭项目。

### 2. 构建 Main.vbp 项目

- a. 通过双击“Main.vbp”项目打开它。
- b. 您很有可能会接受到警告消息：“未能创建引用: X:\...\LCTransTest.dll”，单击“确定”继续。
- c. 转至“项目”->“引用”（在工具栏上）。
- d. 在“引用 - main.vbp”窗口上，确保选择了 Microsoft ActiveX Data Objects 2.7 Library 框。转至“浏览...”。查找在步骤 1 中生成的 LCTransTest.dll 并单击“打开”以添加此引用。在“引用 - main.vbp”窗口中单击“确定”。
- e. 编译该项目。转至“文件”->“建立 main.exe”，然后单击“确定”。
- f. 保存项目。转至“文件”->“保存项目”。
- g. 关闭项目。

### 3. 其它设置

- a. 在 Windows 上，转至“开始”->“设置”->“控制面板”->“管理工具”->“组件服务”。
- b. 在“组件服务”窗口上，在左面板中展开“组件服务”直到看到“COM+ 应用程序”为止。
- c. 右键单击“COM+ 应用程序”，选择“新建”->“应用程序”。
- d. 在弹出窗口“COM 应用程序安装向导”上，单击“下一步”。
- e. 选择“创建空的应用程序”。
- f. 输入“LCTransTest”作为新应用程序的名称。保留“激活类型”为“服务器应用程序”。单击“下一步”。
- g. 单击“下一步”，然后单击“完成”。



- h. 展开“LCTransTest”，右键单击“组件”。转至“新建”->“组件”，选择“导入已注册的组件”，单击“LCTransTest.TestClass”->“下一步”->“完成”。
- i. 展开“组件”。右键单击“LCTransTest.TestClass”。转至“属性”。在“事务”选项卡中，为“事务支持”选择“必需”
- j. 通过右键单击“组件服务”->“计算机”->“我的电脑”重新启动 Microsoft Distributed Transaction Coordinator。选择“停止 MS DTC”。等待直到它停止，然后右键单击“我的电脑”->“启动 MS DTC”以重新启动 DTC。

#### 4. 要以调试方式运行样本

- a. 打开 LCTransTest.vbp。
- b. 在 LCTransTest 中，确保对于“调试”选项卡中的“项目属性\*”，已经选择了“等待创建组件”。（这应该是缺省情况。）
- c. 在“con1.Open connString”这一行上放置断点（通过将光标放在该行上并按 F9 键完成）。
- d. 按 F5 键（或者选择“运行”下拉菜单中的“启动”）。当装入此 dll 并且此方法运行时，调试器将在该断点处停止执行。
- e. 打开 main.vbp。
- f. 在 main.vbp 中，设置命令行自变量。在“项目”属性中“建立”选项卡下面的“命令行自变量”文本框中，请输入以下内容：

```
provider=ibmdadb2;dsn=<dbname>;uid=<userid>;pwd=<password> <filename>
```

其中 <dbname> 是您具有的数据库的名称，而 <filename> 是将包含输出信息的文件的名称（包括路径）。例如，C:\lctoutput.txt。然后单击“确定”。

- g. 在 Visual Basic 中，可以使用“调试”->“单步跳入”<F8> 或“调试”->“单步跳过”<shift + F8> 来运行可执行文件，一次运行一行代码。
- h. 当到达主要可执行文件中称为“transTest.RunTest”的一行并尝试单步跳过它时，另一个 Visual Basic 窗口（已打开的 LCTransTest 项目）将显示出来，而您将停在放置的断点处。然后，可以使用“单步跳入”或“单步跳过”来继续运行 RunTest 方法，一次一行代码。

#### 相关任务:

- 第 233 页的『使用 Visual Basic 构建 ADO 应用程序』
- 第 237 页的『Visual Basic 松散耦合事务项目故障诊断』

#### 相关参考:

- 第 89 页的『Visual Basic 样本』

## Visual Basic 松散耦合事务项目故障诊断

Visual Basic 集成环境的一种功能是每次编译数据链路库（dll）时，Visual Basic 就会为它创建新的全局唯一标识（GUID），并在 Windows 注册表中注册它。在构建了项目 dll 之后，就使用 GUID 向分布式事务协调程序（DTC）注册它。如果重新构建项目 dll，则 Visual Basic 为它提供新的 GUID，并且也会在 Windows 注册表中注册它。因此，DTC 正用来引用项目 dll 的 GUID 现在已经过时，并且在 windows 注册表中该项目有两个不同的条目。

#### 过程:

要避免此问题，在第一次构建项目之后，修改项目属性：

1. 在“组件”选项卡中，选择“二进制文件兼容性”，然后使用“...”按钮来查找刚构建的 dll 的完整路径。
2. 然后单击“确定”按钮，并立即保存项目。

现在，每次重新编译项目 dll 时，它会保留同一个 GUID。但是，如果更改 dll 的接口（如通过添加或删除方法，或者更改现有方法的参数），则必须使用新的 GUID。

如果 windows 注册表中已经有多个 GUID 条目，则执行下列操作：

1. 使用注册表编辑器搜索项目名。
2. 除去除 Visual Basic 最近项目列表中的项目名之外的所有项目名。

连接信息（包括用户标识和密码）在 DTC 和 Visual Basic 应用程序中必须完全相同才能进行松散耦合事务。dll 的 public 方法通常用于直接访问它。但是，当涉及 DTC 时，它会封装 dll 并拦截对这些方法的所有入局调用和来自这些方法的出局结果。这样，DTC 可以知道其中一个方法中的数据库活动应在何时与同一对象的另一数据库活动或不同对象的数据库活动松散耦合。

为了避免此操作可能导致的任何问题，执行下列操作：

1. 在“标识”选项卡中的“DTC 中的应用程序属性”中，选择“此用户：”。
2. 使用“浏览”按钮来查找将运行此项目的用户的标识。
3. 在项目连接字符串中使用同一用户标识和密码。

如果在 Visual Basic 应用程序连接字符串中使用登录至计算机时所使用的用户标识和密码，则不必执行此附加步骤。

要确保松散耦合事务项目正常工作，可以执行下列操作：

1. 查看可执行文件创建的输出文件，并确认数据库更新是否正在进行。
2. 检查 cli 跟踪以获取 ENLIST\_IN\_DTC。

如果这些测试中的任何一个失败，则表示没有向 DTC 正确注册 dll，并且松散耦合事务没有发生。

**相关任务：**

- 第 233 页的『使用 Visual Basic 构建 ADO 应用程序』
- 第 236 页的『使用 Visual Basic 构建松散耦合事务』

**相关参考：**

- 第 89 页的『Visual Basic 样本』

## 使用 Visual Basic 构建 RDO 应用程序

“远程数据对象”（RDO）提供一个通过 ODBC 存取远程数据源的信息模型。RDO 提供一个对象集，它使得与数据库连接、执行查询和存储过程、处理结果和落实服务器更改更容易。RDO 是为存取远程 ODBC 关系数据源而专门设计的，有了它，使用 ODBC 更容易，而不必进行复杂的应用程序编程，它也是通过 ODBC 驱动程序存取关系数据库的主要途径。RDO 实现基于开放式数据库连接（ODBC）API 的瘦代码层，而驱动程序管理器则建立连接、创建结果集和游标并使用最少的工作站资源来执行复杂的过程。

DB2 在 sql1lib\samples\VB 目录中提供了 Visual Basic RDO 样本程序。

#### 过程:

要在 Microsoft Visual Basic 中使用 RDO, 需要建立对 Visual Basic 项目的引用。执行下列各项操作:

1. 从“项目”菜单选择“引用”
2. 选择“Microsoft 远程数据对象 <version\_number>”的复选框
3. 单击“确定”。

其中 <version\_number> 是当前的 RDO 版本。

一个完整的 Visual Basic 程序包括表单和其它图形元素, 您需要在 Visual Basic 环境中查看它。以下是作为 DB2 程序一部分的 Visual Basic 命令, 该程序与 sample 数据库连接, 打开从 EMPLOYEE 表中选择所有列的记录集, 然后在一个消息窗口中逐个显示职员的名字:

```
Dim rdoEn As rdoEngine
Dim rdoEv As rdoEnvironment
Dim rdoCn As rdoConnection
Dim Cnct$
Dim rdoRS As rdoResultset
Dim SQLQueryDB As String
```

指定连接字符串:

```
Cnct$ = "DSN=SAMPLE;UID=;PWD=;"
```

设置 RDO 环境:

```
Set rdoEn = rdoEngine
Set rdoEv = rdoEn.rdoEnvironments(0)
```

与数据库连接:

```
Set rdoCn = rdoEv.OpenConnection("", , , Cnct$)
```

对记录集指定 SELECT 语句:

```
SQLQueryDB = "SELECT * FROM EMPLOYEE"
```

打开记录集, 执行查询:

```
Set rdoRS = rdoCn.OpenResultset(SQLQueryDB)
```

未到达记录集末尾时, 在消息框中一次显示表中一个职员的名字 LASTNAME 和 FIRSTNAME:

```
While Not rdoRS.EOF
MsgBox rdoRS!LASTNAME &"", " &rdoRS!FIRSTNAME
```

移动到记录集中的下一行:

```
rdoRS.MoveNext
Wend
```

关闭程序:

```
rdoRS.Close
rdoCn.Close
rdoEv.Close
```

相关概念:

- 『ActiveX Data Objects and Remote Data Objects』 (*Application Development Guide: Programming Client Applications*)

相关参考:

- 第 89 页的『Visual Basic 样本』

## 使用 Visual Basic 的对象链接与嵌入 (OLE) 自动化

因为 OLE 与语言无关, 所以可以使用任何语言来实现 OLE 自动化 UDF 和存储过程。通过公开 OLE 自动化服务器的方法并将这些方法向 DB2<sup>®</sup> 注册为 UDF, 就可以完成此任务。支持 OLE 自动化服务器开发的应用程序开发环境包括下列语言的某些版本: Microsoft<sup>®</sup> Visual Basic、Microsoft Visual C++、Microsoft Visual J++、Microsoft FoxPro、Borland Delphi、Powersoft PowerBuilder 和 Micro Focus COBOL。另外, 可通过 OLE 自动化来存取为 OLE 正确包装的 Java<sup>™</sup> bean 对象 (例如, 使用 Microsoft Visual J++ 进行包装)。

需要参考适当的应用程序开发环境文档以获取开发 OLE 自动化服务器的更多信息。

### OLE 自动化 UDF 和存储过程

Microsoft Visual Basic 支持创建 OLE 自动化服务器。通过向 Visual Basic 项目添加类模块创建 Visual Basic 新对象类型。通过向类模块添加公共子过程创建方法。可以将这些公共过程作为 OLE 自动化 UDF 和存储过程注册到 DB2。有关创建和构建 OLE 服务器的进一步信息, 参考 Microsoft Visual Basic 手册 *Creating OLE Servers*, Microsoft Corporation, 1995 和 Microsoft Visual Basic 提供的 OLE 样本。

DB2 提供了用 Microsoft Visual Basic 编写的 OLE 自动化 UDF 和存储过程自包含样本, 它们位于目录 `sqllib\samples\ole\msvb` 中。有关构建和运行 OLE 自动化 UDF 和存储过程样本的信息, 请参阅 `sqllib\samples\ole` 中的 README 文件。

相关概念:

- 『OLE automation routine design』 (*Application Development Guide: Programming Server Applications*)
- 『OLE automation routines in BASIC and C++』 (*Application Development Guide: Programming Server Applications*)

相关参考:

- 第 82 页的『对象链接与嵌入 (OLE) 样本』

---

## .NET

### 构建 C# .NET 应用程序

DB2 提供了一个批处理文件 `bldapp.bat` 来编译和链接 DB2 C# .NET 应用程序 (位于 `sqllib\samples\.NET\cs` 目录中) 以及可以使用此文件来构建的样本程序。对于要编译的源文件的名称, 批处理文件采用了一个参数 `%1` (不带 `.cs` 扩展名)。

过程:

要从源文件 DbAuth.cs 构建程序 DbAuth, 请输入:

```
bldapp DbAuth
```

为了确保在运行可执行文件时具有需要的参数, 可以根据输入的数目来指定参数的不同组合:

1. 不带参数。只输入程序名:

```
DbAuth
```

2. 带一个参数。输入程序名和数据库别名:

```
DbAuth <db_alias>
```

3. 带两个参数。输入程序名以及用户标识和密码:

```
DbAuth <userid> <passwd>
```

4. 带三个参数。输入程序名以及数据库别名、用户标识和密码:

```
DbAuth <db_alias> <userid> <passwd>
```

5. 带四个参数。输入程序名以及服务器名、端口号、用户标识和密码:

```
DbAuth <server> <portnum> <userid> <passwd>
```

6. 带五个参数。输入程序名以及数据库别名、服务器名、端口号、用户标识和密码:

```
DbAuth <db_alias> <server> <portnum> <userid> <passwd>
```

要构建和运行 LCTrans 样本程序, 需要遵循 LCTrans.cs 源文件中提供的更详细的指示信息。

#### 相关任务:

- 第 246 页的『构建公共语言运行时 (CLR) .NET 例程』

#### 相关参考:

- 第 67 页的『C# 样本』
- 第 242 页的『C# .NET 应用程序编译和链接选项』

#### 相关样本:

- 『bldapp.bat -- Builds C# applications on Windows』
- 『DbAuth.cs -- How to Grant, display and revoke privileges on database』
- 『LCTrans.cs -- Demonstrates loosely coupled transactions (CSNET)』

## C# .NET 应用程序的批处理文件

```
@echo off
rem BATCH FILE: bldapp.bat
rem Builds C# applications on Windows
rem Usage: bldapp prog_name

rem Default compiler is set to Microsoft C# Compiler
rem To use a different compiler, comment out 'set BLDCOMP=csc'
rem and write 'set BLDCOMP=x' where x is the the required compiler
set BLDCOMP=csc

rem When using the .NET Framework Version 1.0 point to netf10
rem set VERSION=netf10\
rem When using the .NET Framework Version 1.1 point to netf11
set VERSION=netf11\

if exist %1.cs goto build
```

```

:build
if "%1"=="LCTrans" goto LCTransbuild
if "%1"=="SubCOM" goto SubCOMbuild
if "%1"=="RootCOM" goto RootCOMbuild
%BLDCOMP% %1.cs /r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll
goto exit

:RootCOMbuild
%BLDCOMP% /out:RootCOM.dll /target:library %1.cs /r:System.EnterpriseServices.dll
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll /r:System.Data.dll /r:System.dll
/r:SubCOM.dll
goto exit

:SubCOMbuild
%BLDCOMP% /out:SubCOM.dll /target:library %1.cs /r:System.EnterpriseServices.dll
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll /r:System.Data.dll /r:System.Xml.dll
/r:System.dll
goto exit

:LCTransbuild
%BLDCOMP% %1.cs /r:System.EnterpriseServices.dll
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll /r:System.Data.dll /r:System.dll
/r:SubCOM.dll /r:RootCOM.dll
goto exit

:exit
@echo on

```

## C# .NET 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Microsoft C# 编译器在 Windows 上构建 C# 应用程序的编译和链接选项，如在 bldapp.bat 批处理文件中所演示的那样。

<b>bldapp 的编译和链接选项</b>
<p><b>独立 C# 应用程序的编译和链接选项:</b></p> <p><b>%BLDCOMP%</b> 编译器的变量。缺省值是 csc，即 Microsoft C# 编译器。</p> <p><b>/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll</b> 引用正在使用的 .NET Framework 版本的 DB2 数据链路库。</p> <p><b>%VERSION%</b> 应用程序的 .NET Framework 有两个受支持的版本。DB2 在不同的子目录中有每个版本的数据链路库。对于 .NET Framework V1.0，%VERSION% 指向 netf10\ 子目录；对于 .NET Framework V1.1，%VERSION% 指向 netf11\ 子目录。</p>

## bldapp 的编译和链接选项

松散耦合样本程序的编译和链接选项 LCTrans:

**%BLDCOMP%**

编译器的变量。缺省值是 csc, 即 Microsoft C# 编译器。

**/out:RootCOM.dll**

从 RootCOM.cs 源文件输出由 LCTrans 应用程序使用的 RootCOM 数据链路库。

**/out:SubCOM.dll**

从 SubCOM.cs 源文件输出由 LCTrans 应用程序使用的 SubCOM 数据链路库。

**/target:library %1.cs**

从输入源文件 (RootCOM.cs 或 SubCOM.cs) 创建数据链路库。

**/r:System.EnterpriseServices.dll**

引用 Microsoft Windows System EnterpriseServices 数据链路库。

**/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll**

引用正在使用的 .NET Framework 版本的 DB2 数据链路库。

**%VERSION%**

应用程序的 .NET Framework 有两个受支持的版本。DB2 在不同的子目录中有每个版本的数据链路库。对于 .NET Framework V1.0, %VERSION% 指向 netf10 子目录; 对于 .NET Framework V1.1, %VERSION% 指向 netf11 子目录。

**/r:System.Data.dll**

引用 Microsoft Windows System Data 数据链路库。

**/r:System.dll**

引用 Microsoft Windows System 数据链路库。

**/r:System.Xml.dll**

引用 Microsoft Windows System XML 数据链路库 (适用于 SubCOM.cs)。

**/r:SubCOM.dll**

引用 SubCOM 数据链路库 (适用于 RootCOM.cs 和 LCTrans.cs)。

**/r:RootCOM.dll**

引用 RootCOM 数据链路库 (适用于 LCTrans.cs)。

有关其它编译器选项, 参考编译器文档。

相关任务:

- 第 240 页的『构建 C# .NET 应用程序』

相关样本:

- 『bldapp.bat -- Builds C# applications on Windows』

## 构建 Visual Basic .NET 应用程序

DB2 提供了一个批处理文件 bldapp.bat 来编译和链接 DB2 Visual Basic .NET 应用程序 (位于 sqllib\samples\NET\vb 目录中) 以及可以使用此文件来构建的样本程序。对于要编译的源文件的名称 (不带 .vb 扩展名), 批处理文件采用了一个参数 %1。

过程:

要从源文件 DbAuth.vb 构建程序 DbAuth, 请输入:

```
bldapp DbAuth
```

为了确保在运行可执行文件时具有需要的参数, 可以根据输入的数目来指定参数的不同组合:

1. 不带参数。只输入程序名:

```
DbAuth
```

2. 带一个参数。输入程序名和数据库别名:

```
DbAuth <db_alias>
```

3. 带两个参数。输入程序名以及用户标识和密码:

```
DbAuth <userid> <passwd>
```

4. 带三个参数。输入程序名以及数据库别名、用户标识和密码:

```
DbAuth <db_alias> <userid> <passwd>
```

5. 带四个参数。输入程序名以及服务器名、端口号、用户标识和密码:

```
DbAuth <server> <portnum> <userid> <passwd>
```

6. 带五个参数。输入程序名以及数据库别名、服务器名、端口号、用户标识和密码:

```
DbAuth <db_alias> <server> <portnum> <userid> <passwd>
```

要构建和运行 LCTrans 样本程序, 需要遵循 LCTrans.vb 源文件中提供的更详细的指示信息。

#### 相关任务:

- 第 246 页的『构建公共语言运行时 (CLR) .NET 例程』

#### 相关参考:

- 第 91 页的『Visual Basic .NET 样本』
- 第 245 页的『Visual Basic .NET 应用程序编译和链接选项』

#### 相关样本:

- 『bldapp.bat -- Builds Visual Basic .Net applications on Windows』
- 『DbAuth.vb -- How to Grant, display and revoke privileges on database』
- 『LCTrans.vb -- Demonstrates loosely coupled transactions』

## Visual Basic .NET 应用程序的批处理文件

```
@echo off
rem BATCH FILE: bldapp.bat
rem Builds Visual Basic .Net applications on Windows
rem Usage: bldapp prog_name [ db_name [ userid password ]]

rem Default compiler is set to Microsoft Visual Basic .NET Compiler
rem To use a different compiler, comment out 'set BLDCOMP=vbc'
rem and write 'set BLDCOMP=x' where x is the the required compiler
set BLDCOMP=vbc

rem When using the .NET Framework Version 1.0 point to netf10
rem set VERSION=netf10\
rem When using the .NET Framework Version 1.1 point to netf11
set VERSION=netf11\

if exist %1.vb goto build
```



```

:build
if "%1"=="LCTrans" goto LCTransbuild
if "%1"=="SubCOM" goto SubCOMbuild
if "%1"=="RootCOM" goto RootCOMbuild
%BLDCOMP% %1.vb /r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll /r:System.dll
/r:System.Data.dll /r:System.Xml.dll
goto exit

:RootCOMbuild
%BLDCOMP% %1.vb /r:System.EnterpriseServices.dll
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll /target:library /r:System.Data.dll
/r:System.dll /r:SubCOM.dll /out:RootCOM.dll
goto exit

:SubCOMbuild
%BLDCOMP% %1.vb /r:System.EnterpriseServices.dll
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll /target:library /r:System.Data.dll
/r:System.Xml.dll /r:System.dll /out:SubCOM.dll
goto exit

:LCTransbuild
%BLDCOMP% %1.vb /r:System.EnterpriseServices.dll
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll /r:System.Data.dll /r:System.dll
/r:SubCOM.dll /r:RootCOM.dll
goto exit

:exit
@echo on

```

## Visual Basic .NET 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Microsoft Visual .NET 编译器在 Windows 上构建 Visual Basic .NET 应用程序的编译和链接选项，如在 bldapp.bat 批处理文件中所演示的那样。

<b>bldapp 的编译和链接选项</b>
<p><b>独立 VB .NET 应用程序的编译和链接选项:</b></p> <p><b>%BLDCOMP%</b> 编译器的变量。缺省值是 vbc，即 Microsoft Visual Basic .NET 编译器。</p> <p><b>/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll</b> 引用正在使用的 .NET Framework 版本的 DB2 数据链路库。</p> <p><b>%VERSION%</b> 应用程序的 .NET Framework 有两个受支持的版本。DB2 在不同的子目录中有每个版本的数据链路库。对于 .NET Framework V1.0, %VERSION% 指向 netf10\ 子目录; 对于 .NET Framework V1.1, %VERSION% 指向 netf11\ 子目录。</p>

## bldapp 的编译和链接选项

松散耦合样本程序的编译和链接选项 LCTrans:

**%BLDCOMP%**

编译器的变量。缺省值是 vbc, 即 Microsoft Visual Basic .NET 编译器。

**/out:RootCOM.dll**

从 RootCOM.vb 源文件输出由 LCTrans 应用程序使用的 RootCOM 数据链路库。

**/out:SubCOM.dll**

从 SubCOM.vb 源文件输出由 LCTrans 应用程序使用的 SubCOM 数据链路库。

**/target:library %1.cs**

从输入源文件 (RootCOM.vb 或 SubCOM.vb) 创建数据链路库。

**/r:System.EnterpriseServices.dll**

引用 Microsoft Windows System EnterpriseServices 数据链路库。

**/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll**

引用正在使用的 .NET Framework 版本的 DB2 数据链路库。

**%VERSION%**

应用程序的 .NET Framework 有两个受支持的版本。DB2 在不同的子目录中有每个版本的数据链路库。对于 .NET Framework V1.0, %VERSION% 指向 netf10 子目录; 对于 .NET Framework V1.1, %VERSION% 指向 netf11 子目录。

**/r:System.Data.dll**

引用 Microsoft Windows System Data 数据链路库。

**/r:System.dll**

引用 Microsoft Windows System 数据链路库。

**/r:System.Xml.dll**

引用 Microsoft Windows System XML 数据链路库 (适用于 SubCOM.vb)。

**/r:SubCOM.dll**

引用 SubCOM 数据链路库 (适用于 RootCOM.vb 和 LCTrans.vb)。

**/r:RootCOM.dll**

引用 RootCOM 数据链路库 (适用于 LCTrans.vb)。

有关其它编译器选项, 参考编译器文档。

相关任务:

- 第 243 页的『构建 Visual Basic .NET 应用程序』

相关样本:

- 『bldapp.bat -- Builds Visual Basic .Net applications on Windows』

## 构建公共语言运行时 (CLR) .NET 例程

DB2 提供了用于编译和链接 DB2 .NET 程序的批处理文件。这些批处理文件和可以使用这些文件构建的样本程序一同位于 `sqllib\samples\.NET\cs` 和 `sqllib\samples\.NET\vb` 目录中。

批处理文件 `bldrtn.bat` 包含构建 CLR 例程（存储过程和用户定义的函数）的命令。该批处理文件在服务器上构建 .NET 组合件 DLL。它使用两个参数，这两个参数在批处理文件中以变量 `%1` 和 `%2` 表示。

第一个参数 `%1` 指定源文件名。该批处理文件将源文件名用作组合件 DLL 名。第二个参数 `%2` 指定想要连接的数据库的名称。由于必须在数据库所在的那个实例中构建组合件 DLL，所以没有任何有关用户标识和密码的参数。

只有第一个参数即源文件名是必需的。数据库名称是可选的。如果未提供数据库名称，则该程序使用缺省的 `sample` 数据库。

#### 先决条件:

数据库服务器必须运行安装有 Microsoft .NET Framework V1.1 (.NET Framework 1.0 和 .NET Framework 1.1 都受客户机应用程序支持) 的 Windows 操作系统。 .NET Framework 是单独提供或作为 Microsoft .NET Framework 1.1 Software Development Kit 的一部分提供的。

必须安装下列版本的 DB2:

#### 服务器:

DB2 8.2 或更新版本

#### 客户机:

DB2 7.2 或更新版本

必须授予执行例程的 CREATE 语句的权限。有关执行 CREATE 语句所需的特权，请参阅例程类型的 CREATE 语句: CREATE PROCEDURE 或 CREATE FUNCTION。

#### 过程:

下列示例显示如何构建具有存储过程和用户定义的函数的例程组合件 DLL:

#### 存储过程组合件 DLL

要从 VB .NET 源文件 `SpServer.vb` 或 C# 源文件 `SpServer.cs` 构建 `SpServer` 组合件 DLL:

1. 输入批处理文件名和程序名（不带扩展名）:

```
bldrtn SpServer
```

如果连接至 `sample` 数据库之外的数据库，还需要输入数据库名称:

```
bldrtn SpServer database
```

该批处理文件将组合件 DLL (`SpServer.dll`) 复制到 `sqllib\function` 目录。

2. 接着，通过在服务器上运行 `spscat` 脚本对例程进行编目:

```
SpCat
```

此脚本连接至样本数据库，如果先前通过调用 `SpDrop.db2` 对这些例程进行了编目，则取消对它们的编目，然后通过调用 `SpCreate.db2` 对它们进行编目，最后与数据库断开连接。还可以分别调用 `SpDrop.db2` 和 `SpCreate.db2` 脚本。

3. 然后，除非这是第一次构建组合件 DLL，否则停止数据库然后重新启动它以重组新版本的组合件 DLL。必要时，对组合件 DLL 设置文件方式，以便 DB2 实例可以存取它。

一旦构建了组合件 DLL (SpServer)，就可以构建调用它的客户机应用程序 SpClient。

可以通过使用批处理文件 bldapp.bat 来构建 SpClient。

为了确保在运行可执行文件时具有需要的参数，可以根据输入的参数数目来指定参数的不同组合而不是接受缺省值：

1. 不带参数。只输入程序名（以供在服务器实例上本地调用）：

```
SpClient
```

2. 带一个参数。输入程序名和数据库别名（供在服务器实例上本地调用 sample 数据库之外的数据库）：

```
SpClient <db_alias>
```

3. 带三个参数。输入程序名以及数据库别名、用户标识和密码（供从远程客户机调用）：

```
SpClient <db_alias> <userid> <passwd>
```

4. 带五个参数。输入程序名以及数据库别名、服务器名、端口号、用户标识和密码（供从远程客户机调用）：

```
SpClient <db_alias> <server> <portnum> <userid> <passwd>
```

客户机应用程序将存取组合件 DLL (SpServer) 并在服务器数据库上执行一系列例程。然后将输出返回给该客户机应用程序。

### 用户定义的函数组合件 DLL

要从 VB .NET 源文件 UDFsrv.vb 或 C# 源文件 UDFsrv.cs 构建用户定义的函数组合件 DLL UDFsrv，请输入：

```
bldrtn UDFsrv
```

如果连接至 sample 数据库之外的数据库，还需要输入数据库名称：

```
bldrtn UDFsrv database
```

该批处理文件将用户定义的函数组合件 DLL UDFsrv.dll 复制到 sqllib\function 目录中。

一旦构建了 UDFsrv，就可以构建调用它的客户机应用程序 udfcli。

可以通过使用批处理文件 bldapp.bat 来构建 UDFcli。

为了确保在运行可执行文件时具有需要的参数，可以根据输入的参数数目来指定参数的不同组合而不是接受缺省值：

1. 不带参数。只输入程序名（以供在服务器实例上本地调用）：

```
UDFcli
```

2. 带一个参数。输入程序名和数据库别名（供在服务器实例上本地调用 sample 数据库之外的数据库）：

```
UDFcli <db_alias>
```

3. 带三个参数。输入程序名以及数据库别名、用户标识和密码（供从远程客户机调用）：

```
UDFcli <db_alias> <userid> <passwd>
```

4. 带五个参数。输入程序名以及数据库别名、服务器名、端口号、用户标识和密码（供从远程客户机调用）：

```
UDFcli <db_alias> <server> <portnum> <userid> <passwd>
```

调用应用程序从 `udfsrv` 组合件 DLL 调用 `ScalarUDF` 函数。

相关概念：

- 『Common language runtime (CLR) routines』 (*Application Development Guide: Programming Server Applications*)

相关样本：

- 『bldrtn.bat -- Builds C# routines (stored procedures and UDFs)』
- 『SpServer.cs -- C# external code implementation of procedures created in spcat.db2』
- 『SpClient.cs -- Call different types of stored procedures from SpServer.java』
- 『UDFcli.cs -- Client application that calls the user-defined functions』
- 『UDFsrv.cs -- User-defined scalar functions called by udfcli.cs』
- 『bldrtn.bat -- Builds Visual Basic .NET routines (stored procedures and UDFs)』
- 『SpServer.vb -- VB.NET implementation of procedures created in SpCat.db2』
- 『SpClient.vb -- Call different types of stored procedures from SpServer.java』
- 『UDFcli.vb -- Client application that calls the user-defined functions』
- 『UDFsrv.vb -- User-defined scalar functions called by udfcli.vb』

## C# .NET 例程的批处理文件

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Builds C# routines (stored procedures and UDFs)
rem Usage: bldrtn prog_name

rem When using the .NET Framework Version 1.1 point to netf11
set VERSION=netf11\

rem Compile the program.
csc /out:%1.dll /target:library /debug /lib:"%DB2PATH%\bin\netf11\
/reference:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll %1.cs

if exist "%DB2PATH%\function\%1.dll" goto delete else goto copydll

:delete
del "%DB2PATH%\function\%1.dll"
goto copydll

:copydll
rem Copy the routine assembly data link library to the 'function' directory
copy "%1.dll" "%DB2PATH%\function"

@echo on
```

## Visual Basic .NET 例程的批处理文件

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Builds Visual Basic .NET routines (stored procedures and UDFs)
rem Usage: bldrtn prog_name

rem Compile the program.
vbc %1.vb /out:%1.dll /target:library /debug /libpath:"%DB2PATH%\bin\netf11\
/reference:IBM.Data.DB2.dll /reference:System.dll /reference:System.Data.dll

if exist "%DB2PATH%\function\%1.dll" goto delete else goto copydll

:delete
del "%DB2PATH%\function\%1.dll"
goto copydll

:copydll
rem Copy the routine assembly data link library to the 'function' directory
copy "%1.dll" "%DB2PATH%\function"

@echo on
```

## CLR .NET 例程编译和链接选项

下列内容是 DB2 建议用于使用 Microsoft Visual Basic .NET 编译器或 Microsoft C# 编译器在 Windows 上构建“公共语言运行时”（CLR）.NET 例程的编译和链接选项，如在 samples\.NET\cs\bldrtn.bat 和 samples\.NET\vb\bldrtn.bat 批处理文件中所演示的那样。

bldrtn 的编译和链接选项	
使用 <b>Microsoft C#</b> 编译器的编译和链接选项:	
<b>csc</b>	Microsoft C# 编译器。
<b>/out:%1.dll /target:library</b>	将数据链路库作为存储过程组合件 dll 输出。
<b>/debug</b>	使用调试器。
<b>/lib: "%DB2PATH%\bin\netf11\</b>	使用 .NET Framework V1.1 的库路径。
<b>/reference:IBM.Data.DB2.dll</b>	使用 .NET Framework V1.1 的 DB2 数据链路库。
有关其它编译器选项，参考编译器文档。	

### bldrtn 的编译和链接选项

使用 **Microsoft Visual Basic .NET** 编译器的编译和链接选项:

**vbc** Microsoft Visual Basic .NET 编译器。

**/out:%1.dll /target:library**

将数据链路库作为存储过程组合件 dll 输出。

**/debug** 使用调试器。

**/libpath:"%DB2PATH%\bin\netf11\**

使用 .NET Framework V1.1 的库路径。

**/reference:IBM.Data.DB2.dll**

使用 .NET Framework V1.1 的 DB2 数据链路库。

**/reference:System.dll**

引用 Microsoft Windows System 数据链路库。

**/reference:System.Data.dll**

引用 Microsoft Windows System Data 数据链路库。

有关其它编译器选项, 参阅编译器文档。

#### 相关任务:

- 第 246 页的『构建公共语言运行时 (CLR) .NET 例程』

#### 相关样本:

- 『bldrtn.bat -- Builds C# routines (stored procedures and UDFs)』
- 『bldrtn.bat -- Builds Visual Basic .NET routines (stored procedures and UDFs)』

---

## Microsoft Visual C++

本节讨论使用 ActiveX 数据对象 (ADO)、对象链接与嵌入技术 (OLE) 以及嵌入式 SQL 和 DB2 API 来构建应用程序。

DB2 CLI 应用程序和例程的构建信息在 *CLI Guide and Reference* 中。

### 使用 Visual C++ 构建 ADO 应用程序

“ActiveX 数据对象” (ADO) 可用于编写一个应用程序, 以便通过 OLE DB 提供程序存取和处理数据库服务器中的数据。ADO 的主要优点是高速、易用、内存开销较低以及占用磁盘较少。

DB2 在 `sql1lib\samples\VC` 目录中提供了 Visual C++ ADO 样本程序。

#### 过程:

可以使用以下两个与 ODBC 兼容的提供程序当中的任一个:

- IBM OLE DB provider for DB2
- Microsoft OLE DB provider for ODBC

#### 使用 IBM OLE DB provider for DB2

Windows 操作系统上的 DB2 版本 8.2 客户机将可选择安装 IBMDADB2 (即 IBM OLE DB 2.0-compliant provider for DB2)。该提供程序为希望存取 DB2 数据库中数据的客户提供接口。IBM OLE DB provider for DB2 支持下列 ADO 应用程序类型:

- Microsoft Active Server Pages (ASP)
- Microsoft Visual Studio C++ 和 Visual Basic 应用程序
- Microsoft Visual Interdev

有关这些类型的应用程序的详细信息, 请参考 ADO 文档。

### 使用 Microsoft OLE DB provider for ODBC

进行了以下更改之后, 使用 Microsoft OLE DB 提供程序和 Visual C++ 的 DB2 ADO 程序可以象普通的 C++ 程序一样编译。

要让 C++ 源程序作为 ADO 程序运行, 可将下面的 import 语句放在源程序文件开始处:

```
#import "C:\program files\common files\system\ado\msado<VERSION NUMBER>.dll" \  
no_namespace \  
rename( "EOF", "adoEOF")
```

其中 <VERSION NUMBER> 是 ADO 库的版本号。

当编译器时, 用户需要验证 msado<VERSION NUMBER>.dll 在指定的路径中。另一个方法是将 C:\program files\common files\system\ado 添加到环境变量 LIBPATH 中, 然后在源文件中使用这个较短的 import 语句:

```
#import <msado<VERSION NUMBER>.dll> \  
no_namespace \  
rename( "EOF", "adoEOF")
```

这是 DB2 样本程序 BLOBAccess.dsp 中使用的方法。

有了此 IMPORT 语句, DB2 程序就可存取 ADO 库。现在, 可像编译任何其它程序一样编译 Visual C++ 程序。如果还使用另一个编程接口, 如 DB2 API 或 DB2 CLI, 可参考适当的主题以获取构建程序的其它信息。

### 相关概念:

- 『 Purpose of the IBM OLE DB Provider for DB2 』 (*Application Development Guide: Programming Client Applications*)
- 『 Application Types Supported by the IBM OLE DB Provider for DB2 』 (*Application Development Guide: Programming Client Applications*)
- 『 Compilation and Linking of C/C++ Applications and the IBM OLE DB Provider 』 (*Application Development Guide: Programming Client Applications*)
- 『 Connections to Data Sources in C/C++ Applications using the IBM OLE DB Provider 』 (*Application Development Guide: Programming Client Applications*)
- 『 OLE DB Services Automatically Enabled by IBM OLE DB Provider 』 (*Application Development Guide: Programming Client Applications*)
- 『 Large Object Manipulation with the IBM OLE DB Provider 』 (*Application Development Guide: Programming Client Applications*)
- 『 IBM OLE DB Provider Restrictions 』 (*Application Development Guide: Programming Client Applications*)



#### 相关参考:

- 『Data Type Mappings between DB2 and OLE DB』 (*Application Development Guide: Programming Client Applications*)
- 『Data Conversion for Setting Data from OLE DB Types to DB2 Types』 (*Application Development Guide: Programming Client Applications*)
- 『Data Conversion for Setting Data from DB2 Types to OLE DB Types』 (*Application Development Guide: Programming Client Applications*)
- 『IBM OLE DB Provider Support for OLE DB Components and Interfaces』 (*Application Development Guide: Programming Client Applications*)
- 『IBM OLE DB Provider support for OLE DB properties』 (*Application Development Guide: Programming Client Applications*)
- 『IBM OLE DB Provider Support for ADO Methods and Properties』 (*Application Development Guide: Programming Client Applications*)
- 第 92 页的『Visual C++ 样本』

## 使用 Visual C++ 的对象链接与嵌入 (OLE) 自动化

因为 OLE 与语言无关, 所以可以使用任何语言来实现 OLE 自动化 UDF 和存储过程。通过公开 OLE 自动化服务器的方法并将这些方法向 DB2<sup>®</sup> 注册为 UDF, 就可以完成此任务。支持 OLE 自动化服务器开发的应用程序开发环境包括下列语言的某些版本: Microsoft<sup>®</sup> Visual Basic、Microsoft Visual C++、Microsoft Visual J++、Microsoft FoxPro、Borland Delphi、Powersoft PowerBuilder 和 Micro Focus COBOL。另外, 可通过 OLE 自动化来存取为 OLE 正确包装的 Java<sup>™</sup> bean 对象 (例如, 使用 Microsoft Visual J++ 进行包装)。

需要参考适当的应用程序开发环境文档以获取开发 OLE 自动化服务器的更多信息。

### OLE 自动化 UDF 和存储过程

Microsoft Visual C++ 支持创建 OLE 自动化服务器。可使用“Microsoft 基础类”和“Microsoft 基础类应用程序向导”实现服务器, 或作为 Win32 应用程序来实现服务器。服务器可以是 DLL 或 EXE。参考 Microsoft Visual C++ 文档和 Microsoft Visual C++ 提供的 OLE 样本以获取进一步的信息。

DB2 提供了用 Microsoft Visual C++ 编写的 OLE 自动化 UDF 和存储过程自包含样本, 它们位于目录 `sqllib\samples\ole\msvc` 中。有关构建和运行 OLE 自动化 UDF 和存储过程样本的信息, 请参阅 `sqllib\samples\ole` 中的 README 文件。

#### 相关概念:

- 『OLE automation routine design』 (*Application Development Guide: Programming Server Applications*)
- 『OLE automation routines in BASIC and C++』 (*Application Development Guide: Programming Server Applications*)

#### 相关参考:

- 第 82 页的『对象链接与嵌入 (OLE) 样本』

## 在 Windows 上构建 C/C++ 应用程序

DB2 提供了用于编译和链接 DB2 API 及嵌入式 SQL C/C++ 程序的批处理文件。这些批处理文件和可以使用这些文件构建的样本程序一同位于 `sqllib\samples\c` 和 `sqllib\samples\cpp` 目录中。

批处理文件 `bldapp.bat` 包含构建 DB2 API 和嵌入式 SQL 程序的命令。它最多使用四个参数，这些参数在批处理文件中以变量 `%1`、`%2`、`%3` 和 `%4` 表示。

第一个参数 `%1` 指定源文件名。这是不包含嵌入式 SQL 的程序所需的唯一参数。构建嵌入式 SQL 程序需要连接至数据库，因此还提供了三个附加参数：第二个参数 `%2` 指定想要连接的数据库的名称；第三个参数 `%3` 指定数据库的用户标识，而 `%4` 指定密码。

对于嵌入式 SQL 程序，`bldapp` 将这些参数传递到预编译和绑定文件 `embprep.bat`。如果未提供数据库名称，则使用缺省 `sample` 数据库。仅当构建程序所在的实例不同于数据库所在的实例时，才需要用户标识和密码参数。

### 过程:

下列示例显示如何构建和运行 DB2 API 和嵌入式 SQL 应用程序。

要从 `sqllib\samples\c` 中的源文件 `cli_info.c` 或从 `sqllib\samples\cpp` 中的源文件 `cli_info.cxx` 构建 DB2 API 非嵌入式 SQL 样本程序 `cli_info`，请输入：

```
bldapp cli_info
```

得到一个可执行文件 `cli_info.exe`。可在命令行上输入可执行文件名（不带扩展名）来运行该可执行文件：

```
cli_info
```

### 构建和运行嵌入式 SQL 应用程序

从 `sqllib\samples\c` 中的 C 源文件 `tbmod.sqc` 或从 `sqllib\samples\cpp` 中的 C++ 源文件 `tbmod.sqx` 构建嵌入式 SQL 应用程序 `tbmod` 有三种方法：

1. 如果与同一实例中的样本数据库连接，则输入：

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接，还需要输入该数据库名称：

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接，还需要输入该数据库实例的用户标识和密码：

```
bldapp tbmod database userid password
```

得到一个可执行文件 `tbmod.exe`。

运行此嵌入式 SQL 应用程序有三种方法：

1. 如果存取同一实例中的 `sample` 数据库，只须输入可执行文件名：

```
tbmod
```

2. 如果存取同一实例中的另一个数据库，则输入可执行文件名和数据库名称：

```
tbmod database
```

3. 如果存取另一个实例中的数据库，则输入可执行文件名、数据库名称以及该数据库实例的用户标识和密码：

```
tbmod database userid password
```

## 构建和运行多线程应用程序

Windows 上的 C/C++ 多线程应用程序需要用 `-MT` 或 `-MD` 选项进行编译。`-MT` 选项将使用静态库 `LIBCMT.LIB` 进行链接, 而 `-MD` 选项将使用动态库 `MSVCRT.LIB` 进行链接。使用 `-MD` 链接的二进制文件会小一些但依赖于 `MSVCRT.DLL`, 而用 `-MT` 链接的二进制文件要大一些但对于运行时是自包含的。

批处理文件 `bldmt.bat` 使用 `-MT` 选项来构建多线程程序。所有其它编译和链接选项都与批处理文件 `bldapp.bat` 用来构建常规独立应用程序所用的选项相同。

要从 `samples\c\dbthrds.sqc` 或 `samples\cpp\dbthrds.sqx` 源文件构建多线程样本程序 `dbthrds`, 请输入:

```
bldmt dbthrds
```

得到一个可执行文件 `dbthrds.exe`。

运行此多线程应用程序有三种方法:

1. 如果存取同一实例中的 `sample` 数据库, 只须输入可执行文件名 (不需输入扩展名):

```
dbthrds
```

2. 如果存取同一实例中的另一个数据库, 则输入可执行文件名和数据库名称:

```
dbthrds database
```

3. 如果存取另一个实例中的数据库, 则输入可执行文件名、数据库名称以及该数据库实例的用户标识和密码:

```
dbthrds database userid password
```

### 相关参考:

- 第 256 页的『Windows C/C++ 应用程序编译和链接选项』

### 相关样本:

- 『`bldapp.bat` -- Builds C applications on Windows』
- 『`bldmt.bat` -- Builds C multi-threaded applications on Windows』
- 『`cli_info.c` -- Set and get information at the client level (C)』
- 『`dbthrds.sqc` -- How to use multiple context APIs on Windows (C)』
- 『`embprep.bat` -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows』
- 『`tbmod.sqc` -- How to modify table data (C)』
- 『`bldapp.bat` -- Builds C++ applications on Windows』
- 『`bldmt.bat` -- Builds C++ multi-threaded applications on Windows』
- 『`cli_info.C` -- Set and get information at the client level (C++)』
- 『`dbthrds.sqC` -- How to use multiple context APIs on Windows (C++)』
- 『`tbmod.sqC` -- How to modify table data (C++)』

## C/C++ 应用程序的批处理文件

```
@echo off
rem BATCH FILE: bldapp.bat
rem Builds C/C++ applications on Windows
rem Usage: bldapp prog_name [ db_name [ userid password ]]

rem Default compiler is set to Microsoft Visual C++
rem To use a different compiler, comment out 'set BLDCOMP=c1'
rem and uncomment 'set BLDCOMP=ic1' or 'set BLDCOMP=ec1'
rem Microsoft C/C++ Compiler
set BLDCOMP=c1

rem Intel C++ Compiler for 32-bit applications
rem set BLDCOMP=ic1

rem Intel C++ Compiler for Itanium 64-bit applications
rem set BLDCOMP=ec1

if exist "%1.sqx" goto embedded
if exist "%1.sqc" goto embedded
goto non_embedded

:embedded
rem Precompile and bind the program.
rem Note: some .sqc/.sqx files contain no SQL but link in
rem utilemb.sqc/.sqx, so if you get this warning, ignore it:
rem SQL0053W No SQL statements were found in the program.
call embprep %1 %2 %3 %4
rem Compile the program.
if exist "%1.cxx" goto cpp_emb
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.c utilemb.c
goto link_embedded
:cpp_emb
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.cxx utilemb.cxx
rem Link the program.
:link_embedded
link -debug -out:%1.exe %1.obj utilemb.obj db2api.lib
goto exit

:non_embedded
rem Compile the program.
if exist "%1.cxx" goto cpp_non
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.c utilapi.c
goto link_non_embedded
:cpp_non
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.cxx utilapi.cxx
rem Link the program.
:link_non_embedded
link -debug -out:%1.exe %1.obj utilapi.obj db2api.lib
:exit
@echo on
```

## Windows C/C++ 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Microsoft Visual C++ 编译器在 Windows 上构建 C/C++ 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp.bat 批处理文件中所示的那样。

<b>bldapp 的编译和链接选项</b>	
<b>编译选项:</b>	
<b>%BLDCOMP%</b>	编译器的变量。缺省值是 c1, 即 Microsoft Visual C++ 编译器。它还可设置为 icl (即 Intel C++ Compiler for 32-bit applications) 或 ecl (即 Intel C++ Compiler for Itanium 64-bit applications)。
<b>-Zi</b>	启用调试信息
<b>-Od</b>	禁用优化。关闭优化使得使用调试器更容易。
<b>-c</b>	只执行编译; 不链接。批处理文件具有单独的编译和链接步骤。
<b>-W2</b>	输出警告、错误、严重的和不可恢复的错误消息。
<b>-DWIN32</b>	Windows 操作系统需要的编译器选项。
<b>链接选项:</b>	
<b>link</b>	使用链接程序进行链接。
<b>-debug</b>	包括调试信息。
<b>-out:%1.exe</b>	指定文件名
<b>%1.obj</b>	包括对象文件
<b>utilemb.obj</b>	如果是嵌入式 SQL 程序, 应包括嵌入式 SQL 实用程序对象文件以便检查错误。
<b>utilapi.obj</b>	如果不是嵌入式 SQL 程序, 应包括 DB2 API 实用程序对象文件以便检查错误。
<b>db2api.lib</b>	链接 DB2 库。
有关其它编译器选项, 参考编译器文档。	

#### 相关任务:

- 第 254 页的『在 Windows 上构建 C/C++ 应用程序』

#### 相关样本:

- 『bldapp.bat -- Builds C applications on Windows』
- 『bldapp.bat -- Builds C++ applications on Windows』

## 在 Windows 上构建 C/C++ 例程

DB2 提供了用于编译和链接 DB2 API 及嵌入式 SQL C/C++ 程序的批处理文件。这些批处理文件和可以使用这些文件构建的样本程序一同位于 `sqllib\samples\c` 和 `sqllib\samples\cpp` 目录中。

批处理文件 `bldrtn.bat` 包含构建嵌入式 SQL 例程 (存储过程和用户定义的函数) 的命令。该批处理文件在服务器上构建 DLL。它使用两个参数, 这两个参数在批处理文件中以变量 `%1` 和 `%2` 表示。

第一个参数 %1 指定源文件名。该批处理文件将源文件名用作 DLL 名。第二个参数 %2 指定想要连接的数据库的名称。由于必须在数据库所在的那个实例中构建 DLL，所以没有任何有关用户标识和密码的参数。

只有第一个参数即源文件名是必需的。数据库名称是可选的。如果未提供数据库名称，则该程序使用缺省的 `sample` 数据库。

#### 过程:

下列示例显示如何构建具有下列内容的例程 DLL:

- 存储过程
- 非嵌入式 SQL 用户定义的函数 (UDF)
- 嵌入式 SQL 用户定义的函数 (UDF)

#### 存储过程 DLL

要从 C 源文件 `spserver.sqc` 或 C++ 源文件 `spserver.sqx` 构建 `spserver` DLL:

1. 输入批处理文件名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还需要输入该数据库名称:

```
bldrtn spserver database
```

此批处理文件使用模块定义文件 `spserver.def` 构建 DLL，该模块定义文件与样本程序位于同一目录中。该批处理文件将 DLL (`spserver.dll`) 复制到服务器的 `sqllib\function` 路径中。

2. 接着，通过在服务器上运行 `spcat` 脚本对例程进行编目:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 对这些例程进行了编目，则取消对它们的编目，然后通过调用 `spcreate.db2` 对它们进行编目，最后与数据库断开连接。还可以分别调用 `spdrop.db2` 和 `spcreate.db2` 脚本。

3. 接着，停止数据库然后重新启动它以允许识别新的 DDL。如有必要，对 DLL 设置文件方式，以便 DB2 实例可以存取它。

一旦构建了 DLL (`spserver`)，就可以构建调用它的客户机应用程序 `spclient`。

可以通过使用批处理文件 `bldapp.bat` 来构建 `spclient`。

要调用 DLL，通过输入下列命令来运行样本客户机应用程序:

```
spclient database userid password
```

其中，

#### **database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

## password

是用户标识的有效密码。

客户机应用程序将存取 DLL (spserver)，并在服务器数据库上执行一系列例程。然后将输出返回给该客户机应用程序。

## 非嵌入式 SQL UDF DLL

要从源文件 udfsrv.c 构建用户定义的函数 udfsrv，请输入：

```
bldrtn udfsrv
```

批处理文件使用模块定义文件 udfsrv.def (该文件与样本程序文件处于同一目录中) 来构建用户定义的函数 DLL。该批处理文件将用户定义的函数 DLL (udfsrv.dll) 复制到服务器的 sqllib\function 路径中。

一旦构建了 udfsrv，就可构建调用它的客户机应用程序 udfcli。提供了此程序的 DB2 CLI 以及嵌入式 SQL C 和 C++ 版本。

可以使用批处理文件 bldapp 从 sqllib\samples\cli 中的源文件 udfcli.c 构建 DB2 CLI udfcli 程序。

可以使用批处理文件 bldapp 从 sqllib\samples\c 中的源文件 udfcli.sqc 构建嵌入式 SQL C udfcli 程序。

可以使用批处理文件 bldapp 从 sqllib\samples\cpp 中的源文件 udfcli.sqx 构建嵌入式 SQL C++ udfcli 程序。

要运行 UDF，请输入：

```
udfcli
```

调用应用程序从 udfsrv DLL 中调用 ScalarUDF 函数。

## 嵌入式 SQL UDF DLL

要从 sqllib\samples\c 中的 C 源文件 udfemsrv.sqc 或从 sqllib\samples\cpp 中的 C++ 源文件 udfemsrv.sqx 构建嵌入式 SQL 用户定义的函数库 udfemsrv，请输入：

```
bldrtn udfemsrv
```

如果连接另一个数据库，还需要输入该数据库名称：

```
bldrtn udfemsrv database
```

批处理文件使用模块定义文件 udfemsrv.def (该文件与样本程序处于同一目录中) 来构建用户定义的函数 DDL。该批处理文件将用户定义的函数 DLL (udfemsrv.dll) 复制到服务器的 sqllib\function 路径中。

一旦构建了 udfemsrv，就可构建调用它的客户机应用程序 udfemcli。可以使用批处理文件 bldapp 从 sqllib\samples\c 中的 C 源文件 udfemcli.sqc 或从 sqllib\samples\cpp 中的 C++ 源文件 udfemcli.sqx 构建 udfemcli。

要运行 UDF，请输入：

```
udfemcli
```

调用应用程序将调用 `udfemsrv DLL` 中的 UDF。

#### 相关参考:

- 第 261 页的『Windows C/C++ 例程编译和链接选项』

#### 相关样本:

- 『`bldrtn.bat` -- Builds C routines (stored procedures and UDFs) on Windows』
- 『`embprep.bat` -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows』
- 『`spclient.sqc` -- Call various stored procedures (C)』
- 『`spserver.sqc` -- Definition of various types of stored procedures (C)』
- 『`udfcli.sqc` -- Call a variety of types of user-defined functions (C)』
- 『`udfemcli.sqc` -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『`udfemsrv.sqc` -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『`udfsrv.c` -- Defines a variety of types of user-defined functions (C)』
- 『`bldrtn.bat` -- Builds C++ routines (stored procedures and UDFs) on Windows』
- 『`spclient.sqC` -- Call various stored procedures (C++)』
- 『`spserver.sqC` -- Definition of various types of stored procedures (C++)』
- 『`udfcli.sqC` -- Call a variety of types of user-defined functions (C++)』
- 『`udfemcli.sqC` -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『`udfemsrv.sqC` -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『`udfsrv.C` -- Defines a variety of types of user-defined functions (C++)』

## C/C++ 例程的批处理文件

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Builds C/C++ routines (stored procedures and UDFs) on Windows
rem Usage: bldrtn prog_name [ db_name ]

rem Default compiler is set to Microsoft Visual C++
rem To use a different compiler, comment out 'set BLDCOMP=c1'
rem and uncomment 'set BLDCOMP=icl' or 'set BLDCOMP=ec1'
rem Microsoft C/C++ Compiler
set BLDCOMP=c1

rem Intel C++ Compiler for 32-bit applications
rem set BLDCOMP=icl

rem Intel C++ Compiler for Itanium 64-bit applications
rem set BLDCOMP=ec1

if exist "%1.sqc" goto embedded
if exist "%1.sqx" goto embedded
goto compile

:embedded
rem Precompile and bind the program.
call embprep %1 %2

:compile
rem Compile the program.
```



```

if exist "%1.cxx" goto cpp
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 -MD %1.c
goto link_step
:cpp
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 -MD %1.cxx

:link_step
rem Link the program.
link -debug -out:%1.dll -dll %1.obj db2api.lib -def:%1.def

rem Copy the routine DLL to the 'function' directory
copy %1.dll "%DB2PATH%\function"
@echo on

```

## Windows C/C++ 例程编译和链接选项

下列内容是 DB2 建议用于使用 Microsoft Visual C++ 编译器在 Windows 上构建 C/C++ 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn.bat 批处理文件中所演示的那样。

bldrtn 的编译和链接选项	
<b>编译选项:</b>	
<b>%BLDCOMP%</b>	编译器的变量。缺省值是 cl，即 Microsoft Visual C++ 编译器。它还可设置为 icl（即 Intel C++ Compiler for 32-bit applications）或 ecl（即 Intel C++ Compiler for Itanium 64-bit applications）。
<b>-Zi</b>	启用调试信息。
<b>-Od</b>	禁用优化。
<b>-c</b>	只执行编译；不链接。编译和链接是两个独立的步骤。
<b>-W2</b>	输出警告、错误、严重的和不可恢复的错误消息。
<b>-DWIN32</b>	Windows 操作系统需要的编译器选项。
<b>-MD</b>	使用 MSVCRT.LIB 来创建多线程 DLL。
<b>链接选项:</b>	
<b>link</b>	使用链接程序进行链接。
<b>-debug</b>	包括调试信息。
<b>-out:%1.dll</b>	构建 .DLL 文件。
<b>%1.obj</b>	包括对象文件。
<b>db2api.lib</b>	链接 DB2 库。
<b>-def:%1.def</b>	模块定义文件。
有关其它编译器选项，参考编译器文档。	

### 相关任务:

- 第 257 页的『在 Windows 上构建 C/C++ 例程』

相关样本:

- 『 bldrtn.bat -- Builds C routines (stored procedures and UDFs) on Windows 』
- 『 bldrtn.bat -- Builds C++ routines (stored procedures and UDFs) on Windows 』

## 在 Windows 上构建 C/C++ 多连接应用程序

DB2 提供了用于编译和链接 C 和 C++ 嵌入式 SQL 和 DB2 API 程序的批处理文件。这些批处理文件和可以使用这些文件构建的样本程序都位于 `sqllib\samples\c` 和 `sqllib\samples\cpp` 目录中。

批处理文件 `bldmc.bat` 包含用于构建 DB2 多连接程序的命令，需要两个数据库。编译和链接选项与在 `bldapp.bat` 文件中所用的选项相同。

第一个参数 `%1` 指定源文件名。第二个参数 `%2` 指定要连接至的第一个数据库的名称。第三个参数 `%3` 指定要连接至的第二个数据库。这些参数都是必需参数。

**注:** `makefile` 对数据库名称的“sample”和“sample2”的缺省值进行硬编码（分别为 `%2` 和 `%3`），因此，如果要使用 `makefile` 并接受这些缺省值，则只需指定程序名（`%1` 参数）。如果要使用 `bldmc.bat` 脚本，则必须指定所有三个参数。

可选参数对于本地连接不是必需的，但对于从远程客户机至服务器的连接则是必需的。可选参数是：`%4` 和 `%5`（分别用于指定第一个数据库的用户标识和密码）以及 `%6` 和 `%7`（分别用于指定第二个数据库的用户标识和密码）。

**过程:**

对于多连接样本程序 `dbmcon.exe`，需要两个数据库。如果尚未创建 `sample` 数据库，您可以通过在 DB2 命令窗口的命令行上输入 `db2samp1` 来创建它。第二个数据库（此处名为 `sample2`）可使用下列其中一个命令创建:

如果是本地创建数据库:

```
db2 create db sample2
```

如果是远程创建数据库:

```
db2 attach to <node_name>
db2 create db sample2
db2 detach
db2 catalog db sample2 as sample2 at node <node_name>
```

其中 `<node_name>` 是数据库驻留的节点。

多连接还要求 TCP/IP 侦听器正在运行。为了确保它正在运行，执行下列操作:

1. 按如下所示将环境变量 `DB2COMM` 设置为 TCP/IP:

```
db2set DB2COMM=TCP/IP
```

2. 按 `services` 文件中指定的那样，使用 TCP/IP 服务名称来更新数据库管理器配置文件:

```
db2 update dbm cfg using SVCENAME <TCP/IP service name>
```

每个实例都在 `services` 文件中列示了 TCP/IP 服务名称。如果找不到该服务名称或没有更改 `services` 文件的文件许可权，请向系统管理员咨询。

3. 停止数据库管理器然后重新启动它以使这些更改生效:

```
db2stop  
db2start
```

dbmcon.exe 程序是从 samples\c 或 samples\cpp 目录中的五个文件创建的:

**dbmcon.sqc 或 dbmcon.sqx**

用来连接至两个数据库的主源文件。

**dbmcon1.sqc 或 dbmcon1.sqx**

用来创建绑定至第一个数据库的程序包的源文件。

**dbmcon1.h**

包括在主源文件 dbmcon.sqc 或 dbmcon.sqx 中的 dbmcon1.sqc 或 dbmcon1.sqx 的头文件, 用来存取用于创建和删除绑定至第一个数据库的表的 SQL 语句。

**dbmcon2.sqc 或 dbmcon2.sqx**

用来创建绑定至第二个数据库的程序包的源文件。

**dbmcon2.h**

包括在主源文件 dbmcon.sqc 或 dbmcon.sqx 中的 dbmcon2.sqc 或 dbmcon2.sqx 的头文件, 用来存取用于创建和删除绑定至第二个数据库的表的 SQL 语句。

要构建多连接样本程序 dbmcon.exe, 请输入:

```
bldmc dbmcon sample sample2
```

得到一个可执行文件 dbmcon.exe。

要运行可执行文件, 请输入可执行文件名 (不带扩展名):

```
dbmcon
```

程序将演示向两个数据库的一阶段落实。

**相关概念:**

- 第 93 页的『构建文件』

**相关参考:**

- 『svcname - TCP/IP 服务名称配置参数』 (《管理指南: 性能》)
- 第 256 页的『Windows C/C++ 应用程序编译和链接选项』

**相关样本:**

- 『bldmc.bat -- Builds C multi-connection application on Windows』
- 『dbmcon.sqc -- How to use multiple databases (C)』
- 『dbmcon1.h -- Function declarations for the source file, dbmcon1.sqc (C)』
- 『dbmcon1.sqc -- Functions used in the multiple databases program dbmcon.sqc (C)』
- 『dbmcon2.h -- Function declarations for the source file, dbmcon2.sqc (C)』
- 『dbmcon2.sqc -- Functions used in the multiple databases program dbmcon.sqc (C)』
- 『bldmc.bat -- Builds C++ multi-connection application on Windows』
- 『dbmcon.sqC -- How to use multiple databases (C++)』
- 『dbmcon1.h -- Class declaration for the source file, dbmcon1.sqC (C++)』
- 『dbmcon1.sqC -- Functions used in the multiple databases program dbmcon.sqC (C++)』

- 『dbmcon2.h -- Class declaration for the source file, dbmcon2.sqC (C++)』
- 『dbmcon2.sqC -- Functions used in the multiple databases program dbmcon.sqC (C++)』

---

## IBM VisualAge COBOL

### 在 Windows 上配置 IBM COBOL 编译器

如果开发包含嵌入式 SQL 和 DB2 API 调用的应用程序，且正在使用 IBM VisualAge COBOL 编译器，则应注意以下几点。

#### 过程:

- 当使用 DB2 预编译器预编译应用程序并且使用命令行处理器命令 `db2 prep` 时，应使用 `target ibmcob` 选项。
- 在源文件中不要使用制表符。
- 在您的源文件中使用 `PROCESS` 和 `CBL` 关键字来设置编译选项。只能将这些关键字置于 8 至 72 列之间。
- 如果您的应用程序只包含嵌入式 SQL，而不包含 DB2 API 调用，则不需要使用 `pgmname(mixed)` 编译选项。如果使用 DB2 API 调用，则必须使用 `pgmname(mixed)` 编译选项。
- 如果您使用 IBM VisualAge COBOL 编译器的“System/390 主机数据类型支持”功能部件，则应用程序的 DB2 包含文件在以下目录中：

```
%DB2PATH%\include\cobol_i
```

如果要使用提供的批处理文件构建 DB2 样本程序，必须将在批处理文件中指定的包含文件路径更改为指向 `cobol_i` 目录而不是 `cobol_a` 目录。

如果不使用 IBM VisualAge COBOL 编译器的“System/390 主机数据类型支持”功能部件，或者使用此编译器的较早版本，则应用程序的 DB2 包含文件在以下目录中：

```
%DB2PATH%\include\cobol_a
```

`cobol_a` 目录是缺省目录。

- 按如下所示指定包括 `.cbl` 扩展名的 COPY 文件名：  
`COPY "sql.cbl"`。

#### 相关任务:

- 第 264 页的『在 Windows 上构建 IBM COBOL 应用程序』
- 第 268 页的『在 Windows 上构建 IBM COBOL 例程』

#### 相关参考:

- 第 267 页的『Windows IBM COBOL 应用程序编译和链接选项』
- 第 270 页的『Windows IBM COBOL 例程编译和链接选项』

### 在 Windows 上构建 IBM COBOL 应用程序

DB2 提供了用于编译和链接 DB2 API 和嵌入式 SQL 程序的批处理文件。这些批处理文件和可以使用这些文件构建的样本程序一同位于 `sql1ib\samples\cobol` 目录中。

在 Windows 上，DB2 支持两个用于构建 IBM COBOL 应用程序的预编译器，即 DB2 预编译器和 IBM COBOL 预编译器。缺省值为 DB2 预编译器。可通过取消注释正在使用的批处理文件中的相应行来选择 IBM COBOL 预编译器。使用 IBM COBOL 进行的预编译是由编译器本身使用特定的预编译选项完成的。

批处理文件 `bldapp.bat` 包含构建 DB2 应用程序的命令。它最多使用四个参数，这些参数在批处理文件中以变量 `%1`、`%2`、`%3` 和 `%4` 表示。

第一个参数 `%1` 指定源文件名。这是不包含嵌入式 SQL 的程序所需的唯一参数。构建嵌入式 SQL 程序需要连接数据库，因此还需要提供三个可选参数：第二个参数 `%2` 指定您想连接的数据库的名称；第三个参数 `%3` 指定数据库的用户标识；第四个参数 `%4` 指定密码。

对于使用缺省 DB2 预编译器的嵌入式 SQL 程序，`bldapp.bat` 将这些参数传递到预编译和绑定文件 `embprep.bat`。

对于使用 IBM COBOL 预编译器的嵌入式 SQL 程序，`bldapp.bat` 将 `.sqb` 源文件复制到 `.cbl` 源文件。编译器使用特定的预编译选项对 `.cbl` 源文件执行预编译。

对于任一预编译器，如果未提供数据库名称，则使用缺省 `sample` 数据库。仅当构建程序所在的实例不同于数据库所在的实例时，才需要用户标识和密码参数。

#### 过程:

下列示例显示如何构建和运行 DB2 API 和嵌入式 SQL 应用程序。

要从源文件 `client.cbl` 构建非嵌入式 SQL 样本程序 `client`，请输入：

```
bldapp client
```

得到一个可执行文件 `client.exe`。可输入可执行文件名（不带扩展名），来对 `sample` 数据库运行该可执行文件：

```
client
```

#### 构建和运行嵌入式 SQL 应用程序

从源文件 `updat.sqb` 构建嵌入式 SQL 应用程序 `updat` 有三种方法：

1. 如果与同一实例中的样本数据库连接，则输入：

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接，还需要输入该数据库名称：

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接，还需要输入该数据库实例的用户标识和密码：

```
bldapp updat database userid password
```

得到一个可执行文件 `updat`。

有三种方法运行此嵌入式 SQL 应用程序：

1. 如果存取同一实例中的 `sample` 数据库，只须输入可执行文件名：

```
updat
```

2. 如果存取同一实例中的另一个数据库，则输入可执行文件名和数据库名称：

```
updat database
```

3. 如果存取另一个实例中的数据库，则输入可执行文件名、数据库名称以及该数据库实例的用户标识和密码：

```
updat database userid password
```

#### 相关概念：

- 第 93 页的『构建文件』

#### 相关参考：

- 第 267 页的『Windows IBM COBOL 应用程序编译和链接选项』
- 第 71 页的『COBOL 样本』

#### 相关样本：

- 『bldapp.bat -- Builds Windows VisualAge COBOL applications』
- 『client.cbl -- How to set and query a client (IBM COBOL)』
- 『embprep.bat -- To prep and bind a COBOL embedded SQL program on Windows』
- 『updat.sqb -- How to update, delete and insert table data (IBM COBOL)』

## IBM COBOL 应用程序的批处理文件

```
@echo off
rem BATCH FILE: bldapp.bat
rem Builds Windows VisualAge COBOL applications
rem Usage: bldapp prog_name [ db_name [ userid password ] ]

set IBMCOB_PRECOMP=
set EXTRA_COMPFLAG=

rem To use the IBM COBOL precompiler, uncomment the following line.
rem set IBMCOB_PRECOMP=true

rem If using the IBM COBOL precompiler
if "%IBMCOB_PRECOMP%" == "true" goto IBMCOB_precompile_step

rem Using the default DB2 precompiler,
rem If an embedded SQL program, precompile and bind it.
if not exist "%1.sqb" goto compile_step
call embprep %1 %2 %3 %4
goto compile_step

:IBMCOB_precompile_step
rem Using the IBM COBOL precompiler,
rem Copy the <prog_name>.sqb file to <prog_name>.cbl.
if exist "%1.sqb" cp -f %1.sqb %1.cbl
rem Assign input parameters to the EXTRA_COMPFLAG variable
if "%1" == "" goto error
if "%2" == "" goto case1
if "%3" == "" goto case2
if "%4" == "" goto error
goto case3

:case1
set EXTRA_COMPFLAG=-q"SQL('database sample CALL_RESOLUTION DEFERRED')"
goto compile_step
:case2
set EXTRA_COMPFLAG=-q"SQL('database %2 CALL_RESOLUTION DEFERRED')"
goto compile_step
:case3
set EXTRA_COMPFLAG=-q"SQL('database %2 user %3 using %4
CALL_RESOLUTION DEFERRED')"
goto compile_step
```

```

:compile_step
rem Compile the error-checking utility.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" checkerr.cbl

rem Compile the program.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" %1.cbl
    %EXTRA_COMPFLAG%

rem Link the program.
cob2 %1.obj checkerr.obj db2api.lib
goto exit

:error
echo Usage: bldapp prog_name [ db_name [ userid password ]]

:exit
@echo on

```

## Windows IBM COBOL 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 IBM VisualAge COBOL 编译器在 Windows 上构建 COBOL 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 bldapp.bat 批处理文件中所演示的那样。

bldapp 的编译和链接选项	
<p><b>编译选项:</b></p> <p><b>cob2</b> IBM VisualAge COBOL 编译器。</p> <p><b>-qpgmname(mixed)</b> 指示编译器可用大小写混合的名称调用 (CALL) 库入口点。</p> <p><b>-c</b> 只执行编译; 不链接。编译和链接是两个独立的步骤。</p> <p><b>-qlib</b> 指示编译器处理 COPY 语句。</p> <p><b>-Ipath</b> 指定 DB2 包含文件的位置。例如: <code>-I"%DB2PATH%\include\cobol_a"</code>。</p> <p><b>%EXTRA_COMPFLAG%</b> 如果未注释 "set IBMCOB_PRECOMP=true", 则 IBM COBOL 预编译选项与下列模式之一配合使用, 这取决于输入参数:</p> <p><b>-q"SQL('database sample CALL_RESOLUTION DEFERRED')"</b> 使用缺省样本数据库预编译并延迟调用解决方案。</p> <p><b>-q"SQL('database %2 CALL_RESOLUTION DEFERRED')"</b> 使用用户指定的数据库预编译并延迟调用解决方案。</p> <p><b>-q"SQL('database %2 user %3 using %4 CALL_RESOLUTION DEFERRED')"</b> 使用用户指定的数据库、用户标识和密码预编译并延迟调用解决方案。这是远程客户机存取的形式。</p>	
<p><b>链接选项:</b></p> <p><b>cob2</b> 将编译器用作链接程序的前端</p> <p><b>%1.obj</b> 包括该程序的对象文件。</p> <p><b>checkerr.obj</b> 包括错误检查实用程序的对象文件。</p> <p><b>db2api.lib</b> 链接 DB2 库。</p> <p>有关其它编译器选项, 参考编译器文档。</p>	

#### 相关任务:

- 第 264 页的『在 Windows 上构建 IBM COBOL 应用程序』

#### 相关样本:

- 『bldapp.bat -- Builds Windows VisualAge COBOL applications』

## 在 Windows 上构建 IBM COBOL 例程

DB2 提供了用于编译和链接用 IBM COBOL 编写的 DB2 API 和嵌入式 SQL 程序的批处理文件。这些批处理文件和可以使用这些文件构建的样本程序一同位于 `sqllib\samples\cobol` 目录中。

在 Windows 上，DB2 支持两个用于构建 IBM COBOL 应用程序的预编译器，即 DB2 预编译器和 IBM COBOL 预编译器。缺省值为 DB2 预编译器。可通过取消注释正在使用的批处理文件中的相应行来选择 IBM COBOL 预编译器。使用 IBM COBOL 进行的预编译是由编译器本身使用特定的预编译选项完成的。

批处理文件 `bldrtn.bat` 包含构建嵌入式 SQL 例程（存储过程）的命令。该批处理文件在服务器上例程编译成 DLL。它使用两个参数，这两个参数在批处理文件中以变量 `%1` 和 `%2` 表示。

第一个参数 `%1` 指定源文件名。此批处理文件使用源文件名 `%1` 作为 DLL 名。第二个参数 `%2` 指定想要连接的数据库的名称。因为必须在数据库所在的同一实例中构建存储过程，所以不需要用户标识和密码的参数。

只有第一个参数即源文件名是必需的。数据库名称是可选的。如果未提供数据库名称，则该程序使用缺省的 `sample` 数据库。

如果使用缺省 DB2 预编译器，则 `bldrtn.bat` 将这些参数传递到预编译和绑定文件 `embprep.bat`。

如果使用 IBM COBOL 预编译器，则 `bldrtn.bat` 将 `.sqb` 源文件复制到 `.cbl` 源文件。编译器使用特定的预编译选项对 `.cbl` 源文件执行预编译。

#### 过程:

要从源文件 `outsrv.sqb` 构建样本程序 `outsrv`，连接至样本数据库而后输入:

```
bldrtn outsrv
```

如果连接另一个数据库，还需要输入该数据库名称:

```
bldrtn outsrv database
```

批处理文件将 DLL 复制到路径 `sqllib\function` 中的服务器。

一旦构建了 DLL `outsrv`，就可以构建客户机应用程序 `outcli`，该应用程序将调用 DLL 内的例程（该例程与 DLL 同名）。可以使用批处理文件 `bldapp.bat` 构建 `outcli`。

要调用 `outsrv` 例程，通过输入下列命令来运行样本客户机应用程序:

```
outcli database userid password
```

其中，



**database**

是要连接的数据库的名称。该名称可以是 sample、其远程别名或其它名称。

**userid** 是有效的用户标识。

**password**

是用户标识的有效密码。

客户机应用程序将存取 DLL (outsrv)，并在服务器数据库上执行同名的例程，然后将输出返回给客户机应用程序。

**相关概念:**

- 第 93 页的『构建文件』

**相关参考:**

- 第 270 页的『Windows IBM COBOL 例程编译和链接选项』
- 第 71 页的『COBOL 样本』

**相关样本:**

- 『bldrtn.bat -- Builds Windows VisualAge COBOL routines (stored procedures)』
- 『embprep.bat -- To prep and bind a COBOL embedded SQL program on Windows』
- 『outcli.sqb -- Call stored procedures using the SQLDA structure (IBM COBOL)』
- 『outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (IBM COBOL)』

## IBM COBOL 例程的批处理文件

```

@echo off
rem BATCH FILE: bldrtn.bat
rem Builds Windows VisualAge COBOL routines (stored procedures)
rem Usage: bldrtn prog_name [ db_name ]

set IBMCOB_PRECOMP=
set EXTRA_COMPFLAG=

rem To use the IBM COBOL precompiler, uncomment the following line.
rem set IBMCOB_PRECOMP=true

rem If using the IBM COBOL precompiler
if "%IBMCOB_PRECOMP%" == "true" goto IBMCOB_precompile_step

rem Using the default DB2 precompiler,
rem Precompile and bind the program.
call embprep %1 %2
goto compile_step

:IBMCOB_precompile_step
rem Using the IBM COBOL precompiler,
rem Copy the <prog_name>.sqb file to <prog_name>.cbl.
if exist "%1.sqb" cp -f %1.sqb %1.cbl
rem Assign input parameters to the EXTRA_COMPFLAG variable
if "%1" == "" goto error
if "%2" == "" goto case1

set EXTRA_COMPFLAG=-q"SQL('database %2 CALL_RESOLUTION DEFERRED')"
goto compile_step

:case1
set EXTRA_COMPFLAG=-q"SQL('database sample CALL_RESOLUTION DEFERRED')"

```

```

        goto compile_step

:compile_step
rem Compile the stored procedure.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" %1.cbl %EXTRA_COMPFLAG%

rem Link the stored procedure and create a shared library.
ilib /no1 /gi:%1 %1.obj
ilink /free /no1 /dll db2api.lib %1.exp %1.obj iwzrwin3.obj

rem Copy stored procedure to the %DB2PATH%\function directory.
copy %1.dll "%DB2PATH%\function"
goto exit

:error
echo Usage: bldrtn prog_name [ db_name ]

:exit
@echo on

```

## Windows IBM COBOL 例程编译和链接选项

下列内容是 DB2 建议用于使用 IBM VisualAge COBOL 编译器在 Windows 上构建 COBOL 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn.bat 批处理文件中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>cob2</b>	IBM VisualAge COBOL 编译器。
<b>-qpgmname(mixed)</b>	指示编译器可用大小写混合的名称调用 (CALL) 库入口点。
<b>-c</b>	只执行编译; 不链接。此批处理文件有单独的编译和链接步骤。
<b>-qlib</b>	指示编译器处理 COPY 语句。
<b>-Ipath</b>	指定 DB2 包含文件的位置。例如: -I"%DB2PATH%\include\cobol_a"。
<b>%EXTRA_COMPFLAG%</b>	如果未注释 "set IBMCOB_PRECOMP=true", 则 IBM COBOL 预编译选项与下列模式之一配合使用, 这取决于输入参数:
<b>-q"SQL('database sample CALL_RESOLUTION DEFERRED')"</b>	使用缺省样本数据库预编译并延迟调用解决方案。
<b>-q"SQL('database %2 CALL_RESOLUTION DEFERRED')"</b>	使用用户指定的数据库预编译并延迟调用解决方案。

## bldrtn 的编译和链接选项

### 链接选项:

**i1ink** 使用 IBM VisualAge COBOL 链接程序。  
**/free** 自由格式。  
**/no1** 无徽标。  
**/dll** 创建具有源程序名的 DLL。  
**db2api.lib**  
    链接 DB2 库。  
**%1.exp** 包括导出文件。  
**%1.obj** 包括该程序的对象文件。  
**iwzrwin3.obj**  
    包括 IBM VisualAge COBOL 提供的对象文件。

有关其它编译器选项, 参考编译器文档。

### 相关任务:

- 第 268 页的『在 Windows 上构建 IBM COBOL 例程』

### 相关样本:

- 『bldrtn.bat -- Builds Windows VisualAge COBOL routines (stored procedures)』

## Micro Focus COBOL

### 在 Windows 上配置 Micro Focus COBOL 编译器

如果开发包含嵌入式 SQL 和 DB2 API 调用的应用程序, 且正在使用 Micro Focus 编译器, 则应注意以下几点。

### 过程:

- 当使用命令行处理器命令 `db2 prep` 预编译应用程序时, 使用 `target mfcob` 选项。
- 确保 LIB 环境变量指向 `%DB2PATH%\lib`, 如下所示:

```
set LIB="%DB2PATH%\lib;%LIB%"
```

- Micro Focus COBOL 的 DB2 COPY 文件驻留在 `%DB2PATH%\include\cobol_mf` 中。为包括该目录, 应按如下所示设置 COBCPY 环境变量:

```
set COBCPY="%DB2PATH%\include\cobol_mf;%COBCPY%"
```

必须使用调用约定 74 来对所有 DB2 应用程序编程接口进行调用。DB2 COBOL 预编译器自动在 SPECIAL-NAMES 段中插入 CALL-CONVENTION 子句。如果 SPECIAL-NAMES 一段不存在, 则 DB2 COBOL 预编译器将按如下所示创建它:

```
Identification Division  
Program-ID. "static".  
special-names.  
    call-convention 74 is DB2API.
```

而且, 无论何时调用 DB2 API, 预编译器都会自动将符号 DB2API (用于标识调用约定) 置于 “call” 关键字之后。例如, 每当预编译器从嵌入式 SQL 语句生成 DB2 API 运行时调用时, 会发生此情况。

如果在未预编译的应用程序中调用 DB2 API，应以类似上面给定的方式在该应用程序中手工创建 SPECIAL-NAMES 段。如果正直接调用 DB2 API，则将需要在“call”关键字之后手工添加 DB2API 符号。

**相关任务:**

- 第 272 页的『在 Windows 上构建 Micro Focus COBOL 应用程序』
- 第 274 页的『在 Windows 上构建 Micro Focus COBOL 例程』

**相关参考:**

- 第 273 页的『Windows Micro Focus COBOL 应用程序编译和链接选项』
- 第 276 页的『Windows Micro Focus COBOL 例程编译和链接选项』

## 在 Windows 上构建 Micro Focus COBOL 应用程序

DB2 提供了用于编译和链接 DB2 API 和嵌入式 SQL 程序的批处理文件。这些批处理文件与可以使用这些文件构建的样本程序一同位于 `sql1lib\samples\cobol_mf` 目录中。

批处理文件 `bldapp.bat` 包含构建 DB2 应用程序的命令。它最多使用四个参数，这些参数在批处理文件中以变量 `%1`、`%2`、`%3` 和 `%4` 表示。

第一个参数 `%1` 指定源文件名。这是不包含嵌入式 SQL 的程序所需的唯一参数。构建嵌入式 SQL 程序需要连接数据库，因此还需要提供三个可选参数：第二个参数 `%2` 指定您想连接的数据库的名称；第三个参数 `%3` 指定数据库的用户标识；第四个参数 `%4` 指定密码。

对于嵌入式 SQL 程序，`bldapp` 将这些参数传递到预编译和绑定批处理文件 `embprep.bat`。如果未提供数据库名称，则使用缺省 `sample` 数据库。仅当构建程序所在的实例不同于数据库所在的实例时，才需要用户标识和密码参数。

**过程:**

下列示例显示如何构建和运行 DB2 API 和嵌入式 SQL 应用程序。

要从源文件 `client.cbl` 构建非嵌入式 SQL 样本程序 `client`，请输入：

```
bldapp client
```

得到一个可执行文件 `client.exe`。可输入可执行文件名（不带扩展名），来对 `sample` 数据库运行该可执行文件：

```
client
```

### 构建和运行嵌入式 SQL 应用程序

从源文件 `updat.sqb` 构建嵌入式 SQL 应用程序 `updat` 有三种方法：

1. 如果与同一实例中的样本数据库连接，则输入：

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接，还需要输入该数据库名称：

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接，还需要输入该数据库实例的用户标识和密码：

```
bldapp updat database userid password
```

得到一个可执行文件 `updat.exe`。

有三种方法运行此嵌入式 SQL 应用程序：

1. 如果存取同一实例中的 `sample` 数据库，只须输入可执行文件名（不需输入扩展名）：

```
updat
```

2. 如果存取同一实例中的另一个数据库，则输入可执行文件名和数据库名称：

```
updat database
```

3. 如果存取另一个实例中的数据库，则输入可执行文件名、数据库名称以及该数据库实例的用户标识和密码：

```
updat database userid password
```

相关概念：

- 第 93 页的『构建文件』

相关参考：

- 第 273 页的『Windows Micro Focus COBOL 应用程序编译和链接选项』
- 第 71 页的『COBOL 样本』

相关样本：

- 『`bldapp.bat` -- Builds Windows Micro Focus Cobol applications』
- 『`client.cbl` -- How to set and query a client (MF COBOL)』
- 『`updat.sqb` -- How to update, delete and insert table data (MF COBOL)』
- 『`embprep.bat` -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows』

## Micro Focus COBOL 应用程序的批处理文件

```
@echo off
rem BATCH FILE: bldapp.bat
rem Builds Windows Micro Focus Cobol applications
rem Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

rem If an embedded SQL program, precompile and bind it.
if not exist "%1.sqb" goto compile_step
call embprep %1 %2 %3 %4

:compile_step
rem Compile the error-checking utility.
cobol checkerr.cbl;

rem Compile the program.
cobol %1.cbl;

rem Link the program.
cbl1link -l %1.obj checkerr.obj db2api.lib
@echo on
```

## Windows Micro Focus COBOL 应用程序编译和链接选项

下列内容是 DB2 建议用于使用 Micro Focus COBOL 编译器在 Windows 上构建 COBOL 嵌入式 SQL 和 DB2 API 应用程序的编译和链接选项，如在 `bldapp.bat` 批处理文件中所演示的那样。

<b>bldapp</b> 的编译和链接选项	
编译选项:	
<b>cobol</b>	Micro Focus COBOL 编译器。
链接选项:	
<b>cbllink</b>	使用链接程序链接编辑。
<b>-l</b>	链接 <code>lcobol</code> 库。
<b>checkerr.obj</b>	链接错误检查实用程序的对象文件。
<b>db2api.lib</b>	链接 DB2 API 库。
有关其它编译器选项, 参考编译器文档。	

#### 相关任务:

- 第 272 页的『在 Windows 上构建 Micro Focus COBOL 应用程序』

#### 相关样本:

- 『`bldapp.bat -- Builds Windows Micro Focus Cobol applications`』

## 在 Windows 上构建 Micro Focus COBOL 例程

DB2 提供了用于编译和链接使用 Micro Focus COBOL 编写的 DB2 API 及嵌入式 SQL 程序的批处理文件。这些批处理文件与可以使用这些文件构建的样本程序一同位于 `sqllib\samples\cobol_mf` 目录中。

批处理文件 `bldrtn.bat` 包含构建嵌入式 SQL 例程（存储过程）的命令。该批处理文件在服务器上例程编译成 DLL。该批处理文件使用两个参数，这两个参数在批处理文件中以变量 `%1` 和 `%2` 表示。

第一个参数 `%1` 指定源文件名。此批处理文件使用源文件名 `%1` 作为 DLL 名。第二个参数 `%2` 指定想要连接的数据库的名称。因为必须在数据库所在的同一实例中构建存储过程，所以不需要用户标识和密码的参数。

只有第一个参数即源文件名是必需的。数据库名称是可选的。如果未提供数据库名称，则该程序使用缺省的 `sample` 数据库。

#### 过程:

要从源文件 `outsrv.sqb` 构建样本程序 `outsrv`，如果连接样本数据库，则输入:

```
bldrtn outsrv
```

如果连接另一个数据库，还需要输入该数据库名称:

```
bldrtn outsrv database
```

此脚本文件将 DLL 复制到服务器的 `sqllib/function` 路径中。

一旦构建了 DLL (`outsrv`)，就可以构建客户机应用程序 `outcli`，该应用程序将调用 DLL 内的例程（该例程与 DLL 同名）。可以使用批处理文件 `bldapp.bat` 构建 `outcli`。

要调用 `outsrv` 例程，通过输入下列命令来运行样本客户机应用程序：

```
outcli database userid password
```

其中，

**database**

是要连接的数据库的名称。该名称可以是 `sample`，或它的别名，或另一个数据库名称。

**userid** 是有效的用户标识。

**password**

是用户标识的有效密码。

客户机应用程序将存取 DLL (`outsrv`)，它在服务器数据库上执行同名的例程。然后将输出返回给该客户机应用程序。

**相关概念：**

- 第 93 页的『构建文件』

**相关参考：**

- 第 276 页的『Windows Micro Focus COBOL 例程编译和链接选项』
- 第 71 页的『COBOL 样本』

**相关样本：**

- 『`bldrtn.bat` -- Builds Windows Micro Focus Cobol routines (stored procedures)』
- 『`outcli.sqb` -- Call stored procedures using the SQLDA structure (MF COBOL)』
- 『`outsrv.sqb` -- Demonstrates stored procedures using the SQLDA structure (MF COBOL)』
- 『`embprep.bat` -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows』

## Micro Focus COBOL 例程的批处理文件

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Builds Windows Micro Focus Cobol routines (stored procedures)
rem Usage: bldsrv <prog_name> [ <db_name> ]

rem Precompile and bind the program.
call embprep %1 %2

rem Compile the stored procedure.
cobol %1.cbl /case;

rem Link the stored procedure and create a shared library.
cbllink /d %1.obj db2api.lib

rem Copy the stored procedure to the %DB2PATH%\function directory.
copy %1.dll "%DB2PATH%\function"
@echo on
```

## Windows Micro Focus COBOL 例程编译和链接选项

下列内容是 DB2 建议用于使用 Micro Focus COBOL 编译器在 Windows 上构建 COBOL 例程（存储过程和用户定义的函数）的编译和链接选项，如在 bldrtn.bat 批处理文件中所演示的那样。

bldrtn 的编译和链接选项	
编译选项:	
<b>cobol</b>	Micro Focus COBOL 编译器。
<b>/case</b>	防止将外部符号转换为大写。
链接选项:	
<b>cbllink</b>	使用 Micro Focus COBOL 链接程序链接编辑。
<b>/d</b>	创建 .dll 文件。
<b>db2api.lib</b>	链接 DB2 API 库。
有关其它编译器选项，参考编译器文档。	

### 相关任务:

- 第 274 页的『在 Windows 上构建 Micro Focus COBOL 例程』

### 相关样本:

- 『bldrtn.bat -- Builds Windows Micro Focus Cobol routines (stored procedures)』

---

## Object REXX

### 在 Windows 上构建 Object REXX 应用程序

Object REXX 是 REXX 语言的面向对象版本。面向对象扩充已添加到经典 REXX 中，但其现有的函数和指令未更改。Object REXX 解释器是其先前版本的增强版本，它还支持:

- 类、对象和方法
- 消息传递和多态性
- 单继承性和多重继承性

Object REXX 与 classic REXX 完全兼容。在本节中，只要用到 REXX，都指所有 REXX 版本（包括 Object REXX）。

不要预编译或绑定 REXX 程序。

在 Windows 上，REXX 程序不需要以注释开始。然而，为便于移植，建议每个 REXX 程序的第一行第一列以注释开始。这将使程序能够与其它平台上的批处理命令区分开来:

```
/* Any comment will do. */
```

可以在目录 sqllib\samples\rexx 中找到 REXX 样本程序。



过程:

要运行 REXX 样本程序 updat, 请输入:

```
rexx updat.cmd
```

相关概念:

- 『Programming Considerations for REXX』 (*Application Development Guide: Programming Client Applications*)

相关参考:

- 第 85 页的『REXX 样本』



---

## 第 4 部分 附录



---

## 附录 A. DB2 通用数据库技术信息

---

### DB2 文档和帮助

DB2<sup>®</sup> 技术信息可通过下列工具和方法获得:

- DB2 信息中心
  - 主题
  - DB2 工具的帮助
  - 样本程序
  - 教程
- 可下载的 PDF 文件、CD 上的 PDF 文件和印刷书籍
  - 指南
  - 参考手册
- 命令行帮助
  - 命令帮助
  - 消息帮助
  - SQL 状态帮助
- 已安装的源代码
  - 样本程序

可以在线访问 [ibm.com](http://ibm.com)<sup>®</sup> 上的其它 DB2 Universal Database<sup>™</sup> (DB2 通用数据库) 技术信息, 例如, 技术说明、白皮书和 Redbooks<sup>™</sup> (红皮书)。访问位于以下网址的 DB2 信息管理软件资料库站点: [www.ibm.com/software/data/pubs/](http://www.ibm.com/software/data/pubs/)。

### DB2 文档更新

IBM<sup>®</sup> 可能会定期提供 DB2 信息中心的文档修订包和其它文档更新。如果访问 <http://publib.boulder.ibm.com/infocenter/db2help/> 网址中的 DB2 信息中心, 则将始终可以查看最新的信息。如果本地安装了 DB2 信息中心, 则需要手工安装所有更新才能查看它们。文档更新允许您在新信息可供使用时更新从 *DB2 信息中心 CD* 安装的信息。

信息中心的更新比 PDF 或硬拷贝书籍的更新要频繁。要获得最新的 DB2 技术信息, 一提供文档更新时就安装它们, 或者访问 [www.ibm.com](http://www.ibm.com) 站点上的 DB2 信息中心。

#### 相关概念:

- 『CLI sample programs』 (*CLI Guide and Reference, Volume 1*)
- 第 103 页的『Java 样本程序』
- 第 282 页的『DB2 信息中心』

#### 相关任务:

- 第 297 页的『从 DB2 工具调用上下文帮助』
- 第 290 页的『更新安装在计算机或内部网服务器上的 DB2 信息中心』
- 第 298 页的『从命令行处理器调用消息帮助』

- 第 298 页的『从命令行处理器调用命令帮助』
- 第 299 页的『从命令行处理器调用 SQL 状态帮助』

**相关参考:**

- 第 291 页的『DB2 PDF 和印刷文档』

---

## DB2 信息中心

DB2<sup>®</sup> 信息中心使您可以访问充分利用 DB2 系列产品（包括 DB2 Universal Database<sup>™</sup>（DB2 通用数据库）、DB2 Connect<sup>™</sup>、DB2 Information Integrator 和 DB2 Query Patroller<sup>™</sup>）所需的所有信息。DB2 信息中心还包含主要的 DB2 功能部件和组件（包括复制、数据仓储和 DB2 extender）的信息。

如果是在 Mozilla 1.0（或更新版本）或 Microsoft<sup>®</sup> Internet Explorer 5.5（或更新版本）中查看的话，则 DB2 信息中心具有下列功能部件。某些功能部件需要您启用对 JavaScript<sup>™</sup> 的支持：

### 灵活安装选项

可选择使用最适合您的需要的选项来查看 DB2 文档：

- 要轻松确保文档始终是最新的，可直接从 IBM<sup>®</sup> Web 站点上的 DB2 信息中心访问所有文档，网址为：<http://publib.boulder.ibm.com/infocenter/db2help/>
- 要将更新工作量减至最少并使网络通信保持在内部网内，可将 DB2 文档安装在内部网上的单台服务器上
- 要使您有最大的灵活性并减少对网络连接的依赖，可将 DB2 文档安装在您自己的计算机上

**搜索** 可通过在搜索文本字段中输入搜索术语来搜索 DB2 信息中心中的所有主题。可通过用引号将术语括起来以检索确定匹配项，还可以使用通配运算符（\* 和 ?）和布尔运算符（AND、NOT 和 OR）细化搜索。

### 面向任务的目录

可从单个目录查找 DB2 文档中的主题。目录主要是按想要执行的任务的种类组织的，同时也包括有关产品概述、目标、参考信息、索引和词汇表的条目。

- 产品概述描述 DB2 系列中的可用产品之间的关系、其中每个产品提供的功能部件以及其中每个产品的最新发行信息。
- 目标类别（例如，安装、管理和开发）包括一些主题，这些主题使您能够快速地完成并且更好地理解完成这些任务的背景信息。
- 参考主题提供有关主题的详细信息，包括语句和命令语法、消息帮助以及配置参数。

### 显示目录中的当前主题

可通过单击目录框架中的刷新 / 显示当前主题按钮或通过单击内容框架中的在目录中显示按钮来显示当前主题在目录中的位置。如果访问了指向若干个文件中的相关主题的若干个链接，或者是从搜索结果到达主题的，此功能会非常有用。

**索引** 可从索引访问全部文档。索引是按索引项的拼音顺序组织的。

**词汇表** 可使用词汇表来查找在 DB2 文档中使用的术语的定义。词汇表是按词汇表术语的拼音顺序组织的。

## 集成的本地化信息

DB2 信息中心以您在浏览器首选项中设置的首选语言显示信息。如果主题不是以首选语言提供的，则 DB2 信息中心将显示该主题的英文版。

有关 iSeries™ 技术信息，参阅 IBM eServer™ iSeries 信息中心，网址为 [www.ibm.com/eserver/iseries/infocenter/](http://www.ibm.com/eserver/iseries/infocenter/)。

### 相关概念:

- 第 283 页的『DB2 信息中心安装方案』

### 相关任务:

- 第 290 页的『更新安装在计算机或内部网服务器上的 DB2 信息中心』
- 第 290 页的『以首选语言显示 DB2 信息中心中的主题』
- 第 289 页的『调用 DB2 信息中心』
- 第 285 页的『使用“DB2 安装”向导来安装 DB2 信息中心（UNIX）』
- 第 287 页的『使用“DB2 安装”向导来安装 DB2 信息中心（Windows）』

---

## DB2 信息中心安装方案

对于如何访问 DB2® 信息，不同的工作环境可有不同的需求。可以使用三种方法访问 DB2 信息中心：从 IBM® Web 站点访问、从组织网络的服务器访问或从安装在计算机上的版本访问。在所有三种情况中，文档都包含在 DB2 信息中心中，DB2 信息中心是基于主题的信息的结构化 Web，可使用浏览器来查看。缺省情况下，DB2 产品从 IBM Web 站点访问 DB2 信息中心。但是，如果想要从内部网服务器或从您自己的计算机访问 DB2 信息中心，必须使用产品“介质包”中的 DB2 信息中心 CD 来安装 DB2 信息中心。参阅下面的用于访问 DB2 文档的选项的总结及三个方案来帮助确定访问 DB2 信息中心时使用哪个方法最适合您和您的工作环境，以及可能需要考虑哪些安装问题。

### 用于访问 DB2 文档的选项的总结:

下表提供了有关哪些选项可能用来在您的工作环境中访问 DB2 信息中心中的 DB2 产品文档的建议。

因特网访问	内部网访问	建议
是	是	访问 IBM Web 站点上的 DB2 信息中心，或者访问安装在内部网服务器上的 DB2 信息中心。
是	否	访问 IBM Web 站点上的 DB2 信息中心。
否	是	访问安装在内部网服务器上的 DB2 信息中心。
否	否	访问本地计算机上的 DB2 信息中心。

### 方案：访问您的计算机上的 DB2 信息中心:

Tsu-Chen 在一个小镇上开了一家工厂，而小镇没有本地 ISP，所以他不能访问因特网。他购买了 DB2 Universal Database™（DB2 通用数据库）来管理库存、产品订单、银行帐户信息和业务开销。由于以前从未使用过 DB2 产品，所以 Tsu-Chen 需要从 DB2 产品文档了解产品的使用方法。

使用典型安装选项在计算机上安装了 DB2 通用数据库之后，Tsu-Chen 尝试访问 DB2 文档。但是，浏览器显示一条错误消息，提示找不到他尝试打开的页。Tsu-Chen 查阅了 DB2 产品的安装手册，发现如果想要访问计算机上的 DB2 文档，则必须安装 DB2 信息中心。他在介质包中找到了 *DB2 信息中心 CD* 并安装了它。

Tsu-Chen 现在能够从操作系统的应用程序启动程序访问 DB2 信息中心，也能够了解如何使用 DB2 产品来增加业务的成功率。

#### 方案：访问 IBM Web 站点上的 DB2 信息中心：

Colin 是培训公司的一名信息技术顾问。他精通数据库技术和 SQL 并对全北美使用 DB2 通用数据库的企业提供有关这些主题的讲座。Colin 的部分讲座包括将 DB2 文档用作教学工具。例如，在讲授有关 SQL 的课程时，Colin 使用有关 SQL 的 DB2 文档作为教授数据库查询的基本和高级语法的方法。

Colin 授课的大部分企业都访问因特网。当 Colin 安装了 DB2 通用数据库的最新版本时，他会决定配置其移动式计算机以访问 IBM Web 站点上的 DB2 信息中心。此配置允许 Colin 在授课期间在线访问最新的 DB2 文档。

但是，有时在旅行时 Colin 不能访问因特网。这对他来说是个问题，尤其是在他需要访问 DB2 文档来备课时。为避免类似情况，Colin 在他的移动式计算机上安装了 DB2 信息中心的副本。

Colin 可以随时很方便的获得 DB2 文档的副本。通过使用 **db2set** 命令，他可以根据所在位置很容易地将其移动式计算机上的注册表变量配置为访问 IBM Web 站点上或他的移动式计算机上的 DB2 信息中心。

#### 方案：访问内部网服务器上的 DB2 信息中心：

Eva 是一家人寿保险公司的高级数据库管理员。她的管理职责包括在公司的 UNIX<sup>®</sup> 数据库服务器上安装和配置 DB2 通用数据库的最新版本。她的公司最近通知职员，为安全起见，在工作时间将不提供因特网访问。因为公司具有联网环境，所以 Eva 决定在内部网服务器上安装 DB2 信息中心，以便公司中经常使用公司数据仓库的所有职员（销售代表、销售经理和业务分析人员）都可以访问 DB2 文档。

Eva 会指导她的数据库小组使用响应文件在所有职员的计算机上安装 DB2 通用数据库的最新版本，以确保每台计算机都配置为使用内部网服务器的主机名和端口号来访问 DB2 信息中心。

但是，Eva 的小组中的初级数据库管理员 Migual 误解了 Eva 的意思，她在一些职员的计算机上安装了 DB2 信息中心的副本，但没有将 DB2 通用数据库配置为访问内部网服务器上的 DB2 信息中心。为了更正这种情况，Eva 告诉 Migual 使用 **db2set** 命令在这些计算机中的每一台上更改 DB2 信息中心注册表变量（DB2\_DOCHOST 表示主机名，而 DB2\_DOCPORT 表示端口号）。现在，该网络上的所有相应计算机都可以访问 DB2 信息中心，而且职员可在 DB2 文档中查找有关 DB2 问题的答案。

#### 相关概念：

- 第 282 页的『DB2 信息中心』

#### 相关任务：

- 第 290 页的『更新安装在计算机或内部网服务器上的 DB2 信息中心』



- 第 285 页的『使用“DB2 安装”向导来安装 DB2 信息中心 (UNIX)』
- 第 287 页的『使用“DB2 安装”向导来安装 DB2 信息中心 (Windows)』
- 『设置访问 DB2 信息中心的位置: 公共 GUI 帮助』

**相关参考:**

- 『db2set - DB2 Profile Registry Command』 (*Command Reference*)

---

## 使用“DB2 安装”向导来安装 DB2 信息中心 (UNIX)

可使用三种方法访问 DB2 产品文档: 从 IBM Web 站点访问、从内部网服务器访问或从计算机上安装的版本访问。缺省情况下, DB2 产品从 IBM Web 站点访问 DB2 文档。如果想要从内部网服务器或您自己的计算机访问 DB2 文档, 必须从 *DB2 信息中心 CD 安装文档*。使用“DB2 安装”向导, 您可以定义安装首选项并在使用 UNIX 操作系统的计算机上安装 DB2 信息中心。

**先决条件:**

本节列示了在 UNIX 计算机上安装 DB2 信息中心的硬件、操作系统、软件以及通信需求。

- **硬件需求**

需要下列其中一种处理器:

- PowerPC (AIX)
- HP 9000 (HP-UX)
- Intel 32 位 (Linux)
- Solaris UltraSPARC 计算机 (Solaris Operating Environment)

- **操作系统需求**

需要下列其中一个操作系统:

- IBM AIX 5.1 (在 PowerPC 上)
- HP-UX 11i (在 HP 9000 上)
- Red Hat Linux 8.0 (在 Intel 32 位上)
- SuSE Linux 8.1 (在 Intel 32 位上)
- Sun Solaris V8 (在 Solaris Operating Environment UltraSPARC 计算机上)

**注:** DB2 信息中心在支持 DB2 客户机的一部分 UNIX 操作系统上运行。因此, 建议从 IBM Web 站点访问 DB2 信息中心, 或者在内部网服务器上安装并访问 DB2 信息中心。

- **软件需求**

- 支持下列浏览器:

- Mozilla V1.0 或更高版本

- “DB2 安装”向导是一个图形安装程序。必须实现能够呈现图形用户界面的 X Window System 软件才能使“DB2 安装”向导在计算机上运行。必须确保正确导出了显示内容, 才能运行“DB2 安装”向导。例如, 在命令提示符处输入以下命令:

```
export DISPLAY=9.26.163.144:0.
```

- 通信需求
  - TCP/IP

#### 过程:

要使用“DB2 安装”向导安装 DB2 信息中心:

1. 登录系统。
2. 在系统上放入并装上 DB2 信息中心产品 CD。
3. 通过输入以下命令切换到装上 CD 的目录:

```
cd /cd
```

其中 /cd 表示 CD 的安装点。

4. 输入 **.db2setup** 命令来启动“DB2 安装”向导。
5. “IBM DB2 安装启动板”打开。要直接进至 DB2 信息中心的安装, 单击**安装产品**。联机帮助可指导您完成其余步骤。要调用联机帮助, 单击**帮助**。可随时单击**取消**来结束安装。
6. 在**选择您想要安装的产品**页中, 单击**下一步**。
7. 在**欢迎使用“DB2 安装”向导**页中, 单击**下一步**。“DB2 安装”向导将指导您完成程序安装过程。
8. 要继续安装, 必须接受许可协议。在**许可协议**页中, 选择**我接受许可协议中的条款**, 然后单击**下一步**。
9. 在**选择安装操作**页中, 选择**在此计算机上安装 DB2 信息中心**。如果想要在稍后使用响应文件在此计算机或其它计算机上安装 DB2 信息中心, 则选择**将设置保存在响应文件中**。单击**下一步**。
10. 在**选择要安装的语言**页中, 选择将用来安装 DB2 信息中心的语言。单击**下一步**。
11. 在**指定 DB2 信息中心端口**页中, 配置 DB2 信息中心的人局通信。单击**下一步**继续安装。
12. 在**开始复制文件**页中复查您作出的安装选择。要更改任何设置, 单击**上一步**。单击**安装**以将 DB2 信息中心文件复制到计算机上。

还可以使用响应文件安装 DB2 信息中心。

缺省情况下, 安装日志 db2setup.his、db2setup.log 和 db2setup.err 位于 /tmp 目录中。

db2setup.log 文件会捕获所有 DB2 产品安装信息(包括错误)。db2setup.his 文件会记录计算机上的所有 DB2 产品安装。DB2 将 db2setup.log 文件追加至 db2setup.his 文件。db2setup.err 文件捕获 Java 返回的任何错误输出, 例如, 异常和陷阱信息。

当安装完成后, DB2 信息中心将安装在下列其中一个目录中, 这取决于您的 UNIX 操作系统:

- AIX: /usr/opt/db2\_08\_01
- HP-UX: /opt/IBM/db2/V8.1
- Linux: /opt/IBM/db2/V8.1
- Solaris Operating Environment: /opt/IBM/db2/V8.1

#### 相关概念:

- 第 282 页的『DB2 信息中心』

- 第 283 页的『DB2 信息中心安装方案』

#### 相关任务:

- 『使用响应文件安装 DB2 (UNIX)』(《安装与配置补充手册》)
- 第 290 页的『更新安装在计算机或内部网服务器上的 DB2 信息中心』
- 第 290 页的『以首选语言显示 DB2 信息中心中的主题』
- 第 289 页的『调用 DB2 信息中心』
- 第 287 页的『使用“DB2 安装”向导来安装 DB2 信息中心 (Windows)』

---

## 使用“DB2 安装”向导来安装 DB2 信息中心 (Windows)

可使用三种方法访问 DB2 产品文档: 从 IBM Web 站点访问、从内部网服务器访问或从计算机上安装的版本访问。缺省情况下, DB2 产品从 IBM Web 站点访问 DB2 文档。如果想要从内部网服务器或您自己的计算机访问 DB2 文档, 必须从 *DB2 信息中心 CD* 安装 DB2 文档。使用“DB2 安装”向导, 可以定义安装首选项并在使用 Windows 操作系统的计算机上安装 DB2 信息中心。

#### 先决条件:

本节列示了在 Windows 上安装 DB2 信息中心的硬件、操作系统、软件以及通信需求。

#### • 硬件需求

需要下列其中一种处理器:

- 32 位计算机: 奔腾或与奔腾兼容的 CPU

#### • 操作系统需求

需要下列其中一个操作系统:

- Windows 2000
- Windows XP

**注:** DB2 信息中心在支持 DB2 客户机的一部分 Windows 操作系统上运行。因此, 建议从 IBM Web 站点访问 DB2 信息中心, 或者在内部网服务器上安装并访问 DB2 信息中心。

#### • 软件需求

- 支持下列浏览器:

- Mozilla 1.0 或更高版本
- Internet Explorer V5.5 或 V6.0 (对于 Windows XP, 则为 Internet Explorer V6.0)

#### • 通信需求

- TCP/IP

#### 限制:

- 需要具有安装 DB2 信息中心的管理特权的帐户。

#### 过程:

要使用“DB2 安装”向导安装 DB2 信息中心:

1. 使用为 DB2 信息中心安装定义的帐户登录至系统。

2. 将 CD 插入到驱动器中。如果启用了自动运行功能，则它将启动“IBM DB2 安装启动板”。
3. “DB2 安装”向导会确定系统语言并启动该语言的安装程序。如果想要运行英语之外的语言的安装程序，或者安装程序无法自动启动，则可以手工启动“DB2 安装”向导。

要手工启动“DB2 安装”向导：

- a. 单击**开始**并选择**运行**。
- b. 在**打开**字段中，输入以下命令：

```
x:\setup.exe /i 2-letter language identifier
```

其中 *x*：表示 CD 驱动器，而 *2-letter language identifier* 表示将用来运行安装程序的语言。

- c. 单击**确定**。
4. “IBM DB2 安装启动板”打开。要直接进至 DB2 信息中心的安装，单击**安装产品**。联机帮助可指导您完成其余步骤。要调用联机帮助，单击**帮助**。可随时单击**取消**来结束安装。
5. 在**选择您想要安装的产品**页中，单击**下一步**。
6. 在**欢迎使用“DB2 安装”向导**页中，单击**下一步**。“DB2 安装”向导将指导您完成程序安装过程。
7. 要继续安装，必须接受许可协议。在**许可协议**页中，选择**我接受许可协议中的条款**，然后单击**下一步**。
8. 在**选择安装操作**页中，选择**在此计算机上安装 DB2 信息中心**。如果想要在稍后使用响应文件在此计算机或其它计算机上安装 DB2 信息中心，则选择**将设置保存在响应文件中**。单击**下一步**。
9. 在**选择要安装的语言**页中，选择将用来安装 DB2 信息中心的语言。单击**下一步**。
10. 在**指定 DB2 信息中心端口**页中，配置 DB2 信息中心的人局通信。单击**下一步**继续安装。
11. 在**开始复制文件**页中复查您作出的安装选择。要更改任何设置，单击**上一步**。单击**安装**以将 DB2 信息中心文件复制到计算机上。

可以使用响应文件来安装 DB2 信息中心。还可以使用 **db2rspgn** 命令来根据现有安装生成响应文件。

有关安装期间遇到的错误的信息，请参阅‘My Documents\DB2LOG\’目录中的 db2.log 和 db2wi.log 文件。‘My Documents’目录的位置将取决于计算机的设置。

db2wi.log 文件会捕获最新的 DB2 安装信息。db2.log 会捕获 DB2 产品安装的历史。

#### 相关概念：

- 第 282 页的『DB2 信息中心』
- 第 283 页的『DB2 信息中心安装方案』

#### 相关任务：

- 『使用响应文件安装 DB2 产品 (Windows)』(《安装与配置补充手册》)
- 第 290 页的『更新安装在计算机或内部网服务器上的 DB2 信息中心』
- 第 290 页的『以首选语言显示 DB2 信息中心中的主题』

- 第 289 页的『调用 DB2 信息中心』
- 第 285 页的『使用“DB2 安装”向导来安装 DB2 信息中心 (UNIX)』

#### 相关参考:

- 『db2rspgn - Response File Generator Command (Windows)』 (*Command Reference*)

---

## 调用 DB2 信息中心

DB2 信息中心允许您访问使用用于 Linux、UNIX 和 Windows 操作系统的 DB2 系列产品 (例如, DB2 通用数据库、DB2 Connect、DB2 Information Integrator 和 DB2 Query Patroller) 所需的所有信息。

可以从下列其中一个位置调用 DB2 信息中心;

- 安装了 DB2 UDB 客户机或服务器的计算机
- 安装了 DB2 信息中心的内部网服务器或本地计算机
- IBM Web 站点

#### 先决条件:

在调用 DB2 信息中心之前:

- 可选: 配置浏览器以使用首选语言来显示主题
- 可选: 配置 DB2 客户机以使用安装在计算机或内部网服务器上的 DB2 信息中心

#### 过程:

要调用安装了 DB2 UDB 客户机或服务器的计算机上的 DB2 信息中心:

- 从“开始菜单”(Windows 操作系统): 单击**开始** → **程序** → **IBM DB2** → **信息** → **信息中心**。
- 从命令行提示符:
  - 对于 Linux 和 UNIX 操作系统, 发出 **db2icdocs** 命令。
  - 对于 Windows 操作系统, 发出 **db2icdocs.exe** 命令。

要在 Web 浏览器中打开安装在内部网服务器或本地计算机上的 DB2 信息中心:

- 打开 Web 页面 (网址为: <http://<host-name>:<port-number>/>), 其中 <host-name> 表示主机名, 而 <port-number> 表示提供 DB2 信息中心的端口号。

要在 Web 浏览器中打开 IBM Web 站点上的 DB2 信息中心:

- 打开 Web 页面 (网址为: [publib.boulder.ibm.com/infocenter/db2help/](http://publib.boulder.ibm.com/infocenter/db2help/))。

#### 相关概念:

- 第 282 页的『DB2 信息中心』
- 第 283 页的『DB2 信息中心安装方案』

#### 相关任务:

- 第 290 页的『以首选语言显示 DB2 信息中心中的主题』
- 第 297 页的『从 DB2 工具调用上下文帮助』
- 第 290 页的『更新安装在计算机或内部网服务器上的 DB2 信息中心』
- 第 298 页的『从命令行处理器调用命令帮助』
- 『设置访问 DB2 信息中心的位置: 公共 GUI 帮助』

#### 相关参考:

- 『HELP Command』（*Command Reference*）

---

## 更新安装在计算机或内部网服务器上的 DB2 信息中心

<http://publib.boulder.ibm.com/infocenter/db2help/> 提供的 DB2 信息中心将会用新的或更改过的文档定期更新。IBM 还可能提供 DB2 信息中心更新，可以下载这些更新并将它们安装在计算机或内部网服务器上。更新 DB2 信息中心不会更新 DB2 客户机或服务器产品。

#### 先决条件:

必须能够访问连接至因特网的计算机。

#### 过程:

要更新安装在计算机或内部网服务器上的 DB2 信息中心:

1. 打开位于 IBM Web 站点上的 DB2 信息中心：  
<http://publib.boulder.ibm.com/infocenter/db2help/>
2. 在欢迎页面的“服务与支持”标题下面的“下载”部分，单击 **DB2 通用数据库文档** 链接。
3. 通过将最新刷新的文档映像级别与已安装的文档级别进行比较来确定 DB2 信息中心的版本是否已过时。已安装的文档级别列示在 DB2 信息中心欢迎页面上。
4. 如果有较新版本的 DB2 信息中心可用，则下载适用于您的操作系统的最新刷新的 *DB2 信息中心映像*。
5. 要安装刷新过的 *DB2 信息中心映像*，遵循 Web 页面上提供的指示信息。

#### 相关概念:

- 第 283 页的『DB2 信息中心安装方案』

#### 相关任务:

- 第 289 页的『调用 DB2 信息中心』
- 第 285 页的『使用“DB2 安装”向导来安装 DB2 信息中心（UNIX）』
- 第 287 页的『使用“DB2 安装”向导来安装 DB2 信息中心（Windows）』

---

## 以首选语言显示 DB2 信息中心中的主题

DB2 信息中心尝试以您在浏览器首选项中指定的语言显示主题。如果主题未翻译为首选语言，则 DB2 信息中心将显示该主题的英文版。

#### 过程:

要在 Internet Explorer 浏览器中以您的首选语言显示主题:

1. 在 Internet Explorer 中，单击 **工具** → **Internet 选项** → **语言...** 按钮。“语言首选项”窗口打开。
2. 确保您的首选语言被指定为语言列表中的第一个条目。
  - 要将新语言添加至列表，单击 **添加...** 按钮。

注：添加语言并不能保证计算机具有以首选语言显示主题所需的字体。

- 要将语言移至列表顶部，选择该语言并单击**上移**按钮直到该语言成为语言列表中的第一项。

3. 刷新该页面以便以首选语言显示 DB2 信息中心。

要在 Mozilla 浏览器中以首选语言显示主题：

1. 在 Mozilla 中，选择**编辑** → **首选项** → **语言**按钮。“语言”面板将显示在“首选项”窗口中。

2. 确保您的首选语言被指定为语言列表中的第一个条目。

- 要将新语言添加至列表，单击**添加...**按钮以从“添加语言”窗口中选择一种语言。
- 要将语言移至列表顶部，选择该语言并单击**上移**按钮直到该语言成为语言列表中的第一项。

3. 刷新该页面以便以首选语言显示 DB2 信息中心。

相关概念：

- 第 282 页的『DB2 信息中心』

---

## DB2 PDF 和印刷文档

下列各表提供正式书名、书号和 PDF 文件名。要订购硬拷贝书籍，必须知道正式书名。要打印 PDF 文件，必须知道 PDF 文件名。

DB2 文档按下列标题分类：

- 核心 DB2 信息
- 管理信息
- 应用程序开发信息
- 商业智能信息
- DB2 Connect 信息
- 入门信息
- 教程信息
- 可选组件信息
- 发行说明

对于 DB2 资料库中的每本书籍，下表描述了订购该书籍的硬拷贝、打印或查看该书籍的 PDF 所需的信息。DB2 资料库中的每本书籍的完整描述可从 IBM 出版物中心（IBM Publications Center）获取，网址为 [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order)。

### 核心 DB2 信息

这些书籍中的信息对所有 DB2 用户来说都是基础知识，不管您是程序员、数据库管理员或是使用 DB2 Connect、DB2 仓库管理器或其它 DB2 产品的人员，都将会发现此信息很有用。

表 42. 核心 DB2 信息

书名	书号	PDF 文件名
<i>IBM DB2 Universal Database Command Reference</i>	SC09-4828	db2n0x81

表 42. 核心 DB2 信息 (续)

书名	书号	PDF 文件名
《IBM DB2 通用数据库词汇表》	无书号	db2t0c81
《IBM DB2 通用数据库消息参考第 1 卷》	G152-0177, 未提供硬拷贝	db2m1c81
《IBM DB2 通用数据库消息参考第 2 卷》	G152-0178, 未提供硬拷贝	db2m2c81
《IBM DB2 通用数据库新增内容》	S152-0176	db2q0c81

## 管理信息

这些书籍中的信息包括有效地设计、实现和维护 DB2 数据库、数据仓库和联合系统所需的那些主题。

表 43. 管理信息

书名	书号	PDF 文件名
《IBM DB2 通用数据库管理指南: 计划》	S152-0167	db2d1c81
《IBM DB2 通用数据库管理指南: 实现》	S152-0165	db2d2c81
《IBM DB2 通用数据库管理指南: 性能》	S152-0166	db2d3c81
<i>IBM DB2 Universal Database Administrative API Reference</i>	SC09-4824	db2b0x81
<i>IBM DB2 Universal Database Data Movement Utilities Guide and Reference</i>	SC09-4830	db2dmx81
《IBM DB2 通用数据库数据恢复和高可用性指南与参考》	S152-0181	db2hac81
《IBM DB2 通用数据库数据仓库中心管理指南》	S152-0188	db2ddc81
<i>IBM DB2 Universal Database SQL Reference, Volume 1</i>	SC09-4844	db2s1x81
<i>IBM DB2 Universal Database SQL Reference, Volume 2</i>	SC09-4845	db2s2x81
<i>IBM DB2 Universal Database System Monitor Guide and Reference</i>	SC09-4847	db2f0x81

## 应用程序开发信息

这些书籍中的信息对于应用程序开发者或使用 DB2 通用数据库 (DB2 UDB) 的程序员特别有用。您将找到有关受支持的语言和编译器的信息, 以及使用各种受支持的编程接口 (例如, 嵌入式 SQL、ODBC、JDBC、SQLJ 和 CLI) 访问 DB2 UDB 所需的文档。如果正在使用 DB2 信息中心, 还可访问 HTML 版本的源代码以获取样本程序。



表 44. 应用程序开发信息

书名	书号	PDF 文件名
《IBM DB2 通用数据库应用程序开发指南: 构建和运行应用程序》	S152-0168	db2axc81
<i>IBM DB2 Universal Database Application Development Guide: Programming Client Applications</i>	SC09-4826	db2a1x81
<i>IBM DB2 Universal Database Application Development Guide: Programming Server Applications</i>	SC09-4827	db2a2x81
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1</i>	SC09-4849	db2l1x81
<i>IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2</i>	SC09-4850	db2l2x81
<i>IBM DB2 Universal Database Data Warehouse Center Application Integration Guide</i>	SC27-1124	db2adx81
<i>IBM DB2 XML Extender Administration and Programming</i>	SC27-1234	db2sxx81

## 商业智能信息

这些书籍中的信息描述如何使用将增强 DB2 通用数据库的数据仓储功能和分析功能的组件。

表 45. 商业智能信息

书名	书号	PDF 文件名
<i>IBM DB2 Warehouse Manager Standard Edition Information Catalog Center Administration Guide</i>	SC27-1125	db2dix81
<i>IBM DB2 Warehouse Manager Standard Edition Installation Guide</i>	G152-0187	db2idc81
<i>IBM DB2 Warehouse Manager Standard Edition Managing ETI Solution Conversion Programs with DB2 Warehouse Manager</i>	SC18-7727	iwhe1mstx80

## DB2 Connect 信息

此类别中的信息描述如何使用 DB2 Connect 企业版或 DB2 Connect 个人版来存取大型机和中型机服务器上的数据。

表 46. DB2 Connect 信息

书名	书号	PDF 文件名
<i>IBM Connectivity Supplement</i>	无书号	db2h1x81
《IBM DB2 Connect 快速入门 DB2 Connect 企业版》	G152-0271	db2c6c81
《IBM DB2 Connect 快速入门 DB2 Connect 个人版》	G152-0171	db2c1c81
《IBM DB2 Connect 用户指南》	S152-0172	db2c0c81

## 入门信息

安装和配置服务器、客户机以及其它 DB2 产品时，此类别中的信息非常有用。

表 47. 入门信息

书名	书号	PDF 文件名
《IBM DB2 通用数据库快速入门 DB2 客户机版》	G152-0170, 未提供硬拷贝	db2itc81
《IBM DB2 通用数据库快速入门 DB2 服务器版》	G152-0173	db2isc81
《IBM DB2 通用数据库快速入门 DB2 个人版》	G152-0175	db2i1c81
《IBM DB2 通用数据库安装与配 置补充手册》	G152-0174, 未提供硬拷贝	db2iyc81
《IBM DB2 通用数据库快速入门 DB2 Data Links Manager 版》	G152-0169	db2z6c81

## 教程信息

教程信息介绍 DB2 功能部件并指导如何执行各种任务。

表 48. 教程信息

书名	书号	PDF 文件名
《商业智能教程: 数据仓库简 介》	无书号	db2tuc81
《商业智能教程: 数据仓储扩 展课程》	无书号	db2tac81
<i>Information Catalog Center Tutorial</i>	无书号	db2aix81
<i>Video Central for e-business Tutorial</i>	无书号	db2twx81
《Visual Explain 教程》	无书号	db2tvc81

## 可选组件信息

此类别中的信息描述如何使用可选 DB2 组件。

表 49. 可选组件信息

书名	书号	PDF 文件名
《IBM DB2 Cube Views 指南与参考》	S152-0532	db2aac81
IBM DB2 Query Patroller Guide: Installation, Administration and Usage Guide	GC09-7658	db2dwx81
IBM DB2 Spatial Extender and Geodetic Extender User's Guide and Reference	SC27-1226	db2sbn81
IBM DB2 Database Data Links Manager Administration Guide and Reference	SC27-1221	db2z0x82
《DB2 Net Search Extender 管理和用户指南》	S152-0596	不适用

注: 此文档的 HTML 不是从“HTML 文档” CD 安装的。

## 发行说明

发行说明提供了特定于产品发行版和修订包级别的附加信息。发行说明还提供了并入到每个发行版、更新和修订包中的文档更新的总结。

表 50. 发行说明

书名	书号	PDF 文件名
《DB2 发行说明》	请参阅“注”。	请参阅“注”。
《DB2 安装说明》	仅在产品 CD-ROM 上提供。	未提供。

注: 提供有下列格式的发行说明:

- XHTML 和文本格式 (在产品 CD 上)
- PDF 格式 (在 PDF 文档 CD 上)

此外,《发行说明》中讨论已知问题和变通方法和发行版之间的不兼容性的部分还会出现在 DB2 信息中心中。

要在基于 UNIX 的平台上查看文本格式的发行说明,请参阅 Release.Notes 文件。此文件位于 DB2DIR/Readme/%L 目录中,其中 %L 表示语言环境名称,DB2DIR 表示:

- 对于 AIX 操作系统: /usr/opt/db2\_08\_01
- 对于所有其它基于 UNIX 的操作系统: /opt/IBM/db2/V8.1

相关概念:

- 第 281 页的『DB2 文档和帮助』

相关任务:

- 第 296 页的『从 PDF 文件打印 DB2 书籍』
- 第 296 页的『订购印刷的 DB2 书籍』
- 第 297 页的『从 DB2 工具调用上下文帮助』

---

## 从 PDF 文件打印 DB2 书籍

可从 *DB2 PDF* 文档 CD 上的 PDF 文件打印 DB2 书籍。通过使用 Adobe Acrobat Reader, 可打印整本书或特定范围的那些页。

### 先决条件:

确保安装了 Adobe Acrobat Reader。如果需要安装 Adobe Acrobat Reader, 则可从 Adobe Web 站点获得它, 网址为 [www.adobe.com](http://www.adobe.com)。

### 过程:

要从 PDF 文件打印 DB2 书籍:

1. 插入 *DB2 PDF* 文档 CD。在 UNIX 操作系统上, 安装“DB2 PDF 文档” CD。有关如何在 UNIX 操作系统上安装 CD 的详细信息, 参阅《快速入门》一书。
2. 打开 `index.htm`。文件将在浏览器窗口中打开。
3. 单击想要查看的 PDF 的标题。该 PDF 将在 Acrobat Reader 中打开。
4. 选择文件 → 打印以打印想要的书籍的任何部分。

### 相关概念:

- 第 282 页的『DB2 信息中心』

### 相关任务:

- 『装上 CD-ROM (AIX)』 (《DB2 服务器快速入门》)
- 『装上 CD-ROM (HP-UX)』 (《DB2 服务器快速入门》)
- 『装上 CD-ROM (Linux)』 (《DB2 服务器快速入门》)
- 第 296 页的『订购印刷的 DB2 书籍』
- 『安装 CD-ROM (Solaris Operating Environment)』 (《DB2 服务器快速入门》)

### 相关参考:

- 第 291 页的『DB2 PDF 和印刷文档』

---

## 订购印刷的 DB2 书籍

如果喜欢使用硬拷贝书籍, 可以用以下三种方式中的一种订购它们。

### 过程:

可在某些国家或地区订购印刷版书籍。访问您所在国家或地区的 IBM 出版物 Web 站点, 以了解您所在国家或地区是否提供此项服务。如果可以订购这些出版物, 则可以:

- 与 IBM 授权经销商或市场营销代表联系。要查找您当地的 IBM 代表, 查看 IBM 全球联系人目录 (IBM Worldwide Directory of Contacts), 网址为 [www.ibm.com/planetwide](http://www.ibm.com/planetwide)。

- 访问 IBM 出版物中心 (IBM Publications Center)，网址为 <http://www.ibm.com/shop/publications/order>。可能未在所有国家或地区提供从 IBM 出版物中心订购书籍这项功能。

DB2 产品可用时，印刷书籍与 *DB2 PDF 文档 CD* 上以 PDF 格式提供的那些书籍是相同的。印刷书籍中的内容出现在 *DB2 信息中心 CD* 中时也是相同的。但是，DB2 信息中心 CD 中有一些附加内容未出现在 PDF 书籍中的任何位置（例如，SQL 管理例程和 HTML 样本）。并非 DB2 PDF 文档 CD 上提供的所有书籍都可以订购硬拷贝。

**注：**DB2 信息中心的更新比 PDF 或硬拷贝书籍的更新要频繁得多；一提供文档更新就安装它们，或者参阅网址如下的 DB2 信息中心以获取最新信息：  
<http://publib.boulder.ibm.com/infocenter/db2help/>。

#### 相关任务:

- 第 296 页的『从 PDF 文件打印 DB2 书籍』

#### 相关参考:

- 第 291 页的『DB2 PDF 和印刷文档』

---

## 从 DB2 工具调用上下文帮助

上下文帮助提供有关与特定窗口、笔记本、向导或顾问程序相关联的任务或控件的信息。上下文帮助可从具有图形用户界面的 DB2 管理和开发工具获得。有两种类型的上下文帮助:

- 通过位于每个窗口或笔记本上的**帮助按钮**访问的帮助
- 弹出信息，即将鼠标光标放到字段或控件上或在窗口、笔记本、向导或顾问程序中选择了字段或控件并按 F1 键时显示的弹出信息窗口。

**帮助按钮**允许您访问概述、先决条件和任务信息。弹出信息描述各个字段和控件。

#### 过程:

要调用上下文帮助:

- 要获取窗口和笔记本帮助，启动其中一个 DB2 工具，然后打开任意窗口或笔记本。单击窗口或笔记本右下角的**帮助按钮**以调用上下文帮助。

还可从位于每个 DB2 工具中心上方的**帮助**菜单项访问上下文帮助。

在向导和顾问程序中，单击第一页上的“任务概述”链接以查看上下文帮助。

- 要获取有关窗口或笔记本的各个控件的弹出信息帮助，单击该控件，然后按 **F1**。包含有关控件的详细信息的弹出信息将显示在黄色窗口中。

**注：**如果希望只要将鼠标光标放在字段或控件上就显示弹出信息，在“工具设置”笔记本的**文档**页上选择**自动显示弹出信息**复选框。

与弹出信息类似，诊断弹出信息是另一种形式的上下文相关帮助；它们包含数据输入规则。诊断弹出信息显示在输入的数据无效或不充分时出现的紫色窗口中。会对以下各项显示诊断弹出信息:

- 必填字段。
- 其数据遵照精确格式的字段，例如，日期字段。

**相关任务:**

- 第 289 页的『调用 DB2 信息中心』
- 第 298 页的『从命令行处理器调用消息帮助』
- 第 298 页的『从命令行处理器调用命令帮助』
- 第 299 页的『从命令行处理器调用 SQL 状态帮助』
- 『访问 DB2 信息中心: 概念帮助』
- 『如何使用 DB2 UDB 帮助: 公共 GUI 帮助』
- 『设置访问 DB2 信息中心的位置: 公共 GUI 帮助』
- 『设置对 DB2 上下文帮助和文档的访问权: 公共 GUI 帮助』

---

## 从命令行处理器调用消息帮助

消息帮助描述产生消息的原因并描述为解决错误而应采取的任何操作。

**过程:**

要调用消息帮助, 打开命令行处理器并输入:

```
? XXXnnnnn
```

其中 *XXXnnnnn* 表示有效的消息标识。

例如, ? SQL30081 会显示有关 SQL30081 消息的帮助。

**相关概念:**

- 『消息介绍』(《消息参考》第 1 卷)

**相关参考:**

- 『db2 - Command Line Processor Invocation Command』( *Command Reference* )

---

## 从命令行处理器调用命令帮助

命令帮助说明命令行处理器中命令的语法。

**过程:**

要调用命令帮助, 打开命令行处理器并输入:

```
? command
```

其中 *command* 表示一个关键字或整条命令。

例如, ? catalog 显示所有 CATALOG 命令的帮助, 而 ? catalog database 只显示 CATALOG DATABASE 命令的帮助。

**相关任务:**

- 第 297 页的『从 DB2 工具调用上下文帮助』
- 第 289 页的『调用 DB2 信息中心』
- 第 298 页的『从命令行处理器调用消息帮助』

- 第 299 页的『从命令行处理器调用 SQL 状态帮助』

相关参考:

- 『db2 - Command Line Processor Invocation Command』 ( *Command Reference* )

---

## 从命令行处理器调用 SQL 状态帮助

DB2 通用数据库返回可作为 SQL 语句结果的条件的 SQLSTATE 值。SQLSTATE 帮助说明 SQL 状态和 SQL 状态类代码的含义。

过程:

要调用 SQL 状态帮助, 打开命令行处理器并输入:

```
? sqlstate 或 ? class code
```

其中, *sqlstate* 表示有效的 5 位 SQL 状态, *class code* 表示该 SQL 状态的前 2 位。

例如, ? 08003 显示 08003 SQL 状态的帮助, 而 ? 08 显示 08 类代码的帮助。

相关任务:

- 第 289 页的『调用 DB2 信息中心』
- 第 298 页的『从命令行处理器调用消息帮助』
- 第 298 页的『从命令行处理器调用命令帮助』

---

## DB2 教程

DB2® 教程帮助您了解 DB2 通用数据库的各个方面。教程提供了开发应用程序、调整 SQL 查询性能、使用数据仓库、管理元数据和使用 DB2 开发 Web 服务等方面的课程, 这些课程中还提供了逐步指示信息。

开始之前:

可从“信息中心”查看 [XHTML](http://publib.boulder.ibm.com/infocenter/db2help/) 版本的教程, 网址如下:  
<http://publib.boulder.ibm.com/infocenter/db2help/>。

某些教程课程使用了样本数据或代码。有关其特定任务的任何先决条件的描述, 请参阅每个教程。

**DB2 通用数据库教程:**

单击以下列表中的教程标题以查看该教程。

《商业智能教程: 数据仓库中心介绍》

使用“数据仓库中心”来执行介绍性的数据仓储任务。

《商业智能教程: 数据仓储扩展课程》

使用“数据仓库中心”来执行高级数据仓储任务。

《信息目录中心教程》

使用“信息目录中心”来创建和管理信息目录以查找并使用元数据。

---

## DB2 故障诊断信息

提供有大量故障诊断和问题确定信息，可帮助您使用 DB2® 产品。

### DB2 文档

DB2 信息中心以及构成 DB2 资料库的 PDF 书籍中处处可找到故障诊断信息。可参阅 DB2 信息中心导航树（在浏览器窗口的左窗格中）的“支持和故障诊断”分支以查看 DB2 故障诊断文档的完整列表。

### DB2 技术支持 Web 站点

如果您遇到了问题并且想要获取查找可能的原因和解决方案的帮助，请参阅 DB2 技术支持（DB2 Technical Support）Web 站点。该“技术支持”站点具有指向最新 DB2 出版物、技术说明、授权程序分析报告（APAR）、修订包的链接、内部 DB2 错误代码的最新列表以及其它资源。可搜索此知识库并查找问题的可能解决方案。

访问位于以下网址的 DB2 技术支持 Web 站点：  
<http://www.ibm.com/software/data/db2/udb/win02unix/support>

### DB2 问题确定教程系列

要查找有关如何快速标识和解决在使用 DB2 产品时可能遇到的问题的信息，参阅 DB2 问题确定教程系列 Web 站点。有一个教程介绍可用的 DB2 问题确定设施和工具并帮助您决定何时使用它们。其它教程处理相关主题，例如“数据库引擎问题确定”、“性能问题确定”和“应用程序问题确定”。

查看 DB2 技术支持站点上的 DB2 问题确定教程的完整集合，网址如下：  
<http://www.ibm.com/software/data/support/pdm/db2tutorials.html>

### 相关概念:

- 第 282 页的『DB2 信息中心』
- 『Introduction to problem determination - DB2 Technical Support tutorial』  
(*Troubleshooting Guide*)

---

## 辅助功能

辅助功能部件可帮助那些身体有某些缺陷（如活动不方便或视力不太好）的用户成功地使用软件产品。以下列表指定 DB2® V8 产品中的主要辅助功能部件:

- 所有 DB2 功能可使用键盘（而不是鼠标）导航来实现。有关更多信息，请参阅第 301 页的『键盘输入和导航』。
- 可定制 DB2 用户界面上的字体大小和颜色。有关更多信息，请参阅第 301 页的『界面显示的辅助功能』。
- DB2 产品支持使用 Java™ Accessibility API 的辅助功能应用程序。有关更多信息，请参阅第 301 页的『与辅助技术的兼容性』。
- DB2 文档是以易使用格式提供的。有关更多信息，请参阅第 301 页的『文档的辅助功能』。



## 键盘输入和导航

### 键盘输入

只使用键盘就可以操作 DB2 工具。使用键或键组合就可以执行使用鼠标所能完成的操作。标准操作系统击键用于标准操作系统操作。

有关使用键或键组合执行操作的更多信息，请参阅 [键盘快捷方式和加速键：公共 GUI 帮助](#)。

### 键盘导航

可使用键或键组合来导航 DB2 工具用户界面。

有关使用键或键组合来导航 DB2 工具的更多信息，请参阅 [键盘快捷方式和加速键：公共 GUI 帮助](#)。

### 键盘焦点

在 UNIX<sup>®</sup> 操作系统中，击键操作起作用的活动窗口的区域将突出显示。

## 界面显示的辅助功能

DB2 工具所具有的功能部件使视力不太好的用户更易使用。这些辅助功能方面的增强包括了对可定制字体属性的支持。

### 字体设置

可使用“工具设置”笔记本来选择菜单和对话框窗口中文本的颜色、大小和字体。

有关指定字体设置的更多信息，请参阅 [更改菜单和文本的字体：公共 GUI 帮助](#)。

### 不依赖于颜色

不需要分辨颜色就可以使用此产品中的任何功能。

## 与辅助技术的兼容性

DB2 工具界面支持 Java Accessibility API，它使您能够将屏幕阅读器和其它辅助技术与 DB2 产品配合使用。

## 文档的辅助功能

DB2 的相关文档是以 XHTML 1.0 格式提供的，它在大部分 Web 浏览器中是可查看的。XHTML 允许您根据浏览器中设置的显示首选项来查看文档。还允许您使用屏幕阅读器和其它辅助技术。

语法图是以点分十进制格式提供的。仅当使用屏幕阅读器访问联机文档时，此格式才可用。

### 相关概念：

- [第 302 页的『点分十进制语法图』](#)

### 相关任务：

- [『键盘快捷方式和加速键：公共 GUI 帮助』](#)
- [『更改菜单和文本的字体：公共 GUI 帮助』](#)

## 点分十进制语法图

语法图是以点分十进制的格式为使用屏幕阅读器访问信息中心的用户提供的。

在点分十进制格式中，每个语法元素写在单独的一行上。如果两个或多个语法元素总是一起出现（或总是一起不出现），它们可显示在同一行上，这是因为可将它们视作单个复合语法元素。

每一行以点分十进制编号开始；例如，3、3.1 或 3.1.1。要正确地听到这些数字，确保屏幕阅读器设置为读出标点。具有相同点分十进制编号的所有语法元素（例如，具有编号 3.1 的所有语法元素）是互斥的替代项。如果听到行 3.1 USERID 和 3.1 SYSTEMID，就知道语法可能包括 USERID 或 SYSTEMID，但不会同时包括这两者。

点分十进制编号级别表示嵌套级别。例如，如果具有点分十进制编号 3 的语法元素后跟点分十进制编号为 3.1 的一系列语法元素，则编号为 3.1 的所有语法元素是编号为 3 的语法元素的下级。

某些单词和符号用在点分十进制编号的旁边以添加有关这些语法元素的信息。这些单词和符号有时可能会出现在元素本身的开头。为易于识别，如果该单词或符号是语法元素的一部分，它的前面会加上反斜杠 (\) 字符。\* 符号可用在点分十进制编号的旁边以指示该语法元素重复。例如，点分十进制编号为 3 的语法元素 \*FILE 的格式为 3 \\* FILE。3\* FILE 这一格式指示语法元素 FILE 重复。格式 3\* \\* FILE 指示语法元素 \* FILE 重复。

用来分隔一串语法元素的字符（例如，逗号）在语法中刚好显示在它们要分隔的项之前。这些字符可与每一项显示在同一行上，或显示在单独一行上并带有与相关项相同的点分十进制编号。该行还可显示另一个符号，该符号给出有关语法元素的信息。例如，行 5.1\*、5.1 LASTRUN 和 5.1 DELETE 意味着如果使用多个 LASTRUN 和 DELETE 语法元素，必须用逗号分隔这些元素。如果未指定分隔符，则假定使用空格来分隔每个语法元素。

如果语法元素前面有 % 符号，这表示在别处定义的引用。% 符号之后的字符串是语法段的名称，而非文字。例如，行 2.1 %OPI 意味着您应引用单独的语法分段 OPI。

下列单词和符号用在点分十进制编号的旁边：

- ? 表示可选语法元素。后跟 ? 符号的点分十进制编号指示具有相应点分十进制编号的所有语法元素及任何下级语法元素都是可选的。如果只有一个带有点分十进制编号的语法元素，则 ? 符号与该语法元素显示在同一行上（例如，5? NOTIFY）。如果有多个带有点分十进制编号的语法元素，则 ? 符号单独显示在一行上，后跟可选语法元素。例如，如果您听到行 5 ?、5 NOTIFY 和 UPDATE，就知道语法元素 NOTIFY 和 UPDATE 是可选的；即，您可选择其中一项或全部都不选。? 符号相当于路线图中的支路。
- ! 表示缺省语法元素。后跟 ! 符号的点分十进制编号和语法元素指示该语法元素是共享同一点分十进制编号的所有语法元素的缺省选项。只有共享同一点分十进制编号的语法元素的其中一个可指定 ! 符号。例如，如果听到行 2? FILE、2.1! (KEEP) 和 2.1 (DELETE)，就知道 (KEEP) 是 FILE 关键字的缺省选项。在此示例中，如果包括 FILE 关键字但未指定选项，将应用缺省选项 KEEP。缺省选项还会应用于下一个较高的点分十进制编号。在此示例中，如果省略了 FILE 关键字，将使用缺省值 FILE(KEEP)。但是，如果听到行 2? FILE、2.1、2.1.1! (KEEP) 和 2.1.1 (DELETE)，

则缺省选项 KEEP 仅应用于下一个较高的点分十进制编号 2.1（它没有相关联的关键字），而不会应用于 2? FILE。如果省略了关键字 FILE，则不会使用任何值。

- \* 表示可重复零次或多次的语法元素。后跟 \* 符号的点分十进制编号指示此语法元素可使用零次或多次；即，它是可选的而且可以重复。例如，如果听到行 5.1\* data area，就知道可以包括一个数据区、多个数据区或者不包括数据区。如果听到行 3\*、3 HOST 和 3 STATE，就知道可包括 HOST 和 / 或 STATE 或者不包括任何内容。

注:

1. 如果点分十进制编号的旁边有星号（\*）且只有一项带有该点分十进制编号，可重复同一项多次。
  2. 如果点分十进制编号的旁边有星号且有若干项带有该点分十进制编号，可使用列表中的多项，但每项只能使用一次。在先前示例中，可以写为 HOST STATE，但不能写为 HOST HOST。
  3. \* 符号相当于路线语法图中的回路。
- + 表示必须被包括一次或多次的语法元素。后跟 + 符号的点分十进制编号指示此语法元素必须被包括一次或多次；即，它必须至少被包括一次，而且可以重复。例如，如果听到行 6.1+ data area，就知道必须至少包括一个数据区。如果听到行 2+、2 HOST 和 2 STATE，就知道必须包括 HOST 和 / 或 STATE。与 \* 符号类似，如果 + 符号是带有该点分十进制编号的唯一项，则它只能重复特定项。与 \* 符号一样，+ 符号相当于路线语法图中的回路。

相关概念:

- 第 300 页的『辅助功能』

相关任务:

- 『键盘快捷方式和加速键: 公共 GUI 帮助』

相关参考:

- 『How to read the syntax diagrams』 (*SQL Reference, Volume 2*)

---

## DB2 通用数据库产品的 Common Criteria 认证

DB2 通用数据库正在进行 Common Criteria 评估保证级别 4 (EAL4) 的评估认证。有关 Common Criteria 的更多信息，请参阅 Common Criteria Web 站点：<http://niap.nist.gov/cc-scheme/>



---

## 附录 B. 声明

IBM 可能在所有国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：** International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Canada Limited  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario

L6G 1C7  
CANADA

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息可能包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，与实际商业企业所用的名称和地址的任何雷同纯属巧合。

版权许可：

本信息可能包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明：

©（贵公司的名称）（年）。此部分代码是根据 IBM 公司的样本程序衍生出来的。

© Copyright IBM Corp.（输入年份）。All rights reserved.

---

## 商标

下列各项是国际商业机器公司在美国和 / 或其他国家或地区的商标, 且已在 DB2 UDB 文档库中的至少一份文档中使用。

ACF/VTAM	iSeries
AISPO	LAN Distance
AIX	MVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	NetView
BookManager	OS/390
C Set++	OS/400
C/370	PowerPC
CICS	pSeries
Database 2	QBIC
DataHub	QMF
DataJoiner	RACF
DataPropagator	RISC System/6000
DataRefresher	RS/6000
DB2	S/370
DB2 Connect	SP
DB2 Extender	SQL/400
DB2 OLAP Server	SQL/DS
DB2 Information Integrator	System/370
DB2 Query Patroller	System/390
DB2 Universal Database	SystemView
Distributed Relational	Tivoli
Database Architecture	VisualAge
DRDA	VM/ESA
eServer	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WebSphere
IBM	WIN-OS/2
IMS	z/OS
IMS/ESA	zSeries

下列各项是其他公司的商标或注册商标, 且已在 DB2 UDB 文档库中的至少一份文档中使用:

Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

Intel 和 Pentium 是 Intel Corporation 在美国和 / 或其他国家或地区的商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。



# 索引

## [ A ]

安全性  
    样本 87  
安装  
    信息中心 283, 285, 287

## [ B ]

帮助  
    显示 289, 290  
    用于命令  
        调用 298  
    用于消息  
        调用 298  
    用于 SQL 语句  
        调用 299  
绑定  
    SQL 过程 129  
绑定实用程序  
    样本数据库 46  
备份  
    SQL 过程 130  
编目  
    样本数据库 46  
编译器  
    构建文件 93  
    使用 AIX IBM COBOL 174  
    使用 AIX Micro Focus COBOL 180  
    使用 HP-UX Micro Focus  
        COBOL 198  
    使用 Solaris Micro Focus COBOL 227  
    使用 Windows IBM COBOL 264  
    使用 Windows Micro Focus  
        COBOL 271  
    受支持 AIX 版本 8  
    受支持 HP-UX 版本 10  
    受支持 Linux 版本 12  
    受支持 Solaris 版本 17  
    受支持 Windows 版本 19  
    makefile 95  
表函数  
    对象链接与嵌入  
        样本 83  
    OLE DB 232  
不可用 300

## [ C ]

操作系统  
    受支持 AIX 版本 8

操作系统 (续)  
    受支持 HP-UX 版本 10  
    受支持 Linux 版本 12  
    受支持 Solaris 版本 17  
    受支持 Windows 版本 19  
    受 DB2 支持 7  
    DB2 安装路径 48  
插件  
    安全性样本 87  
    Java 样本 80  
程序  
    样本 57  
重新绑定  
    SQL 过程 131  
存储过程  
    使用 Visual Basic 进行 OLE 自动化  
        240  
    使用 Visual C++ 进行 OLE 自动化  
        253  
    AIX C++ 配置文件 172  
    CALL 语句 124  
错误检查  
    实用程序文件 98

## [ D ]

打印  
    PDF 文件 296  
点分十进制语法图 302  
调用  
    命令帮助 298  
    消息帮助 298  
    SQL 语句帮助 299  
调用级接口 (CLI)  
    DB2 AD 客户机支持 3  
调用 SQL 过程  
    客户机应用程序 128  
订购 DB2 书籍 296  
动态配置  
    样本 75  
对象链接与嵌入  
    数据库表函数  
        描述 232  
        样本文件 83  
    样本 82  
自动化  
    使用 Visual Basic 240  
    使用 Visual C++ 253  
    DB2 AD 客户机支持 3  
多连接应用程序  
    构建文件 93

多连接应用程序 (续)  
    构建 UNIX C 142  
    构建 UNIX C++ 149  
    构建 Windows C/C++ 262  
多线程应用程序  
    构建文件 93  
    使用 AIX C 构建 165  
    使用 AIX C++ 构建 169  
    使用 HP-UX C 构建 191  
    使用 HP-UX C++ 构建 197  
    使用 Linux C 构建 207  
    使用 Linux C++ 构建 211  
    使用 Solaris C 构建 221  
    使用 Solaris C++ 构建 226  
    使用 Windows C/C++ 构建 254

## [ F ]

服务器  
    受 DB2 支持 7  
辅助功能  
    点分十进制语法图 302  
    功能部件 300  
复原  
    SQL 过程 130

## [ G ]

更新  
    HTML 文档 290  
共享库  
    重新构建例程 24  
    替换 AIX 161  
构建文件 93  
故障诊断  
    教程 300  
    联机信息 300  
管理规范, Windows 233  
    样本 92

## [ H ]

环境  
    应用程序开发  
        设置 23  
环境变量  
    UNIX 31

## [ J ]

键盘快捷方式

支持 300

教程 299

故障诊断和问题确定 300

脚本

运行命令处理器 (CLP) 123

## [ K ]

开发中心

DB2 AD 客户机支持 3

可移植性

迁移应用程序 52

库, 共享的

重新构建例程 24

替换 AIX 161

## [ L ]

例程

重新构建共享库 24

构建文件 93

样本程序文件

SQL 过程 87

在 AIX 上装入 COBOL 共享库 161

AIX 入口点 159

AIX 上的 CREATE 语句 160

例程的入口点, AIX 159

联机

帮助, 存取 297

## [ M ]

命令帮助

调用 298

命令行处理器 (CLP)

样本文件 70

运行脚本 123

DB2 AD 客户机支持 3

## [ P ]

配置文件

对于 AIX 上的 VisualAge C++ 170

批处理文件 93

## [ Q ]

迁移应用程序

应用程序 48

在 DB2 的两个版上运行 53

32 位环境至 64 位环境 50

应用程序可移植性 52

迁移应用程序 (续)

Java 例程 49

Java 应用程序 49

Java applet 49

嵌入式 SQL

AIX C++ 配置文件 171

DB2 AD 客户机支持 3

## [ R ]

日志管理

用户出口样本文件 81

## [ S ]

实例

数据库管理器 5

数据库管理器

实例 5

松散耦合的事务

Visual Basic

故障诊断 237

在 Windows 上构建 236

Visual Basic 样本 89

## [ W ]

文档

显示 289

文件扩展名

样本 57

问题确定

教程 300

联机信息 300

## [ X ]

消息帮助

调用 298

信息中心

安装 283, 285, 287

## [ Y ]

样本

程序

Java 样本目录 103

程序文件 57

动态重新配置 75

对象链接与嵌入 82

数据库表函数 83

命令行处理器 (CLP) 70

日志管理用户出口 81

支持的语言 57

样本 (续)

C 62

COBOL 71

C++ 65

Java 插件 80

Java WebSphere 80

JDBC 75

Perl 83

PHP 84

SQLJ 78

样本数据库

绑定 46

编目 46

创建 44

设置 43

在主机系统上创建 45

印刷书籍, 订购 296

应用程序开发

环境设置 23

迁移应用程序

在 DB2 的两个版上运行 53

样本数据库设置 43

Java 环境设置 26

Perl

构建应用程序 133

PHP

构建应用程序 135

UNIX 环境设置 30

Windows 环境设置 38

用户出口程序

样本文件 81

用户定义的函数 (UDF)

使用 Visual Basic 进行 OLE 自动化

240

使用 Visual C++ 进行 OLE 自动化

253

AIX C++ 配置文件 173

预编译

SQL 过程 129

预编译器

DB2 AD 客户机支持 3

远程数据对象

使用 Visual Basic 构建 238

Visual Basic 样本 89

## [ Z ]

主机系统

创建样本数据库 45

受支持服务器 7

## [ 数字 ]

32 位应用程序

迁移至 64 位环境 50

## A

- ActiveX 数据对象
  - 使用 Visual Basic 构建 233
  - 使用 Visual C++ 构建 251
  - DB2 AD 客户机支持 3
  - Visual Basic 样本 89
  - Visual C++ 样本 92
- AIX
  - C 多线程应用程序
    - 构建 165
  - C 例程
    - 编译和链接选项 164
  - C 应用程序
    - 编译和链接选项 162
  - C++ 存储过程
    - 使用配置文件构建 172
  - C++ 例程
    - 编译和链接选项 168
  - C++ 嵌入式 SQL
    - 使用配置文件构建 171
  - C++ 应用程序
    - 编译和链接选项 166
  - C++ 用户定义的函数
    - 使用配置文件构建 173
  - C++ API 应用程序
    - 使用配置文件构建 171
  - IBM COBOL 例程
    - 编译和链接选项 179
    - 构建 178
  - IBM COBOL 应用程序
    - 编译和链接选项 177
    - 构建 175
  - Java
    - 环境设置 34
  - Micro Focus COBOL 例程
    - 编译和链接选项 183
  - Micro Focus COBOL 应用程序
    - 编译和链接选项 181
  - REXX 应用程序
    - 构建 184
- API
  - AIX C++ 配置文件 171
- applet
  - 构建 JDBC 106
  - 构建 SQLJ 111
  - 使用要点 104
  - JDBC 样本 75
  - SQLJ 样本 78

## C

- C
  - 错误检查实用程序文件 98
  - 多连接应用程序
    - 在 UNIX 上构建 142

## C (续)

- 多连接应用程序 (续)
  - 在 Windows 上构建 262
- 多线程应用程序
  - AIX 165
  - HP-UX 191
  - Linux 207
  - Solaris 221
  - Windows 254
- 构建文件 93
- 例程
  - 在 UNIX 上构建 144
  - 在 Windows 上构建 257
  - AIX 上的编译选项 164
  - HP-UX 上的编译选项 190
  - Linux 上的编译选项 206
  - Solaris 上的编译选项 220
  - Windows 上的编译选项 261
- 受支持 AIX 版本 8
- 受支持 HP-UX 版本 10
- 受支持 Linux 版本 12
- 受支持 Solaris 版本 17
- 受支持 Windows 版本 19
- 样本 62
- 应用程序
  - 在 UNIX 上构建 141
  - 在 Windows 上构建 254
  - AIX 上的编译选项 162
  - HP-UX 上的编译选项 188
  - Linux 上的编译选项 204
  - Solaris 上的编译选项 218
  - Windows 上的编译选项 256
- makefile 95
- CALL 语句
  - 命令行处理器 124
- CLI (调用级接口)
  - 样本程序文件 69
- CLR (公共语言运行时)
  - 例程
    - 编译和链接选项 250
    - 在 Windows 上构建 246
- COBOL 语言
  - 错误检查实用程序文件 98
  - 构建文件 93
  - 受支持 AIX 版本 8
  - 受支持 HP-UX 版本 10
  - 受支持 Linux 版本 12
  - 受支持 Solaris 版本 17
  - 受支持 Windows 版本 19
  - 样本 71
  - AIX
    - 安装和运行于 161
    - IBM 编译器 174
    - Micro Focus 编译器 180
  - HP-UX
    - 使用 Micro Focus 编译器 198

## COBOL 语言 (续)

- IBM COBOL 例程
  - 在 AIX 上构建 178
  - 在 Windows 上构建 268
  - AIX 上的编译选项 179
  - Windows 上的编译选项 270
- IBM COBOL 应用程序
  - 在 AIX 上构建 175
  - 在 Windows 上构建 264
  - AIX 上的编译选项 177
  - Windows 上的编译选项 267
- Linux
  - Micro Focus 编译器 212
- makefile 95
- Micro Focus 例程
  - 在 UNIX 上构建 155
  - 在 Windows 上构建 274
  - AIX 上的编译选项 183
  - HP-UX 上的编译选项 200
  - Linux 上的编译选项 215
  - Solaris 上的编译选项 230
  - Windows 上的编译选项 276
- Micro Focus 应用程序
  - 在 UNIX 上构建 154
  - 在 Windows 上构建 272
  - AIX 上的编译选项 181
  - HP-UX 上的编译选项 199
  - Linux 上的编译选项 214
  - Solaris 上的编译选项 228
  - Windows 上的编译选项 273
- Solaris Operating Environment
  - Micro Focus 编译器 227
- Windows
  - IBM 编译器 264
  - Micro Focus 编译器 271
- CREATE 语句
  - 和 AIX 例程 160
- CREATE PROCEDURE 语句
  - 具有 SQL 过程 127
- C# .NET
  - 批处理文件 93
  - 受支持 Windows 版本 19
  - 样本 67
  - 应用程序
    - 编译和链接选项 242
    - 在 Windows 上构建 240
- C++
  - 错误检查实用程序文件 98
  - 多连接应用程序
    - 在 UNIX 上构建 149
    - 在 Windows 上构建 262
  - 多线程应用程序
    - AIX 169
    - HP-UX 197
    - Linux 211
    - Solaris Operating Environment 226

## C++ (续)

多线程应用程序 (续)

Windows 254

构建文件 93

例程

在 UNIX 上构建 151

在 Windows 上构建 257

AIX 上的编译选项 168

HP-UX 上的编译选项 196

Linux 上的编译选项 210

Solaris 上的编译选项 225

Windows 上的编译选项 261

使用 Visual C++ 进行 OLE 自动化  
253

受支持 AIX 版本 8

受支持 HP-UX 版本 10

受支持 Linux 版本 12

受支持 Solaris 版本 17

受支持 Windows 版本 19

样本 65

应用程序

在 UNIX 上构建 147

在 Windows 上构建 254

AIX 上的编译选项 166

HP-UX 上的编译选项 193

Linux 上的编译选项 209

Solaris 上的编译选项 223

Windows 上的编译选项 256

AIX 上的 VisualAge 配置文件 170

makefile 95

## D

DB2 个人开发者版 vii

DB2 教程 299

DB2 书籍

打印 PDF 文件 296

DB2 通用开发者版 vii

DB2 信息中心 282

调用 289

DB2 CLI

样本程序文件 69

DB2INSTANCE 环境变量 44

DB2INSTPROF

和数据库管理器 5

DB2PATH

和数据库管理器 5

## E

EXTERNAL NAME 子句

CREATE 语句 160

## F

FORTTRAN 语言

DB2 支持 7

## H

HP-UX

C 多线程应用程序

构建 191

C 例程

编译和链接选项 190

C 应用程序

编译和链接选项 188

C++ 例程

编译和链接选项 196

C++ 应用程序

编译和链接选项 193

Java

环境设置 34

Micro Focus COBOL 例程

编译和链接选项 200

Micro Focus COBOL 应用程序

编译和链接选项 199

HTML 文档

更新 290

## J

Java

插件样本文件 80

构建

JDBC 应用程序 107

JDBC applet 106

SQLJ 应用程序 110, 112

SQLJ applet 111

构建 JDBC 例程 108

构建 SQLJ 例程 117

环境设置 26

迁移应用程序 49

受支持 AIX JDK 8

受支持 HP-UX JDK 10

受支持 Linux JDK 12

受支持 Solaris JDK 17

样本

目录 103

AIX 环境设置 34

applet, 使用要点 104

DB2 AD 客户机支持 3

HP-UX 环境设置 34

JDBC 样本 75

Linux

环境设置 36

makefile 95

Solaris Operating Environment

设置 37

Java (续)

SQLJ 样本 78

UNIX 环境设置 32

WebSphere 样本文件 80

Windows

设置 41

受支持 JDK 版本 19

JDBC (Java 数据库连接)

构建例程 108

构建应用程序 107

构建 applet 106

样本 75

applet, 使用要点 104

DB2 AD 客户机支持 3

JDK\_PATH, 数据库管理器配置关键字 25

## K

KEEPFENCED 数据库管理器配置关键字  
25

Kerberos

安全性协议

样本 87

## L

Linux

C 多线程应用程序

构建 207

C 例程

编译和链接选项 206

C 应用程序

编译和链接选项 204

C++ 例程

编译和链接选项 210

C++ 应用程序

编译和链接选项 209

Java

环境设置 36

Micro Focus COBOL

配置编译器 212

Micro Focus COBOL 例程

编译和链接选项 215

Micro Focus COBOL 应用程序

编译和链接选项 214

## M

makefile 95

Microsoft Transaction Server

Visual Basic 样本 89

MQ 用户定义的函数

设置 28

## N

NOCONVERT 选项 231

## O

Object REXX for Windows 276

样本 85

OLE DB 提供程序

使用 Visual Basic 233

使用 Visual C++ 251

## P

Perl

构建应用程序 133

样本 83

DB2 支持 7

PHP

构建应用程序 135

样本 84

DB2 支持 7

## R

REXX 语言

构建 AIX 应用程序 184

构建 Windows 应用程序 276

受支持 AIX 版本 8

受支持 Windows 版本 19

样本 85

DB2 支持 7

## S

Solaris Operating Environment

例程

C 编译和链接选项 220

C++ 编译和链接选项 225

应用程序

C 编译和链接选项 218

C++ 编译和链接选项 223

C 多线程应用程序

构建 221

Java 设置 37

Micro Focus COBOL 例程

编译和链接选项 230

Micro Focus COBOL 应用程序

编译和链接选项 228

SQL 过程

备份和复原 130

重新绑定 131

创建 127

客户机应用程序 128

样本程序文件 87

SQL 过程 (续)

预编译和绑定选项 129

CALL 语句 124

SQL 语句帮助

调用 299

SQL 92 和 MVS 一致性标志

DB2 AD 客户机支持 3

SQLJ

例程

UNIX 上的编译选项 120

Windows 上的编译选项 122

应用程序

UNIX 上的编译选项 115

Windows 上的编译选项 117

SQLJ (Java 嵌入式 SQL)

程序

构建 110

构建例程 117

构建文件 93

样本 78

应用程序

构建 112

applet

构建 111

applet, 使用要点 104

DB2 AD 客户机支持 3

## U

UNIX

应用程序开发

环境变量设置 31

设置 30

C

构建多连接应用程序 142

C 例程

构建 144

C 应用程序

构建 141

C++

构建多连接应用程序 149

C++ 例程

构建 151

C++ 应用程序

构建 147

Java 设置 32

Micro Focus COBOL 例程

构建 155

Micro Focus COBOL 应用程序

构建 154

SQLJ 例程

编译选项 120

SQLJ 应用程序

编译选项 115

## V

Visual Basic

构建 ADO 应用程序 233

构建 RDO 应用程序 238

受支持 Windows 版本 19

松散耦合的事务

故障诊断 237

在 Windows 上构建 236

样本 89

OLE 自动化 240

Visual Basic .NET

构建应用程序 243

批处理文件 93

样本 91

应用程序

编译和链接选项 245

Visual C++

构建 ADO 应用程序 251

样本 92

OLE 自动化 253

## W

WCHARTYPE CONVERT

预编译器选项 231

wchar\_t 数据类型

转换预编译选项 231

WebSphere MQ 用户定义的函数

设置 28

Windows

管理规范 233

样本 92

应用程序开发

环境设置 38

C/C++ 例程

编译和链接选项 261

构建 257

C/C++ 应用程序

编译和链接选项 256

构建 254

IBM COBOL 例程

编译和链接选项 270

构建 268

IBM COBOL 应用程序

编译和链接选项 267

构建 264

Java

设置 41

Micro Focus COBOL 例程

编译和链接选项 276

构建 274

Micro Focus COBOL 应用程序

编译和链接选项 273

构建 272

Windows (续)

SQLJ 例程

编译选项 122

SQLJ 应用程序

编译选项 117

## [ 特别字符 ]

.NET

例程

编译和链接选项 250

在 Windows 上构建 246

批处理文件 93

C# 样本 67

C# 应用程序

编译和链接选项 242

在 Windows 上构建 240

Visual Basic 样本 91

Visual Basic 应用程序

编译和链接选项 245

在 Windows 上构建 243

---

## 与 IBM 联系

在中国，请致电下列其中一个号码以与 IBM 联系：

- 800-810-1818 或 (010) 84981188 分机 5151，可获得售前客户服务
- 800-810-1818 或 (010) 84981188 分机 5200，可获得售后客户服务
- 800-810-1818 或 (010) 84981188 分机 5017，可获得市场营销与销售的信息

要查找您所在国家或地区的 IBM 营业处，可在网上查看 IBM 全球联系人目录 (Directory of Worldwide Contacts)，网址为：<http://www.ibm.com/planetwide>

---

## 产品信息

有关 DB2 通用数据库产品的信息可通过万维网获取，网址为：<http://www-900.ibm.com/cn/software/db2/>

此站点包含有关 DB2 产品家族、DB2 解决方案、技术前沿与趋势、DB2 服务、成功案例、市场活动、培训与认证、DB2 开发者园地、合作伙伴、下载中心、资料库、第三方分析报告、殊荣与奖项、DB2 新闻以及如何购买 DB2 的最新信息。

有关如何在中国以外的国家或地区与 IBM 联系的信息，请访问 IBM Worldwide 页面，网址为：[www.ibm.com/planetwide](http://www.ibm.com/planetwide)









部件号: CT2TUSC

中国印刷

S152-0168-01



(1P) P/N: CT2TUSC



Spine information:



**IBM® DB2 Universal  
Database™**  
**DB2 通用数据库**

**构建和运行应用程序**

版本 8.2