



# 应用程序开发指南：构建和运行应用程序

版本 8





# 应用程序开发指南：构建和运行应用程序

版本 8

在使用本资料及其支持的产品之前，务必阅读声明中的一般信息。

本文档包含 IBM 的专利信息。它在许可证协议下提供，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。

可以在线方式或通过您当地的 IBM 代表订购 IBM 出版物。

- 要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)。

在美国或加拿大，要从“DB2 市场营销和销售中心”订购 DB2 出版物，请致电 1-800-IBM-4YOU（426-4968）。

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。

---

# 目录

关于本书 . . . . .	vii
----------------	-----

---

## 第 1 部分 应用程序开发环境 . . . . . 1

第 1 章 DB2 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 支持的用于构建和运行应用程序的软件 . . . . .	8
按平台分类的受支持软件 . . . . .	9
AIX 支持的用于构建和运行应用程序的软件 . . . . .	9
HP-UX 支持的用于构建和运行应用程序的软件 . . . . .	10
Linux 支持的用于构建和运行应用程序的软件 . . . . .	11
Solaris 支持的用于构建和运行应用程序的软件 . . . . .	13
Windows 支持的用于构建和运行应用程序的软件 . . . . .	14
第 2 章 设置 . . . . .	17
一般设置信息 . . . . .	17
设置应用程序开发环境 . . . . .	17
更新数据库管理器配置文件 . . . . .	19
设置 Java 环境 . . . . .	20
设置 SQL 过程环境 . . . . .	21
UNIX . . . . .	22
设置 UNIX 应用程序开发环境 . . . . .	22
UNIX 环境变量设置 . . . . .	23
Java . . . . .	24
SQL 过程 . . . . .	28
Windows . . . . .	31
设置 Windows 应用程序开发环境 . . . . .	32
Java . . . . .	35
SQL 过程 . . . . .	37
样本数据库 . . . . .	39
设置样本数据库 . . . . .	39
创建样本数据库 . . . . .	39
在“主机”或 AS/400 和 iSeries 服务器上	
创建样本数据库 . . . . .	41
编目样本数据库 . . . . .	42

绑定样本数据库实用程序 . . . . .	42
迁移应用程序 . . . . .	48
下一步 . . . . .	51

---

## 第 3 章 样本程序和相关文件 . . . . . 53

样本文件 . . . . .	53
样本程序: 结构和设计 . . . . .	58
按语言和应用程序接口分类的样本程序 . . . . .	65
C/C++ 样本 . . . . .	65
DB2 CLI 样本 . . . . .	68
JDBC 样本 . . . . .	70
SQLJ 样本 . . . . .	72
Java WebSphere 样本 . . . . .	74
Java 插件样本 . . . . .	75
COBOL 样本 . . . . .	75
SQL 过程样本 . . . . .	80
Visual Basic 样本 . . . . .	82
Visual C++ 样本 . . . . .	83
Windows Management Instrumentation 样本 . . . . .	83
“对象链接与嵌入” (OLE) 样本 . . . . .	84
对象链接与嵌入数据库 (OLE DB) 表函数	
样本 . . . . .	85
命令行处理器 (CLP) 样本 . . . . .	85
REXX 样本 . . . . .	86
日志管理用户出口样本 . . . . .	88
构建文件、Makefile 和错误检查实用程序 . . . . .	89
构建文件 . . . . .	89
Makefile . . . . .	92
错误检查实用程序 . . . . .	95

---

## 第 2 部分 构建和运行独立于平台的应用程序 . . . . . 99

### 第 4 章 Java . . . . . 101

Java 样本程序 . . . . .	101
Java applet 注意事项 . . . . .	103
JDBC . . . . .	105
构建 JDBC applet . . . . .	105
构建 JDBC 应用程序 . . . . .	106
构建 JDBC 例程 . . . . .	107
SQLJ . . . . .	110

构建 SQLJ 程序 . . . . .	110
构建 SQLJ applet . . . . .	111
构建 SQLJ 应用程序 . . . . .	113
SQLJ 应用程序和 Applet 的 UNIX 构建脚本 . . . . .	114
UNIX 的 SQLJ 应用程序选项 . . . . .	116
SQLJ 应用程序和 Applet 的 Windows 批处理文件 . . . . .	116
Windows 的 SQLJ 应用程序选项 . . . . .	118
构建 SQLJ 例程 . . . . .	119
SQLJ 例程的 UNIX 构建脚本 . . . . .	120
UNIX 的 SQLJ 存储过程选项 . . . . .	122
SQLJ 例程的 Windows 批处理文件 . . . . .	123
Windows 的 SQLJ 存储过程选项 . . . . .	125
<b>第 5 章 SQL 过程 . . . . .</b>	<b>127</b>
创建 SQL 过程 . . . . .	127
在 UNIX 上使用客户机应用程序调用 SQL 过程 . . . . .	128
在 Windows 上使用客户机应用程序调用 SQL 过程 . . . . .	129
保留 SQL 过程的中间文件 . . . . .	131
为 SQL 过程定制预编译和绑定选项 . . . . .	132
备份和复原 SQL 过程 . . . . .	133
分发可编译 SQL 过程 . . . . .	134
重新绑定 SQL 过程 . . . . .	135
<b>第 6 章 调用存储过程 . . . . .</b>	<b>137</b>
使用 CALL 语句调用存储过程 . . . . .	137
<hr/>	
<b>第 3 部分 构建和运行特定于平台的应用程序 . . . . .</b>	<b>139</b>
<b>第 7 章 AIX . . . . .</b>	<b>141</b>
重要注意事项 . . . . .	141
例程的 AIX 导出文件 . . . . .	142
AIX 例程和 CREATE 语句 . . . . .	142
替换 AIX 共享库 . . . . .	143
在 AIX 上安装 COBOL 的注意事项 . . . . .	143
IBM C . . . . .	144
在 AIX 上构建 C 应用程序 . . . . .	144
为 C 应用程序构建脚本 . . . . .	145
AIX C 应用程序编译和链接选项 . . . . .	147
在 AIX 上构建 C 例程 . . . . .	148
为 C 例程构建脚本 . . . . .	151
AIX C 例程编译和链接选项 . . . . .	152

在 AIX 上构建 C 多线程应用程序 . . . . .	153
为 C 多线程应用程序构建脚本 . . . . .	154
VisualAge C++ . . . . .	155
在 AIX 上构建 C++ 应用程序 . . . . .	155
为 C++ 应用程序构建脚本 . . . . .	157
AIX C++ 应用程序编译和链接选项 . . . . .	158
在 AIX 上构建 C++ 例程 . . . . .	159
为 C++ 例程构建脚本 . . . . .	162
AIX C++ 例程编译和链接选项 . . . . .	163
在 AIX 上构建 C++ 多线程应用程序 . . . . .	164
为 C++ 多线程应用程序构建脚本 . . . . .	165
VisualAge C++ 配置文件 . . . . .	166
使用配置文件构建 VisualAge C++ 程序 . . . . .	166
使用配置文件构建 C++ DB2 API 应用程序 . . . . .	167
使用配置文件构建 C++ 嵌入式 SQL 应用程序 . . . . .	168
使用配置文件构建 C++ 存储过程 . . . . .	169
使用配置文件构建 C++ 用户定义函数 . . . . .	170
IBM COBOL Set AIX 版 . . . . .	171
在 AIX 上配置 IBM COBOL 编译器 . . . . .	171
在 AIX 上构建 IBM COBOL 应用程序 . . . . .	172
为 IBM COBOL 应用程序构建脚本 . . . . .	174
AIX IBM COBOL 应用程序编译和链接选项 . . . . .	175
在 AIX 上构建 IBM COBOL 例程 . . . . .	175
为 IBM COBOL 例程构建脚本 . . . . .	177
AIX IBM COBOL 例程编译和链接选项 . . . . .	178
Micro Focus COBOL . . . . .	179
在 AIX 上配置 Micro Focus COBOL 编译器 . . . . .	179
在 AIX 上构建 Micro Focus COBOL 应用程序 . . . . .	179
为 Micro Focus COBOL 应用程序构建脚本 . . . . .	181
AIX Micro Focus COBOL 应用程序编译和链接选项 . . . . .	182
在 AIX 上构建 Micro Focus COBOL 例程 . . . . .	182
为 Micro Focus COBOL 例程构建脚本 . . . . .	184
AIX Micro Focus COBOL 例程编译和链接选项 . . . . .	185
REXX . . . . .	186
在 AIX 上构建 REXX 应用程序 . . . . .	186
<b>第 8 章 HP-UX . . . . .</b>	<b>189</b>
HP-UX C . . . . .	189

在 HP-UX 上构建 C 应用程序 . . . . .	189	在 Linux 上构建 C++ 例程. . . . .	235
为 C 应用程序构建脚本. . . . .	191	为 C++ 例程构建脚本 . . . . .	239
HP-UX C 应用程序编译和链接选项 . . . . .	192	Linux C++ 例程编译和链接选项 . . . . .	240
在 HP-UX 上构建 C 例程 . . . . .	193	在 Linux 上构建 C++ 多线程应用程序	241
为 C 例程构建脚本 . . . . .	197	为 C++ 多线程应用程序构建脚本. . . . .	241
HP-UX C 例程编译和链接选项 . . . . .	198		
在 HP-UX 上构建 C 多线程应用程序 . . . . .	199	<b>第 10 章 Solaris . . . . . 243</b>	
为 C 多线程应用程序构建脚本 . . . . .	200	Forte C . . . . .	243
HP-UX C++ . . . . .	200	在 Solaris 上构建 C 应用程序. . . . .	243
在 HP-UX 上构建 C++ 应用程序. . . . .	200	为 C 应用程序构建脚本. . . . .	245
为 C++ 应用程序构建脚本. . . . .	202	Solaris C 应用程序编译和链接选项 . . . . .	246
HP-UX C++ 应用程序编译和链接选项 . . . . .	203	在 Solaris 上构建 C 例程 . . . . .	247
在 HP-UX 上构建 C++ 例程 . . . . .	204	为 C 例程构建脚本 . . . . .	251
为 C++ 例程构建脚本 . . . . .	208	Solaris C 例程编译和链接选项. . . . .	252
HP-UX C++ 例程编译和链接选项. . . . .	209	在 Solaris 上构建 C 多线程应用程序 . . . . .	253
在 HP-UX 上构建 C++ 多线程应用程序	210	为 C 多线程应用程序构建脚本 . . . . .	255
为 C++ 多线程应用程序构建脚本. . . . .	211	Forte C++ . . . . .	255
Micro Focus COBOL . . . . .	212	在 Solaris 上构建 C++ 应用程序. . . . .	255
在 HP-UX 上配置 Micro Focus COBOL		为 C++ 应用程序构建脚本. . . . .	257
编译器 . . . . .	212	Solaris C++ 应用程序编译和链接选项 . . . . .	258
在 HP-UX 上构建 Micro Focus COBOL		在 Solaris 上构建 C++ 例程 . . . . .	259
应用程序. . . . .	213	为 C++ 例程构建脚本 . . . . .	263
为 Micro Focus COBOL 应用程序构建脚		Solaris C++ 例程编译和链接选项. . . . .	264
本 . . . . .	215	在 Solaris 上构建 C++ 多线程应用程序	265
HP-UX Micro Focus COBOL 应用程序编		为 C++ 多线程应用程序构建脚本. . . . .	267
译和链接选项 . . . . .	216	Micro Focus COBOL . . . . .	267
在 HP-UX 上构建 Micro Focus COBOL		在 Solaris 上配置 Micro Focus COBOL 编	
例程 . . . . .	216	译器 . . . . .	267
为 Micro Focus COBOL 例程构建脚本	218	在 Solaris 上构建 Micro Focus COBOL 应	
HP-UX Micro Focus COBOL 例程编译和		用程序 . . . . .	268
链接选项. . . . .	219	为 Micro Focus COBOL 应用程序构建脚	
		本 . . . . .	270
<b>第 9 章 Linux . . . . . 221</b>		Solaris Micro Focus COBOL 应用程序编译	
Linux C . . . . .	221	和链接选项 . . . . .	270
在 Linux 上构建 C 应用程序 . . . . .	221	在 Solaris 上构建 Micro Focus COBOL 例	
为 C 应用程序构建脚本. . . . .	223	程 . . . . .	271
Linux C 应用程序编译和链接选项 . . . . .	224	为 Micro Focus COBOL 例程构建脚本	273
在 Linux 上构建 C 例程 . . . . .	225	Solaris Micro Focus COBOL 例程编译和链	
为 C 例程构建脚本 . . . . .	228	接选项 . . . . .	274
Linux C 例程编译和链接选项 . . . . .	229		
在 Linux 上构建 C 多线程应用程序. . . . .	230	<b>第 11 章 Windows 操作系统 . . . . . 275</b>	
为 C 多线程应用程序构建脚本 . . . . .	231	WCHARTYPE CONVERT 预编译选项 . . . . .	275
Linux C++ . . . . .	232	对象链接与嵌入数据库 (OLE DB) 表函数	276
在 Linux 上构建 C++ 应用程序 . . . . .	232	Windows Management	
为 C++ 应用程序构建脚本. . . . .	233	Instrumentation (WMI) . . . . .	277
Linux C++ 应用程序编译和链接选项. . . . .	234	Microsoft Visual Basic . . . . .	277

使用 Visual Basic 构建 ADO 应用程序	277
使用 Visual Basic 构建 RDO 应用程序	280
使用 Visual Basic 的对象链接与嵌入 (OLE) 自动化	282
Microsoft Visual C++	283
使用 Visual C++ 构建 ADO 应用程序	283
使用 Visual C++ 的对象链接与嵌入 (OLE) 自动化	285
在 Windows 上构建 C/C++ 应用程序	285
C/C++ 应用程序的批处理文件	287
Windows C/C++ 应用程序编译和链接选项	288
在 Windows 上构建 C/C++ 例程	289
C/C++ 例程的批处理文件	293
Windows C/C++ 例程编译和链接选项	294
IBM VisualAge COBOL	295
在 Windows 上配置 IBM COBOL 编译器	295
在 Windows 上构建 IBM COBOL 应用程序	296
IBM COBOL 应用程序的批处理文件	297
Windows IBM COBOL 应用程序编译和链接选项	298
在 Windows 上构建 IBM COBOL 例程	299
IBM COBOL 例程的批处理文件	300
Windows IBM COBOL 例程编译和链接选项	301
Micro Focus COBOL	301
在 Windows 上配置 Micro Focus COBOL 编译器	301
在 Windows 上构建 Micro Focus COBOL 应用程序	302
Micro Focus COBOL 应用程序的批处理文件	304
Windows Micro Focus COBOL 应用程序编译和链接选项	304
在 Windows 上构建 Micro Focus COBOL 例程	305
Micro Focus COBOL 例程的批处理文件	307
Windows Micro Focus COBOL 例程编译和链接选项	307
Object REXX	308
在 Windows 上构建 Object REXX 应用程序	308

## 第 4 部分 附录 . . . . . 309

### 附录 A. “DB2 通用数据库” 技术信息 . . 311

“DB2 通用数据库” 技术信息概述	311
DB2 文档的修订包	311
DB2 技术信息类别	312
从 PDF 文件打印 DB2 书籍	318
订购打印的 DB2 书籍	319
访问联机帮助	320
通过从浏览器访问 “DB2 信息中心” 来查找主题	321
通过从管理工具访问 “DB2 信息中心” 来查找产品信息	323
直接从 DB2 HTML 文档 CD 联机查看技术文档	324
更新安装在机器上的 HTML 文档	325
将文件从 DB2 HTML 文档 CD 复制到 Web 服务器	326
对于使用 Netscape 4.x 搜索 DB2 文档进行故障诊断	327
搜索 DB2 文档	328
联机 DB2 故障诊断信息	329
易使用性	329
键盘输入和导航	330
界面显示的易使用性	330
备用警告信号	330
与辅助技术的兼容性	330
可访问文档	330
DB2 教程	330
从浏览器访问的 DB2 信息中心	331

### 附录 B. 声明 . . . . . 333

商标	336
----	-----

### 索引 . . . . . 339

### 与 IBM 联系 . . . . . 345

产品信息	345
------	-----



---

## 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。为以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供了建议，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 应用程序中使用动态和静态 SQL 的特定问题。
  - 受支持的主机语言和解释语言，如 C/C++、COBOL、Perl 和 REXX，以及如何在这些语言编写的应用程序中使用嵌入式 SQL。
  - Java (JDBC 和 SQLj)，以及构建使用 WebSphere Application Server 的 Java 应用程序的注意事项。
  - IBM OLE DB Provider for DB2 Server。提供有关对 OLE DB 服务、组件和特性的 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行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身份列、声明的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。

- 存取主机和 iSeries 环境的应用程序。描述有关存取主机和 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服务器端编程（包括例程、大对象、用户定义类型和触发器）的信息。它包含有关下列各项的信息：
  - 例程（存储过程、用户定义函数和方法），包括：
    - 例程性能、安全性、库管理注意事项和限制。
    - 注册和编写例程，包括 CREATE 语句和调试。
    - 过程参数方式和参数控制。
    - 过程结果集。
    - UDF 功能部件，包括暂存区、标量和表函数。
    - SQL 过程，包括调试和条件控制。
    - 参数样式、权限和外部例程的绑定。
    - 有关 C、Java 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，以及将单值类型和 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符以及对对象标识符列的约束。
  - 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、主机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：

- 如何设置应用程序开发环境（包括 Java 和 SQL 过程）、如何设置样本数据库以及如何从先前版本的 DB2 迁移应用程序的特定指示信息。
- 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 支持

DB2 应用程序开发客户机 . . . . .	3	Linux 支持的用于构建和运行应用程序的软	
数据库管理器实例 . . . . .	5	件 . . . . .	11
DB2 支持的服务器 . . . . .	7	Solaris 支持的用于构建和运行应用程序的软	
DB2 支持的用于构建和运行应用程序的软件 . . . . .	8	件 . . . . .	13
按平台分类的受支持软件 . . . . .	9	Windows 支持的用于构建和运行应用程序的	
AIX 支持的用于构建和运行应用程序的软件 . . . . .	9	软件. . . . .	14
HP-UX 支持的用于构建和运行应用程序的软			
件 . . . . .	10		

《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 来构建应用程序:

- AIX
- HP-UX
- Linux
- Solaris 操作环境
- Windows 操作系统

---

## DB2 应用程序开发客户机

DB2 应用程序开发 (DB2 AD) 客户机提供工具和环境，以便您开发可访问 DB2 服务器和实现了“分布式关系数据库体系结构”(DRDA)的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

适用于本书所描述平台的 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 调用层接口 (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，而从各个其它

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（如果有的话）开发的应用程序。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。
- **Java 开发工具箱（JDK）**，IBM 的 JDK 1.3.1 和 Java 运行时环境（JRE）1.3.1 AIX 版、IBM 开发者工具箱和运行时环境版本 1.3.1 Linux 版和 Windows 版、HP-UX Software Developer's Kit 和 Runtime Environment 1.3.1 for HP-UX 以及 Sun 公司的 Java 开发工具箱 1.3.1 Solaris 版。除了 Solaris，如果任何选择安装的组件需要 Java 运行，则将安装每个操作系统各自的 JDK。如果未选择安装任何上述组件，仍可以选择安装 JDK。在 Solaris 上，随 DB2 一起交付的 JDK 1.3.1 必须由用户安装。
- Windows 操作系统上的“**ActiveX 数据对象（ADO）**和“**对象连接与嵌入（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**。
- **DB2 开发中心**，支持例程（存储过程和用户定义函数）和结构化类型快速开发的图形应用程序。“开发中心”提供了一个开发环境，它支持从工作站一直到 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”、“命令中心”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 开发加载项”：
  - Microsoft Visual C++，版本 6
  - Microsoft Visual Basic，版本 6
  - Microsoft Visual InterDev，版本 6
- **交互式 SQL**，可通过“命令中心”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特定查询。
- **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，可以从匿名 FTP 站点的 PDF 文件 prepapi.pdf 获得，该站点的网址为：



<ftp://ftp.software.ibm.com/ps/products/db2/info/>

- **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 版和 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 9 页的『AIX 支持的用于构建和运行应用程序的软件』
- 第 10 页的『HP-UX 支持的用于构建和运行应用程序的软件』
- 第 11 页的『Linux 支持的用于构建和运行应用程序的软件』
- 第 13 页的『Solaris 支持的用于构建和运行应用程序的软件』
- 第 14 页的『Windows 支持的用于构建和运行应用程序的软件』

---

## 数据库管理器实例

DB2<sup>®</sup> 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。

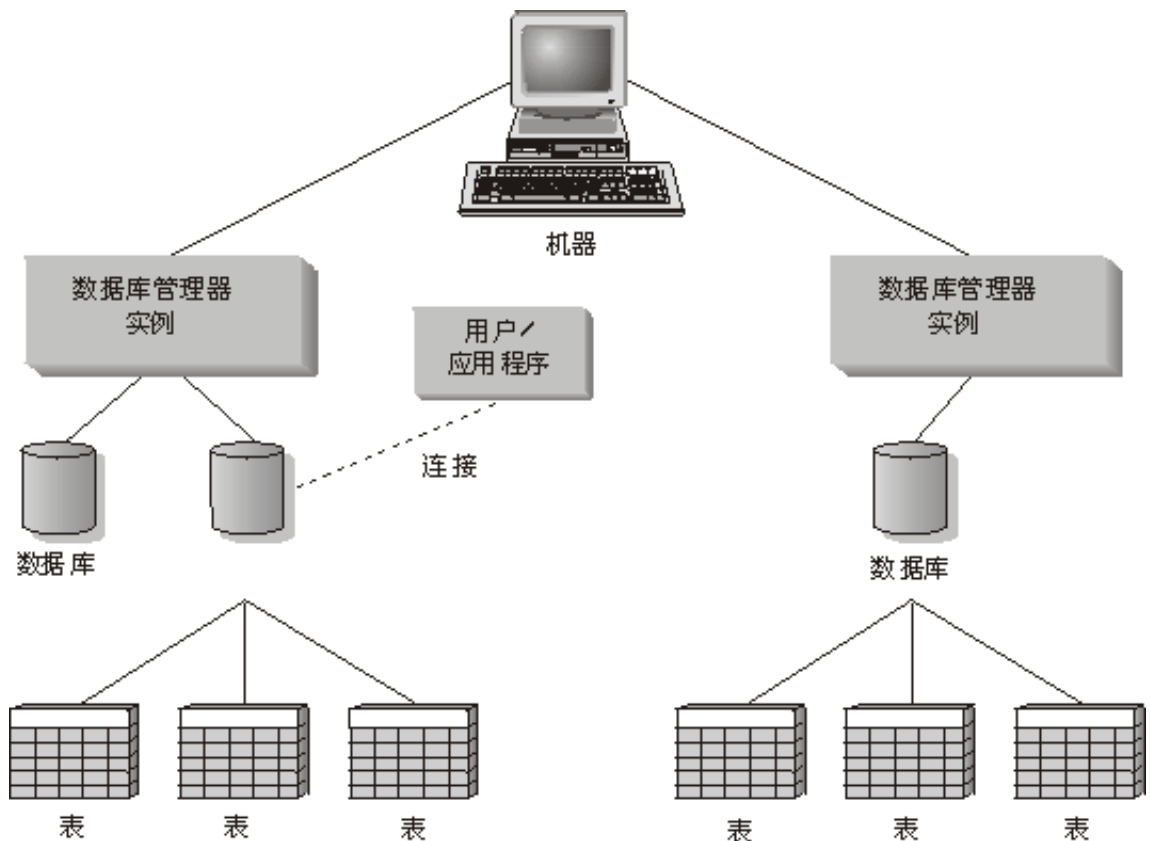


图 1. 数据库管理器实例

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。

借助于 UNIX® 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 8.1。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。

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 8.1。

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/sql1ib，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。

---

## 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/2000/XP/.NET Server 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 版和 OS/390 版
  - DB2 AS/400 版和 iSeries 版
  - DB2 VSE 版和 VM 版（以前是 SQL/DS VM 版和 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。

---

## DB2 支持的用于构建和运行应用程序的软件

“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
- Windows 操作系统

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。但是，DB2 支持在所有受支持的 64 位操作系统环境（除了 Linux IA64 之外）上运行 32 位应用程序和例程（存储过程 and 用户定义函数）。

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。

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 0.7.6 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 的应用程序存取 DB2 的方法。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 ODBC 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。

[www.ibm.com/software/data/db2/udb/winon2unix/support](http://www.ibm.com/software/data/db2/udb/winon2unix/support)

#### 相关参考:

- 第 9 页的『AIX 支持的用于构建和运行应用程序的软件』
- 第 10 页的『HP-UX 支持的用于构建和运行应用程序的软件』
- 第 11 页的『Linux 支持的用于构建和运行应用程序的软件』
- 第 13 页的『Solaris 支持的用于构建和运行应用程序的软件』
- 第 14 页的『Windows 支持的用于构建和运行应用程序的软件』

---

## 按平台分类的受支持软件

### AIX 支持的用于构建和运行应用程序的软件

DB2 AIX 版支持下列操作系统:

#### AIX/6000

带有维护包 9 的版本 4.3.3 和更新版本

带有维护包 1 的版本 5.1.0 和更新版本

DB2 AIX 版支持下列编程语言和编译器:

**C** IBM C AIX 版的版本 5.0

**C++** IBM VisualAge C++ 版本 5.0

#### COBOL

IBM COBOL Set AIX 版的版本 1.1

Micro Focus COBOL Server Express 版本 2.0.10

## Fortran

IBM XL Fortran AIX 版的版本 4.1（仅限于 32 位）和版本 5.1.0（适用于 32 位和 64 位）

**Java** IBM 的“Java 开发工具箱（JDK）版本 1.3.1 和 Java 运行时环境（JRE）版本 1.3.1 AIX 版”（必要时由 DB2 安装）

注：如果任何选择安装的组件需要 Java 运行，则 DB2 将安装此 JDK。  
如果未选择安装任何组件，仍可以选择安装 JDK。

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

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

IBM Object REXX AIX 版的版本 1.1

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，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 8 页的『DB2 支持的用于构建和运行应用程序的软件』
- 『分区 DB2 服务器的安装需求（AIX）』（《DB2 服务器快速入门》）
- 第 146 页的『AIX C 应用程序编译和链接选项』
- 第 152 页的『AIX C 例程编译和链接选项』
- 第 157 页的『AIX C++ 应用程序编译和链接选项』
- 第 163 页的『AIX C++ 例程编译和链接选项』
- 第 174 页的『AIX IBM COBOL 应用程序编译和链接选项』
- 第 177 页的『AIX IBM COBOL 例程编译和链接选项』
- 第 182 页的『AIX Micro Focus COBOL 应用程序编译和链接选项』
- 第 184 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 支持的用于构建和运行应用程序的软件

DB2 HP-UX 版支持下列操作系统：

### HP-UX

版本 11 和 11i

DB2 HP-UX 版支持下列编程语言和编译器:

**C** HP C 编译器版本 B.11.11.02

**C++** HP aC++ 版本 A.03.31

### **COBOL**

Micro Focus COBOL 版本 4.1

### **Fortran**

HP Fortran/9000 版本 10.0

HP-UX F77 B.11.00.01

**Java** Hewlett-Packard 公司的 Software Developer's Kit 和 Runtime Environment 1.3.1 for HP-UX 11.0 和 11i PA-RISC (必要时由 DB2 安装)

注: 如果任何选择安装的组件需要 Java 运行, 则 DB2 将安装此 SDK。  
如果未选择安装任何组件, 仍可以选择安装 SDK。

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

有关 DB2 HP-UX 版软件支持更新的内容, 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 8 页的『DB2 支持的用于构建和运行应用程序的软件』
- 『分区 DB2 服务器的安装需求 (HP-UX)』 (《DB2 服务器快速入门》)
- 第 192 页的『HP-UX C 应用程序编译和链接选项』
- 第 198 页的『HP-UX C 例程编译和链接选项』
- 第 203 页的『HP-UX C++ 应用程序编译和链接选项』
- 第 208 页的『HP-UX C++ 例程编译和链接选项』
- 第 215 页的『HP-UX Micro Focus COBOL 应用程序编译和链接选项』
- 第 218 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 支持的用于构建和运行应用程序的软件**

**DB2 Linux Intel x86 版 (32 位体系结构)** 支持以下操作系统环境:

Linux kernel 版本 2.4.9 或更新版本、glibc 版本 2.2.4 或更新版本以及 rpm (必需安装)

**DB2 Linux on S/390 版** 支持以下操作系统环境:

下列其中一项:

- RedHat v7.2
- SuSE SLES-7 Linux S/390 版

DB2 Linux Intel x86 版和 S/390 版支持下列编程语言和编译器:

**C/C++** 对于 Intel 上的 Linux: GNU/Linux gcc 和 g++ 版本 2.95.3 和 2.96

对于 Linux/390: GNU/Linux gcc 和 g++ 版本 2.95.3

有关对 Intel 上的 Linux 的 gcc 和 g++ 版本 3.0 支持和对 Linux/390 的未来版本支持信息, 请访问 DB2 应用程序开发 Web 站点:

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

**Java** 对于 Intel 上的 Linux: IBM 开发者工具箱和运行时环境 Linux 版 Java 2 技术版版本 1.3.1 的 32 位版本 (必要时由 DB2 安装)

对于 Linux/390: IBM zSeries 开发者工具箱 Linux 版 Java 2 技术版 (级别为 Sun 1.3.1 SDK, 必要时由 DB2 安装)

**注:** 如果任何选择安装的组件需要 Java 运行, 则 DB2 将安装适当的“开发者工具箱”。如果未选择安装任何组件, 仍可以选择安装“开发者工具箱”。

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

**REXX** 对于 Intel 上的 Linux: Object REXX Interpreter for Linux 版本 2.1

对于 Linux/390: Object REXX 2.2.0 for Linux/390

**DB2 Linux IA64 版支持以下操作系统环境:**

下列其中一项:

- Red Hat 7.2
- SuSE SLES-7 Linux IA64 版

DB2 Linux IA64 版支持下列编程语言和编译器:

**C** GNU/Linux gcc 版本 3.0.2

**C++** GNU/Linux g++ 版本 3.0.2

**Java** IBM 开发者工具箱和运行时环境 Linux 版 Java 2 技术版版本 1.3.1 的 64 位版本 (必要时由 DB2 安装)。要使用此 JDK, 还必须先安装 gcc 3.0.2 和 gcc3 libstdc++ 运行时库。



**注：** 如果任何选择安装的组件需要 Java 运行，则 DB2 将安装此“开发者工具箱”。如果未选择安装任何组件，仍可以选择安装“开发者工具箱”。

## **Perl** Perl 5.6

**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函数）。

有关 DB2 Linux 版软件更新的内容，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 8 页的『DB2 支持的用于构建和运行应用程序的软件』
- 『分区 DB2 服务器的安装需求（Linux）』（《DB2 服务器快速入门》）
- 第 224 页的『Linux C 应用程序编译和链接选项』
- 第 229 页的『Linux C 例程编译和链接选项』
- 第 234 页的『Linux C++ 应用程序编译和链接选项』
- 第 239 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》）

## **Solaris** 支持的用于构建和运行应用程序的软件

DB2 Solaris 版支持下列操作系统：

### **Solaris**

Solaris 7、Solaris 8 和 Solaris 9

DB2 Solaris 版支持下列编程语言和编译器：

**C** Forte C 版本 5.0、6 和 6.1

**注：** 这些编译器版本过去经常称为“SPARCompiler”。

**C++** Forte C++ 版本 5.0、6 和 6.1

**注：** 这些编译器版本过去经常称为“SPARCompiler”。

### **COBOL**

Micro Focus COBOL Server Express 版本 2.0.10

## Fortran

SPARCompiler Fortran 版本 4.2 和版本 5.0

**Java** Sun 公司的 Java 开发工具箱 (JDK) 版本 1.3.1 Solaris 版 (随 DB2 一起交付)

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

有关 DB2 Solaris 版软件更新的内容, 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 8 页的『DB2 支持的用于构建和运行应用程序的软件』
- 『分区 DB2 服务器的安装需求 (Solaris 操作环境)』(《DB2 服务器快速入门》)
- 第 246 页的『Solaris C 应用程序编译和链接选项』
- 第 252 页的『Solaris C 例程编译和链接选项』
- 第 258 页的『Solaris C++ 应用程序编译和链接选项』
- 第 264 页的『Solaris C++ 例程编译和链接选项』
- 第 270 页的『Solaris Micro Focus COBOL 应用程序编译和链接选项』
- 第 273 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)』(《DB2 服务器快速入门》)

## Windows 支持的用于构建和运行应用程序的软件

**DB2 Windows 32 位版操作系统支持下列操作系统:**

**Microsoft Windows XP**

**Microsoft Windows .NET**

**Microsoft Windows 2000**

**Microsoft Windows NT**

带有服务包 6a 的版本 4.0 或更新版本。

**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 版本 4.2 和版本 5.0

Microsoft Visual Basic 6.0 专业版

**C/C++** Microsoft Visual C++ 版本 5.0 和 6.0

Intel C++ 编译器 32 位版应用程序版本 5 或更新版本

## **COBOL**

Micro Focus COBOL 版本 4.0.20

Micro Focus COBOL Net Express 版本 3.1.0

IBM VisualAge COBOL 版本 2.0

**REXX** IBM Object REXX Windows NT/95 版版本 1.1

有关获取 IBM Object REXX Windows 版的信息，访问：

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

**Java** IBM 开发者工具箱和运行时环境 Windows 版 Java 2 技术版版本 1.3.1 的 32 位版本（必要时由 DB2 安装）

**注：**如果任何选择安装的组件需要 Java 运行，则 DB2 将安装此“开发者工具箱”。如果未选择安装任何组件，仍可以选择安装“开发者工具箱”。

Sun 公司的 Java 开发工具箱（JDK）1.3.1 Win32 版

**Perl** Perl 5.004\_04 和 DBI 0.93

**Microsoft Windows 脚本编制主机**

版本 5.1

**DB2 Windows 64 位版操作系统支持下列操作系统：**

**Microsoft Windows XP 64 位版**

**Microsoft Windows .NET Server 64 位版**

DB2 Windows 版 64 位操作系统支持下列编程语言：

**C/C++** Intel C++ 编译器 Itanium 版本 6.0

Microsoft C/C++ 编译器 Intel Itanium 版体系结构

**Java** IBM 开发者工具箱和运行时环境 Windows 版 Java 2 技术版版本 1.3.1 的 64 位版本（必要时由 DB2 安装）

**注：**如果任何选择安装的组件需要 Java 运行，则 DB2 将安装此“开发者工具箱”。如果未选择安装任何组件，仍可以选择安装“开发者工具箱”。

## Microsoft Windows 脚本编制主机

版本 5.1

注: Windows .NET Server 包括下列所有服务器:

- Windows .NET Web 服务器
- Windows .NET 标准服务器
- Windows .NET 企业服务器
- Windows .NET 数据中心服务器

有关 DB2 Windows 版软件支持更新的内容, 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 8 页的『DB2 支持的用于构建和运行应用程序的软件』
- 『DB2 服务器的安装需求 (Windows)』 (《DB2 服务器快速入门》)
- 第 288 页的『Windows C/C++ 应用程序编译和链接选项』
- 第 294 页的『Windows C/C++ 例程编译和链接选项』
- 第 298 页的『Windows IBM COBOL 应用程序编译和链接选项』
- 第 300 页的『Windows IBM COBOL 例程编译和链接选项』
- 第 304 页的『Windows Micro Focus COBOL 应用程序编译和链接选项』
- 第 307 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》)
- 『分区 DB2 服务器的安装需求 (Windows)』 (《DB2 服务器快速入门》)

---

## 第 2 章 设置

一般设置信息 . . . . .	17	Windows . . . . .	31
设置应用程序开发环境 . . . . .	17	设置 Windows 应用程序开发环境 . . . . .	32
更新数据库管理器配置文件 . . . . .	19	Java . . . . .	35
设置 Java 环境 . . . . .	20	设置 Windows Java 环境 . . . . .	35
设置 SQL 过程环境 . . . . .	21	Windows Java 环境设置 . . . . .	36
UNIX . . . . .	22	SQL 过程 . . . . .	37
设置 UNIX 应用程序开发环境 . . . . .	22	设置 Windows SQL 过程环境 . . . . .	37
UNIX 环境变量设置 . . . . .	23	样本数据库 . . . . .	39
Java . . . . .	24	设置样本数据库 . . . . .	39
设置 UNIX Java 环境 . . . . .	24	创建样本数据库 . . . . .	39
设置 AIX Java 环境 . . . . .	25	在“主机”或 AS/400 和 iSeries 服务器上	
设置 HP-UX Java 环境 . . . . .	26	创建样本数据库 . . . . .	41
设置 Linux Java 环境 . . . . .	26	编目样本数据库 . . . . .	42
设置 Solaris Java 环境 . . . . .	27	绑定样本数据库实用程序 . . . . .	42
SQL 过程 . . . . .	28	迁移应用程序 . . . . .	48
设置 UNIX SQL 过程环境 . . . . .	28	下一步 . . . . .	51
UNIX 缺省 DB2 SQLROUTINE 编译命			
令值 . . . . .	30		

---

### 一般设置信息

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

### 设置应用程序开发环境

为构建和运行 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。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，请查看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, 否则更新数据库管理器配置文件
2. 如果要使用 DB2 CLI、Java 或 SQL 过程来编程, 则必须用下面各部分中的指示信息配置环境后才能执行任何特定于平台的更改。
  - 设置 CLI 环境
  - 设置 Java 环境
  - 设置 “SQL 过程” 环境
3. 用下面各主题中的指示信息配置操作系统环境:
  - 设置 UNIX 环境
  - 设置 Windows 环境
4. 可选: 设置样本数据库

## 相关概念:

- 第 5 页的『数据库管理器实例』
- 第 44 页的『迁移应用程序』

## 相关任务:

- 第 19 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 20 页的『设置 Java 环境』
- 第 21 页的『设置 SQL 过程环境』
- 第 22 页的『设置 UNIX 应用程序开发环境』
- 第 32 页的『设置 Windows 应用程序开发环境』
- 第 39 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 8 页的『DB2 支持的用于构建和运行应用程序的软件』
- 第 9 页的『AIX 支持的用于构建和运行应用程序的软件』
- 第 10 页的『HP-UX 支持的用于构建和运行应用程序的软件』
- 第 11 页的『Linux 支持的用于构建和运行应用程序的软件』
- 第 13 页的『Solaris 支持的用于构建和运行应用程序的软件』
- 第 14 页的『Windows 支持的用于构建和运行应用程序的软件』

## 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。

对于例程（存储过程和 UDF），关键字 `KEEPFENCED` 的缺省值为 `YES`。这将使例程进程保持为活动状态。如果要开发例程，您可能想要测试多次装入同一共享库。此缺省设置可能会干扰重新装入库。开发例程时，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`。

对于线程安全的例程，用来运行例程的进程在完成后仍留在实例上（仅用来调用例程的进程内的线程将终止）。由于此原因，当开发新例程时，应将例程定义为非线程安全的。然后，如果适当的话，使例程能够在处于生产方式时以线程安全方式运行。

**注：**在先前 DB2 版本中，`KEEPFENCED` 称为 `KEEPDARI`。

为进行 Java 应用程序开发，需要用“Java 开发工具箱”（JDK）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，输入：

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

例如，要将关键字 `KEEPFENCED` 设置为 `NO`：

```
db2 update dbm cfg using KEEPFENCED 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，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。

### 相关任务:

- 第 20 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

可以用适用于您平台的“Java 开发工具箱” (JDK) 来开发存取 DB2 数据库的 Java 程序。该 JDK 包括“Java 数据库连接” (JDBC), 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。

DB2 Java 嵌入式 SQL (SQLJ) 支持作为 DB2 AD 客户机的一部分提供。利用 DB2 SQLJ 支持和 DB2 JDBC 支持, 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:

- DB2 SQLJ 转换程序 sqlj, 它使用 Java 源语句替换 SQLJ 程序中的嵌入式 SQL 语句, 并生成一个序列化概要文件, 该概要文件包含有关在 SQLJ 程序中找到的 SQL 操作的信息。
- DB2 SQLJ 概要文件定制器 db2profc, 它预编译存储在序列化概要文件中的 SQL 语句, 将它们定制成运行时函数调用, 并在该 DB2 数据库中生成一个程序包。
- DB2 SQLJ 概要文件打印机 db2profp, 它以平面文本格式打印概要文件的 DB2 定制版本的内容。

过程:

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, 请参考 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：

- 设置 UNIX Java 环境
- 设置 Windows Java 环境

有关最新的 DB2 Java 应用程序开发更新内容，访问以下 Web 页：

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

#### 相关任务：

- 第 24 页的『设置 UNIX Java 环境』
- 第 35 页的『设置 Windows Java 环境』

#### 相关参考：

- 『db2profc - DB2 SQLj Profile Customizer Command』（*Command Reference*）
- 『db2profp - DB2 SQLj Profile Printer Command』（*Command Reference*）

## 设置 SQL 过程环境

编译器配置是用两个 DB2 注册表变量完成的：

#### **DB2\_SQLROUTINE\_COMPILER\_PATH**

指定了设置编译器环境变量的脚本的路径名。

#### **DB2\_SQLROUTINE\_COMPILE\_COMMAND**

指定了完整的命令，DB2 使用该命令来编译 SQL 过程生成的 C 文件。

可以使用 db2set 命令或使用“开发中心”的“SQL 存储过程构建选项”对话框来设置这些 DB2 注册表变量的值。使用“SQL 存储过程构建选项”对话框就不再需要物理访问数据库服务器以及为使更改生效而重新启动数据库服务器。

下面的各步骤将演示 db2set 命令。

#### 先决条件：

要获得 SQL 过程支持，必须在服务器上安装“应用程序开发客户机”和 DB2 支持的 C 或 C++ 编译器。

#### 过程：

要配置 SQL 过程环境，根据您的平台执行下列其中一项主题中的步骤：

- 设置 UNIX SQL 过程环境
- 设置 Windows SQL 过程环境

### 相关概念:

- 『DB2 注册表和环境变量』（《管理指南：性能》）

### 相关任务:

- 第 28 页的『设置 UNIX SQL 过程环境』
- 第 37 页的『设置 Windows SQL 过程环境』
- 第 127 页的『创建 SQL 过程』
- 第 137 页的『使用 CALL 语句调用存储过程』
- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』
- 第 135 页的『重新绑定 SQL 过程』

---

## UNIX

有关 UNIX DB2 CLI 设置信息，参见 *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:

为正在使用的外壳程序运行正确的脚本:

对于 **bash** 或 **Korn** 外壳程序:

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

对于 **C** 外壳程序:

```
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。

如果要使用 ODBC 或 DB2 CLI、Java 或 SQL 过程，则执行下列主题中的步骤:

- 设置 UNIX ODBC 环境
- 设置 UNIX Java 环境
- 设置 UNIX SQL 过程环境

### 相关概念:

- 第 23 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 24 页的『设置 UNIX Java 环境』
- 第 28 页的『设置 UNIX SQL 过程环境』

#### 相关参考:

- 第 9 页的『AIX 支持的用于构建和运行应用程序的软件』
- 第 10 页的『HP-UX 支持的用于构建和运行应用程序的软件』
- 第 11 页的『Linux 支持的用于构建和运行应用程序的软件』
- 第 13 页的『Solaris 支持的用于构建和运行应用程序的软件』

## UNIX 环境变量设置

根据所在的 UNIX® 平台，在 db2profile（对于 bash 或 korn 外壳程序）或 db2cshrc（对于 C 外壳程序）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 .profile（bash 或 korn 外壳程序）或 .login（C 外壳程序）文件中。

#### AIX:

- PATH，将包括几个 DB2® 目录（sqlllib/bin 也在内）
- LIBPATH，包括目录 sqlllib/lib（参见下面的注释）

#### 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（参见下面的注释）

#### 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（参见下面的注释）

**注：**如果正在 64 位 DB2 实例中运行本地 32 位应用程序，则 LIBPATH、SHLIB\_PATH 和 LD\_LIBRARY\_PATH 应包含 sqlllib/lib32 而不是 sqlllib/lib。

sqlllib/userprofile 和 sqlllib/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。这些用户脚本只适用于从 DB2 版本 7 开始的版本。

#### 相关任务:

- 第 22 页的『设置 UNIX 应用程序开发环境』

## Java

### 设置 UNIX Java 环境

要在具有 DB2 JDBC 支持的 UNIX 上运行 JDBC 和 SQLJ 程序，应将更新 Java 环境的命令包括在数据库管理器文件 db2profile 和 db2cshrc 中。当创建 DB2 实例时，修改 .bashrc、.profile 和 / 或 .cshrc 以便:

1. 将 THREADS\_FLAG 设置为“native”。（仅限于 HP-UX、Linux 和 Solaris）
2. CLASSPATH 包括:
  - “.”（当前目录）
  - 文件 sqllib/java/db2java.zip
  - 文件 sqllib/java/db2jcc.jar

要构建 SQLJ 程序，还须更新 CLASSPATH 以包括文件:

```
sqllib/java/sqlj.zip
```

要运行 SQLJ 程序，还须更新 CLASSPATH 以包括文件:

```
sqllib/java/runtime.zip
```

#### 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安装有 JDK 的路径。为此，可在服务器命令行上输入如下命令:

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

其中 /home/db2inst/jdk13 是安装有 JDK 的路径。

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 字段出现在输出开始部分附近。

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 25 页的『设置 AIX Java 环境』
- 第 26 页的『设置 HP-UX Java 环境』
- 第 26 页的『设置 Linux Java 环境』
- 第 27 页的『设置 Solaris Java 环境』
- 第 19 页的『更新数据库管理器配置文件』

#### 相关参考：

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

### 设置 AIX Java 环境

#### 过程：

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

1. IBM 的“Java 开发工具箱（JDK）版本 1.3.1 和 Java 运行时环境（JRE）版本 1.3.1 AIX 版”。
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 101 页的『Java 样本程序』
- 第 103 页的『Java applet 注意事项』

#### 相关任务：

- 第 39 页的『设置样本数据库』

#### 相关参考：

- 第 9 页的『AIX 支持的用于构建和运行应用程序的软件』

## 设置 HP-UX Java 环境

### 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. Hewlett-Packard 公司的 Software Developer's Kit 和 Runtime Environment 1.3.1 for HP-UX 11.0 和 11i PA-RISC。
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。

要运行 Java 例程（存储过程和用户定义函数），应确保共享库路径类似于以下内容:

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

其中 \$JAVADIR 通常设置为 /opt/java1.3（Java SDK 的缺省位置）。

注: DB2 不支持例程的“热点 Java 虚拟机”。

### 相关概念:

- 第 101 页的『Java 样本程序』
- 第 103 页的『Java applet 注意事项』

### 相关任务:

- 第 39 页的『设置样本数据库』

### 相关参考:

- 第 10 页的『HP-UX 支持的用于构建和运行应用程序的软件』

## 设置 Linux Java 环境

### 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:
  - 对于 Intel 上的 Linux: “IBM 开发者工具箱和运行时环境 Linux 版 Java 2 技术版版本 1.3.1 的 32 位版本”

- 对于 Linux/390: IBM zSeries 开发者工具箱 Linux 版 Java 2 技术版
  - 对于 IA64 上的 Linux: “IBM 开发者工具箱和运行时环境 Linux 版 Java 2 技术版版本 1.3.1 的 64 位版本”
2. 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 必须能够装入这些库和 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 `/lib` 或 `/usr/lib` 中的从属库。

可以将 Java 共享库的位置添加到 `/etc/ld.so.conf`，但是我们建议在 `/usr/lib` 中创建指向这些库的符号链接。对于 IBM JDK 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 JDK 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。

如果您决定将 Java 共享库的位置添加到 `/etc/ld.so.conf`，必须以 `root` 用户身份运行以下命令来刷新运行时链接程序高速缓存。

```
bash# ldconfig
```

#### 相关概念:

- 第 101 页的『Java 样本程序』
- 第 103 页的『Java applet 注意事项』

#### 相关任务:

- 第 39 页的『设置样本数据库』

#### 相关参考:

- 第 11 页的『Linux 支持的用于构建和运行应用程序的软件』

## 设置 Solaris Java 环境

### 过程:

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

1. Sun 公司的 Java 开发工具箱（JDK）版本 1.3.1 Solaris 版。
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 101 页的『Java 样本程序』
- 第 103 页的『Java applet 注意事项』

**相关任务：**

- 第 39 页的『设置样本数据库』

**相关参考：**

- 第 13 页的『Solaris 支持的用于构建和运行应用程序的软件』

## SQL 过程

### 设置 UNIX SQL 过程环境

SQL 过程需要使用编译器配置设置服务器上的两个环境变量：  
DB2\_SQLROUTINE\_COMPILER\_PATH 和  
DB2\_SQLROUTINE\_COMPILE\_COMMAND。

**限制：**

实例所有者必须属于受防护标识的主组。

**过程：**

第一次编译存储过程时，DB2 将会生成可执行脚本文件 \$HOME/sqlllib/function/routine/sr\_cpath（它包含这些编译器环境变量的缺省值）。如果缺省值不适用于您的编译器，则执行下列操作之一：

- 编辑 sr\_cpath 脚本文件。
- 将 DB2\_SQLROUTINE\_COMPILER\_PATH DB2 注册表变量设置为包含另一个可执行脚本（它指定了期望的设置）的全路径名。

“应用程序开发客户机”的安装提供了一个缺省的编译命令，该命令为各服务器平台支持的至少一个编译器工作。

**AIX** IBM C AIX 版的版本 5.0



## HP-UX

HP aC++ 版本 A.03.31

**Linux** GNU/Linux g++

## Solaris

Forte C++ 版本 5.0、6 和 6.1

要使用其它编译器或者定制缺省命令，必须按如下所示设置 DB2\_SQLROUTINE\_COMPILE\_COMMAND DB2 注册表变量：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=<compile_command>
```

其中 <compile\_command> 是 C 或 C++ 编译命令，包括创建例程所需的选项和参数。要使设置生效，必须执行 db2start 命令。

在编译命令中，使用关键字 SQLROUTINE\_FILENAME 来替换生成的 SQC、C、PDB、DEF、EXP、消息日志和共享库文件的文件名。仅对 AIX 使用关键字 SQLROUTINE\_ENTRY 来替换入口点名称。可以在为“IBM C AIX 版的版本 5.0”设置 DB2\_SQLROUTINE\_COMPILE\_COMMAND 缺省值的命令中看到使用这两个关键字的示例：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=xlc -I$HOME/sqllib/include \  
SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp -e SQLROUTINE_ENTRY \  
-o SQLROUTINE_FILENAME -L$HOME/sqllib/lib -ldb2
```

### 相关任务:

- 第 131 页的『保留 SQL 过程的中间文件』
- 第 132 页的『为 SQL 过程定制预编译和绑定选项』
- 第 133 页的『备份和复原 SQL 过程』
- 第 127 页的『创建 SQL 过程』
- 第 137 页的『使用 CALL 语句调用存储过程』
- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』

### 相关参考:

- 第 30 页的『UNIX 缺省 DB2 SQLROUTINE 编译命令值』
- 第 9 页的『AIX 支持的用于构建和运行应用程序的软件』
- 第 10 页的『HP-UX 支持的用于构建和运行应用程序的软件』
- 第 11 页的『Linux 支持的用于构建和运行应用程序的软件』
- 第 13 页的『Solaris 支持的用于构建和运行应用程序的软件』

## UNIX 缺省 DB2 SQLROUTINE 编译命令值

以下是为受支持的 UNIX 服务器平台上的 C 或 C++ 编译器设置 DB2\_SQLROUTINE\_COMPILE\_COMMAND 的缺省值的命令:

**AIX** 要使用 “IBM C AIX 版的版本 5.0” :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=xlc -I$HOME/sqlllib/include \
SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp \
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sqlllib/lib -ldb2
```

要使用 IBM VisualAge C++ 5.0:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=xlc -I$HOME/sqlllib/include \
SQLROUTINE_FILENAME.C -bE:SQLROUTINE_FILENAME.exp \
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sqlllib/lib -ldb2
```

如果 DB2 注册表变量 DB2\_SQLROUTINE\_COMPILE\_COMMAND 未设置, 这将是缺省编译命令。

**注:** 要在 AIX 上编译 64 位 SQL 过程, 可将 -q64 选项添加到以上命令中。

要使用 “IBM VisualAge C++ AIX 版的版本 5” 增量编译器:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

在 vacbld 命令之后如果没有指定配置文件, DB2 将会在第一次尝试创建 SQL 过程时创建以下缺省配置文件:

```
$HOME/sqlllib/function/routine/sqlproc.icc
```

为 DB2\_SQLROUTINE\_COMPILE\_COMMAND 设置 DB2 注册表值时, 可以指定自己的配置文件:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld \
$HOME/sqlllib/function/sqlproc.icc"
```

## HP-UX

要使用 “HP C 编译器版本 B.11.11.02” :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc +DAportable +ul -Aa +z \
-I$HOME/sqlllib/include -c SQLROUTINE_FILENAME.c; \
ld -b -o SQLROUTINE_FILENAME SQLROUTINE_FILENAME.o \
-L$HOME/sqlllib/lib -ldb2
```

要使用 “HP aC++ 版本 A.03.31” :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=aCC +DAportable +ul +z -ext \
-I$HOME/sqlllib/include SQLROUTINE_FILENAME.C -b \
-o SQLROUTINE_FILENAME -L$HOME/sqlllib/lib -ldb2
```

如果 DB2 注册表变量 DB2\_SQLROUTINE\_COMPILE\_COMMAND 未设置, 这将是缺省编译命令。

注: 要在 HP-UX 上编译 64 位 SQL 过程, 则去掉 +DAportable 选项并将 +DA2.0W 选项添加到上述命令中。

**Linux** 要使用 GNU/Linux gcc:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -fpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

要使用 GNU/Linux g++:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=g++ -fpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.C \  
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

如果 DB2 注册表变量 DB2\_SQLROUTINE\_COMPILE\_COMMAND 未设置, 这将是缺省编译命令。

## Solaris

要使用 “Forte C 版本 5.0” :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -xarch=v8plusa -Kpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib \  
-R$HOME/sql1lib/lib -ldb2
```

要使用 “Forte C++ 版本 5.0” :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=CC -xarch=v8plusa -Kpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.C \  
-G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib \  
-R$HOME/sql1lib/lib -ldb2
```

如果 DB2 注册表变量 DB2\_SQLROUTINE\_COMPILE\_COMMAND 未设置, 这将是缺省编译命令。

注:

1. 编译器选项 -xarch=v8plusa 已添加到缺省的编译器命令以避免这样一个问题: 链接 libdb2.so 时无法产生有效的可执行文件。
2. 要编译 Solaris 上的 64 位 SQL 过程, 应将 -xarch=v8plusa 选项去掉并添加 -xarch=v9 到上述命令。

**相关任务:**

- 第 28 页的『设置 UNIX SQL 过程环境』

---

## Windows

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

## 设置 Windows 应用程序开发环境

当在 Windows XP、Windows .Net Server、Windows NT 或 Windows 2000 上安装“DB2 AD 客户机”时，安装程序将用环境变量 INCLUDE、LIB、PATH、DB2PATH 和 DB2INSTANCE 更新配置注册表。缺省实例为 DB2。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 只应在用户环境级别定义。如果使用 DB2INSTDEF 注册表变量（它定义未设置 DB2INSTANCE 时使用的缺省实例名），则不需要它。

### 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 .NET Server 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。

例如，Microsoft Visual C++ 编译器环境设置文件 Vc\bin\vcvars32.bat 具有以下命令:

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

要让 DB2 使用此文件，首先按如下所示将 %INCLUDE%（它设置 %DB2PATH%\INCLUDE 路径）从该列表的尾部移至列表的开头:

```
set INCLUDE=%INCLUDE%;%MSVCDir%\INCLUDE;%MSVCDir%\...\ATL\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，必须确保 LIB 环境变量指向 %DB2PATH%\lib\Win64。这意味着必须更改其缺省设置。缺省情况下，环境变量指向 %DB2PATH%\lib，它针对的是在 32 位环境上开发 32 位应用程序或在 64 位环境上开发 64 位应用程序。

- 确保在远程数据库的服务器上已设置 DB2COMM 环境变量。
- 确保在服务器端已启动安全性服务，以便进行 SERVER 认证；而在客户端的安全性服务取决于 CLIENT 认证的级别。要手工启动安全性服务，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，您才会想要将安全性服务设置为自动启动。要自动启动安全性服务，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 .NET Server

1. 单击“开始”按钮。
2. 单击“设置”。
3. 单击“控制面板”。
4. 单击“管理工具”。
5. 单击“服务”。
6. 在“服务”窗口中，突出显示“DB2 安全服务器”。
7. 如果它未列示设置“已启动”和“自动”，则从顶部菜单单击“操作”。

8. 单击“特性”。
9. 确保您在“常规”选项卡中。
10. 从“启动类型”下拉菜单中选择“自动”。
11. 单击“确定”。
12. 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 .NET Server、Windows NT 或 Windows 2000 环境上的数据库管理器是作为服务实现的，因此，尽管可能发生了问题，启动服务时也不会返回错误或警告。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，用户应一直检查事件日志或“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、Java 或 SQL 过程，则继续进行适当的任务：

- 设置 Windows CLI 环境
- 设置 Windows Java 环境
- 设置 Windows SQL 过程环境

### 相关任务：

- 『Setting Up the Windows CLI Environment』（*CLI Guide and Reference, Volume 1*）

- 第 35 页的『设置 Windows Java 环境』
- 第 37 页的『设置 Windows SQL 过程环境』

#### 相关参考:

- 第 14 页的『Windows 支持的用于构建和运行应用程序的软件』

## Java

### 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。

#### 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. 下列其中一项:
  - “IBM 开发者工具箱和运行时环境 Windows 版 Java 2 技术版版本 1.3.1”
  - Sun 公司的 Java 开发工具箱 (JDK) 1.3.1 Win32 版
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安装有 JDK 的路径。为此，可在服务器命令行上输入如下命令:

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

其中 `c:\jdk13` 是安装有 JDK 的路径。

如果安装有 JDK 的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 开发工具箱”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 JDK。

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 36 页的『Windows Java 环境设置』
- 第 101 页的『Java 样本程序』
- 第 103 页的『Java applet 注意事项』

#### 相关任务:

- 第 39 页的『设置样本数据库』

#### 相关参考:

- 第 14 页的『Windows 支持的用于构建和运行应用程序的软件』

### Windows Java 环境设置

要在具有 DB2® JDBC 支持的受支持 Windows® 平台上运行 JDBC 和 SQLJ 程序，安装 DB2 时将会自动更新 CLASSPATH 以包括：

- “.”（当前目录）
- sqllib\java\db2java.zip 文件
- sqllib\java\db2jcc.jar 文件

要构建 SQLJ 程序，还须更新 CLASSPATH 以包括文件：

```
sqllib\java\sqlj.zip
```

要运行 SQLJ 程序，还须更新 CLASSPATH 以包括文件：

```
sqllib\java\runtime.zip
```



注: “DB2 版本 8” 不支持 Microsoft Software Developer’s Kit for Java。它既不能用于 SQLJ 定制也不能用于类型 2 JDBC 应用程序。

相关任务:

- 第 35 页的『设置 Windows Java 环境』

## SQL 过程

### 设置 Windows SQL 过程环境

下列 Windows 操作系统将支持 SQL 过程: Windows NT、Windows 2000、Windows XP 和 Windows .NET Server。

“SQL 过程” 需要使用服务器上的编译器配置设置两个环境变量: DB2\_SQLROUTINE\_COMPILER\_PATH 和 DB2\_SQLROUTINE\_COMPILE\_COMMAND。如果编译器的环境变量设置为 SYSTEM 变量, 则不需要任何配置。

限制:

当尝试设置变量 DB2\_SQLROUTINE\_COMPILER\_PATH 时, 64 位 Windows 存在问题, 因为它需要文件的完整路径且不会允许自变量 / 开关。所以, 如果用户指定了下列内容, 该变量就不起作用:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="C:\MsSdk64\SetEnv.bat /XP64 /RETAIL"
```

解决办法是创建另一批处理文件, 它以适当的标志调用 Microsoft 的设置批处理文件, 例如:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="C:\MsSdk64\SetEnvXP64.bat"
```

其中 C:\MsSdk64\SetEnvXP64.bat 的内容将为:

```
call C:\MsSdk64\SetEnv.bat /XP64 /RETAIL
```

此问题不会发生在 Windows 32 位环境上, 因为 vcvars32.bat 不需要也不接受任何参数。

过程:

假定 C++ 编译器安装在 C: 驱动器上, 按如下所示设置 DB2\_SQLROUTINE\_COMPILER\_PATH DB2 注册表变量:

对于 Microsoft Visual C++ 版本 5.0:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\devstudio\vc\bin\vcvars32.bat"
```

对于 Microsoft Visual C++ 版本 6.0:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\Micros~1\vc98\bin\vcvars32.bat"
```

如果需要的话, 请更改驱动器或者路径以反映 C++ 编译器在您的系统上的位置。

对于编译命令, 使用关键字 `SQLROUTINE_FILENAME` 来替换生成的 `SQC`、`C`、`PDB`、`DEF`、`EXP`、消息日志和共享库文件的文件名。要使用 Microsoft Visual C++ 版本 5.0 和 6.0:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="cl -Od -W2 /TC -D_X86_=1  
-I%DB2PATH%\include SQLROUTINE_FILENAME.c /link -d11  
-def:SQLROUTINE_FILENAME.def /out:SQLROUTINE_FILENAME.d11  
%DB2PATH%\lib\db2api.lib"
```

如果 `DB2` 注册表变量 `DB2_SQLROUTINE_COMPILE_COMMAND` 未设置, 这将是缺省编译命令。

以下是“Microsoft Visual C++ 版本 6.0”编译器的环境设置命令的一个示例。可以通过剪切并将下列命令粘贴到批处理文件中并在 `DB2` 命令窗口中运行该文件来执行下列命令。请确定已进行了所有必需的更改, 包括对于您特定环境的路径设置。

```
@echo on  
rem Setting the SQL PROCEDURE environment:  
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\Micros~1\vc98\bin\vcvars32.bat"  
db2set DB2_SQLROUTINE_COMPILE_COMMAND="cl -Od -W2 /TC -D_X86_=1  
-I%DB2PATH%\include SQLROUTINE_FILENAME.c /link -d11  
-def:SQLROUTINE_FILENAME.def /out:SQLROUTINE_FILENAME.d11  
%DB2PATH%\lib\db2api.lib"  
@echo off
```

### 相关任务:

- 第 131 页的『保留 SQL 过程的中间文件』
- 第 132 页的『为 SQL 过程定制预编译和绑定选项』
- 第 133 页的『备份和复原 SQL 过程』
- 第 127 页的『创建 SQL 过程』
- 第 137 页的『使用 `CALL` 语句调用存储过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』

### 相关参考:

- 第 14 页的『Windows 支持的用于构建和运行应用程序的软件』

---

## 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样本数据库。如果要使用远程客户机存取服务器上的样本数据库，则需要在客户机工作站上对样本数据库编目。

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样本数据库，则需要将数据库实用程序（包括 DB2 CLI 实用程序文件）绑定至样本数据库。

#### 过程:

以下是设置样本数据库的步骤:

1. 创建样本数据库
2. 编目样本数据库
3. 绑定样本数据库实用程序

#### 相关任务:

- 第 39 页的『创建样本数据库』
- 第 42 页的『编目样本数据库』
- 第 42 页的『绑定样本数据库实用程序』

### 创建样本数据库

使用 `db2samp1` 命令在命令行上创建样本数据库。

#### 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 `db2samp1`（创建样本数据库的程序）的位置。`db2profile` 或 `db2cshrc` 文件将把 `db2samp1` 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, db2samp1 位于:  
\$HOME/sql1lib/bin

其中, \$HOME 是 DB2 实例所有者的主目录。

- 在 Windows 上, db2samp1 位于:  
%DB2PATH%\bin

其中, %DB2PATH% 是安装有 DB2 的路径。

2. 确保将 DB2INSTANCE 环境变量设置为要创建样本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 外壳程序或 Korn 外壳程序, 则可输入以下命令进行设置:

```
DB2INSTANCE=instance_name
export DB2INSTANCE
```

如果使用 C 外壳程序, 则可输入以下命令进行设置:

```
setenv DB2INSTANCE instance_name
```

- 在 Windows 上, 输入:

```
set DB2INSTANCE=instance_name
```

其中 *instance\_name* 是数据库实例的名称。

3. 输入 db2samp1, 后跟您想创建的样本数据库的位置, 以便创建样本数据库。在 UNIX 平台上, 这是一个 *path* (即路径, 如 “\$HOME” ), 且输入形式如下:

```
db2samp1 path
```

例如:

```
db2samp1 $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 41 页的『在 “主机” 或 AS/400 和 iSeries 服务器上创建样本数据库』

- 第 42 页的『编目样本数据库』
- 第 42 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

如果要对“主机”服务器（如 DB2 UDB z/OS 版和 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 版和 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样本数据库。
2. 连接样本数据库。
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 42 页的『编目样本数据库』
- 第 42 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.sqlC -- How to move table data (C++)』

## 编目样本数据库

要从远程客户机存取服务器上的样本数据库，需要在客户机工作站上对样本数据库编目。

不需要在服务器工作站上编目样本数据库，因为创建它时就已在服务器上编目了它。

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编目样本数据库，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，还必须编目远程节点。

### 相关任务:

- 『编目客户机上的 TCP/IP 节点』（《安装和配置补遗》）
- 『使用 CLP 编目数据库』（《安装和配置补遗》）
- 『在 DB2 客户机上编目 NetBIOS 节点』（《安装和配置补遗》）
- 『在客户机上编目“命名管道”节点』（《安装和配置补遗》）
- 『在 DB2 客户机上编目 APPC 节点』（《安装和配置补遗》）
- 第 42 页的『绑定样本数据库实用程序』

## 绑定样本数据库实用程序

如果要从正在运行另一 DB2 版本或正在另一操作系统上运行的远程客户机存取服务器上的样本数据库，则需要将数据库实用程序（包括 DB2 CLI 实用程序）绑定至样本数据库。

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指定 BIND 命令来显式地完成绑定。

### 过程:

视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，连接样本数据库:

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

其中 *userid* 和 *password* 是样本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 菜单选择 DB2 命令窗口。

将显示该命令窗口。

4. 输入如下命令，连接样本数据库:

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

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

5. 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% 是安装有 DB2 的路径。

6. 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 版和 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 客户机存取 DB2 z/OS 版和 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 客户机存取 DB2 z/OS 版和 OS/390 版服务器, 则输入:  

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

相关参考:

- 『 BIND Command 』 ( *Command Reference* )

---

## 迁移应用程序

“DB2 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> 版本 2.1.x

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, 必须首先迁移至支持此迁移的上述受支持版本之一, 然后从该版本迁移至 DB2 版本 8。

从下列特定环境进行迁移时应注意:

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



1. 类型 4 驱动程序压缩文档为 db2jcc.jar。在与 applet 相关联的 .html 文件中，将压缩文档从 db2java.zip 更改为 db2jcc.jar。将 db2jcc.jar 复制至 Web 服务器。
2. 类型 4 驱动程序类名称为 com.ibm.db2.jcc.DB2Driver。在 applet.java 文件中，将类型 3 驱动程序类名称 COM.ibm.db2.jdbc.net.DB2Driver 更改为类型 4 类名称。如果 applet 使用 javax.sql.DataSource 来获取连接，则可能没有对驱动程序类的引用。
3. 类型 3 和类型 4 驱动程序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 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。  
 类型 4 驱动程序客户机直接连接至 DB2 服务器，所以 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，则用类型 4 驱动程序的新 URL 更新 .java 文件和（如果必要的话）.html 文件。
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。

**注：**类型 4 驱动程序不支持连接池（javax.sql.ConnectionPoolDataSource 和 javax.sql.PooledConnection）的“JDBC 2.0 可选程序包 API”。如果 applet 使用这些类型 3 功能部件，则它不能迁移至类型 4 驱动程序。

## 32 位至 64 位环境

可能不必将应用程序从 32 位迁移至 64 位。“DB2 版本 8”支持在所有 64 位 DB2 实例（除了 Linux IA64 之外）中运行现有 32 位本地应用程序。为此，用户必须重新绑定 32 位应用程序，然后以适当的库路径运行它。例如，要在 AIX 上的 64 位 DB2 实例中运行 32 位应用程序：

- 在 bash 或 Korn 外壳程序上：
 

```
export LIBPATH=$HOME/sqlllib/lib32
my32bitapp
```
- 在 C 外壳程序上：

```
setenv LIBPATH $HOME/sql1lib/lib32
my32bitapp
```

如果想要将 32 位应用程序迁移至 64 位操作环境而同时又仍在 32 位服务器上运行，则使用 **LONGERROR** 预编译选项来准备移植应用程序。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，否则修剪主机变量的 **long** 类型的使用。或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。

## HP-UX

如果您要将 DB2 从 HP-UX 版本 10 或更早版本迁移至 HP-UX 版本 11，则您的 DB2 程序必须用 HP-UX 版本 11 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、用户定义函数和用户出口程序。另外，在 HP-UX 版本 11 上编译的 DB2 程序可能无法在 HP-UX 版本 10 或更早版本上运行。在 HP-UX 版本 10 上编译和运行的 DB2 程序可以与 HP-UX 版本 11 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® 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。

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：

- 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，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，则对应用程序中指定该 DB2 版本的路径进行编码。例如，如果 AIX® 应用程序需要“DB2 版本 5”，则编码 `/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 `sqllib/lib`；在 Windows 操作系统上，该目录为 `sqllib\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 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 `char` 类型，而不是其它变体（如无符号 `char`）。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 6.1、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。

在 UNIX 平台上，如果您具有来自先前 DB2 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 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）中的下列系统命令之一：

**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	OFFidSTR	LENstrTBL	OFFstrTBL
0x00000003	0x000000284	0x0000007f	0x0000002cb
***Import File Strings***			
INDEX	PATH	BASE	MEMBER
0	/home/db2inst/sqllib/lib:/usr/lib:/lib		
1		libc.a	shr.o
2		libdb2.a	shr.o
-----			

第 0（零）行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。

视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？

当安装 DB2 版本 1、2、5、6.1 或者 7 中之一时，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？

可使用环境变量 `LIBPATH`（在 AIX 上）、`SHLIB_PATH`（在 HP-UX 上）和 `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。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/sqllib/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/sqllib/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/sqllib/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/sqllib/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:

- 『迁移建议』（《DB2 服务器快速入门》）

- 『JDBC 2.1 Core API Restrictions by the DB2 JDBC Type 4 Driver』 (*Application Development Guide: Programming Client Applications*)
- 『JDBC 2.1 Optional Package API Support by the DB2 JDBC Type 4 Driver』 (*Application Development Guide: Programming Client Applications*)

#### 相关任务:

- 第 17 页的『设置应用程序开发环境』
- 『创建 DB2 文件的链接』 (《安装和配置补遗》)
- 『迁移数据库』 (《DB2 服务器快速入门》)
- 『迁移说明表』 (《DB2 服务器快速入门》)
- 『迁移实例 (UNIX)』 (《DB2 服务器快速入门》)
- 『使 V6 或 V7 DB2 服务器脱机以进行 DB2 迁移』 (《DB2 服务器快速入门》)
- 『迁移 DB2 (Windows)』 (《DB2 服务器快速入门》)
- 『迁移 DB2 (UNIX)』 (《DB2 服务器快速入门》)

#### 相关参考:

- 『迁移限制』 (《DB2 服务器快速入门》)
- 『发行版之间的版本 8 不兼容性』 (《管理指南: 计划》)
- 『发行版之间的版本 7 不兼容性』 (《管理指南: 计划》)

---

## 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 . . . . .	53	Visual C++ 样本 . . . . .	83
样本程序: 结构和设计. . . . .	58	Windows Management Instrumentation 样本	83
按语言 and 应用程序接口分类的样本程序 . . . . .	65	“对象链接与嵌入” (OLE) 样本 . . . . .	84
C/C++ 样本 . . . . .	65	对象链接与嵌入数据库 (OLE DB) 表函数	
DB2 CLI 样本 . . . . .	68	样本. . . . .	85
JDBC 样本 . . . . .	70	命令行处理器 (CLP) 样本 . . . . .	85
SQLJ 样本 . . . . .	72	REXX 样本 . . . . .	86
Java WebSphere 样本 . . . . .	74	日志管理用户出口样本. . . . .	88
Java 插件样本 . . . . .	75	构建文件、Makefile 和错误检查实用程序 . . . . .	89
COBOL 样本. . . . .	75	构建文件 . . . . .	89
SQL 过程样本 . . . . .	80	Makefile . . . . .	92
Visual Basic 样本 . . . . .	82	错误检查实用程序 . . . . .	95

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® 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，以及用作了解 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，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 `sqlllib/samples (UNIX)` 和 `sqlllib\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编目存储过程和 UDF 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 `README` 文件。

提供了大多数样本程序文件的 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得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 README 文件。README 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。还可以存取所列示样本目录中的样本文件。对于目录路径，使用了 UNIX 式斜杠，如 samples/c（在目录只用于 Windows 的地方除外，如 samples\VB\ADO）。

表 1. 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
CLI samples/cli	README	README	README	README	README
JDBC samples/java	README	README	README	README	README
SQLJ samples/java	README	README	README	README	README
IBM COBOL samples/cobol	README	n/a	n/a	n/a	README
Micro Focus COBOL samples/cobol_mf	README	README	n/a	README	README
Visual Basic samples\VB\ADO	n/a	n/a	n/a	n/a	ReadMe.txt
SQL 过程 samples/sqlproc	README	README	README	README	README
CLP samples/clp	README	README	README	README	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 55 页的表 2.

按程序组分类的样本文件扩展名

表 3.

表 2. 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)
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\AD0 samples\VB\MTS samples\VB\RD0		.bas .frm .vbp
Visual C++	samples\VC\AD0		.cpp .dsp .dsw

表 3. 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	.cadsm (UNIX 和 Windows) .cdisk (UNIX 和 Windows) .ctape (UNIX) .cxbsa (UNIX 和 Windows)

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 操作环境和 Windows 32 位操作系统提供了此样本。

## Java™ 样本

即是“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

即是“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

即是“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。

可在安装了 DB2 的目录的 `samples` 子目录中找到这些样本程序。每种受支持的语言都有一些子目录。以下示例显示如何在每个受支持的平台上找到用 C 或 C++ 编写的样本程序。

- **在 UNIX 平台上。**

可在数据库实例目录下的 `sqllib/samples/c` 中找到嵌入式 SQL 和 DB2 API 程序的 C 源代码; DB2 CLI 程序的 C 源代码在 `sqllib/samples/cli` 中。有关样本表中那些程序的其它信息, 参考 DB2 实例下适当的 `samples` 子目录中的 README 文件。README 文件将包含本书中未列出的任何附加样本。

- **在 Windows 操作系统上。**

可以在 DB2 安装目录下的 `sqllib\samples\c` 中找到嵌入式 SQL 和 DB2 API 程序的 C 源代码; DB2 CLI 程序的 C 源代码在 `sqllib\samples\cli` 中。有关样本表中样本程序的其它信息, 参考适当 `sqllib\samples` 子目录中的 README 文件。README 文件将包含本书中未列出的任何附加样本。

样本程序目录在大多数平台上一般是只读的。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。

**相关概念:**

- 第 89 页的『构建文件』
- 第 92 页的『Makefile』
- 第 95 页的『错误检查实用程序』
- 第 58 页的『样本程序: 结构和设计』

**相关参考:**

- 第 65 页的『C/C++ 样本』
- 第 68 页的『DB2 CLI 样本』
- 第 70 页的『JDBC 样本』
- 第 72 页的『SQLJ 样本』
- 第 80 页的『SQL 过程样本』
- 第 82 页的『Visual Basic 样本』
- 第 83 页的『Visual C++ 样本』
- 第 84 页的『“对象链接与嵌入”(OLE)样本』
- 第 85 页的『对象链接与嵌入数据库(OLE DB)表函数样本』
- 第 85 页的『命令行处理器(CLP)样本』
- 第 88 页的『日志管理用户出口样本』
- 第 75 页的『COBOL 样本』

- 第 74 页的『Java WebSphere 样本』
- 第 75 页的『Java 插件样本』
- 第 83 页的『Windows Management Instrumentation 样本』
- 第 86 页的『REXX 样本』

---

## 样本程序：结构 and 设计

对用 C、CLI、C++、Java 和 Visual Basic ADO 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 Java，第一个字母为大写）。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：

<b>前缀</b>	DB2 级别
<b>il</b>	安装映象级别
<b>cl</b>	客户机级别
<b>in</b>	实例级别
<b>db</b>	数据库级别
<b>ts</b>	表空间级别
<b>tb</b>	表级别
<b>dt</b>	数据类型级别

各级别显示了一个层次结构。安装映象级别是 DB2 的最上面一层。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（从而保存有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：

<b>前缀</b>	编程方法
<b>fn</b>	SQL 函数
<b>sp</b>	存储过程
<b>ud</b>	用户定义函数

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。

下面是在此设计中实现的样本程序。

**注：**Java™ 程序名的第一个字符（有时还有其它字符）为大写，且在教程样本名中不存在下划线。Visual Basic 样本有某些字符（但不是第一个字符）为大写。

表 4. DB2 通用数据库中的样本程序设计

样本程序	描述
教程样本	
tut_mod	如何修改表数据。  受支持的语言 / API: C、C++、CLI 和 JDBC
tut_read	如何读取表。  受支持的语言 / API: C、C++、CLI 和 JDBC
tut_use	如何使用数据库。  受支持的语言 / API: C、C++ 和 CLI
安装映象级别	
ilinfo	如何获取 / 设置安装级别信息。  受支持的语言 / API: CLI 和 JDBC
客户机级别	
cliExeSQL	如何执行 SQL 语句。  受支持的语言 / API: Visual Basic/ADO
cli_info	如何获取 / 设置客户机级别信息。  受支持的语言 / API: C、C++、CLI 以及 Visual Basic/ADO
clihandl	如何分配和释放句柄。  受支持的语言 / API: CLI
clisqlca	如何使用 SQLCA 数据。  受支持的语言 / API: CLI
clisnap	如何捕获客户机级别快照。  受支持的语言 / API: C 和 C++
实例级别	

表 4. DB2 通用数据库中的样本程序设计 (续)

样本程序	描述
inattach	如何连接实例和从实例折离。  受支持的语言 / API: C 和 C++
inauth	如何显示实例级别的权限。  受支持的语言 / API: C 和 C++
ininfo	如何获取和设置实例级别信息。  受支持的语言 / API: C、C++ 和 CLI
insnap	如何捕获实例级别快照。  受支持的语言 / API: C 和 C++
instart	如何停止和启动当前本地实例。  受支持的语言 / API: C 和 C++
数据库级别	
dbauth	如何授予 / 显示 / 取消数据库级别的权限。  受支持的语言 / API: C、C++、JDBC 和 SQLJ
dbcfg	如何配置数据库和数据库管理器参数。  受支持的语言 / API: C 和 C++
dbconn	如何连接数据库和断开与数据库的连接。  受支持的语言 / API: C、C++、CLI、JDBC、SQLJ 和 Visual Basic/ADO
dbcreate	如何创建和删除数据库。  受支持的语言 / API: C 和 C++
dbinfo	如何获取和设置数据库级别的信息。  受支持的语言 / API: C、C++、CLI、JDBC 和 Visual Basic/ADO
dbinline	如何使用直接插入 SQL 过程语言。  受支持的语言 / API: C
dbCommit	如何动态控制数据库级别的自动落实。  受支持的语言 / API: Visual Basic/ADO
dbmcon	如何连接多个数据库和断开与多个数据库的连接。  受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ



表 4. DB2 通用数据库中的样本程序设计 (续)

样本程序	描述
dbmconx	如何连接多个数据库和断开与多个数据库的连接。  受支持的语言 / API: CLI
dbmigrat	如何迁移数据库。  受支持的语言 / API: C 和 C++
dbnative	如何转换具有 ODBC 转义子句的语句。  受支持的语言 / API: CLI 和 JDBC
dbpkg	如何使用程序包。  受支持的语言 / API: C 和 C++
dbrecov	如何恢复数据库。  受支持的语言 / API: C 和 C++
dbsample	如何创建包括“主机”和 AS/400® 表和视图的样本数据库。  受支持的语言 / API: C
DbSeq	如何创建、改变和删除数据库中的序列。  受支持的语言 / API: JDBC
dbsnap	如何捕获数据库级别快照。  受支持的语言 / API: C 和 C++
dbthrs	如何使用线程。  受支持的语言 / API: C 和 C++
dbuse	如何使用数据库对象。  受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
dbusemx	如何将数据库对象与嵌入式 SQL 配合使用。  受支持的语言 / API: CLI
表空间级别	
tscreate	如何创建 / 删除缓冲池和表空间。  受支持的语言 / API: C 和 C++
tsinfo	如何获取表空间级别的信息。  受支持的语言 / API: C 和 C++

表 4. DB2 通用数据库中的样本程序设计 (续)

样本程序	描述
表级别	
tbconstr	如何使用表约束。 受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
tbcreate	如何创建、改变和删除表。 受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
TbGenCol	如何使用生成的列。 受支持的语言 / API: JDBC
tbident	如何使用身份列。 受支持的语言 / API: C、JDBC 和 SQLJ
tbinfo	如何获取和设置表级别的信息。 受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
tbmod	如何修改表中的信息。 受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
tbmove	如何移动表数据。 受支持的语言 / API: C 和 C++
tbpriv	如何授予 / 显示 / 取消表级别特权。 受支持的语言 / API: C、C++、JDBC 和 SQLJ
tbread	如何读取表中的信息。 受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
tbreorg	如何重组表。 受支持的语言 / API: C 和 C++
tbsavept	如何使用外部保存点。 受支持的语言 / API: C
tbtemp	如何使用声明的临时表。 受支持的语言 / API: C 和 JDBC
tbtrig	如何对表使用触发器。 受支持的语言 / API: C、C++、JDBC 和 SQLJ

表 4. DB2 通用数据库中的样本程序设计 (续)

样本程序	描述
<b>数据类型级别</b>	
dtformat	如何使用装入和导入数据格式扩展名。  受支持的语言 / API: C
dtHier	如何检索分层数据。  受支持的语言 / API: Visual Basic/ADO
dtinfo	如何获取数据类型的信息。  受支持的语言 / API: CLI 和 JDBC
dtlob	如何读写 LOB 数据。  受支持的语言 / API: C、C++、CLI、JDBC 和 Visual Basic/ADO
dtstruct	如何创建、使用和删除结构化类型和类型表的层次结构。  受支持的语言 / API: C++
dtudt	如何创建 / 使用 / 删除用户定义单值类型。  受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
<b>DB2 函数</b>	
fnuse	如何使用 SQL 函数。  受支持的语言 / API: C 和 C++
<b>存储过程</b>	
spcall	如何调用存储过程。  受支持的语言 / API: CLI 和 Visual Basic/ADO
spclient	调用 spserver 中的存储过程的客户机应用程序。  受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
spserver	spclient 调用的存储过程例程。  受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
<b>用户定义函数</b>	
udfcli	调用 udfsrv 中的 UDF 的客户机应用程序。  受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ
udfsrv	udfcli 调用的用户定义函数库。  受支持的语言 / API: C、C++、CLI、JDBC 和 SQLJ

表 4. DB2 通用数据库中的样本程序设计 (续)

样本程序	描述
udfjcli	调用 udfjsrv 中的“参数样式”Java 用户定义函数的客户机应用程序。 受支持的语言 / API: JDBC 和 SQLJ
udfjsrv	udfjcli 调用的“参数样式”Java 用户定义函数库。 受支持的语言 / API: JDBC 和 SQLJ
udfsqlcl	调用 udfsqlsv 中的 Java 用户定义函数的客户机应用程序。 受支持的语言 / API: JDBC
udfsqlsv	udfsqlcl 调用的带有 SQL 语句的 Java 用户定义函数库。 受支持的语言 / API: JDBC
udfUse	如何创建和使用 UDT 和 UDF。 受支持的语言 / API: Visual Basic/ADO

相关概念:

- 第 89 页的『构建文件』
- 第 92 页的『Makefile』
- 第 95 页的『错误检查实用程序』
- 第 53 页的『样本文件』

相关参考:

- 第 65 页的『C/C++ 样本』
- 第 68 页的『DB2 CLI 样本』
- 第 70 页的『JDBC 样本』
- 第 72 页的『SQLJ 样本』
- 第 80 页的『SQL 过程样本』
- 第 82 页的『Visual Basic 样本』
- 第 83 页的『Visual C++ 样本』
- 第 84 页的『“对象链接与嵌入”（OLE）样本』
- 第 85 页的『对象链接与嵌入数据库（OLE DB）表函数样本』
- 第 85 页的『命令行处理器（CLP）样本』
- 第 88 页的『日志管理用户出口样本』
- 第 75 页的『COBOL 样本』

- 第 74 页的『Java WebSphere 样本』
- 第 75 页的『Java 插件样本』
- 第 83 页的『Windows Management Instrumentation 样本』
- 第 86 页的『REXX 样本』

按语言和应用程序接口分类的样本程序

C/C++ 样本

注: 文件扩展名:

- C** .c (非嵌入式 SQL)
- .sqc (嵌入式 SQL)
- C++** .C (非嵌入式 SQL)
- .sqC (嵌入式 SQL)

表 5. C/C++ 样本程序文件

样本程序名	程序描述
教程样本 — 演示基本数据库操作的程序。	
tut_mod.sqc 和 tut_mod.sqC	如何修改表数据。
tut_read.sqc 和 tut_read.sqC	如何读取表。
tut_use.sqc 和 tut_use.sqC	如何使用数据库。
客户机级别 — 涉及 DB2 客户机级别的样本。	
cli_info.c 和 cli_info.C	如何获取和设置客户机级别信息。
clisnap.c 和 clisnap.C	如何捕获客户机级别快照。
实例级别 — 涉及 DB2 实例级别的样本。	
inattach.c 和 inattach.C	如何连接实例和从实例折离。
inauth.sqc 和 inauth.sqC	如何显示实例级别的权限。
ininfo.c 和 ininfo.C	如何获取和设置实例级别信息。
insnap.c 和 insnap.C	如何捕获实例级别快照。

表 5. C/C++ 样本程序文件 (续)

样本程序名	程序描述
instart.c 和 instart.C	如何停止和启动当前本地实例。
数据库级别 — 涉及 DB2 数据库对象的样本。	
dbauth.sqc 和 dbauth.sqC	如何授予 / 显示 / 取消数据库级别的权限。
dbcfg.sqc 和 dbcfg.sqC	如何配置数据库和数据库管理器参数。
dbconn.sqc 和 dbconn.sqC	如何连接数据库和断开与数据库的连接。
dbcreate.c 和 dbcreate.C	如何创建和删除数据库。
dbinfo.c 和 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。
dbmcon.sqc 和 dbmcon.sqC	如何连接多个数据库及断开与多个数据库的连接。
dbmcon1.h 和 dbmcon1.h	dbmcon1.sqc 和 dbmcon1.sqC 的头文件
dbmcon1.sqc 和 dbmcon1.sqC	dbmcon.sqc 和 dbmcon.sqC 的支持文件。
dbmcon2.h 和 dbmcon2.h	dbmcon2.sqc 和 dbmcon2.sqC 的头文件
dbmcon2.sqc 和 dbmcon2.sqC	dbmcon.sqc 和 dbmcon.sqC 的支持文件。
dbmigrat.c 和 dbmigrat.C	如何迁移数据库。
dbpkg.sqc 和 dbpkg.sqC	如何使用程序包。
dbrecov.sqc 和 dbrecov.sqC	如何恢复数据库。
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 和 dbsnap.C	如何捕获数据库级别快照。
dbthrds.sqc 和 dbthrds.sqC	如何使用线程。
dbuse.sqc 和 dbuse.sqC	如何使用数据库对象。

表 5. C/C++ 样本程序文件 (续)

样本程序名	程序描述
表空间级别 — 涉及 DB2 表空间级别的样本。	
tscreate.sqc 和 tscreate.sqC	如何创建和删除缓冲池和表空间。
tsinfo.sqc 和 tsinfo.sqC	如何获取表空间级别的信息。
表级别 — 涉及 DB2 表对象的样本。	
tbconstr.sqc 和 tbconstr.sqC	如何使用表约束。
tbcreate.sqc 和 tbcreate.sqC	如何创建、改变和删除表。
tbident.sqc	如何使用身份列。
tbinfo.sqc 和 tbinfo.sqC	如何获取和设置表级别的信息。
tbmod.sqc 和 tbmod.sqC	如何修改表中的信息。
tbmove.sqc 和 tbmove.sqC	如何移动表数据。
tbpriv.sqc 和 tbpriv.sqC	如何授予 / 显示 / 取消表级别特权。
tbread.sqc 和 tbread.sqC	如何读取表中的信息。
tbreorg.sqc 和 tbreorg.sqC	如何重组表。
tbsavept.sqc	如何使用外部保存点。
tbtemp.sqc	如何使用声明的临时表。
tbtrig.sqc 和 tbtrig.sqC	如何对表使用触发器。
数据类型级别 — 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 和 dtlob.sqC	如何读写 LOB 数据。
dtstruct.sqC	如何创建、使用和删除结构化类型和类型表的层次结构。
dtudt.sqc 和 dtudt.sqC	如何创建、使用和删除用户定义单值类型。

**DB2 函数级别**

表 5. C/C++ 样本程序文件 (续)

样本程序名	程序描述
fnuse.sqc 和 fnuse.sqC	如何使用 SQL 函数。
存储过程级别 — 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 和 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 和 spserver.sqC	在服务器上构建和运行的存储过程。
UDF 级别 — 演示用户定义函数的样本。	
udfcli.sqc 和 udfcli.sqC	调用 udfsrv.c 和 udfsrv.C 中的用户定义函数的客户机应用程序。
udfsrv.c 和 udfsrv.C	udfcli.sqc 和 udfcli.sqC 调用的用户定义函数 ScalarUDF。
udfemcli.sqc 和 udfemcli.sqC	调用嵌入式 SQL 用户定义函数库 udfemsrv 的客户机应用程序。
udfemsrv.sqc 和 udfemsrv.sqC	udfemcli 调用的嵌入式 SQL 用户定义函数库。
其它	
utilsnap.c 和 utilsnap.C	快照监视器样本的实用程序。

- 相关概念:
- 第 53 页的『样本文件』
  - 第 58 页的『样本程序: 结构和设计』

DB2 CLI 样本

表 6. 样本 CLI 程序文件

样本程序名	程序描述
教程样本 — 演示基本数据库操作的程序。	
tut_mod.c	如何修改表数据。
tut_read.c	如何读取表。
tut_use.c	如何使用数据库。



表 6. 样本 CLI 程序文件 (续)

样本程序名	程序描述
安装映象级别 — 涉及 DB2 和 CLI 安装映象级别的样本。	
ilinfo.c	如何获取和设置安装级别信息（如 CLI 驱动程序的版本）。
客户机级别 — 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。
clihandl.c	如何分配和释放句柄。
clisqlca.c	如何使用 SQLCA 数据。
实例级别 — 涉及 DB2 实例级别的样本。	
ininfo.c	如何获取和设置实例级别信息。
数据库级别 — 涉及 DB2 数据库对象的样本。	
dbcongui.c	如何借助“图形用户界面”（GUI）来连接数据库。
dbconn.c	如何连接数据库和断开连接。
dbinfo.c	如何获取和设置数据库级别的信息。
dbmcon.c	如何连接多个数据库及断开与多个数据库的连接。
dbmconx.c	如何使用嵌入式 SQL 连接多个数据库及断开与多个数据库的连接。
dbmconx1.h	dbmconx1.sqc 的头文件。
dbmconx1.sqc	dbmconx 程序的嵌入式 SQL 文件。
dbmconx2.h	dbmconx2.sqc 的头文件。
dbmconx2.sqc	dbmconx 程序的嵌入式 SQL 文件。
dbnative.c	如何将包含 ODBC 转义子句的语句转换为数据源特定格式。
dbuse.c	如何使用数据库对象。
dbusemx.sqc	如何将数据库对象与嵌入式 SQL 配合使用。
表级别 — 涉及 DB2 表对象的样本。	
tbconstr.c	如何使用表约束。
tbcreate.c	如何创建、改变和删除表。
tbinfo.c	如何获取和设置表级别的信息。
tbmod.c	如何修改表中的信息。
tbread.c	如何读取表中的信息。
数据类型级别 — 涉及数据类型的样本。	
dtinfo.c	如何获取数据类型的信息。
dtlob.c	如何读写 LOB 数据。
dtudt.c	如何创建、使用和删除用户定义单值类型。
存储过程级别 — 演示存储过程的样本。	

表 6. 样本 CLI 程序文件 (续)

样本程序名	程序描述
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。
spcliress.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> — 演示用户定义函数的样本。	
udfcli.c	调用 udfsrv.c 中用户定义函数的客户机应用程序。
udfsrv.c	udfcli.c 调用的用户定义函数 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。

相关概念:

- 第 53 页的『样本文件』
- 第 58 页的『样本程序: 结构和设计』

JDBC 样本

表 7. 样本 JDBC 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>安装映象级别</b> — 涉及 DB2 安装映象级别的样本。	
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 / 取消数据库级别的权限。
DbConn.java	如何连接数据库和断开与数据库的连接。
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及断开与多个数据库的连接。

表 7. 样本 JDBC 程序文件 (续)

样本程序名	程序描述
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。
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。
表级别 — 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	如何使用生成的列。
TbIdent.java	如何使用“身份列”。
TbInfo.java	如何获取和设置表级别的信息。
TbMod.java	如何修改表中的信息。
TbPriv.java	如何授予 / 显示 / 取消表级别特权。
TbRead.java	如何读取表中的信息。
TbTemp.java	如何使用“声明的临时表”。
TbTrig.java	如何对表使用触发器。
数据类型级别 — 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单值类型。
applet — 演示 applet 的样本。	
Applt.java	如何使用 applet。
存储过程 — 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.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。
UDF — 演示用户定义函数的样本。	
UDFcli.java	调用用户定义函数库 UDFsrv 的客户机应用程序。
UDFsrv.java	UDFcli.java 调用的用户定义函数。
udfcats	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。

表 7. 样本 JDBC 程序文件 (续)

样本程序名	程序描述
UDFjcli.java	调用用户定义函数库 UDFjsrv 的客户机应用程序。
UDFjsrv.java	UDFjcli.java 调用的用户定义函数。
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。

相关概念:

- 第 101 页的『Java 样本程序』
- 第 53 页的『样本文件』
- 第 58 页的『样本程序: 结构和设计』

相关参考:

- 第 72 页的『SQLJ 样本』
- 第 74 页的『Java WebSphere 样本』
- 第 75 页的『Java 插件样本』

SQLJ 样本

表 8. 样本 SQLJ 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 / 取消数据库级别的权限。
DbConn.sqlj	如何连接数据库和断开连接。
DbMCon.java	如何连接多个数据库及断开与多个数据库的连接。

表 8. 样本 *SQLJ* 程序文件 (续)

样本程序名	程序描述
DbMCon1.sqlj	DbMCon.java 的支持文件。
DbMCon2.sqlj	DbMCon.java 的支持文件。
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	如何使用身份列。
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 / 取消表级别特权。
TbRead.sqlj	如何读取表中的信息。
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单值类型。
<b>applet 级别</b> — 演示 applet 的样本。	
Applt.sqlj	如何使用 applet。
<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.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> — 演示用户定义函数的样本。	
UDFcli.sqlj	调用用户定义函数库 UDFsrv 的客户机应用程序。
UDFsrv.java	UDFcli 调用的用户定义函数。
udfcats	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.sqlj	调用用户定义函数库 UDFjsrv 的客户机应用程序。
UDFjsrv.java	UDFjcli 调用的用户定义函数。
udfjcats	UDFjsrv 程序的 UDF 目录脚本。此脚本调用 UDFjDrop.db2 和 UDFjCreate.db2。

表 8. 样本 SQLJ 程序文件 (续)

样本程序名	程序描述
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。

相关概念:

- 第 101 页的『Java 样本程序』
- 第 53 页的『样本文件』
- 第 58 页的『样本程序: 结构和设计』

相关参考:

- 第 70 页的『JDBC 样本』
- 第 74 页的『Java WebSphere 样本』
- 第 75 页的『Java 插件样本』

Java WebSphere 样本

表 9. 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）交互作用来存取存储在 DB2 中的数据。AccessEmployee.ear 文件位于 samples/java/WebSphere 目录中。

相关概念:

- 第 101 页的『Java 样本程序』
- 第 53 页的『样本文件』
- 第 58 页的『样本程序: 结构和设计』

相关参考:

- 第 70 页的『JDBC 样本』
- 第 72 页的『SQLJ 样本』
- 第 75 页的『Java 插件样本』

# Java 插件样本

表 10. 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。
Example3Folder.java	如何在“控制中心”树中的“数据库对象”下添加新对象。
Example4.java	如何除去“数据库”和“实例”对象的“配置”菜单项。
Example5.java	如何从“表”除去“改变”操作。
Example6.java	如何禁用“数据库配置”缺省按钮。

## 相关概念:

- 『引入“控制中心”的插件体系结构』（《管理指南：实现》）
- 『“控制中心”插件性能注意事项』（《管理指南：实现》）
- 『编译和运行示例插件』（《管理指南：实现》）
- 第 101 页的『Java 样本程序』
- 第 53 页的『样本文件』
- 第 58 页的『样本程序：结构和设计』
- 『编写作为“控制中心”扩展的插件』（《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 70 页的『JDBC 样本』
- 第 72 页的『SQLJ 样本』
- 第 74 页的『Java WebSphere 样本』

# COBOL 样本

注：在用于 C、CLI、C++ 和 Java 样本的 DB2 级别设计中未结构化 COBOL 样本。

表 11. 不带有任何嵌入式 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>• sqlqryc — 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>
dbcat.cbl	<ul style="list-style-type: none"> <li>• sqlecadb — 编目数据库</li> <li>• sqledcls — 关闭数据库目录扫描</li> <li>• sqledgne — 获取下一个数据库目录条目</li> <li>• sqledosd — 打开数据库目录扫描</li> <li>• sqleuncd — 取消编目数据库</li> </ul>
dbcmr.cbl	<ul style="list-style-type: none"> <li>• sqledcgd — 更改数据库注释</li> <li>• sqledcls — 关闭数据库目录扫描</li> <li>• sqledgne — 获取下一个数据库目录条目</li> <li>• sqledosd — 打开数据库目录扫描</li> <li>• sqleisig — 安装信号处理程序</li> </ul>
dbconf.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> <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>



表 11. 不带有任何嵌入式 SQL 的 COBOL DB2 API 样本程序 (续)

样本程序	包括的 API
dbinst.cbl	<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.cbl	<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.cbl	<ul style="list-style-type: none"> <li>• sqleatin — 连接</li> <li>• sqlmonss — 获取快照</li> </ul>
dbstart.cbl	<ul style="list-style-type: none"> <li>• sqlepstart — 启动数据库管理器</li> </ul>
dbstop.cbl	<ul style="list-style-type: none"> <li>• sqlefrce — 强制应用程序</li> <li>• sqlepstp — 停止数据库管理器</li> </ul>
dcscat.cbl	<ul style="list-style-type: none"> <li>• sqlegdad — 编目 DCS 数据库</li> <li>• sqlegdcl — 关闭 DCS 目录扫描</li> <li>• sqlegdel — 取消编目 DCS 数据库</li> <li>• sqlegdge — 获取数据库的 DCS 目录条目</li> <li>• sqlegdgt — 获取 DCS 目录条目</li> <li>• sqlegdsc — 打开 DCS 目录扫描</li> </ul>
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>• sqlemgdb — 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>

表 11. 不带有任何嵌入式 SQL 的 COBOL DB2 API 样本程序 (续)

样本程序	包括的 API
nodecat.cbl	<ul style="list-style-type: none"> <li>• sqlectnd — 编目节点</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>• sqleuncn — 取消编目节点</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>• sqlesact — 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 12. 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>• sqluadaw — 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>• sqluexpr — 导出</li> <li>• sqluimpr — 导入</li> </ul>
impexp.sqb	<ul style="list-style-type: none"> <li>• sqluexpr — 导出</li> <li>• sqluimpr — 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 — 取装表空间容器查询</li> <li>• sqlbotcq — 打开表空间容器查询</li> <li>• sqlbtcq — 表空间容器查询</li> <li>• sqlefmem — 可用内存</li> </ul>

表 12. 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 — 取装表空间查询</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>• sqluexpr — 导出</li> <li>• sqluload — 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 — 取装表空间容器查询</li> <li>• sqlbftpq — 取装表空间查询</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>• sqlbtcq — 表空间容器查询</li> <li>• sqlfmem — 可用内存</li> </ul>

表 13. 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。

表 13. 不带有任何 DB2 API 的 COBOL 嵌入式 SQL 样本程序 (续)

样本程序名	程序描述
lobloc.sqb	演示 LOB 定位器的用法。
openftch.sqb	演示如何使用静态 SQL 取装、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。
trigsql.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。

相关概念:

- 第 53 页的『样本文件』

SQL 过程样本

表 14. SQL 过程样本程序文件

样本程序名	程序描述
basecase.db2	UPDATE_SALARY 过程提高“sample”数据库的“staff”表中由“empno” IN 参数标识的员工的工资。该过程根据使用“rating” IN 参数的 CASE 语句确定提高的数量。
basecase.sqc	调用 UPDATE_SALARY 过程。
baseif.db2	UPDATE_SALARY_IF 过程提高“sample”数据库的“staff”表中由“empno” IN 参数标识的员工的工资。该过程根据使用“rating” IN 参数的 IF 语句确定提高的数量。
baseif.sqc	调用 UPDATE_SALARY_IF 过程。
dynamic.db2	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。
dynamic.sqc	调用 CREATE_DEPT_TABLE 过程。
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。

表 14. SQL 过程样本程序文件 (续)

样本程序名	程序描述
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 参数指示的部门的员工工资。
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 参数指示的部门的员工工资。
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。
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，它可接收结果集。
resultset.db2	MEDIAN_RESULT_SET 过程从 “sample” 数据库的 “staff” 表中获取由 “dept” IN 参数指示的部门的员工的平均工资。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员工的结果集。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。
whiles.db2	DEPT_MEDIAN 过程从 “sample” 数据库的 “staff” 表中获取由 “dept” IN 参数指示的部门的员工的平均工资。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。

相关概念:

- 第 53 页的『样本文件』

## Visual Basic 样本

表 15. Visual Basic ADO 样本程序文件

样本程序名	程序描述
客户机级别	
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。
数据库级别	
dbConn.bas	如何连接数据库和断开与数据库的连接。
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。
数据类型级别	
dtHier.bas	如何检索分层数据。
dtLob.bas	如何读写 LOB 数据。
存储过程	
spCall.bas	如何调用存储过程。
用户定义函数	
udfUse.bas	如何创建和使用 UDT 和 UDF。

表 16. Visual Basic RDO 和 MTS 样本程序文件

样本程序名	程序描述
Bank.vbp	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。此程序的文件在 samples\VB\RDO 中。
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，在样本数据库中创建了一个临时表 DB2MTS。此程序的文件在 samples\VB\MTS 中。</p>
db2mts.vbp	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。此程序的文件在 samples\VB\MTS 中。

相关概念:

- 第 53 页的『样本文件』
- 第 58 页的『样本程序：结构 and 设计』

相关参考:

- 第 83 页的『Windows Management Instrumentation 样本』

Visual C++ 样本

表 17. 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样本数据库的 ORG 表中的数据。

相关概念:

- 第 53 页的『样本文件』

Windows Management Instrumentation 样本

表 18. Windows Management Instrumentation 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。
listsvr.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。
rollfwd.db.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 53 页的『样本文件』
- 第 277 页的『Windows Management Instrumentation (WMI)』

相关参考:

- 第 82 页的『Visual Basic 样本』

### “对象链接与嵌入”（OLE）样本

表 19. “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 通讯录（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 自动化用户定义函数，使用实例变量演示暂存区（用 Visual Basic 实现）。
ccounter	计数器 OLE 自动化用户定义函数（用 Visual C++ 实现）。
salarysrv	一个 OLE 自动化存储过程，它计算样本数据库中 STAFF 表的平均薪水（用 Visual Basic 实现）。
salarycltvc	一个 Visual C++ DB2 CLI 样本，它调用 Visual Basic 存储过程 salarysrv。
salarycltvb	一个 Visual Basic DB2 CLI 样本，它调用 Visual Basic 存储过程 salarysrv。
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）。

相关概念:

- 第 53 页的『样本文件』

相关参考:

- 第 85 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

表 20. “对象链接与嵌入数据库”（OLE DB）表函数

样本程序名	程序描述
jet.db2	Microsoft.Jet.OLEDB.3.51 提供程序
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 服务器的 Microsoft OLE DB 提供程序

相关概念:

- 第 53 页的『样本文件』

相关参考:

- 第 84 页的『“对象链接与嵌入”（OLE）样本』

## 命令行处理器（CLP）样本

表 21. 命令行处理器（CLP）样本脚本。

样本文件名	文件描述
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。
join.db2	演示表的外连接。
stock.db2	演示触发器的用法。
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。

相关概念:

- 第 53 页的『样本文件』

## REXX 样本

表 22. 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	演示 API: GET DATABASE CONFIGURATION DEFAULTS
d_dbmcon	演示 API: GET DATABASE MANAGER CONFIGURATION DEFAULTS
db_udcs	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	演示 GET AUTHORIZATIONS API
dbcat	演示下列 API:  CATALOG DATABASE CLOSE DATABASE DIRECTORY SCAN GET NEXT DATABASE DIRECTORY ENTRY OPEN DATABASE DIRECTORY SCAN UNCATALOG DATABASE
dbcmt	演示下列 API:  CHANGE DATABASE COMMENT GET ERROR MESSAGE INSTALL SIGNAL HANDLER
dbconf	演示下列 API:  CREATE DATABASE DROP DATABASE GET DATABASE CONFIGURATION RESET DATABASE CONFIGURATION UPDATE DATABASE CONFIGURATION
dbinst	演示下列 API:  ATTACH TO INSTANCE DETACH FROM INSTANCE GET INSTANCE
dbmconf	演示下列 API:  GET DATABASE MANAGER CONFIGURATION RESET DATABASE MANAGER CONFIGURATION UPDATE DATABASE MANAGER CONFIGURATION
dbstart	演示 START DATABASE MANAGER API
dbstat	演示下列 API:  REORGANIZE TABLE RUN STATISTICS

表 22. REXX 样本程序文件。(续)

样本文件名	文件描述
dbstop	演示下列 API: FORCE USERS STOP DATABASE MANAGER
dcscat	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	演示使用动态 SQL 的“CURSOR”的用法
ebcdicdb	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	演示 EXPORT 和 IMPORT API
lobeval	演示延迟数据库中 LOB 的求值
lobfile	演示 LOB 文件句柄的用法
lobloc	演示 LOB 定位器的用法
lobval	演示 LOB 的用法
migrate	演示 MIGRATE DATABASE API
nodecat	演示下列 API: CATALOG NODE CLOSE NODE DIRECTORY SCAN GET NEXT NODE DIRECTORY ENTRY OPEN NODE DIRECTORY SCAN UNCATALOG NODE
quitab	演示 API: QUIESCE TABLESPACES FOR TABLE
rechist	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	演示 RESTART DATABASE API
sqlcsrx	整理顺序的示例
updat	使用动态 SQL 来更新数据库

相关概念:

- 第 53 页的『样本文件』

相关任务:

- 第 186 页的『在 AIX 上构建 REXX 应用程序』
- 第 308 页的『在 Windows 上构建 Object REXX 应用程序』

日志管理用户出口样本

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 23. “日志管理用户出口” 样本程序文件。

样本文件名	文件描述
db2uext2.cadsm	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 db2uext2.c，并作为 C 程序编译。在 UNIX 和 Windows 操作系统上可用。 注：必须使用 xlc_r 或 xlc_r（而不是 xlc 或 xlc）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 index.storsys.ibm.com 匿名 FTP 服务器上是可用的。
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 db2uext2.c，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 db2uext2.c，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 db2uext2.c，并作为 C 程序编译。在 UNIX 和 Windows 操作系统上可用。

相关概念:

- 第 53 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》）

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 上则称为批处理文件。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 为受支持的平台上的每一种语言提供了构建文件，这些文件位于每种语言的样本程序目录下，用它们构建的各种类型程序可在这些受支持的平台上运行。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。在 Windows 上，构建文件具有 .bat（批处理）扩展名，它不包括在该表中。对于 UNIX 平台，没有扩展名。

表 24. 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	UNIX C/C++ 多线程应用程序
bldcli	sqlproc samples 子目录中 SQL 过程的 CLI 客户机应用程序。
bldevm	AIX 和 Windows 事件监视器程序 evm（仅限于 C 语言）

注: bldcli 文件与 samples/cli 目录中的 bldapp 文件相同。赋给它另一个名称，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。用户还可以存取适当的样本目录中的文本文件。

表 25. 按语言 and 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 bldevm	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.bat bldmc.bat bldevm.bat
C++ samples/cpp	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.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	n/a	n/a	n/a	bldapp.bat bldrtn.bat
Micro Focus COBOL samples/cobol_mf	bldapp bldrtn	bldapp bldrtn	n/a	bldapp bldrtn	bldapp.bat bldrtn.bat

构建文件在文档中用于构建应用程序和例程，原因是它们非常清楚地演示了 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。这些增加的变量名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，应将数据库名以硬编码的方式写入构建文件来替换样本。

用于构建嵌入式 SQL 程序的构建文件调用另一个文件 `embprep`，该文件包含构建嵌入式 SQL 程序所需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 `db2prof` 定制器的用户标识和密码（即使存取的是本地数据库）。这遵从类型 4 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、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 92 页的『`Makefile`』
- 第 95 页的『错误检查实用程序』
- 第 53 页的『样本文件』

#### 相关参考:

- 第 9 页的『`AIX` 支持的用于构建和运行应用程序的软件』
- 第 10 页的『`HP-UX` 支持的用于构建和运行应用程序的软件』
- 第 11 页的『`Linux` 支持的用于构建和运行应用程序的软件』
- 第 13 页的『`Solaris` 支持的用于构建和运行应用程序的软件』
- 第 14 页的『`Windows` 支持的用于构建和运行应用程序的软件』

## 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 开发工具箱”（JDK）一起交付的。但是，DB2® 提供了 `makefile` 作为 Java 样本的选项，以便用户在想要利用 `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 26. 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	n/a	n/a	n/a	makefile
Micro Focus COBOL samples/cobol_mf	makefile	makefile	n/a	makefile	makefile
SQL 过程 samples/sqlproc	makefile	makefile	makefile	makefile	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（参见 `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 设置为样本数据库，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 上，该变量设置为 `rm -f`；而在 Windows<sup>®</sup> 上，它设置为 `del`。

#### 相关概念:

- 第 89 页的『构建文件』
- 第 95 页的『错误检查实用程序』
- 第 53 页的『样本文件』

#### 相关参考:

- 第 9 页的『AIX 支持的用于构建和运行应用程序的软件』
- 第 10 页的『HP-UX 支持的用于构建和运行应用程序的软件』
- 第 11 页的『Linux 支持的用于构建和运行应用程序的软件』
- 第 13 页的『Solaris 支持的用于构建和运行应用程序的软件』
- 第 14 页的『Windows 支持的用于构建和运行应用程序的软件』

## 错误检查实用程序

“DB2<sup>®</sup> AD 客户机”提供了几种实用程序文件。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:

**checkerr.cb1**

用于 IBM COBOL 程序

**checkerr.cb1**

用于 Micro Focus COBOL 程序

**utilcli.c**

用于 CLI 程序

**utilcli.h**

utilcli.c 的头文件

**utilapi.c**

用于 C 非嵌入式 SQL 程序

**utilapi.h**

utilapi.c 的头文件

**utilemb.sqc**

用于 C 嵌入式 SQL 程序

**utilemb.h**

utilemb.sqc 的头文件

**utilapi.C**

用于 C++ 非嵌入式 SQL 程序

**utilapi.h**

utilapi.C 的头文件

**utilemb.sqC**

用于 C++ 嵌入式 SQL 程序

**utilemb.h**

utilemb.sqC 的头文件

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 SqlInfoPrint() 函数和 TransRollback() 函数的 EMB\_SQL\_CHECK 宏:

```
#define EMB_SQL_CHECK( MSG_STR )          \
    if( SqlInfoPrint( MSG_STR, &sqlca, __LINE__, __FILE__ ) != 0 ) \
        TransRollback( );
```

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, 并打印出一条适当的错误消息。

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 89 页的『构建文件』
- 第 92 页的『`Makefile`』
- 第 53 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 . . . . .	101	UNIX 的 SQLJ 应用程序选项 . . . . .	116
Java applet 注意事项 . . . . .	103	SQLJ 应用程序和 Applet 的 Windows 批 处理文件 . . . . .	116
JDBC . . . . .	105	Windows 的 SQLJ 应用程序选项 . . . . .	118
构建 JDBC applet . . . . .	105	构建 SQLJ 例程 . . . . .	119
构建 JDBC 应用程序 . . . . .	106	SQLJ 例程的 UNIX 构建脚本 . . . . .	120
构建 JDBC 例程 . . . . .	107	UNIX 的 SQLJ 存储过程选项 . . . . .	122
SQLJ . . . . .	110	SQLJ 例程的 Windows 批处理文件 . . . . .	123
构建 SQLJ 程序 . . . . .	110	Windows 的 SQLJ 存储过程选项 . . . . .	125
构建 SQLJ applet . . . . .	111		
构建 SQLJ 应用程序 . . . . .	113		
SQLJ 应用程序和 Applet 的 UNIX 构建脚 本 . . . . .	114		

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，访问以下 Web 页：

<http://www.ibm.com/software/data/db2/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 samples 目录结构：

- 在 UNIX 上：

#### **sql1ib/samples/java**

在所有子目录中都包含 Java 样本程序的 README 文件。

#### **sql1ib/samples/java/jdbc**

包含 JDBC 样本程序文件。

#### **sql1ib/samples/java/sqlj**

包含 SQLJ 样本程序。

#### **sql1ib/samples/java/WebSphere**

包含 WebSphere 样本程序。

#### **sql1ib/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**

在所有子目录中都包含 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 并不与 Java 广泛配合使用，因而“Java 开发工具箱”（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函数））。

#### **相关任务:**

- 第 20 页的『设置 Java 环境』
- 第 105 页的『构建 JDBC applet』
- 第 106 页的『构建 JDBC 应用程序』
- 第 107 页的『构建 JDBC 例程』
- 第 111 页的『构建 SQLJ applet』

- 第 113 页的『构建 SQLJ 应用程序』
- 第 119 页的『构建 SQLJ 例程』
- 第 110 页的『构建 SQLJ 程序』

#### 相关参考:

- 第 70 页的『JDBC 样本』
- 第 72 页的『SQLJ 样本』
- 第 74 页的『Java WebSphere 样本』
- 第 75 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. 如果正在使用现在不建议使用的类型 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:

- 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 服务器的级别早于修订包 2:

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

为了测试您的 JDBC 环境，可以使用样本文件 db2JDBCVersion.java，该文件在 sqllib\samples\java (Windows) 中或者在 sqllib/samples/java 中 (UNIX)。db2JDBCVersion 程序检查当前正在使用的是哪个版本的 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 迁移至类型 4 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封装到一个 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一起封装。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，参见 JDK 版本 1.3 文档。

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

```
java.lang.ClassNotFoundException: Applt_SJProfile0
```

一个解决办法是，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。可使用下列命令之一转换概要文件：

```
profconv Applt_SJProfile0.ser
```

或

```
java sqlj.runtime.profile.util.SerProfileToClass Applt_SJProfile0.ser
```

最后，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. 您可能希望将文件 db2java.zip（对于 SQLJ applet，还有文件 runtime.zip）放入可从 Web 站点装入的几个 applet 共享的目录中。在 Windows® 操作系统上，这些文件位于 sqllib\java 目录中，而在 UNIX 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，参见 JDK 版本 1.3 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 命令来删掉 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（具体说来就是在 Domino™ Go Webserver 上）运行 DB2 Java applet 的信息，访问：

<http://www.ibm.com/software/data/db2/db2lotus/gojava.htm>

#### 相关任务:

- 第 20 页的『设置 Java 环境』
- 第 105 页的『构建 JDBC applet』
- 第 111 页的『构建 SQLJ applet』

---

## JDBC

### 构建 JDBC applet

Applt 演示一个存取 DB2 数据库的动态 SQL Java applet。

#### 过程:

可以使用现在不建议使用的类型 3 驱动程序（也称为“net”驱动程序）或使用类型 4 驱动程序。关于与这两种驱动程序连接的各节将在后面提供。强烈建议将 applet 迁移至类型 4 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，或通过在您的客户机工作目录中输入以下命令来使用“Java 开发工具箱”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”。

#### 与类型 4 驱动程序连接

要与类型 4 驱动程序连接，根据 Applt.html 文件中的指示信息修改该文件（指定的 TCP/IP 端口号应为数据库端口号“50000”除外）。

## 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 `sqllib\java\db2java.zip` 或 UNIX 上的 `sqllib/java/db2java.zip` 复制到 `Applt.class` 和 `Applt.html` 所在的目录中。

如果正在使用类型 4 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 103 页的『Java applet 注意事项』

### 相关任务:

- 第 106 页的『构建 JDBC 应用程序』
- 第 107 页的『构建 JDBC 例程』
- 第 111 页的『构建 SQLJ applet』

### 相关参考:

- 第 70 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 JDK 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 外壳程序：

```
export LIBPATH=$HOME/sql1lib/lib32
```

- 如果使用 C 外壳程序：

```
setenv LIBPATH $HOME/sql1lib/lib32
```

**相关任务：**

- 第 105 页的『构建 JDBC applet』
- 第 107 页的『构建 JDBC 例程』
- 第 113 页的『构建 SQLJ 应用程序』

**相关参考：**

- 第 70 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 语句的用户定义函数
- 带有 SQL 语句的用户定义函数

**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 sqllib\function 目录中或 UNIX 上的 sqllib/function 目录中。
3. 接着, 通过在服务器上运行 spcat 脚本来编目例程。输入:

```
spcat
```

此脚本连接至样本数据库, 如果先前通过调用 Spdrop.db2 编目了这些例程, 则取消对它们的编目, 然后通过调用 Spcreate.db2 来编目它们, 最后与数据库断开连接。还可以分别运行 Spdrop.db2 和 Spcreate.db2 脚本。

4. 接着, 停止并重新启动数据库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。
5. 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 语句的用户定义函数

UDFsrv 是一个不包含 SQL 语句的用户定义函数库。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:

1. 使用如下命令编译 UDFsrv.java 来生成文件 UDFsrv.class:

```
javac UDFsrv.java
```

2. 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。
3. 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 (一旦注册了这些话)。

### 带有 SQL 语句的用户定义函数

UDFsqlsv 是一个包含 SQL 语句的用户定义函数库。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:

1. 使用以下命令编译 UDFsqlsv.java 来生成文件 UDFsqlsv.class:

```
javac UDFsqlsv.java
```



2. 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。
3. 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 105 页的『构建 JDBC applet』
- 第 106 页的『构建 JDBC 应用程序』
- 第 119 页的『构建 SQLJ 例程』

#### 相关参考:

- 第 70 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)』

### 构建 SQLJ 程序

DB2 提供了构建 SQLJ 样本程序的构建文件。对于 applet 和应用程序，可以使用 UNIX 上的 `blsqlj` 脚本或 Windows 上的 `blsqlj.bat` 批处理文件。对于例程（存储过程 and 用户定义函数），可以使用 UNIX 上的 `blsqljs` 脚本或 Windows 上的 `blsqljs.bat` 批处理文件。

随 DB2 一起交付的 SQLJ 转换程序将已转换的 `.java` 文件编译为 `.class` 文件。因此，构建文件不使用 Java 编译器。

**注：**在 DB2 的先前版本中，`db2prof` 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。新格式遵从 DB2 类型 4 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` 是在 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”。

#### 过程:

要使用 IBM Java 开发工具箱 UNIX 版和 Windows 版操作系统构建和运行 SQLJ 程序，必须针对您的操作系统使用以下命令关闭 JDK 的即时编译器:

#### UNIX:

对于 bash 或 Korn 外壳程序:

```
export JAVA_COMPILER=NONE
```

对于 C 外壳程序:

```
setenv JAVA_COMPILER NONE
```

#### Windows:

```
SET JAVA_COMPILER=NONE
```

要构建不同类型的 DB2 SQLJ 程序，参见下列主题:

- 构建 SQLJ applet
- 构建 SQLJ 应用程序
- 构建 SQLJ 例程

#### 相关概念:

- 第 101 页的『Java 样本程序』

- 第 103 页的『Java applet 注意事项』

#### 相关任务:

- 第 111 页的『构建 SQLJ applet』
- 第 113 页的『构建 SQLJ 应用程序』
- 第 119 页的『构建 SQLJ 例程』

#### 相关参考:

- 第 72 页的『SQLJ 样本』

## 构建 SQLJ applet

下列步骤显示如何构建 Applet 样本，它将演示存取 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。

该构建文件使用多达 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。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，参见构建文件。

#### 过程:

可以使用现在不建议使用的类型 3 驱动程序（也称为“net”驱动程序）或使用类型 4 驱动程序。将在下面提供关于与这两种驱动程序连接的各节内容。强烈建议将 applet 迁移至类型 4 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 开发工具箱”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该号码。不要使用数据库端口号“50000”。

## 与类型 4 驱动程序连接

要与类型 4 驱动程序连接，根据 Applet.html 文件中的指示信息修改该文件（指定的 TCP/IP 端口号应为数据库端口号“50000”除外）。

### 构建 applet

1. 使用如下命令构建该 applet:

```
bldsqlj Applet <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：

Applet.html	Applet.class
Applet_Cursor1.class	Applet_Cursor2.class
Applet_SJProfileKeys.class	Applet_SJProfile0.ser

3. 如果正在使用类型 3 驱动程序，则将 Windows 上的 sqllib\java\runtime.zip 和 sqllib\java\db2java.zip 或 UNIX 上的 sqllib/java/runtime.zip 和 sqllib/java/db2java.zip 复制到 Applet.class 和 Applet.html 所在的目录中。  
如果正在使用类型 4 驱动程序，则将 Windows 上的 sqllib\java\runtime.zip 和 sqllib\java\db2jcc.jar 或 UNIX 上的 sqllib/java/runtime.zip 和 sqllib/java/db2jcc.jar 复制到 Applet.class 和 Applet.html 所在的目录中。
4. 在客户机上，启动 Web 浏览器（它必须支持 JDK 1.3）或 applet 查看器，并装入 Applet.html。

还可使用 Java makefile 构建此程序。

### 相关概念:

- 第 103 页的『Java applet 注意事项』

### 相关任务:

- 第 105 页的『构建 JDBC applet』
- 第 113 页的『构建 SQLJ 应用程序』
- 第 119 页的『构建 SQLJ 例程』

### 相关参考:

- 第 115 页的『UNIX 的 SQLJ 应用程序选项』

- 第 118 页的『Windows 的 SQLJ 应用程序选项』
- 第 72 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。

该构建文件使用多达 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。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，参见构建文件。

过程:

要使用构建文件 bldsqlj (UNIX) 或 bldsqlj.bat (Windows) 构建 TbMod，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 JDK 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 外壳程序:
 

```
export LIBPATH=$HOME/sql1lib/lib32
```
- 如果使用 C 外壳程序:

```
setenv LIBPATH $HOME/sql1lib/lib32
```

#### 相关任务:

- 第 106 页的『构建 JDBC 应用程序』
- 第 111 页的『构建 SQLJ applet』
- 第 119 页的『构建 SQLJ 例程』

#### 相关参考:

- 第 115 页的『UNIX 的 SQLJ 应用程序选项』
- 第 118 页的『Windows 的 SQLJ 应用程序选项』
- 第 72 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 UNIX Java embedded SQL (SQLJ) applications
# 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
#       password    = $PSWD variable
#       server_name = local hostname
#       port_number = 50000
#       db_name     = sample

# To add defaults for user ID (USER) and password (PSWD)
# Uncomment the following and add the values in the quotes
# USER=""
# PSWD=""

# Translate and compile the SQLJ source file
# and bind the package to the database.
if ( (( $# == 1 )) && [ $USER != "" && $PSWD != "" ] ) ||
    (( $# >= 3 && $# <= 6 ))
then
    sqlj $1.sqlj
```

```

        if (($# == 1))
        then
            db2profcc -url=jdbc:db2://$(hostname):50000/sample \
                -user=$USER -password=$PSWD \
                -preoptions="package using $1" $1_SJProfile0
        elif (($# == 3))
        then
            db2profcc -url=jdbc:db2://$(hostname):50000/sample -user=$2 \
                -password=$3 -preoptions="package using $1" $1_SJProfile0
        elif (($# == 4))
        then
            db2profcc -url=jdbc:db2://$4:50000/sample -user=$2 -password=$3 \
                -preoptions="package using $1" $1_SJProfile0
        elif (($# == 5))
        then
            db2profcc -url=jdbc:db2://$4:$5/sample -user=$2 -password=$3 \
                -preoptions="package using $1" $1_SJProfile0
        else
            db2profcc -url=jdbc:db2://$4:$5/$6 -user=$2 -password=$3 \
                -preoptions="package using $1" $1_SJProfile0
        fi
    else
        echo 'Usage: bldsqlj prog_name (requires hardcoding user ID and password)'
        echo '      bldsqlj prog_name userid password'
        echo '      bldsqlj prog_name userid password server_name'
        echo '      bldsqlj prog_name userid password server_name port_number'
        echo '      bldsqlj prog_name userid password server_name port_number'
        echo '      db_name'
        echo ''
        echo '      Defaults:'
        echo '      userid      = '$USER
        echo '      password    = '$PSWD
        echo '      server_name = '$(hostname)
        echo '      port_number = 50000'
        echo '      db_name     = sample'
    fi
fi

```

## UNIX 的 SQLJ 应用程序选项

下表包含在 UNIX 上的 bldsqlj 构建脚本中使用的 SQLJ 转换程序和预编译选项。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 的转换程序和预编译选项			
<b>sqlj</b>	SQLJ 转换程序（也编译该程序）。		
<b>\$1.sqlj</b>	SQLJ 源文件。		
<b>db2profcb</b>	DB2 Java 版概要文件定制器。		
<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>-preoptions</b>	用字符串 "package using %1" 指定数据库的程序包名，其中 %1 是 SQLJ 源文件名。		
<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 113 页的『构建 SQLJ 应用程序』

相关参考:

- 第 122 页的『UNIX 的 SQLJ 存储过程选项』

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 Windows Java embedded SQL (SQLJ) applications and applets
rem To add defaults for user ID (USER) and password (PSWD)
rem Uncomment the following and add the appropriate values
rem set USER=
rem set PSWD=

goto start
:usage
echo Usage: bldsqlj prog_name (requires hardcoding user ID and password)
echo          bldsqlj prog_name userid password
```



```

echo          blsqlj 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      = %USER%
echo          password    = %PSWD%
echo          server_name = %COMPUTERNAME%
echo          port_number = 50000
echo          db_name     = sample
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 "%USER%" == "" goto usage
if "%PSWD%" == "" goto usage
if "%COMPUTERNAME%" == "" goto nohostname
sqlj %1.sqlj
db2profrc -url=jdbc:db2://%COMPUTERNAME%:50000/sample -user=%USER%
          -password=%PSWD% -preoptions="package using %1" %1_SJProfile0
goto continue

:case3
if "%COMPUTERNAME%" == "" goto nohostname
sqlj %1.sqlj
db2profrc -url=jdbc:db2://%COMPUTERNAME%:50000/sample -user=%2
          -password=%3 -preoptions="package using %1" %1_SJProfile0
goto continue

:case4
sqlj %1.sqlj
db2profrc -url=jdbc:db2://%4:50000/sample -user=%2 -password=%3
          -preoptions="package using %1" %1_SJProfile0
goto continue

:case5
sqlj %1.sqlj
db2profrc -url=jdbc:db2://%4:%5/sample -user=%2 -password=%3
          -preoptions="package using %1" %1_SJProfile0
goto continue

:case6
sqlj %1.sqlj
db2profrc -url=jdbc:db2://%4:%5/%6 -user=%2 -password=%3

```

```
-preoptions="package using %1" %1_SJProfile0
goto continue

:continue
rem Put any post building steps here
goto exit

:nohostname
echo Local server name (hostname) could not be determined.
echo.
goto usage

:exit
@echo on
```

Windows 的 SQLJ 应用程序选项

下表包含在 Windows 操作系统上的 bldsqlj.bat 批处理文件中使用的 SQLJ 转换程序和预编译选项。这些是 DB2 建议您用来构建 SQLJ 例程（存储过程和用户定义函数）的选项。

bldsqlj.bat 的转换程序和预编译选项			
<b>sqlj</b>	SQLJ 转换程序（也编译该程序）。		
<b>%1.sqlj</b>	SQLJ 源文件。		
<b>db2profrc</b>	DB2 Java 版概要文件定制器。		
<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>-preoptions</b>	用字符串 "package using %1" 指定数据库的程序包名，其中 %1 是 SQLJ 源文件名。		
<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 113 页的『构建 SQLJ 应用程序』

### 相关参考:

- 第 124 页的『Windows 的 SQLJ 存储过程选项』

### 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。

该构建文件使用多达 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。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，参见构建文件。

### 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` 脚本来编目例程。输入:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 Spdrop.db2 编目了这些例程，则取消对它们的编目，然后通过调用 Spcreate.db2 来编目它们，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，停止并重新启动数据库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. 构建和运行 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 107 页的『构建 JDBC 例程』
- 第 111 页的『构建 SQLJ applet』
- 第 113 页的『构建 SQLJ 应用程序』

#### 相关参考:

- 第 122 页的『UNIX 的 SQLJ 存储过程选项』
- 第 124 页的『Windows 的 SQLJ 存储过程选项』
- 第 72 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 UNIX Java embedded SQL (SQLJ) routines
# 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
#      password    = $PSWD variable
#      server_name = local hostname
#      port_number = 50000
#      db_name     = sample

# To add defaults for user ID (USER) and password (PSWD)
# Uncomment the following and add the values in the quotes
# USER=""
# PSWD=""

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Translate and compile the SQLJ source file
# and bind the package to the database.
if ( (( $# == 1 )) && [ $USER != "" && $PSWD != "" ] ) ||
  (( $# >= 3 && $# <= 6 ))
then
  sqlj $1.sqlj

  if (($# == 1))
  then
    db2profrc -url=jdbc:db2://$(hostname):50000/sample \
      -user=$USER -password=$PSWD \
      -preoptions="package using $1" $1_SJProfile0
  elif (($# == 3))
  then
    db2profrc -url=jdbc:db2://$(hostname):50000/sample -user=$2 \
      -password=$3 -preoptions="package using $1" $1_SJProfile0
  elif (($# == 4))
  then
    db2profrc -url=jdbc:db2://$4:50000/sample -user=$2 -password=$3 \
      -preoptions="package using $1" $1_SJProfile0
  elif (($# == 5))
  then
    db2profrc -url=jdbc:db2://$4:$5/sample -user=$2 -password=$3 \
      -preoptions="package using $1" $1_SJProfile0
  else
    db2profrc -url=jdbc:db2://$4:$5/$6 -user=$2 -password=$3 \
      -preoptions="package using $1" $1_SJProfile0
  fi

  # Copy the *.class and *.ser files to the 'function' directory.
  rm -f $DB2PATH/function/$1*.class
  rm -f $DB2PATH/function/$1*.ser
  cp $1*.class $DB2PATH/function

```

```

        cp $1*.ser $DB2PATH/function
    else
        echo 'Usage: bldsqljs prog_name (requires hardcoding user ID and password)'
        echo '      bldsqljs prog_name userid password'
        echo '      bldsqljs prog_name userid password server_name'
        echo '      bldsqljs prog_name userid password server_name port_number'
        echo '      bldsqljs prog_name userid password server_name port_number'
        echo '      db_name'
        echo ''
        echo '      Defaults:'
        echo '          userid      = '$USER
        echo '          password    = '$PSWD
        echo '          server_name = '$(hostname)
        echo '          port_number = 50000'
        echo '          db_name     = sample'
    fi

```

## UNIX 的 SQLJ 存储过程选项

下表包含在 UNIX 上的 bldsqljs 构建脚本中使用的 SQLJ 转换程序和预编译选项。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函数）的选项。

bldsqljs 的转换程序和预编译选项	
<b>sqlj</b>	SQLJ 转换程序（也编译该程序）。
<b>\$1.sqlj</b>	SQLJ 源文件。
<b>db2profrc</b>	DB2 Java 版概要文件定制器。
<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>-prepoptions</b>	用字符串 "package using %1" 指定数据库的程序包名，其中 %1 是 SQLJ 源文件名。
<b>\$1_SJProfile0</b>	指定该程序的序列化概要文件。

### 相关任务:

- 第 119 页的『构建 SQLJ 例程』

### 相关参考:

- 第 115 页的『UNIX 的 SQLJ 应用程序选项』

### 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 Windows Java embedded SQL (SQLJ) routines

rem To add defaults for user ID (USER) and password (PSWD)
rem Uncomment the following and add the appropriate values
rem set USER=
rem set PSWD=

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      = %USER%
echo          password    = %PSWD%
echo          server_name = %COMPUTERNAME%
echo          port_number = 50000
echo          db_name     = sample
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 "%USER%" == "" goto usage
if "%PSWD%" == "" goto usage
if "%COMPUTERNAME%" == "" goto nohostname
sqlj %1.sqlj
db2profrc -url=jdbc:db2://%COMPUTERNAME%:50000/sample -user=%USER%
          -password=%PSWD% -preoptions="package using %1" %1_SJProfile0
```

```

        goto continue

:case3
    if "%COMPUTERNAME%" == "" goto nohostname
sqlj %1.sqlj
    db2profrc -url=jdbc:db2://%COMPUTERNAME%:50000/sample -user=%2
        -password=%3 -preoptions="package using %1" %1_SJProfile0
    goto continue

:case4
sqlj %1.sqlj
    db2profrc -url=jdbc:db2://%4:50000/sample -user=%2 -password=%3
        -preoptions="package using %1" %1_SJProfile0
    goto continue

:case5
sqlj %1.sqlj
    db2profrc -url=jdbc:db2://%4:%5/sample -user=%2 -password=%3
        -preoptions="package using %1" %1_SJProfile0
    goto continue

:case6
sqlj %1.sqlj
    db2profrc -url=jdbc:db2://%4:%5/%6 -user=%2 -password=%3
        -preoptions="package using %1" %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 Local server name (hostname) could not be determined.
echo.
goto usage

:exit
@echo on

```

## Windows 的 SQLJ 存储过程选项

下表包含在 Windows 操作系统上的 bldsqljs.bat 批处理文件中使用的 SQLJ 转换程序和预编译选项。这些是 DB2 建议您用来构建 SQLJ 例程（存储过程和用户定义函数）的选项。



bldsqljs.bat 的转换程序和预编译选项	
<b>sqlj</b>	SQLJ 转换程序（也编译该程序）。
<b>%1.sqlj</b>	SQLJ 源文件。
<b>db2prof</b>	DB2 Java 版概要文件定制器。
<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>-preproptions</b>	用字符串 "package using %1" 指定数据库的程序包名，其中 %1 是 SQLJ 源文件名。
<b>%1_SJProfile0</b>	指定该程序的序列化概要文件。

相关任务:

- 第 119 页的『构建 SQLJ 例程』

相关参考:

- 第 118 页的『Windows 的 SQLJ 应用程序选项』

相关样本:

- 『bldsqljs.bat -- Builds a Java embedded SQL (SQLJ) stored procedure on Windows』



---

# 第 5 章 SQL 过程

创建 SQL 过程 . . . . .	127	保留 SQL 过程的中间文件 . . . . .	131
在 UNIX 上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 . . . . .	132
在 Windows 上使用客户机应用程序调用 SQL 过程 . . . . .	129	备份和复原 SQL 过程 . . . . .	133
		分发可编译 SQL 过程 . . . . .	134
		重新绑定 SQL 过程 . . . . .	135

---

## 创建 SQL 过程

“DB2 命令行处理器”脚本（扩展名为 .db2 的脚本，在 UNIX 上，它们位于 `sqllib/samples/sqlproc` 目录中，而在 Windows 上，位于 `sqllib\samples\sqlproc` 目录中）将执行 `CREATE PROCEDURE` 语句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：.sqlc（表示 C 嵌入式 SQL）、.c（表示 DB2 CLI）或 .java（表示 JDBC）。

### 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* 是样本数据库所在的实例的用户标识和密码。

要执行 `resultset.db2` 脚本文件中包含的 `CREATE PROCEDURE` 语句，输入以下命令：

```
db2 -td@ -vf resultset.db2
```

现在，SQL 过程就可供调用了。

### 相关任务:

- 第 137 页的『使用 `CALL` 语句调用存储过程』
- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』
- 第 135 页的『重新绑定 SQL 过程』

### 相关样本:

- 『`resultset.db2` -- To create the `MEDIAN_RESULT_SET` SQL procedure』

---

## 在 UNIX 上使用客户机应用程序调用 SQL 过程

可以通过构建和运行客户机应用程序来在 UNIX 上调用 SQL 过程。DB2 在 `sqllib/samples/sqlproc` 目录中提供了样本客户机程序来调用可以从也是所提供的样本命令行处理器脚本创建的 SQL 过程。存在 DB2 CLI、C 嵌入式 SQL 以及 JDBC 的客户机源文件。

还提供了构建脚本（除了 JDBC）来创建这些应用程序接口的客户机程序：

- `bldcli` 包含构建 SQL 过程的 DB2 CLI 客户机应用程序的命令。这与 `sqllib/samples/cli` 目录中的 `bldapp` 脚本是同一个文件。
- `bldapp` 包含构建 SQL 过程的 C 嵌入式 SQL 客户机应用程序的命令。这与 `sqllib/samples/c` 目录中的 `bldapp` 脚本是同一个文件。

### 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 `resultset.c` 构建 DB2 CLI 客户机应用程序 `resultset`，输入：

```
bldcli resultset
```

此命令创建可执行文件 `resultset`。

要调用该存储过程，通过输入可执行文件名、要连接的数据库的名称和该数据库实例的用户标识和密码来运行样本客户机应用程序：

```
resultset 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`，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
```

产生可执行文件 `basecase`。

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：

```
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`。

要调用存储过程，对应用程序运行 `java` 解释器：

```
java NestedSP userid password database
```

#### 相关任务:

- 第 137 页的『使用 `CALL` 语句调用存储过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』
- 第 135 页的『重新绑定 SQL 过程』

#### 相关样本:

- 『`basecase.sqc` -- To call the `UPDATE_SALARY` SQL procedure』
- 『`rsultset.c` -- To call the `MEDIAN_RESULT_SET` SQL procedure』

---

## 在 Windows 上使用客户机应用程序调用 SQL 过程

可以通过构建和运行客户机应用程序来在 Windows 上调用 SQL 过程。DB2 在 `sqllib\samples\sqlproc` 目录中提供了样本客户机程序来调用可以从也是提供的样本命令行处理器脚本创建的 SQL 过程。存在 DB2 CLI、C 嵌入式 SQL 以及 JDBC 的客户机源文件。

提供了批处理文件（除了 JDBC）来创建这些应用程序接口的客户机程序：

- `bldcli.bat` 包含构建 SQL 过程的 DB2 CLI 客户机应用程序的命令。这与 `sqllib\samples\cli` 目录中的 `bldapp.bat` 是同一个文件。
- `bldapp.bat` 包含构建 SQL 过程的 C 嵌入式 SQL 客户机应用程序的命令。这与 `sqllib\samples\c` 目录中的 `bldapp.bat` 是同一个文件。

#### 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`，输入：

```
bldcli rsultset
```

此命令将创建可执行文件 `rsultset.exe`。

要调用该存储过程，通过输入可执行文件名、要连接的数据库的名称和该数据库实例的用户标识和密码来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`，输入批处理文件名、可执行文件名、正在连接的数据库以及包含数据库的实例的用户标识和密码：

```
bldapp basecase database userid password
```

产生可执行文件 `basecase.exe`。

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：

```
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`。

要调用存储过程，对应用程序运行 `java` 解释器：

```
java NestedSP userid password database
```

### **相关任务：**

- 第 137 页的『使用 `CALL` 语句调用存储过程』
- 第 128 页的『在 `UNIX` 上使用客户机应用程序调用 `SQL` 过程』
- 第 134 页的『分发可编译 `SQL` 过程』
- 第 135 页的『重新绑定 `SQL` 过程』

### **相关样本：**

- 『`basecase.sqc` -- To call the `UPDATE_SALARY` `SQL` procedure』
- 『`rsultset.c` -- To call the `MEDIAN_RESULT_SET` `SQL` procedure』

---

## 保留 SQL 过程的中间文件

在发出 CREATE PROCEDURE 语句时，DB2 将创建许多中间文件，正常情况下，如果 DB2 成功完成该语句，这些中间文件将被删除。如果 SQL 过程没有如期望的那样执行，您可能会发现保留这些文件是很有用的，可便于您检查 DB2 创建的 SQC、C 和消息日志文件。

### 过程:

要保存在成功执行 CREATE PROCEDURE 语句期间 DB2 创建的文件，必须将服务器上的 DB2\_SQLROUTINE\_KEEP\_FILES DB2 注册表变量的值设置为“1”、“y”或“yes”，如以下命令所示：

```
db2set DB2_SQLROUTINE_KEEP_FILES=1
```

然后停止并重新启动 DB2 以使更改生效。

这样，您就必须手工删除可能在未成功创建 SQL 过程时留下来的中间文件。这些文件存放在以下目录中：

### UNIX

```
$HOME/sqlllib/function/routine/sqlproc/<db_name>/<schema_name>/tmp
```

### Windows

```
sqlllib\function\routine\sqlproc\<db_name>\<schema_name>\tmp
```

其中 <db\_name> 和 <schema\_name> 分别是用来创建 SQL 过程的数据库和模式。

### 相关任务:

- 第 132 页的『为 SQL 过程定制预编译和绑定选项』
- 第 133 页的『备份和复原 SQL 过程』
- 第 127 页的『创建 SQL 过程』
- 第 137 页的『使用 CALL 语句调用存储过程』
- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』
- 第 135 页的『重新绑定 SQL 过程』

---

## 为 SQL 过程定制预编译和绑定选项

可通过设置 DB2\_SQLROUTINE\_PREPOPTS DB2 注册表变量来定制预编译和绑定选项。在过程级别，则不能定制这些选项。

### 过程:

要为 SQL 过程指定定制的预编译选项，使用以下命令将 DB2 预编译器将要使用的预编译选项列表放在 DB2 注册表中:

```
db2set DB2_SQLROUTINE_PREPOPTS=options
```

其中 *options* 指定 DB2 预编译器将要使用的预编译选项列表。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}
EXPLAINSAP {NO | YES | ALL}
FEDERATED {NO | YES}
INSERT {DEF | BUF}
ISOLATION {CS |RR |UR |RS |NC}
QUERYOPT optimization-level
SYNCPOINT {ONEPHASE | TWOPHASE | NONE}
```

**示例。**要让程序包对日期使用 ISO 格式并将“可重复读取”用作隔离级别，则按如下指定选项:

```
db2set DB2_SQLROUTINE_PREPOPTS="DATETIME ISO ISOLATION RR"
```

然后停止并重新启动 DB2 以使更改生效。

### 相关任务:

- 第 131 页的『保留 SQL 过程的中间文件』
- 第 133 页的『备份和复原 SQL 过程』
- 第 127 页的『创建 SQL 过程』
- 第 137 页的『使用 CALL 语句调用存储过程』
- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』
- 第 135 页的『重新绑定 SQL 过程』



---

## 备份和复原 SQL 过程

当创建 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，将会除去文件系统上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。  
原因代码: “7”。

则使用 `db2stop` 停止 DB2，并使用 `db2start` 重新启动它。

### 相关任务:

- 第 131 页的『保留 SQL 过程的中间文件』
- 第 132 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 137 页的『使用 CALL 语句调用存储过程』
- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』
- 第 135 页的『重新绑定 SQL 过程』

---

## 分发可编译 SQL 过程

定义 SQL 过程时，该过程被转换为 C 程序，还对目标数据库预编译并绑定它，并且编译和链接它以创建共享库。编译和链接步骤要求 C 或 C++ 编译器在数据库服务器上是可用的。但是，一旦定义了 SQL 过程，就可以用已编译的格式将它分发到使用相同操作系统和相同版本 DB2 的 DB2 数据库服务器上，但不需要对 C 或 C++ 编译器有存取权。在这些情况下，DB2 允许用户以已编译格式从一个数据库中抽取 SQL 过程，再将 SQL 过程以已编译格式安装到另一服务器上的另一个数据库中。

DB2 为抽取和安装都提供了命令行接口和编程接口。命令行接口由两个 CLP 命令 GET ROUTINE 和 PUT ROUTINE 组成。编程接口由两个内置存储过程 GET\_ROUTINE\_SAR 和 PUT\_ROUTINE\_SAR 组成。

### 过程:

要将编译过的 SQL 过程从一个数据库服务器分发到另一个数据库服务器，请执行以下步骤:

1. 构建应用程序，包括定义属于应用程序一部分的 SQL 过程。
2. 测试过程后，请将每个过程的已编译版本抽取到一个不同的文件，可通过发出 GET ROUTINE 命令或调用 GET\_ROUTINE\_SAR 存储过程来实现。复制文件到分布式媒体中（如果需要的话）。

**示例。**假定 SCHEMA1.MYPROC 是存在于源服务器上的 SQL 过程。以下命令将创建一个名为 myproc.sar 的文件，它包含过程的可分发表示:

```
GET ROUTINE INTO myproc.sar FROM PROCEDURE SCHEMA1.MYPROC
```

3. 通过发出 PUT ROUTINE 命令或调用 PUT\_ROUTINE\_SAR 存储过程，使用在先前步骤中创建的文件，在每台服务器上安装每个过程的已编译版本。

**示例。**以下命令根据目标服务器上的先前示例安装过程:

```
PUT ROUTINE FROM myproc.sar
```

每个数据库服务器都必须具有相同的操作系统和 DB2 级别。

**注:** 如果 GET ROUTINE 或 PUT ROUTINE 操作（或其相应过程）无法成功执行，则它将总是返回错误以及提供有关故障原因的信息的诊断文本。例如，如果提供给 GET ROUTINE 的过程名不标识 SQL 过程，则将返回诊断“-204, 42704”文本，其中“-204”和“42704”分别为 SQLCODE 和 SQLSTATE，它们标识问题的原因。此示例中的 SQLCODE SQLSTATE 指示未定义 GET ROUTINE 命令中提供的过程名。

### 相关任务:

- 第 21 页的『设置 SQL 过程环境』
- 第 127 页的『创建 SQL 过程』
- 第 137 页的『使用 CALL 语句调用存储过程』
- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 135 页的『重新绑定 SQL 过程』

#### 相关参考:

- 『GET\_ROUTINE\_SAR procedure』（*SQL Reference, Volume 1*）
- 『PUT\_ROUTINE\_SAR procedure』（*SQL Reference, Volume 1*）
- 『GET ROUTINE Command』（*Command Reference*）
- 『PUT ROUTINE Command』（*Command Reference*）

---

## 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 MYSCHEMA.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', 'MYSCHEMA.MYPROC', 'CONSERVATIVE')
```

其中 'P' 指示 'MYSCHEMA.MYPROC' 是一个过程名。第一个参数的值 'SP' 指示 'MYSCHEMA.MYPROC' 是一个特定过程名。'CONSERVATIVE' 指示应该应用保守重新绑定语义。参见下面相关链接中的 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 21 页的『设置 SQL 过程环境』
- 第 131 页的『保留 SQL 过程的中间文件』
- 第 132 页的『为 SQL 过程定制预编译和绑定选项』
- 第 133 页的『备份和复原 SQL 过程』
- 第 127 页的『创建 SQL 过程』
- 第 137 页的『使用 CALL 语句调用存储过程』
- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』

- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』

**相关参考:**

- 『REBIND Command』（*Command Reference*）

---

## 第 6 章 调用存储过程

使用 CALL 语句调用存储过程 . . . . . 137

本章提供了使用 CALL 语句在命令行上调用 DB2 存储过程的详细信息。

有关最新的 DB2 应用程序开发情况的更新，请访问以下 Web 页：

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

---

### 使用 CALL 语句调用存储过程

可以通过从 DB2 命令行处理器界面使用 call SQL 语句调用存储过程。必须在目录中定义调用的存储过程。

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* 是样本数据库所在的实例的用户标识和密码。

要使用 call 语句，输入存储过程名加上任何 IN 或 INOUT 参数，并输入 “?” 作为每个 OUT 参数的占位符。

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。例如，在 SQL 过程源文件 *whiles.db2* 中，以 DEPT\_MEDIAN 过程的 CREATE PROCEDURE 语句开头：

```
CREATE PROCEDURE DEPT_MEDIAN  
(IN deptNumber SMALLINT, OUT medianSalary DOUBLE)
```

要调用此过程，需要对 IN 参数 deptNumber 指定一个有效的 SMALLINT 值，并对 OUT 参数指定一个问号 “?”。DEPT\_MEDIAN 过程将存取样本数据库的 STAFF 表。变量 deptNumber 被指定给 STAFF 表的 DEPT 列，因此您可以从 DEPT 列中获取有效值：例如，值 “51”。

现在，可以输入带有过程名的 call 语句，并对 IN 参数指定了值且对 OUT 参数的值指定了一个问号 “?”。过程参数必须括在圆括号中，如下所示：

```
db2 call dept_median (51, ?)
```

**注:** 在 UNIX 平台上, 圆括号对于命令外壳程序具有特殊意义, 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: MEDIAN_SALARY  
参数值: +1.765450000000000E+004
```

B20000I SQL 命令成功完成。

#### 相关任务:

- 第 128 页的『在 UNIX 上使用客户机应用程序调用 SQL 过程』
- 第 129 页的『在 Windows 上使用客户机应用程序调用 SQL 过程』
- 第 134 页的『分发可编译 SQL 过程』

#### 相关样本:

- 『whiles.db2 -- To create the DEPT\_MEDIAN SQL procedure 』
- 『whiles.sqc -- To call the DEPT\_MEDIAN SQL procedure 』

---

## 第 3 部分 构建和运行特定于平台的应用程序





---

## 第 7 章 AIX

重要注意事项 . . . . .	141	使用配置文件构建 C++ 存储过程. . . . .	169
例程的 AIX 导出文件 . . . . .	142	使用配置文件构建 C++ 用户定义函数 . . . . .	170
AIX 例程和 CREATE 语句. . . . .	142	IBM COBOL Set AIX 版 . . . . .	171
替换 AIX 共享库. . . . .	143	在 AIX 上配置 IBM COBOL 编译器 . . . . .	171
在 AIX 上安装 COBOL 的注意事项. . . . .	143	在 AIX 上构建 IBM COBOL 应用程序 . . . . .	172
IBM C . . . . .	144	为 IBM COBOL 应用程序构建脚本. . . . .	174
在 AIX 上构建 C 应用程序 . . . . .	144	AIX IBM COBOL 应用程序编译和链接选	
为 C 应用程序构建脚本. . . . .	145	项 . . . . .	175
AIX C 应用程序编译和链接选项 . . . . .	147	在 AIX 上构建 IBM COBOL 例程 . . . . .	175
在 AIX 上构建 C 例程 . . . . .	148	为 IBM COBOL 例程构建脚本 . . . . .	177
为 C 例程构建脚本 . . . . .	151	AIX IBM COBOL 例程编译和链接选项 . . . . .	178
AIX C 例程编译和链接选项 . . . . .	152	Micro Focus COBOL . . . . .	179
在 AIX 上构建 C 多线程应用程序 . . . . .	153	在 AIX 上配置 Micro Focus COBOL 编译	
为 C 多线程应用程序构建脚本 . . . . .	154	器 . . . . .	179
VisualAge C++ . . . . .	155	在 AIX 上构建 Micro Focus COBOL 应用	
在 AIX 上构建 C++ 应用程序 . . . . .	155	程序 . . . . .	179
为 C++ 应用程序构建脚本. . . . .	157	为 Micro Focus COBOL 应用程序构建脚	
AIX C++ 应用程序编译和链接选项 . . . . .	158	本 . . . . .	181
在 AIX 上构建 C++ 例程 . . . . .	159	AIX Micro Focus COBOL 应用程序编译和	
为 C++ 例程构建脚本 . . . . .	162	链接选项. . . . .	182
AIX C++ 例程编译和链接选项 . . . . .	163	在 AIX 上构建 Micro Focus COBOL 例程 . . . . .	182
在 AIX 上构建 C++ 多线程应用程序 . . . . .	164	为 Micro Focus COBOL 例程构建脚本 . . . . .	184
为 C++ 多线程应用程序构建脚本. . . . .	165	AIX Micro Focus COBOL 例程编译和链接	
VisualAge C++ 配置文件 . . . . .	166	选项 . . . . .	185
使用配置文件构建 VisualAge C++ 程序 . . . . .	166	REXX . . . . .	186
使用配置文件构建 C++ DB2 API 应用程		在 AIX 上构建 REXX 应用程序. . . . .	186
序 . . . . .	167		
使用配置文件构建 C++ 嵌入式 SQL 应用			
程序 . . . . .	168		

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：

<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。

AIX 要求您提供导出文件，该文件指定库中的哪些全局函数是可从外部调用的。此文件必须包括库中所有例程的名称。其它 UNIX 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 142 页的『AIX 例程和 CREATE 语句』

## 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 142 页的『例程的 AIX 导出文件』

## 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。

## 在 AIX 上安装 COBOL 的注意事项

由于 AIX 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所在的路径完全不同。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 22 页的『设置 UNIX 应用程序开发环境』
- 第 171 页的『在 AIX 上配置 IBM COBOL 编译器』
- 第 179 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。

### 在 AIX 上构建 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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`，输入：

```
bldapp cli_info
```

产生可执行文件 `cli_info`。

要运行可执行文件，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，输入：

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名:

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码:

```
bldapp tbmod database userid password
```

产生可执行文件 `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样本数据库，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，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，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 89 页的『构建文件』

相关任务:

- 第 148 页的『在 AIX 上构建 C 例程』

相关参考:

- 第 146 页的『AIX C 应用程序编译和链接选项』
- 第 65 页的『C/C++ 样本』

相关样本:

- 『bldapp -- Builds AIX C application programs』
- 『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』
- 『tbmod.sqc -- How to modify table data (C)』

## 为 C 应用程序构建脚本

```
#!/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/sqllib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If embedded SQL program, precompile and bind it.
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	
编译选项:	
<b>xlc</b>	IBM C 编译器。
<b>\$EXTRA_CFLAG</b>	如果取消了 “BUILD_64BIT=true” 的注释，则包含值 “-q64” ；否则，它不包含任何值。
<b>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。例如: \$HOME/sqlllib/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。
链接选项:	
<b>xlc</b>	使用编译器作为链接程序的前端。
<b>\$EXTRA_CFLAG</b>	如果取消了 “BUILD_64BIT=true” 的注释，则包含值 “-q64” ；否则，它不包含任何值。
<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>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>-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/sqlllib/lib。如果不指定 -L 选项，则编译器将采用以下路径: /usr/lib:/lib。
有关其它编译程序选项，参考编译程序文档。	

**相关任务:**

- 第 144 页的『在 AIX 上构建 C 应用程序』

**相关参考:**

- 第 152 页的『AIX C 例程编译和链接选项』

**相关样本:**

- 『bldapp -- Builds AIX C application programs』

## 在 AIX 上构建 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，则该程序使用缺省的样本数据库。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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (UDF)

### 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至样本数据库，则输入构建脚本名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还须输入该数据库名:

```
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` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 编目了这些例程，则取消对它们的编目，然后通过调用 `spcreate.db2` 来编目它们，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. 最后停止并重新启动数据库以便可识别新的共享库。

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。该名称可以是样本，或它的别名，或另一个数据库名。

**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` 构建用户定义函数程序 `udfsrv`，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。该名称可以是样本，或它的别名，或另一个数据库名。

**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 用户定义函数程序 `udfemsrv`，如果连接至样本数据库，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，还须输入该数据库名：

```
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。该名称可以是样本，或它的别名，或另一个数据库名。

**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函数。然后将输出返回给该客户机应用程序。

### **相关概念:**

- 第 89 页的『构建文件』

### **相关任务:**

- 第 144 页的『在 AIX 上构建 C 应用程序』

### 相关参考:

- 第 152 页的『AIX C 例程编译和链接选项』
- 第 65 页的『C/C++ 样本』

### 相关样本:

- 『bldrtn -- Builds AIX C routines (stored procedures and UDFs)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spclient.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrow.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』
- 『spserver.sqc -- A variety of 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 -- Call a variety of types of user-defined functions (C)』

## 为 C 例程构建脚本

```
#!/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/sql/lib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
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>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>	如果取消了“BUILD_64BIT=true”的注释，则包含值“-q64”；否则，它不包含任何值。
<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。

bldrtn 的编译和链接选项	
链接选项:	
<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>	如果取消了“BUILD_64BIT=true”的注释，则包含值“-q64”；否则，它不包含任何值。
<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/sqlllib/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 148 页的『在 AIX 上构建 C 例程』

相关参考:

- 第 146 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)』

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 xlc\_r 编译器而不是 xlc 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。

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 sqlllib/samples/c 目录中。

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。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。参数 \$3 指定数据库的用户标识, 而 \$4 指定用户标识的密码。只有第一个参数即源文件名是必需的。数据库名、用户标识和密码是可选的。如果未提供数据库名, 则该程序使用缺省的样本数据库。

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 dbthrrs.sqc 构建多线程样本程序 dbthrrs, 输入:

```
bldmt dbthrrs
```

产生可执行文件 dbthrrs。要对样本数据库运行该可执行文件, 输入该可执行文件名称:

```
dbthrrs
```

### 相关概念:

- 第 89 页的『构建文件』

### 相关任务:

- 第 144 页的『在 AIX 上构建 C 应用程序』

### 相关参考:

- 第 146 页的『AIX C 应用程序编译和链接选项』
- 第 65 页的『C/C++ 样本』

### 相关样本:

- 『bldmt -- Builds AIX C multi-threaded applications』
- 『dbthrrs.sqc -- How to use multiple context APIs on UNIX (C)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

## 为 C 多线程应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldmt
# Builds AIX C multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib
```

```

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
xlc_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

# Link the program.
xlc_r $EXTRA_CFLAG -o $1 $1.o -L$DB2PATH/lib -ldb2

```

---

## VisualAge C++

### 在 AIX 上构建 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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`，输入：

```
bldapp cli_info
```

产生可执行文件 `cli_info`。可以输入如下命令，对样本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.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，输入：

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp tbmod database userid password
```

产生可执行文件 `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样本数据库，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，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，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 89 页的『构建文件』

### 相关任务：

- 第 159 页的『在 AIX 上构建 C++ 例程』

### 相关参考：

- 第 157 页的『AIX C++ 应用程序编译和链接选项』
- 第 65 页的『C/C++ 样本』

### 相关样本：

- 『`bldapp -- Builds AIX C++ applications`』
- 『`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 』

## 为 C++ 应用程序构建脚本

```
#!/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

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If an embedded SQL program, precompile and bind it.
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 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。

bldapp 的编译和链接选项	
编译选项:	
<b>x1C</b>	VisualAge C++ 编译器。
<b>EXTRA_CFLAG</b>	如果取消了“BUILD_64BIT=true”的注释, 则包含值“-q64”; 否则, 它不包含任何值。
<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。
链接选项:	
<b>x1C</b>	使用编译器作为链接程序的前端。
<b>EXTRA_CFLAG</b>	如果取消了“BUILD_64BIT=true”的注释, 则包含值“-q64”; 否则, 它不包含任何值。
<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。
有关其它编译程序选项, 参考编译程序文档。	

相关任务:

- 第 155 页的『在 AIX 上构建 C++ 应用程序』
- 第 168 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 167 页的『使用配置文件构建 C++ DB2 API 应用程序』

相关参考:

- 第 163 页的『AIX C++ 例程编译和链接选项』

相关样本:

- 『 bldapp -- Builds AIX C++ applications 』

## 在 AIX 上构建 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，则该程序使用缺省的样本数据库。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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (UDF)

### 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至样本数据库，则输入构建脚本名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还须输入该数据库名:

```
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` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 编目了这些例程，则取消对它们的编目，然后通过调用 `spcreate.db2` 来编目它们，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. 最后停止并重新启动数据库以便可识别新的共享库。

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。该名称可以是样本，或它的别名，或另一个数据库名。

**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` 构建用户定义函数程序 `udfsrv`，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。该名称可以是样本，或它的别名，或另一个数据库名。

**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 用户定义函数程序 `udfemsrv`，如果连接至样本数据库，则输入构建脚本名和程序名：

```
bldrtn udfemsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
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。该名称可以是样本，或它的别名，或另一个数据库名。

**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函数。然后将输出返回给该客户机应用程序。

#### 相关概念：

- 第 89 页的『构建文件』

#### 相关任务：

- 第 155 页的『在 AIX 上构建 C++ 应用程序』

#### 相关参考：

- 第 163 页的『AIX C++ 例程编译和链接选项』
- 第 65 页的『C/C++ 样本』

### 相关样本:

- 『 bldrtn -- Builds AIX C++ routines (stored procedures and UDFs) 』
- 『 spclient.sqC -- Call various stored procedures (C++) 』
- 『 spserver.sqC -- A variety of 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 -- Call a variety of types of user-defined functions (C++) 』
- 『 embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs 』
- 『 spcat -- To catalog C stored procedures on UNIX 』
- 『 spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc 』
- 『 spdrops.db2 -- How to uncatalog the stored procedures contained in spserver.sqc 』

### 为 C++ 例程构建脚本

```
#!/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

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2
fi
```

```
# Compile the program.
x1C_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.C

# Link using export file $1.exp, creating shared library $1
x1C_r $EXTRA_CFLAG -qmkshrobj -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>	如果取消了“BUILD_64BIT=true”的注释，则包含值“-q64”；否则，它不包含任何值。
<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。

bldrtn 的编译和链接选项	
链接选项:	
<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>	如果取消了“BUILD_64BIT=true”的注释，则包含值“-q64”；否则，它不包含任何值。
<b>-qmkshrobj</b>	创建共享库。
<b>-o \$1</b>	指定输出为共享库文件。
<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/sql/lib/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 DB2 库。
<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 148 页的『在 AIX 上构建 C 例程』
- 第 169 页的『使用配置文件构建 C++ 存储过程』
- 第 170 页的『使用配置文件构建 C++ 用户定义函数』

相关参考:

- 第 157 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)』

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 xlc\_r 编译器而不是 xlc 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。



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。

脚本 `bldmt` 包含构建多线程应用程序的命令。

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。参数 `$3` 指定数据库的用户标识，而 `$4` 指定用户标识的密码。只有第一个参数即源文件名是必需的。数据库名、用户标识和密码是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

除了以上讨论的 `xlc_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嵌入式 SQL 脚本文件 `bldapp` 所用的相同。

### 过程:

要从源文件 `dbthrrds.sqC` 构建多线程样本程序 `dbthrrds`，输入:

```
bldmt dbthrrds
```

产生可执行文件 `dbthrrds`。要对样本数据库运行该可执行文件，输入该可执行文件名称:

```
dbthrrds
```

### 相关概念:

- 第 89 页的『构建文件』

### 相关任务:

- 第 155 页的『在 AIX 上构建 C++ 应用程序』

### 相关参考:

- 第 157 页的『AIX C++ 应用程序编译和链接选项』
- 第 65 页的『C/C++ 样本』

### 相关样本:

- 『`bldmt` -- Builds AIX C++ multi-threaded applications』
- 『`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』

## 为 C++ 多线程应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldmt
# Builds AIX C++ multi-threaded applications
```

```
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqlllib

# Precompile and bind the program.
./embprep $1 $2 $3 $4

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# Compile the program.
xlc_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.C

# Link the program.
xlc_r $EXTRA_CFLAG -o $1 $1.o -L$DB2PATH/lib -ldb2
```

---

## 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。参见 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++ 用户定义函数

### 相关任务:

- 第 168 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 167 页的『使用配置文件构建 C++ DB2 API 应用程序』
- 第 169 页的『使用配置文件构建 C++ 存储过程』
- 第 170 页的『使用配置文件构建 C++ 用户定义函数』
- 第 144 页的『在 AIX 上构建 C 应用程序』
- 第 148 页的『在 AIX 上构建 C 例程』
- 第 155 页的『在 AIX 上构建 C++ 应用程序』
- 第 159 页的『在 AIX 上构建 C++ 例程』

## 使用配置文件构建 C++ DB2 API 应用程序

sqllib/samples/c 和 sqllib/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 外壳程序:

```
export API=cli_info
```
  - 对于 C 外壳程序:

```
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的 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。

产生可执行文件 cli\_info。可输入可执行文件名来运行该程序:

```
cli_info
```

### 相关任务:

- 第 168 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 169 页的『使用配置文件构建 C++ 存储过程』
- 第 170 页的『使用配置文件构建 C++ 用户定义函数』

## 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 tbmod.sqc 构建嵌入式 SQL 应用程序 tbmod, 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 外壳程序:
 

```
export EMB=tbmod
```
  - 对于 C 外壳程序:
 

```
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。

产生可执行文件 tbmod。可输入可执行文件名来运行该程序:

```
tbmod
```

### 相关任务:

- 第 167 页的『使用配置文件构建 C++ DB2 API 应用程序』
- 第 169 页的『使用配置文件构建 C++ 存储过程』
- 第 170 页的『使用配置文件构建 C++ 用户定义函数』

## 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 外壳程序:

```
export STP=spserver
```

- 对于 C 外壳程序:

```
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 脚本来编目共享库中的存储过程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 spdrops.db2 编目了这些存储过程，则取消对它们的编目，然后通过调用 spcreate.db2 来编目它们，最后与数据库断开连接。还可以分别调用 spdrops.db2 和 spcreate.db2 脚本。

最后停止并重新启动数据库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。 该名称可以是样本、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 168 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 167 页的『使用配置文件构建 C++ DB2 API 应用程序』
- 第 170 页的『使用配置文件构建 C++ 用户定义函数』

## 使用配置文件构建 C++ 用户定义函数

`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++ 构建用户定义函数。

**过程:**

要从源文件 `udfsrv.c` 使用配置文件构建用户定义函数程序 `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` 外壳程序:

```
export UDF=udfsrv
```
  - 对于 C 外壳程序:

```
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: vacbld 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 168 页的『使用配置文件构建 C++ 嵌入式 SQL 应用程序』
- 第 167 页的『使用配置文件构建 C++ DB2 API 应用程序』
- 第 169 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/sqlllib/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 143 页的『在 AIX 上安装 COBOL 的注意事项』

#### 相关任务:

- 第 22 页的『设置 UNIX 应用程序开发环境』
- 第 172 页的『在 AIX 上构建 IBM COBOL 应用程序』
- 第 175 页的『在 AIX 上构建 IBM COBOL 例程』

## 在 AIX 上构建 IBM COBOL 应用程序

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 `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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`，输入：

```
bldapp client
```

产生可执行文件 `client`。可以输入如下命令，对样本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，输入：

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp updat database userid password
```

产生可执行文件 `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样本数据库，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，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，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 89 页的『构建文件』

#### 相关任务：

- 第 175 页的『在 AIX 上构建 IBM COBOL 例程』

#### 相关参考：

- 第 174 页的『AIX IBM COBOL 应用程序编译和链接选项』
- 第 75 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) 』

## 为 IBM COBOL 应用程序构建脚本

```
#!/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/sqlllib

# 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 AIX IBM 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。

bldapp 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>-I\$DB2PATH/include/cobol_a</b>	指定 DB2 包含文件的位置。例如: \$HOME/sqlllib/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。
链接选项:	
<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/sqlllib/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 172 页的『在 AIX 上构建 IBM COBOL 应用程序』

相关参考:

- 第 177 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 sqlllib/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。数据库名是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

### 过程:

要从源文件 outsrv.sqb 构建样本程序 outsrv，连接样本程序而后输入:

```
bldrtn outsrv
```

如果连接另一个数据库，还须输入该数据库名:

```
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。该名称可以是样本、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 89 页的『构建文件』

### 相关任务:

- 第 172 页的『在 AIX 上构建 IBM COBOL 应用程序』

### 相关参考:

- 第 177 页的『AIX IBM COBOL 例程编译和链接选项』
- 第 75 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)』

## 为 IBM COBOL 例程构建脚本

```
#!/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。本书假定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/sqlllib/include/cobol_a。
链接选项:	
<b>cob2</b>	使用编译器来链接编辑。
<b>-o \$1</b>	指定输出为共享库文件。
<b>\$1.o</b>	指定存储过程的对象文件。
<b>checkerr.o</b>	包括该实用程序的对象文件以便检查错误。
<b>-bnoentry</b>	不要指定共享库的缺省入口点。
<b>-bE:\$1.exp</b>	指定导出文件。该导出文件包含存储过程的列表。
<b>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。例如: \$HOME/sqlllib/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 例程』

相关参考:

- 第 174 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)』

### 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 `sqllib/include/cobol_mf` 中。

要包括此目录，输入:

- 在 `bash` 或 `Korn` 外壳程序上:

```
export COBCPY=$COBCPY:$HOME/sqllib/include/cobol_mf
```

- 在 `C` 外壳程序上:

```
setenv COBCPY $COBCPY:$HOME/sqllib/include/cobol_mf
```

注: 您可能想要在 `.profile` 或 `.login` 文件中设置 `COBCPY`。

相关概念:

- 第 143 页的『在 AIX 上安装 COBOL 的注意事项』

相关任务:

- 第 22 页的『设置 UNIX 应用程序开发环境』
- 第 179 页的『在 AIX 上构建 Micro Focus COBOL 应用程序』
- 第 182 页的『在 AIX 上构建 Micro Focus COBOL 例程』

### 在 AIX 上构建 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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`，输入：

```
bldapp client
```

产生可执行文件 `client`。可以输入如下命令，对样本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，输入：

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp updat database userid password
```

产生可执行文件 `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样本数据库，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，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，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 89 页的『构建文件』

#### 相关任务:

- 第 182 页的『在 AIX 上构建 Micro Focus COBOL 例程』

#### 相关参考:

- 第 182 页的『AIX Micro Focus COBOL 应用程序编译和链接选项』
- 第 75 页的『COBOL 样本』

#### 相关样本:

- 『bldapp -- Builds AIX 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』

### 为 Micro Focus COBOL 应用程序构建脚本

```
#!/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/sqllib

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
    embprep $1 $2 $3 $4
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# 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。

bldapp 的编译和链接选项	
编译选项:	
cob	COBOL 编译器。
-c	只执行编译；不链接。
-x	当与 -c 配合使用时，产生对象文件。
链接选项:	
cob	使用编译器作为链接程序的前端。
-x	产生可执行程序。
-o \$1	指定可执行程序。
\$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/sqllib/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。
-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，参考编译器文档。	

## 相关任务:

- 第 179 页的『在 AIX 上构建 Micro Focus COBOL 应用程序』

## 相关参考:

- 第 184 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』

# 在 AIX 上构建 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。数据库名是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

### 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，还须输入该数据库名：

```
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。该名称可以是样本，或它的别名，或另一个数据库名。

**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。

### 相关概念:

- 第 89 页的『构建文件』

### 相关任务:

- 第 179 页的『在 AIX 上构建 Micro Focus COBOL 应用程序』

#### 相关参考:

- 第 184 页的『AIX Micro Focus COBOL 例程编译和链接选项』
- 第 75 页的『COBOL 样本』

#### 相关样本:

- 『bldrtn -- Builds AIX 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』

## 为 Micro Focus COBOL 例程构建脚本

```
#!/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

# Precompile and bind the program.
./embprep $1 $2

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# 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	
编译选项:	
<b>cob</b>	COBOL 编译器。
<b>-c</b>	只执行编译; 不链接。本书假定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。
链接选项:	
<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/sql/lib/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 182 页的『在 AIX 上构建 Micro Focus COBOL 例程』

**相关参考:**

- 第 182 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
- 单继承和多继承

Object REXX 与 经典 REXX 完全兼容。在本节中，只要提到 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 外壳程序，输入：

```
export LIBPATH=$LIBPATH:/lib:/usr/lib:/usr/opt/db2_08_01/lib
```

对于 C 外壳程序，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. 在外壳程序命令提示符下，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，则可在外壳程序命令提示符下输入 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，在外壳程序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
```

在外壳程序命令提示符下输入 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`，输入：

```
updat.cmd
```

**相关任务:**

- 第 22 页的『设置 UNIX 应用程序开发环境』

**相关参考:**

- 第 86 页的『REXX 样本』





---

## 第 8 章 HP-UX

HP-UX C . . . . .	189	在 HP-UX 上构建 C++ 多线程应用程序 . . . . .	210
在 HP-UX 上构建 C 应用程序 . . . . .	189	为 C++ 多线程应用程序构建脚本. . . . .	211
为 C 应用程序构建脚本. . . . .	191	Micro Focus COBOL . . . . .	212
HP-UX C 应用程序编译和链接选项 . . . . .	192	在 HP-UX 上配置 Micro Focus COBOL	
在 HP-UX 上构建 C 例程 . . . . .	193	编译器 . . . . .	212
为 C 例程构建脚本 . . . . .	197	在 HP-UX 上构建 Micro Focus COBOL	
HP-UX C 例程编译和链接选项 . . . . .	198	应用程序. . . . .	213
在 HP-UX 上构建 C 多线程应用程序 . . . . .	199	为 Micro Focus COBOL 应用程序构建脚	
为 C 多线程应用程序构建脚本 . . . . .	200	本 . . . . .	215
HP-UX C++. . . . .	200	HP-UX Micro Focus COBOL 应用程序编	
在 HP-UX 上构建 C++ 应用程序. . . . .	200	译和链接选项 . . . . .	216
为 C++ 应用程序构建脚本. . . . .	202	在 HP-UX 上构建 Micro Focus COBOL	
HP-UX C++ 应用程序编译和链接选项 . . . . .	203	例程 . . . . .	216
在 HP-UX 上构建 C++ 例程 . . . . .	204	为 Micro Focus COBOL 例程构建脚本 . . . . .	218
为 C++ 例程构建脚本 . . . . .	208	HP-UX Micro Focus COBOL 例程编译和	
HP-UX C++ 例程编译和链接选项. . . . .	209	链接选项. . . . .	219

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：

<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。

#### 在 HP-UX 上构建 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。

构建脚本 `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，因此还须提供三个可选参数：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, 则使用缺省样本数据库。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`, 输入:

```
bldapp cli_info
```

产生可执行文件 `cli_info`。

要运行可执行文件, 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, 输入:

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接, 还须输入该数据库名:

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接, 还须输入该数据库实例的用户标识和密码:

```
bldapp tbmod database userid password
```

产生可执行文件 `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样本数据库, 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, 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, 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 89 页的『构建文件』

#### 相关参考:

- 第 65 页的『C/C++ 样本』
- 第 192 页的『HP-UX C 应用程序编译和链接选项』

#### 相关样本:

- 『bldapp -- Builds HP-UX C applications』
- 『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』
- 『tbmod.sqc -- How to modify table data (C)』

### 为 C 应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds HP-UX C applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
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 $1.o utilemb.o -L$DB2PATH/lib -ldb2
else
# Link the program with utilapi.o.
cc $EXTRA_CFLAG -o $1 $1.o utilapi.o -L$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:	
cc	C 编译器。
\$EXTRA_CFLAG	包含不同的值，这取决于是否启用了 32 位还是 64 位支持。对于 32 位，值为 +DAportable；对于 64 位，值为 +DA2.0W。
+DAportable (仅适用于 32 位)	生成跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
+DA2.0W (仅适用于 64 位)	必须用来生成 64 位代码。
-Ae	启用 HP ANSI 扩展方式。
-I\$DB2PATH/include	指定 DB2 包含文件的位置。
-c	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	
<b>链接选项:</b>	
<b>cc</b>	将编译器用作链接程序的前端。
<b>\$EXTRA_CFLAG</b>	包含不同的值, 这取决于是否启用了 32 位还是 64 位支持。对于 32 位, 值为 +DAportable; 对于 64 位, 值为 +DA2.0W。
<b>+DAportable (仅适用于 32 位)</b>	使用跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性, 则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码, 但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
<b>+DA2.0W (仅适用于 64 位)</b>	必须用来生成 64 位代码。
<b>-o \$1</b>	指定可执行文件。
<b>\$1.o</b>	指定该程序的对象文件。
<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>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。例如: -L\$DB2PATH/lib。如果不指定 -L 选项, 则假定为 /usr/lib:/lib。
<b>-ldb2</b>	链接 DB2 库。
有关其它编译程序选项, 参考编译程序文档。	

### 相关任务:

- 第 189 页的『在 HP-UX 上构建 C 应用程序』

### 相关样本:

- 『bldapp -- Builds HP-UX C applications』

## 在 HP-UX 上构建 C 例程

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。

脚本文件 bldrtn 包含构建例程（存储过程和用户定义函数）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，则该程序使用缺省的样本数据库。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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (UDF)

### 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至样本数据库，则输入构建文件名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还须输入该数据库名:

```
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` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 编目了这些例程，则取消对它们的编目，然后通过调用 `spcreate.db2` 来编目它们，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. 最后停止并重新启动数据库以便可识别新的共享库。

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。该名称可以是样本，或它的别名，或另一个数据库名。

**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。

### 非嵌入式 SQL UDF 共享库

要从源文件 **udfsrv.c** 构建用户定义函数程序 **udfsrv**，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。

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** 通过使用脚本文件 **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** 通过使用脚本文件 **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。该名称可以是样本，或它的别名，或另一个数据库名。

**userid** 是有效的用户标识。

**password**

是有效的密码。

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 用户定义函数程序 **udfemsrv**，如果连接至样本数据库，则输入构建脚本名和程序名：

```
bldrtn udfemsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
bldrtn udfemsrv database
```

该脚本文件将 UDF 库复制到 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 通过使用脚本文件 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。该名称可以是样本，或它的别名，或另一个数据库名。

**userid** 是有效的用户标识。

**password**

是有效的密码。

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函数。然后将输出返回给该客户机应用程序。

**相关概念：**

- 第 89 页的『构建文件』

**相关参考：**

- 第 65 页的『C/C++ 样本』
- 第 198 页的『HP-UX C 例程编译和链接选项』

**相关样本：**

- 『bldrtn -- Builds HP-UX C routines (stored procedures and UDFs)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spclient.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』



- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』
- 『spserver.sqc -- A variety of 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 -- Call a variety of types of user-defined functions (C)』

## 为 C 例程构建脚本

```

#!/bin/sh
# SCRIPT: bldrtn
# Builds HP-UX C routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
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 $1.o -L$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>\$EXTRA_CFLAG</b>	包含不同的值，这取决于是否启用了 32 位还是 64 位支持。对于 32 位，值为 +DAportable; 对于 64 位，值为 +DA2.0W。
<b>+DAportable (仅适用于 32 位)</b>	生成跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
<b>+DA2.0W (仅适用于 64 位)</b>	必须用来生成 64 位代码。
<b>+u1</b>	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。
<b>+z</b>	生成与位置无关的代码。
<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。例如: -I\$DB2PATH/include。
<b>-D_POSIX_C_SOURCE=199506L</b>	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）。
<b>-c</b>	只执行编译；不链接。本书假定编译和链接是两个独立的步骤。
<b>链接选项:</b>	
<b>ld</b>	使用链接程序进行链接。
<b>-b</b>	创建一个共享库而不是通常的可执行文件。
<b>-o \$1</b>	指定输出为共享库文件。
<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/sql/lib/lib。如果不指定 -L 选项，则假定为 /usr/lib:/lib。
<b>-ldb2</b>	链接 DB2 库。
<b>-lpthread</b>	与 POSIX 线程库进行链接。
有关其它编译程序选项，参考编译程序文档。	

#### 相关任务:

- 第 193 页的『在 HP-UX 上构建 C 例程』

#### 相关样本:

- 『bldrtn -- Builds HP-UX C routines (stored procedures and UDFs)』

## 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。

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。

脚本文件 `bldmt` 包含构建多线程应用程序的命令。它使用多达四个参数，这些参数以变量 `$1`、`$2`、`$3` 和 `$4` 表示。

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。第三个参数 `$3` 指定数据库的用户标识，而第四个参数 `$4` 指定密码。只有第一个参数即源文件名是必需的。数据库名、用户标识和密码是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

#### 过程:

要从源文件 `dbthrds.sqc` 构建样本程序 `dbthrds`，输入：

```
bldmt dbthrds
```

产生可执行文件 `dbthrds`。要对样本数据库运行该可执行文件，输入该可执行文件名称：

```
dbthrds
```

#### 相关概念:

- 第 89 页的『构建文件』

#### 相关参考:

- 第 65 页的『C/C++ 样本』

#### 相关样本:

- 『bldmt -- Builds HP-UX C multi-threaded applications』

- 『dbthrd.sqc -- How to use multiple context APIs on UNIX (C)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

## 为 C 多线程应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldmt
# Builds HP-UX C multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# To compile 64-bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -D_POSIX_C_SOURCE=199506L -c $1.c

# Link the program
cc $EXTRA_CFLAG -o $1 $1.o -L$DB2PATH/lib -ldb2 -lpthread
```

---

## HP-UX C++

### 在 HP-UX 上构建 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。

构建脚本 `bldapp` 包含构建 DB2 API 和嵌入式 SQL 应用程序的命令。该脚本使用多达四个参数。

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，因此还须

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，则使用缺省样本数据库。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，输入：

```
bldapp cli_info
```

产生可执行文件 cli\_info。可以输入如下命令，对样本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.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，输入：

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp tbmod database userid password
```

产生可执行文件 tbmod。

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样本数据库，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，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，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 89 页的『构建文件』

#### 相关参考:

- 第 65 页的『C/C++ 样本』
- 第 203 页的『HP-UX C++ 应用程序编译和链接选项』

#### 相关样本:

- 『bldapp -- Builds HP-UX C++ applications』
- 『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』

### 为 C++ 应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldapp
# Builds HP-UX C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
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 $1.o utilemb.o -L$DB2PATH/lib -ldb2
else
# Link the program with utilapi.o.
aCC $EXTRA_CFLAG -o $1 $1.o utilapi.o -L$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	包含不同的值，这取决于是否启用了 32 位还是 64 位支持。对于 32 位，值为 +DAportable；对于 64 位，值为 +DA2.0W。
+DAportable (仅适用于 32 位)	生成跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
+DA2.0W (仅适用于 64 位)	必须用来生成 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。本书假定编译和链接是两个独立的步骤。

bldapp 的编译和链接选项	
<b>链接选项:</b>	
<b>aCC</b>	将 HP aC++ 编译器用作链接程序的前端。
<b>\$EXTRA_CFLAG</b>	包含不同的值，这取决于是否启用了 32 位还是 64 位支持。对于 32 位，值为 +DAportable; 对于 64 位，值为 +DA2.0W。
<b>+DAportable (仅适用于 32 位)</b>	使用跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
<b>+DA2.0W (仅适用于 64 位)</b>	必须用来生成 64 位代码。
<b>-o \$1</b>	指定可执行文件。
<b>\$1.o</b>	指定该程序的对象文件。
<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>-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 200 页的『在 HP-UX 上构建 C++ 应用程序』

**相关样本:**

- 『bldapp -- Builds HP-UX C++ applications』

**在 HP-UX 上构建 C++ 例程**

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 sql1lib/samples/cpp 目录中。

脚本 bldrtn 包含构建例程（存储过程或用户定义函数）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。



数据库参数是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。并且，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。

### 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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (UDF)

### 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至样本数据库，则输入构建文件名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还须输入该数据库名:

```
bldrtn spserver database
```

该脚本将存储过程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` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 编目了这些例程，则取消对它们的编目，然后通过调用 `spcreate.db2` 来编目它们，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. 最后停止并重新启动数据库以便可识别新的共享库。

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`。

要调用存储过程，可输入如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。该名称可以是样本，或它的别名，或另一个数据库名。

**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** 构建用户定义函数程序 **udfsrv**，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。

一旦构建了 **udfsrv**，就可构建调用它的客户机应用程序 **udfcli**。可从源文件 **udfcli.sqC** 通过使用脚本 **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。该名称可以是样本，或它的别名，或另一个数据库名。

**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 用户定义函数程序 **udfemsrv**，如果连接样本数据库，则输入构建脚本名和程序名：

```
bldrtn udfemsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
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 `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。该名称可以是样本，或它的别名，或另一个数据库名。

**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函数。然后将输出返回给该客户机应用程序。

**相关概念：**

- 第 89 页的『构建文件』

**相关参考：**

- 第 65 页的『C/C++ 样本』
- 第 208 页的『HP-UX C++ 例程编译和链接选项』

**相关样本：**

- 『`bldrtn` -- Builds HP-UX C++ routines (stored procedures and UDFs)』
- 『`spclient.sqC` -- Call various stored procedures (C++)』
- 『`spserver.sqC` -- A variety of 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` -- Call a variety of types of user-defined functions (C++)』
- 『`embprep` -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

- 『spcat -- To catalog C stored procedures on UNIX』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』

## 为 C++ 例程构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds HP-UX C++ routines (stored procedures and UDFs)
# Usage: bldsrv <prog_name> [ <db_name> ]

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
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 $1.o -L$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>aCC</b>	HP aC++ 编译器。
<b>\$EXTRA_CFLAG</b>	包含不同的值，这取决于是否启用了 32 位还是 64 位支持。对于 32 位，值为 +DAportable; 对于 64 位，值为 +DA2.0W。
<b>+DAportable (仅适用于 32 位)</b>	生成跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
<b>+DA2.0W (仅适用于 64 位)</b>	必须用来生成 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。本书假定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:	
<b>aCC</b>	将 HP aC++ 编译器用作链接程序的前端。
<b>\$EXTRA_CFLAG</b>	包含不同的值，这取决于是否启用了 32 位还是 64 位支持。对于 32 位，值为 +DAportable; 对于 64 位，值为 +DA2.0W。
<b>+DAportable</b> (仅适用于 32 位)	使用跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
<b>+DA2.0W</b> (仅适用于 64 位)	必须用来生成 64 位代码。
<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>-b</b>	创建一个共享库而不是通常的可执行文件。
<b>-o \$1</b>	指定可执行文件。
<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: -L\$DB2PATH/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 204 页的『在 HP-UX 上构建 C++ 例程』

相关样本:

- 『bldrtn -- Builds HP-UX C++ routines (stored procedures and UDFs)』

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。

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。

脚本 `bldmt` 包含构建多线程应用程序的命令。它使用多达四个参数，这些参数以变量 `$1`、`$2`、`$3` 和 `$4` 表示。

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。第三个参数 `$3` 指定数据库的用户标识，而第四个参数 `$4` 指定密码。只有第一个参数即源文件名是必需的。数据库名、用户标识和密码是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

### 过程:

要从源文件 `dbthrrds.sqlC` 构建样本程序 `dbthrrds`，输入:

```
bldmt dbthrrds
```

产生可执行文件 `dbthrrds`。要对样本数据库运行该可执行文件，输入该可执行文件名称:

```
dbthrrds
```

### 相关概念:

- 第 89 页的『构建文件』

### 相关参考:

- 第 65 页的『C/C++ 样本』

### 相关样本:

- 『`bldmt -- Builds HP-UX C++ multi-threaded applications`』
- 『`dbthrrds.sqlC -- How to use multiple context APIs on UNIX (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

## 为 C++ 多线程应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldmt
# Builds HP-UX C++ multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true
```

```

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
aCC $EXTRA_CFLAG -ext -mt -I$DB2PATH/include -c $1.C

# Link the program
aCC $EXTRA_CFLAG -mt -o $1 $1.o -L$DB2PATH/lib -ldb2

```

---

## Micro Focus COBOL

### 在 HP-UX 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。
- 为了使用内置预编译器前端、运行时解释器或动画制作程序调试器，则必须通过执行 Micro Focus 提供的 `mkrts` 命令来将“DB2 类属 API”入口点添加至 Micro Focus 运行时模块 `rts32`。您还需运行 `mkcheck` 来更新 `check` 文件。如果未运行此文件，则您将在 `SQLGSTRT` 中接收到 173 错误。

在构建您的 COBOL 应用程序之前，必须运行 `mkrts` 和 `mkcheck` 来设置您的环境。在运行 `mkrts` 和 `mkcheck` 之前，必须在下列步骤中设置 COBOPT:

1. 作为 root 用户登录。
2. 在目录 `$COBDIR/src/rts` 下输入:

```

COBOPT=/opt/IBM/db2/V8.1/lib/db2mkrts.args; export COBOPT
ksh mkrts
mv $COBDIR/rts32 $COBDIR/rts32.orig
cp rts32 $COBDIR/rts32

```

其中 `$COBDIR` 是安装有 COBOL 编译器的目录，通常为 `/opt/cobol/cobdir`。



3. 还必须重新构建 Hewlett-Packard 公司在交付本产品时提供的 check 可执行文件。如果未重新构建位于 \$COBDIR 目录中的 check 可执行文件, 则使用 cob -C SQL 所进行的编译尝试将失败, 并收到一个运行时系统 173 错误, 这是因为 DB2 预处理器要调用 DB2 库。要重新构建 check 文件, 应以 root 用户的身份切换到 \$COBDIR 目录下的 src/sql 目录中, 然后运行 mkcheck 脚本。该脚本运行完毕后, 需将生成的 check 可执行文件移动到 \$COBDIR 目录下。在 \$COBDIR/src/sql 目录下输入:

```
COBOPT=/opt/IBM/db2/V8.1/lib/db2mkrts.args; export COBOPT
ksh mkcheck
mv $COBDIR/check $COBDIR/check.orig
cp check $COBDIR/check
```

现在, 可带以下目录中提供的自变量执行 mkrts 命令:

```
/opt/IBM/db2/V8.1/lib/db2mkrts.args
```

- 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 外壳程序上, 输入:

```
export COBCPY=$COBCPY:$HOME/sqllib/include/cobol_mf
```

- 在 C 外壳程序上, 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 216 页的『在 HP-UX 上构建 Micro Focus COBOL 例程』
- 第 213 页的『在 HP-UX 上构建 Micro Focus COBOL 应用程序』

## 在 HP-UX 上构建 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。

构建脚本 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，因此还须提供三个可选参数：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文件 embprep。如果未提供数据库名，则使用缺省样本数据库。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，输入：

```
bldapp client
```

产生可执行文件 client。可以输入如下命令，对样本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，输入：

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp updat database userid password
```

产生可执行文件 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样本数据库，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，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，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 89 页的『构建文件』

#### 相关参考:

- 第 215 页的『HP-UX Micro Focus COBOL 应用程序编译和链接选项』
- 第 75 页的『COBOL 样本』

#### 相关样本:

- 『bldapp -- Builds HP-UX 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』

### 为 Micro Focus COBOL 应用程序构建脚本

```
#!/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/sqllib

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
    embprep $1 $2 $3 $4
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the checkerr.cbl error checking utility.
cob +DAportable -cx checkerr.cbl

# Compile the program.
cob +DAportable -cx $1.cbl

# Link the program.
cob +DAportable -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。

bldapp 的编译和链接选项	
编译选项:	
<b>cob</b>	Micro Focus COBOL 编译器。
<b>+DAportable</b>	生成跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
<b>-cx</b>	编译为对象模块。
链接选项:	
<b>cob</b>	使用编译器作为链接程序的前端。
<b>+DAportable</b>	使用跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
<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 213 页的『在 HP-UX 上构建 Micro Focus COBOL 应用程序』

相关样本:

- 『bldapp -- Builds HP-UX Micro Focus COBOL applications』

在 HP-UX 上构建 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 `sql1lib/samples/cobol_mf` 目录中。

脚本 `bldrtn` 包含构建例程（存储过程）的命令。该脚本将例程编译成服务器上可由客户机应用程序调用的共享库。它使用一或两个参数，这些参数在 `bldrtn` 中以变量 `$1` 和 `$2` 表示。

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。数据库名是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。该脚本将源文件名 `$1` 用作共享库名。

### 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，还须输入该数据库名:

```
bldrtn outsrv database
```

该脚本将共享库复制到 `sqllib/function` 目录中。

一旦构建了共享库 `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，通过输入以下命令来运行客户机应用程序:

```
outcli database userid password
```

其中，

#### **database**

是要连接的数据库的名称。该名称可以是样本，或它的别名，或另一个数据库名。

**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`，它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 89 页的『构建文件』

### 相关参考:

- 第 218 页的『HP-UX Micro Focus COBOL 例程编译和链接选项』
- 第 75 页的『COBOL 样本』

#### 相关样本:

- 『bldrtn -- Builds HP-UX 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』

### 为 Micro Focus COBOL 例程构建脚本

```
#!/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

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
embprep $1 $2
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the program.
cob +DAportable +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.
# 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>+DAportable</b>	生成跨 PA_RISC 1.x 和 2.0 工作站以及服务器兼容的代码。如果考虑到应用程序的可移植性，则应使用此选项。在不使用此选项的情况下进行构建将生成对构建机器的处理器级别较好地进行了优化的代码，但对较旧的 PA-RISC 处理器级别则不起作用。参见编译器文档以获取更多信息。
<b>+z</b>	生成与位置无关的代码。
<b>-cx</b>	编译为对象模块。
链接选项:	
<b>ld</b>	使用链接程序进行链接。
<b>-b</b>	创建一个共享库而不是通常的可执行文件。
<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>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。
<b>-ldb2</b>	与 DB2 共享库链接。
<b>-ldb2gmf</b>	与 Micro Focus COBOL 的 DB2 异常处理程序库链接。
<b>-L\$COBDIR/coblib</b>	指定 COBOL 运行时库的位置。
<b>-lcobol</b>	链接 COBOL 库。
<b>-lcrtn</b>	链接 crt 库。
有关其它编译器选项，参考编译器文档。	

相关任务:

- 第 216 页的『在 HP-UX 上构建 Micro Focus COBOL 例程』

相关样本:

- 『bldrtn -- Builds HP-UX Micro Focus COBOL routines (stored procedures)』





---

# 第 9 章 Linux

Linux C . . . . .	221	Linux C++ . . . . .	232
在 Linux 上构建 C 应用程序 . . . . .	221	在 Linux 上构建 C++ 应用程序 . . . . .	232
为 C 应用程序构建脚本 . . . . .	223	为 C++ 应用程序构建脚本 . . . . .	233
Linux C 应用程序编译和链接选项 . . . . .	224	Linux C++ 应用程序编译和链接选项 . . . . .	234
在 Linux 上构建 C 例程 . . . . .	225	在 Linux 上构建 C++ 例程 . . . . .	235
为 C 例程构建脚本 . . . . .	228	为 C++ 例程构建脚本 . . . . .	239
Linux C 例程编译和链接选项 . . . . .	229	Linux C++ 例程编译和链接选项 . . . . .	240
在 Linux 上构建 C 多线程应用程序 . . . . .	230	在 Linux 上构建 C++ 多线程应用程序 . . . . .	241
为 C 多线程应用程序构建脚本 . . . . .	231	为 C++ 多线程应用程序构建脚本 . . . . .	241

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：

<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。

### 在 Linux 上构建 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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`, 输入:

```
bldapp cli_info
```

产生可执行文件 `cli_info`。

要运行可执行文件, 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, 输入:

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接, 还须输入该数据库名:

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接, 还须输入该数据库实例的用户标识和密码:

```
bldapp tbmod database userid password
```

产生可执行文件 `tbmod`。

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样本数据库, 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, 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, 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 89 页的『构建文件』

### 相关参考:

- 第 65 页的『C/C++ 样本』
- 第 224 页的『Linux C 应用程序编译和链接选项』

### 相关样本:

- 『`bldapp -- Builds Linux C applications`』

- 『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』
- 『tbmod.sqc -- How to modify table data (C)』

## 为 C 应用程序构建脚本

```
#!/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/sql1ib

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.c error-checking utility.
    gcc -I$DB2PATH/include -c utilemb.c
else
    # Compile the utilapi.c error-checking utility.
    gcc -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
gcc -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
    # Link the program with utilemb.o.
    gcc -o $1 $1.o utilemb.o $EXTRA_LFLAG \
        -L$DB2PATH/lib -ldb2
else
    # Link the program with utilapi.o.
    gcc -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>-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。
链接选项:	
<b>gcc</b>	将编译程序用作链接程序的前端。
<b>-o \$1</b>	指定可执行文件。
<b>\$1.o</b>	指定对象文件。
<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 “RUNTIME=true” 未注释，则包含值 “-Wl,-rpath,\$DB2PATH/lib” ；否则，不包含任何值。
<b>-Wl,-rpath,\$DB2PATH/lib</b>	指定运行时 DB2 共享库的位置。例如：\$HOME/sql1lib/lib。
<b>-L\$DB2PATH/lib</b>	指定链接时 DB2 共享库的位置。例如：\$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 221 页的『在 Linux 上构建 C 应用程序』

### 相关样本:

- 『bldapp -- Builds Linux C applications』

## 在 Linux 上构建 C 例程

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。

脚本 `bldrtn` 包含构建例程（存储过程或用户定义函数）的命令。该脚本文件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。

数据库参数是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (UDF)

### 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至样本数据库，则输入构建脚本名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还须输入该数据库名:

```
bldrtn spserver database
```

该脚本将存储过程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` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 编目了这些例程，则取消对它们的编目，然后通过调用 `spcreate.db2` 来编目它们，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. 最后停止并重新启动数据库以便可识别新的共享库。

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`。

要调用存储过程，可输入如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。该名称可以是样本，或它的别名，或另一个数据库名。

**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` 构建用户定义函数程序 `udfsrv`，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。该名称可以是样本，或它的别名，或另一个数据库名。

**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 用户定义函数程序 `udfemsrv`，如果连接至样本数据库，则输入构建脚本名和程序名：

```
bldrtn udfemsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
bldrtn udfemsrv database
```

该脚本文件将 UDF 库复制到 `sqllib/function` 目录中。

如有必要，对 UDF 设置文件方式，以便客户机应用程序数据库管理器可以存取它。

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` 通过使用脚本文件 `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。该名称可以是样本，或它的别名，或另一个数据库名。

**userid** 是有效的用户标识。

### **password**

是有效的密码。

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函数。然后将输出返回给该客户机应用程序。

### 相关概念:

- 第 89 页的『构建文件』

### 相关参考:

- 第 65 页的『C/C++ 样本』

- 第 229 页的『Linux C 例程编译和链接选项』

#### 相关样本:

- 『bldrtn -- Builds Linux C routines (stored procedures or UDFs)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spclient.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』
- 『spserver.sqc -- A variety of 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 -- Call a variety of types of user-defined functions (C)』

### 为 C 例程构建脚本

```
#!/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/sql/lib

# Set the runtime path since routines run as setuid.
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 -fpic -I$DB2PATH/include -c $1.c -D_REENTRANT

# Link the program and create a shared library
gcc -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>-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:	
<b>gcc</b>	将编译程序用作链接程序的前端。
<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>	包含值 “-Wl,-rpath,\$DB2PATH/lib” 以设置运行时路径，因为例程是作为 setuid 运行的。
<b>-Wl, -rpath, \$DB2PATH/lib</b>	指定运行时 DB2 共享库的位置。例如: \$HOME/sql1lib/lib。
<b>-L\$DB2PATH/lib</b>	指定链接时 DB2 共享库的位置。例如: \$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。
<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 225 页的『在 Linux 上构建 C 例程』

相关样本:

- 『bldrtn -- Builds Linux C routines (stored procedures or UDFs)』

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 -lpthread 链接。

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 sql1lib/samples/c 目录中。

脚本 bldmt 包含构建多线程应用程序的命令。

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。第三个参数 \$3 指定数据库的用户标识，而第四个参数 \$4 指定密码。只有第一个参数即源文件名是必需的。数据库名、用户标识和密码是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

### 过程:

要从源文件 dbthrrds.sqc 构建样本程序 dbthrrds，输入:

```
bldmt dbthrrds
```

产生可执行文件 dbthrrds。要对样本数据库运行该可执行文件，输入:

```
dbthrrds
```

### 相关概念:

- 第 89 页的『构建文件』

### 相关参考:

- 第 65 页的『C/C++ 样本』

### 相关样本:

- 『bldmt -- Builds Linux C multi-threaded applications』
- 『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』

## 为 C 多线程应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldmt
# Builds Linux C multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi
```

```
# If an embedded SQL program, precompile and bind it.
if [ -f "$1.sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
gcc -I$DB2PATH/include -c $1.c -D_REENTRANT

# Link the program.
gcc -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2 -lpthread
```

---

## Linux C++

### 在 Linux 上构建 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。

`sqllib/samples/cpp` 中的构建脚本 `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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`，输入：

```
bldapp cli_info
```

产生可执行文件 `cli_info`。可以输入如下命令，对样本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.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, 输入:

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接, 还须输入该数据库名:

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接, 还须输入该数据库实例的用户标识和密码:

```
bldapp tbmod database userid password
```

产生可执行文件 `tbmod`。

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样本数据库, 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, 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, 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 89 页的『构建文件』

相关参考:

- 第 65 页的『C/C++ 样本』
- 第 234 页的『Linux C++ 应用程序编译和链接选项』

相关样本:

- 『bldapp -- Builds Linux C++ applications』
- 『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』

## 为 C++ 应用程序构建脚本

```
#!/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/sql/lib

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.C error-checking utility.
    g++ -I$DB2PATH/include -c utilemb.C
else
    # Compile the utilapi.C error-checking utility.
    g++ -I$DB2PATH/include -c utilapi.C
fi

# Compile the program.
g++ -I$DB2PATH/include -c $1.C

if [ -f $1".sqc" ]
then
    # Link the program with utilemb.o
    g++ -o $1 $1.o utilemb.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2
else
    # Link the program with utilapi.o
    g++ -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	
编译选项:	
<b>g++</b>	GNU/Linux C++ 编译器。
<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:	
<b>g++</b>	将编译程序用作链接程序的前端。
<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 “RUNTIME=true” 未注释, 则包含值 “-Wl,-rpath,\$DB2PATH/lib” ; 否则, 不包含任何值。
<b>-Wl,-rpath,\$DB2PATH/lib</b>	指定运行时 DB2 共享库的位置。例如: \$HOME/sqllib/lib。
<b>-L\$DB2PATH/lib</b>	指定链接时 DB2 共享库的位置。例如: \$HOME/sqllib/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 232 页的『在 Linux 上构建 C++ 应用程序』

相关样本:

- 『bldapp -- Builds Linux C++ applications』

在 Linux 上构建 C++ 例程

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。

脚本 bldrtn 包含构建例程（存储过程和用户定义函数）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。它使用一或两个参数。

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, 则该程序使用缺省的样本数据库。

## 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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (UDF)

## 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至样本数据库, 则输入构建文件名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库, 还须输入该数据库名:

```
bldrtn spserver database
```

该脚本文件将存储过程复制到服务器的 `sqllib/function` 路径中。

2. 接着, 通过在服务器上运行 `spcat` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库, 如果先前通过调用 `spdrop.db2` 编目了这些例程, 则取消对它们的编目, 然后通过调用 `spcreate.db2` 来编目它们, 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. 最后停止并重新启动数据库以便可识别新的共享库。

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`。

要调用存储过程, 可输入如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。该名称可以是样本, 或它的别名, 或另一个数据库名。

**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` 构建用户定义函数程序 `udfsrv`，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。

一旦构建了 `udfsrv`，就可构建调用它的客户机应用程序 `udfcli`。可从源文件 `udfcli.sqC` 通过使用脚本 `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。该名称可以是样本，或它的别名，或另一个数据库名。

**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 用户定义函数程序 `udfemsrv`，如果连接样本数据库，则输入构建脚本名和程序名：

```
bldrtn udfemsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
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 `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。该名称可以是样本，或它的别名，或另一个数据库名。

**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函数。然后将输出返回给该客户机应用程序。

**相关概念：**

- 第 89 页的『构建文件』

**相关参考：**

- 第 65 页的『C/C++ 样本』
- 第 239 页的『Linux C++ 例程编译和链接选项』

**相关样本：**

- 『bldrtn -- Builds Linux C++ routines (stored procedures and UDFs)』
- 『spclient.sqC -- Call various stored procedures (C++)』
- 『spserver.sqC -- A variety of 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 -- Call a variety of types of user-defined functions (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqC』

- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc  
』

## 为 C++ 例程构建脚本

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Linux C++ routines (stored procedures and UDFs)
# Usage: bldsrv <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql/lib

# Set the runtime path since routines run as setuid.
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++ -fpic -I$DB2PATH/include -c $1.C -D_REENTRANT

# Link the program and create a shared library.
g++ -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>g++</b>	GNU/Linux C++ 编译器。
<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 ( <code>THREADSAFE</code> ) 或在引擎本身当中运行 ( <code>NOT FENCED</code> )。
链接选项:	
<b>g++</b>	将编译程序用作链接程序的前端。
<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>	包含值 “ <code>-Wl,-rpath,\$DB2PATH/lib</code> ” 以设置运行时路径, 因为例程是作为 <code>setuid</code> 运行的。
<b>-Wl,-rpath,\$DB2PATH/lib</b>	指定运行时 DB2 共享库的位置。例如: <code>\$HOME/sql1lib/lib</code> 。
<b>-L\$DB2PATH/lib</b>	指定链接时 DB2 静态库和共享库的位置。例如: <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 235 页的『在 Linux 上构建 C++ 例程』

相关样本:

- 『`bldrtn -- Builds Linux C++ routines (stored procedures and UDFs)`』

## 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。

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。

脚本 `bldmt` 包含构建嵌入式 SQL 多线程程序的命令。它使用多达四个参数。

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。第三个参数 `$3` 指定数据库的用户标识，而第四个参数 `$4` 指定密码。只有第一个参数即源文件名是必需的。数据库名、用户标识和密码是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

过程:

要从源文件 `dbthrd.sqC` 构建样本程序 `dbthrds`，输入:

```
bldmt dbthrds
```

产生可执行文件 `dbthrds`。要对样本数据库运行该可执行文件，输入:

```
dbthrds
```

相关概念:

- 第 89 页的『构建文件』

相关参考:

- 第 65 页的『C/C++ 样本』

相关样本:

- 『`bldmt -- Builds Linux C++ multi-threaded applications`』
- 『`dbthrd.sqC -- How to use multiple context APIs on UNIX (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

## 为 C++ 多线程应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldmt
# Builds Linux C++ multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
```

```

DB2PATH=$HOME/sqllib

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
g++ -I$DB2PATH/include -c $1.C -D_REENTRANT

# Link the program.
g++ -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2 -lpthread

```

---

## 第 10 章 Solaris

Forte C . . . . .	243	在 Solaris 上构建 C++ 多线程应用程序 . . . . .	265
在 Solaris 上构建 C 应用程序 . . . . .	243	为 C++ 多线程应用程序构建脚本 . . . . .	267
为 C 应用程序构建脚本 . . . . .	245	Micro Focus COBOL . . . . .	267
Solaris C 应用程序编译和链接选项 . . . . .	246	在 Solaris 上配置 Micro Focus COBOL 编	
在 Solaris 上构建 C 例程 . . . . .	247	译器 . . . . .	267
为 C 例程构建脚本 . . . . .	251	在 Solaris 上构建 Micro Focus COBOL 应	
Solaris C 例程编译和链接选项 . . . . .	252	用程序 . . . . .	268
在 Solaris 上构建 C 多线程应用程序 . . . . .	253	为 Micro Focus COBOL 应用程序构建脚	
为 C 多线程应用程序构建脚本 . . . . .	255	本 . . . . .	270
Forte C++ . . . . .	255	Solaris Micro Focus COBOL 应用程序编译	
在 Solaris 上构建 C++ 应用程序 . . . . .	255	和链接选项 . . . . .	270
为 C++ 应用程序构建脚本 . . . . .	257	在 Solaris 上构建 Micro Focus COBOL 例	
Solaris C++ 应用程序编译和链接选项 . . . . .	258	程 . . . . .	271
在 Solaris 上构建 C++ 例程 . . . . .	259	为 Micro Focus COBOL 例程构建脚本 . . . . .	273
为 C++ 例程构建脚本 . . . . .	263	Solaris Micro Focus COBOL 例程编译和链	
Solaris C++ 例程编译和链接选项 . . . . .	264	接选项 . . . . .	274

本章提供在 Solaris 操作环境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：

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

---

### Forte C

注：Forte C 以前称为 “SPARCompiler C”。

DB2 CLI 应用程序和例程的构建信息在 *CLI Guide and Reference* 中。

#### 在 Solaris 上构建 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。

构建脚本 `bldapp` 包含构建 DB2 应用程序的命令。它使用多达四个参数。

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，因此还须提供三个可选参数：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, 则使用缺省样本数据库。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, 输入:

```
bldapp cli_info
```

产生可执行文件 cli\_info。

要运行可执行文件, 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, 输入:

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接, 还须输入该数据库名:

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接, 还须输入该数据库实例的用户标识和密码:

```
bldapp tbmod database userid password
```

产生可执行文件 tbmod。

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样本数据库, 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, 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, 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 89 页的『构建文件』

#### 相关参考:

- 第 65 页的『C/C++ 样本』
- 第 246 页的『Solaris C 应用程序编译和链接选项』

#### 相关样本:

- 『bldapp -- Builds Solaris C applications』
- 『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』
- 『tbmod.sqc -- How to modify table data (C)』

### 为 C 应用程序构建脚本

```
#!/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

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-R$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
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 $EXTRA_LFLAG \
-L$DB2PATH/lib -ldb2
else
# Link the program with utilapi.o
cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilapi.o $EXTRA_LFLAG \
-L$DB2PATH/lib -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。

bldapp 的编译和链接选项	
编译选项:	
<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	
<b>链接选项:</b>	
<b>cc</b>	将编译程序用作链接程序的前端。
<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>	多线程支持中的链接。它是使用 libdb2 进行链接所需要的。 <b>注:</b> 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。
<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 “RUNTIME=true” 未注释，则包含值 “-R\$DB2PATH/lib”；否则，不包含任何值。
<b>-R\$DB2PATH/lib</b>	指定运行时 DB2 共享库的位置。例如: \$HOME/sqlllib/lib。
<b>-L\$DB2PATH/lib</b>	指定链接时 DB2 共享库的位置。例如: \$HOME/sqlllib/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 243 页的『在 Solaris 上构建 C 应用程序』

**相关样本:**

- 『bldapp -- Builds Solaris C applications』

## 在 Solaris 上构建 C 例程

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 sqlllib/samples/c 目录中。

脚本 `bldrtn` 包含构建例程（存储过程和用户定义函数）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，则该程序使用缺省的样本数据库。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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (UDF)

### 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至样本数据库，则输入构建脚本名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还须输入该数据库名:

```
bldrtn spserver database
```

该脚本文件将共享库复制到服务器上的 `sqllib/function` 目录中。

2. 接着，通过在服务器上运行 `spcat` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 编目了这些例程，则取消对它们的编目，然后通过调用 `spcreate.db2` 来编目它们，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. 最后停止并重新启动数据库以便可识别新的共享库。

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。该名称可以是样本, 或它的别名, 或另一个数据库名。

**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` 构建用户定义函数程序 `udfsrv`, 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。

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` 通过使用脚本文件 `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。该名称可以是样本, 或它的别名, 或另一个数据库名。

**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 用户定义函数程序 `udfemsrv`，如果连接至样本数据库，则输入构建脚本名和程序名：

```
bldrtn udfemsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
bldrtn udfemsrv database
```

该脚本文件将 UDF 库复制到 `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` 通过使用脚本文件 `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。该名称可以是样本，或它的别名，或另一个数据库名。

**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函数。然后将输出返回给该客户机应用程序。

### 相关概念:

- 第 89 页的『构建文件』

### 相关参考:

- 第 65 页的『C/C++ 样本』
- 第 252 页的『Solaris C 例程编译和链接选项』

### 相关样本:

- 『`bldrtn -- Builds Solaris C routines (stored procedures or UDFs)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

- 『spcat -- To catalog C stored procedures on UNIX』
- 『spclient.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』
- 『spserver.sqc -- A variety of 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 -- Call a variety of types of user-defined functions (C)』

## 为 C 例程构建脚本

```

#!/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/sql/lib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# Set the runtime path since routines run as setuid.
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 $EXTRA_LFLAG \
    -L$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
```

**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>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 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:	
<b>cc</b>	将编译程序用作链接程序的前端。
<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>	这是必需的，因为 DB2 库是使用 -mt 进行链接的。
<b>-G</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>	包含值 “-R\$DB2PATH/lib” 以设置运行时路径，因为例程是作为 setuid 运行的。
<b>-R\$DB2PATH/lib</b>	指定运行时 DB2 共享库的位置。例如: \$HOME/sql/lib/lib。
<b>-L\$DB2PATH/lib</b>	指定链接时 DB2 共享库的位置。例如: \$HOME/sql/lib/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 225 页的『在 Linux 上构建 C 例程』

相关样本:

- 『bldrtn -- Builds Linux C routines (stored procedures or UDFs)』

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 -pthread 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。

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。

脚本 `bldmt` 包含构建多线程应用程序的命令。

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。第三个参数 `$3` 指定数据库的用户标识，而第四个参数 `$4` 指定密码。只有第一个参数即源文件名是必需的。数据库名、用户标识和密码是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

### 过程:

要从源文件 `dbthrrds.sqc` 构建样本程序 `dbthrrds`，输入：

```
bldmt dbthrrds
```

产生可执行文件 `dbthrrds`。要对样本数据库运行该可执行文件，输入：

```
dbthrrds
```

**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`。请参见下面有关 `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 89 页的『构建文件』
- 『`db2osconf - Utility for Kernel Parameter Values Command`』（*Command Reference*）

### 相关参考:

- 第 65 页的『C/C++ 样本』

### 相关样本:

- 『`bldmt -- Builds Solaris C multi-threaded applications`』
- 『`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`』

## 为 C 多线程应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldmt
# Builds Solaris C multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqlllib

# To compile 64-bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-R$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
cc -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -I$DB2PATH/include -c $1.c

# Link the program.
cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2
```

---

## Forte C++

注: Forte C++ 以前称为 SPARCompiler C++

## 在 Solaris 上构建 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 sqlllib/samples/cpp 目录中。

构建脚本 `bldapp` 包含构建 DB2 API 和嵌入式 SQL 应用程序的命令。它使用多达四个参数。

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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`，输入：

```
bldapp cli_info
```

产生可执行文件 `cli_info`。可以输入如下命令，对样本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.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，输入：

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp tbmod database userid password
```

产生可执行文件 `tbmod`。

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样本数据库，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，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，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 89 页的『构建文件』

#### 相关参考：

- 第 65 页的『C/C++ 样本』
- 第 258 页的『Solaris C++ 应用程序编译和链接选项』

#### 相关样本：

- 『bldapp -- Builds Solaris C++ applications』
- 『cli\_info.C -- Set and get information at the client level (C++)』
- 『tbmod.sqlC -- How to modify table data (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

## 为 C++ 应用程序构建脚本

```
#!/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

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-R$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi
```

```

# If an embedded SQL program, precompile and bind it.
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 $EXTRA_LFLAG \
        -L$DB2PATH/lib -ldb2
else
    # Link the program with utilapi.o
    CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilapi.o $EXTRA_LFLAG \
        -L$DB2PATH/lib -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。

bldapp 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>-I\$DB2PATH/include</b>	指定 DB2 包含文件的位置。例如: \$HOME/sqlllib/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	
<b>链接选项:</b>	
<b>CC</b>	将编译程序用作链接程序的前端。
<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>	多线程支持中的链接。它是使用 libdb2 进行链接所需要的。 <b>注:</b> 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。
<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 “RUNTIME=true” 未注释，则包含值 “-R\$DB2PATH/lib”；否则，不包含任何值。
<b>-R\$DB2PATH/lib</b>	指定运行时 DB2 共享库的位置。例如: \$HOME/sql1lib/lib。
<b>-L\$DB2PATH/lib</b>	指定链接时 DB2 共享库的位置。例如: \$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 255 页的『在 Solaris 上构建 C++ 应用程序』

**相关样本:**

- 『bldapp -- Builds Solaris C++ applications』

**在 Solaris 上构建 C++ 例程**

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 sql1lib/samples/c 目录中。

脚本 `bldrtn` 包含构建例程（存储过程或用户定义函数）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。

数据库参数是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (UDF)

### 存储过程共享库

要从源文件 `spserver.sqc` 构建样本程序 `spserver`:

1. 如果连接至样本数据库，则输入构建文件名和程序名:

```
bldrtn spserver
```

如果连接另一个数据库，还须输入该数据库名:

```
bldrtn spserver database
```

该脚本文件将共享库复制到服务器上的 `sqllib/function` 目录中。

2. 接着，通过在服务器上运行 `spcat` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 编目了这些例程，则取消对它们的编目，然后通过调用 `spcreate.db2` 来编目它们，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. 最后停止并重新启动数据库以便可识别新的共享库。

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。该名称可以是样本, 或它的别名, 或另一个数据库名。

**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` 构建用户定义函数程序 `udfsrv`, 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。

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。该名称可以是样本, 或它的别名, 或另一个数据库名。

**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 用户定义函数程序 `udfemsrv`, 如果连接样本数据库, 则输入构建脚本名和程序名:

```
bldrtn udfemsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
bldrtn udfemsrv database
```

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，以便客户机应用程序数据库管理器可以存取它。

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。该名称可以是样本，或它的别名，或另一个数据库名。

**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函数。然后将输出返回给该客户机应用程序。

**相关概念:**

- 第 89 页的『构建文件』

**相关参考:**

- 第 65 页的『C/C++ 样本』
- 第 264 页的『Solaris C++ 例程编译和链接选项』

**相关样本:**

- 『bldrtn -- Builds Solaris C++ routines (stored procedures or UDFs)』
- 『spclient.sqC -- Call various stored procedures (C++)』
- 『spserver.sqC -- A variety of types of stored procedures (C++)』
- 『udfcli.sqC -- Call a variety of types of user-defined functions (C++)』

- 『udfsrv.C -- Call a variety of types of user-defined functions (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』

## 为 C++ 例程构建脚本

```

#!/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/sqlllib

# To compile 64-bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# Set the runtime path since routines run as setuid.
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 $EXTRA_LFLAG \
    -L$DB2PATH/lib -ldb2

# Copy the shared library to the sqlllib/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>CC</b>	C++ 编译器。
<b>-xarch=\$CFLAG_ARCH</b>	此选项确保编译器在使用 <code>libdb2.so</code> 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 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:	
<b>CC</b>	将编译程序用作链接程序的前端。
<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>	这是必需的，因为 DB2 库是使用 -mt 进行链接的。
<b>-G</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>	包含值 “-R\$DB2PATH/lib” 以设置运行时路径，因为例程是作为 setuid 运行的。
<b>-R\$DB2PATH/lib</b>	指定运行时 DB2 共享库的位置。例如: \$HOME/sql1lib/lib。
<b>-L\$DB2PATH/lib</b>	指定链接时 DB2 共享库的位置。例如: \$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 259 页的『在 Solaris 上构建 C++ 例程』

相关样本:

- 『bldrtn -- Builds Solaris C++ routines (stored procedures or UDFs)』

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 传送到预处理器，并将 -pthread 传送到链接程序。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。

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。

脚本 `bldmt` 包含构建多线程应用程序的命令。它使用多达四个参数。

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。第三个参数 `$3` 指定数据库的用户标识，而第四个参数 `$4` 指定密码。只有第一个参数即源文件名是必需的。数据库名、用户标识和密码是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

### 过程:

要从源文件 `dbthrrds.sqlC` 构建样本程序 `dbthrrds`，输入:

```
bldmt dbthrrds
```

产生可执行文件 `dbthrrds`。要对样本数据库运行该可执行文件，输入:

```
dbthrrds
```

**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`。请参见下面有关 `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 89 页的『构建文件』
- 『db2osconf - Utility for Kernel Parameter Values Command』（*Command Reference*）

### 相关参考:

- 第 65 页的『C/C++ 样本』

### 相关样本:

- 『bldmt -- Builds Solaris C++ multi-threaded applications』
- 『dbthrrds.sqlC -- How to use multiple context APIs on UNIX (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

## 为 C++ 多线程应用程序构建脚本

```
#!/bin/sh
# SCRIPT: bldmt
# Builds Solaris C++ multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# To compile 64-bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-R$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
CC -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -I$DB2PATH/include -c $1.C

# Link the program.
CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2
```

---

## Micro Focus COBOL

### 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，输入：

- 在 `bash` 或 `Korn` 外壳程序上：

```
export COBCPY=$COBCPY:$HOME/sqllib/include/cobol_mf
```

- 在 `C` 外壳程序上：

```
setenv COBCPY $COBCPY:$HOME/sqllib/include/cobol_mf
```

注：应在 `.profile` 文件中设置 `COBCPY`。

#### 相关任务：

- 第 271 页的『在 Solaris 上构建 Micro Focus COBOL 例程』
- 第 268 页的『在 Solaris 上构建 Micro Focus COBOL 应用程序』

## 在 Solaris 上构建 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。

构建脚本 `bldapp` 包含构建 DB2 应用程序的命令。它使用多达四个参数。

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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`，输入：

```
bldapp client
```

产生可执行文件 `client`。可以输入如下命令，对样本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, 输入:

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接, 还须输入该数据库名:

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接, 还须输入该数据库实例的用户标识和密码:

```
bldapp updat database userid password
```

产生可执行文件 `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样本数据库, 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, 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, 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 89 页的『构建文件』

### 相关参考:

- 第 270 页的『Solaris Micro Focus COBOL 应用程序编译和链接选项』
- 第 75 页的『COBOL 样本』

### 相关样本:

- 『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』

## 为 Micro Focus COBOL 应用程序构建脚本

```
#!/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/sqllib

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
    ./embprep $1 $2 $3 $4
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# 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:	
<b>cob</b>	将编译程序用作链接程序的前端。
<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>	指定链接时 DB2 静态库和共享库的位置。例如: \$HOME/sqlllib/lib。
<b>-ldb2</b>	链接 DB2 库。
<b>-ldb2gmf</b>	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 268 页的『在 Solaris 上构建 Micro Focus COBOL 应用程序』

相关样本:

- 『bldapp -- Builds Solaris Micro Focus COBOL applications』

在 Solaris 上构建 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 sqlllib/samples/cobol\_mf 目录中。

脚本 bldrtn 包含构建例程（存储过程）的命令。该脚本文件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。此脚本文件将源文件名 \$1 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例程，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。数据库名是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

过程:

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。

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：

```
bldsrv outsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
bldsrv outsrv database
```

此脚本文件将共享库复制到服务器的 `sqllib/function` 路径中。

一旦构建了共享库 `outsrv`，就可以构建调用库中同名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。该名称可以是样本，或它的别名，或另一个数据库名。

**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`，它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 89 页的『构建文件』

**相关参考：**

- 第 273 页的『Solaris Micro Focus COBOL 例程编译和链接选项』
- 第 75 页的『COBOL 样本』

相关样本:

- 『 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 』

## 为 Micro Focus COBOL 例程构建脚本

```
#!/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

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
./embprep $1 $2
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the program.
cob -cx $1.cbl

# Link the program.
cob -x -o $1 $1.o -Q -G -L$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>-x</b>	当与 <b>-G</b> 选项配合使用时生成共享库。
<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 -G</b>	生成共享库。
<b>-L\$DB2PATH/lib</b>	指定 DB2 运行时共享库的位置。例如: \$HOME/sql/lib/lib。如果不指定 <b>-L</b> 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 271 页的『在 Solaris 上构建 Micro Focus COBOL 例程』

相关样本:

- 『bldrtn -- Builds Solaris Micro Focus COBOL routines (stored procedures)』

---

# 第 11 章 Windows 操作系统

- WCHARTYPE CONVERT 预编译选项 . . . 275
- 对象链接与嵌入数据库 (OLE DB) 表函数 . . . 276
- Windows Management Instrumentation (WMI) . . . 277
- Microsoft Visual Basic . . . 277
  - 使用 Visual Basic 构建 ADO 应用程序 . . . 277
  - 使用 Visual Basic 构建 RDO 应用程序 . . . 280
  - 使用 Visual Basic 的对象链接与嵌入 (OLE) 自动化 . . . 282
- Microsoft Visual C++. . . 283
  - 使用 Visual C++ 构建 ADO 应用程序 . . . 283
  - 使用 Visual C++ 的对象链接与嵌入 (OLE) 自动化 . . . 285
  - 在 Windows 上构建 C/C++ 应用程序 . . . 285
  - C/C++ 应用程序的批处理文件 . . . 287
  - Windows C/C++ 应用程序编译和链接选项 . . . 288
  - 在 Windows 上构建 C/C++ 例程 . . . 289
  - C/C++ 例程的批处理文件 . . . 293
  - Windows C/C++ 例程编译和链接选项 . . . 294
- IBM VisualAge COBOL . . . 295
  - 在 Windows 上配置 IBM COBOL 编译器 . . . 295
  - 在 Windows 上构建 IBM COBOL 应用程序 . . . 296
  - IBM COBOL 应用程序的批处理文件 . . . 297

- Windows IBM COBOL 应用程序编译和链接选项 . . . 298
- 在 Windows 上构建 IBM COBOL 例程 . . . 299
- IBM COBOL 例程的批处理文件 . . . 300
- Windows IBM COBOL 例程编译和链接选项 . . . 301
- Micro Focus COBOL . . . 301
  - 在 Windows 上配置 Micro Focus COBOL 编译器 . . . 301
  - 在 Windows 上构建 Micro Focus COBOL 应用程序 . . . 302
  - Micro Focus COBOL 应用程序的批处理文件 . . . 304
  - Windows Micro Focus COBOL 应用程序编译和链接选项 . . . 304
  - 在 Windows 上构建 Micro Focus COBOL 例程 . . . 305
  - Micro Focus COBOL 例程的批处理文件 . . . 307
  - Windows Micro Focus COBOL 例程编译和链接选项 . . . 307
- Object REXX . . . 308
  - 在 Windows 上构建 Object REXX 应用程序 . . . 308

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:

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

---

## WCHARTYPE CONVERT 预编译选项

WCHARTYPE 预编译选项采用多字节格式或使用 `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。

`WCHAR_T` 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。

如果需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 INSERT INTO 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函数），情况也是如此。

#### 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。



### 相关参考:

- 第 85 页的『对象链接与嵌入数据库 (OLE DB) 表函数样本』

---

## Windows Management Instrumentation (WMI)

Windows® Management Instrumentation (WMI) 是 Microsoft 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的模型。

DB2® 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%` 是安装有 DB2 的路径。

使用 `cscript` 命令来运行“Visual Basic 脚本”样本。例如，要运行 `listsvr` 样本脚本，输入：

```
cscript listsvr.vbs
```

### 相关参考:

- 第 83 页的『Windows Management Instrumentation 样本』

---

## Microsoft Visual Basic

### 使用 Visual Basic 构建 ADO 应用程序

“ActiveX 数据对象” (ADO) 可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 8.1 客户机”将自动可选择地安装 IBM DADB2, 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 存取 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 clstr As String
clstr = "Provider=ibmdadb2; DSN=db2alias; UID=userid; PWD=password"
c1.Open clstr
...
```

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、方法和特性。

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 存取存取数据库

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 样本数据库的、作为程序一部分的 Visual Basic 命令。

不指定用户标识或密码打开样本数据库；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*)

#### 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 82 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 sqllib\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，该程序与样本数据库连接，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姓的 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
```

相关参考:

- 第 82 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 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® 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™ 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, 请参见 `sqllib\samples\ole` 中的 README 文件。

相关参考:

- 第 84 页的『“对象链接与嵌入” (OLE) 样本』

---

## 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。

DB2 在 `sqllib\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 8.1 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 活动服务器页 (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, 可参考本章中适当章节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 83 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 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® 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™ 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，请参见 `sqllib\samples\ole` 中的 README 文件。

#### 相关参考:

- 第 84 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。它使用多达四个参数，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，则使用缺省样本数据库。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`，输入：

```
bldapp cli_info
```

产生可执行文件 `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，输入：

```
bldapp tbmod
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp tbmod database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp tbmod database userid password
```

产生可执行文件 `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样本数据库，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，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，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 89 页的『构建文件』

#### 相关参考：

- 第 288 页的『Windows C/C++ 应用程序编译和链接选项』
- 第 65 页的『C/C++ 样本』

#### 相关样本：

- 『bldapp.bat -- Builds C applications on Windows』
- 『cli\_info.c -- Set and get information at the client level (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』
- 『cli\_info.C -- Set and get information at the client level (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 Windows Microsoft Visual C++ applications
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.
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 批处理文件中所演示的那样。

bldapp 的编译和链接选项	
编译选项:	
%BLDCOMP%	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++ 编译器 32 位版应用程序）或 ecl（即 Intel C++ 编译器 Itanium 64 位版应用程序）。
-Zi	启用调试信息
-Od	禁用优化。关闭优化使得使用调试器更容易。
-c	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。
-W2	输出警告、错误、严重的和不可恢复的错误消息。
-DWIN32	Windows 操作系统需要的编译器选项。

bldapp 的编译和链接选项	
链接选项:	
<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。
有关其它编译程序选项, 参考编译程序文档。	

相关任务:

- 第 285 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。数据库名是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

### 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 用户定义函数 (UDF)
- 嵌入式 SQL 用户定义函数 (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，还须输入该数据库名:

```
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` 脚本来编目例程:

```
spcat
```

此脚本连接至样本数据库，如果先前通过调用 `spdrop.db2` 编目了这些例程，则取消对它们的编目，然后通过调用 `spcreate.db2` 来编目它们，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，停止并重新启动数据库以允许识别新的 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。该名称可以是样本，或它的别名，或另一个数据库名。

**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 构建用户定义函数 udfsrv，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）来构建用户定义函数 DLL。该批处理文件将用户定义函数 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，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 用户定义函数库 udfemsrv，输入：

```
bldrtn udfemsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
bldrtn udfmsrv database
```

批处理文件使用模块定义文件 `udfmsrv.def`（该文件与样本程序处于同一目录中）来构建用户定义函数 DDL。该批处理文件将用户定义函数 DLL（`udfmsrv.dll`）复制到服务器的 `sqllib\function` 路径中。

一旦构建了 `udfmsrv`，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，输入：

```
udfemcli
```

调用应用程序将调用 `udfmsrv` DLL 中的 UDF。

#### 相关概念：

- 第 89 页的『构建文件』

#### 相关参考：

- 第 294 页的『Windows C/C++ 例程编译和链接选项』
- 第 65 页的『C/C++ 样本』

#### 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』
- 『`spcat` -- To catalog C stored procedures on UNIX』
- 『`spclient.sqc` -- Call various stored procedures (C)』
- 『`spcreate.db2` -- How to catalog the stored procedures contained in `spserver.sqc`』
- 『`spdrop.db2` -- How to uncatalog the stored procedures contained in `spserver.sqc`』
- 『`spserver.sqc` -- A variety of 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)』
- 『`udfmsrv.sqc` -- Call a variety of types of embedded SQL user-defined functions. (C)』



- 『udfsrv.c -- Call 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 -- A variety of 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 -- Call 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 Windows Microsoft Visual C++ routines (stored procedures and UDFs)
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=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.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 %1.c
goto link_step
:cpp
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %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>%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++ 编译器 32 位版应用程序) 或 ecl (即 Intel C++ 编译器 Itanium 64 位版应用程序)。
<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。本书假定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。
链接选项:	
<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。
有关其它编译程序选项, 参考编译程序文档。	

相关任务:

- 第 289 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』

---

## 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 296 页的『在 Windows 上构建 IBM COBOL 应用程序』

- 第 299 页的『在 Windows 上构建 IBM COBOL 例程』

## 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 `sqllib\samples\cobol` 目录中。

批处理文件 `bldapp.bat` 包含构建 DB2 应用程序的命令。它使用多达四个参数，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，因此还须提供三个可选参数：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文件 `embprep.bat`。如果未提供数据库名，则使用缺省样本数据库。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`，输入：

```
bldapp client
```

产生可执行文件 `client.exe`。可输入可执行文件名（不带扩展名），来对样本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，输入：

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp updat database userid password
```

产生可执行文件 `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样本数据库, 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, 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, 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 89 页的『构建文件』

相关参考:

- 第 298 页的『Windows IBM COBOL 应用程序编译和链接选项』
- 第 75 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> ]]

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.
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
```

```
rem Link the program.
cob2 %1.obj checkerr.obj db2api.lib
@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	
编译选项:	
cob2	IBM VisualAge COBOL 编译器。
-qpgmname(mixed)	指示编译器可用大小写混合的名称调用（CALL）库入口点。
-c	只执行编译；不链接。本书假定编译和链接是两个独立的步骤。
-q1lib	指示编译器处理 COPY 语句。
-Ipath	指定 DB2 包含文件的位置。例如：-I"%DB2PATH%\include\cobol_a"。
checkerr.cbl	编译错误检查实用程序。
链接选项:	
cob2	将编译器用作链接程序的前端
%1.obj	包括该程序的对象文件。
checkerr.obj	包括错误检查实用程序的对象文件。
db2api.lib	链接 DB2 库。
有关其它编译程序选项，参考编译程序文档。	

相关任务:

- 第 296 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。

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将例程编译成 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。数据库名是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

### 过程:

要从源文件 `outsrv.sqb` 构建样本程序 `outsrv`，连接样本程序而后输入：

```
bldrtn outsrv
```

如果连接另一个数据库，还须输入该数据库名：

```
bldrtn outsrv database
```

该批处理文件将 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。该名称可以是样本、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 89 页的『构建文件』

#### 相关参考:

- 第 300 页的『Windows IBM COBOL 例程编译和链接选项』
- 第 75 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> ]

rem Precompile and bind the program.
call embprep %1 %2

rem Compile the stored procedure.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" %1.cb1

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"
@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 and 用户定义函数）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"。
链接选项:	
<b>ilink</b>	使用 IBM VisualAge COBOL 链接程序。
<b>/free</b>	自由格式。
<b>/no1</b>	无标志。
<b>/dll</b>	创建具有源程序名的 DLL。
<b>db2api.lib</b>	链接 DB2 库。
<b>%1.exp</b>	包括导出文件。
<b>%1.obj</b>	包括该程序的对象文件。
<b>iwzrwin3.obj</b>	包括 IBM VisualAge COBOL 提供的对象文件。
有关其它编译程序选项, 参考编译程序文档。	

相关任务:

- 第 299 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 302 页的『在 Windows 上构建 Micro Focus COBOL 应用程序』
- 第 305 页的『在 Windows 上构建 Micro Focus COBOL 例程』

## 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 sqlllib\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。它使用多达四个参数，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，因此还须提供三个可选参数：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，则使用缺省样本数据库。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`，输入：

```
bldapp client
```

产生可执行文件 `client.exe`。可输入可执行文件名（不带扩展名），来对样本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，输入：

```
bldapp updat
```

2. 如果与同一实例中的另一个数据库连接，还须输入该数据库名：

```
bldapp updat database
```

3. 如果与另一个实例中的数据库连接，还须输入该数据库实例的用户标识和密码：

```
bldapp updat database userid password
```

产生可执行文件 `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样本数据库，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，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，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 89 页的『构建文件』

## 相关参考:

- 第 304 页的『Windows Micro Focus COBOL 应用程序编译和链接选项』
- 第 75 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.
cbllink -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。

bldapp 的编译和链接选项	
编译选项:	
cobol	Micro Focus COBOL 编译器。

bldapp 的编译和链接选项	
链接选项:	
<b>cbllink</b>	使用链接程序链接编辑。
<b>-l</b>	链接 lcobol 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。
有关其它编译程序选项，参考编译程序文档。	

相关任务:

- 第 302 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将例程编译成 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。数据库名是可选的。如果未提供数据库名，则该程序使用缺省的样本数据库。

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，还须输入该数据库名：

```
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。该名称可以是样本，或它的别名，或另一个数据库名。

**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 89 页的『构建文件』

**相关参考：**

- 第 307 页的『Windows Micro Focus COBOL 例程编译和链接选项』
- 第 75 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.
cbl1link /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>	防止外部符号转换为大写。
链接选项:	
<b>cbl1link</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。
有关其它编译程序选项，参考编译程序文档。	

相关任务:

- 第 305 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
- 单继承和多继承

Object REXX 与 classic REXX 完全兼容。在本节中，只要提到 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`，输入：

```
rexx updat.cmd
```

相关参考：

- 第 86 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 通用数据库” 技术信息概述

可以下列格式获取 “DB2 通用数据库” 技术信息:

- 书籍 (PDF 和硬拷贝格式)
- 主题树 (HTML 格式)
- DB2 工具的帮助 (HTML 格式)
- 样本程序 (HTML 格式)
- 命令行帮助
- 教程

本节是有关所提供技术信息以及可如何访问这些信息的概述。

### DB2 文档的修订包

IBM 可能会阶段性地提供文档修订包。文档修订包使您可以在新信息可供使用时更新从 *DB2 HTML 文档 CD* 中安装的信息。

**注:** 如果您安装了文档修订包, 则您的 HTML 文档将包含比 DB2 的印刷或联机 PDF 手册更新的信息。

### DB2 技术信息类别

DB2 技术信息是按下列标题分类的:

- 核心 DB2 信息
- 管理信息
- 应用程序开发信息
- 商务智能信息
- 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，下表描述了订购硬拷贝、打印或查看 PDF 或者找出该书的 HTML 目录所需的信息。DB2 资料库中每本书的完整描述可从 IBM 出版物中心（IBM Publications Center）获得，网址为 [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order)。

HTML 文档 CD 的安装目录对于各个信息类别来说是不同的：

```
htmlcdpath/doc/htmlcd/%L/category
```

其中：

- *htmlcdpath* 是安装了 HTML CD 的目录。
- *%L* 是语言标识符。例如，en\_US。
- *category* 是类别标识符。例如，core 表示核心 DB2 信息。

在下表中的 PDF 文件名列中，文件名第六个位置的字符指示书籍的语言版本。例如，文件名 db2d1e80 标识英文版本的《管理指南：计划》，而文件名 db2d1g80 标识该书的德语版本。下列字母用在文件名的第六个字符处以指示语言版本：

语言	标识符
阿拉伯语	w
巴西葡萄牙语	b
保加利亚语	u
克罗地亚语	9
捷克语	x
丹麦语	d
荷兰语	q
英语	e
芬兰语	y
法语	f
德语	g
希腊语	a
匈牙利语	h
意大利语	i
日语	j
韩国语	k
挪威语	n
波兰语	p
葡萄牙语	v
罗马尼亚语	8
俄语	r
简体中文	c
斯洛伐克语	7
斯洛文尼亚语	l
西班牙语	z

瑞典语	s
繁体中文	t
土耳其语	m

无书号指示该书只有联机版本而没有印刷版本。

### 核心 DB2 信息

此类别中的信息包括对所有 DB2 用户都很重要的 DB2 主题。不管您是程序员、数据库管理员或您将使用 DB2 Connect、DB2 仓库管理器或其它 DB2 产品，都将会发现此类别中的信息很有用。

此类别的安装目录为 doc/htmlcd/%L/core。

表 27. 核心 DB2 信息

书名	书号	PDF 文件名
<i>IBM DB2 Universal Database Command Reference</i>	SC09-4828	db2n0x80
《IBM DB2 通用数据库词汇表》	无书号	db2t0c80
《IBM DB2 通用数据库主索引》	S152-0192	db2w0c80
《IBM DB2 通用数据库消息参考第 1 卷》	G152-0177	db2m1c80
《IBM DB2 通用数据库消息参考第 2 卷》	G152-0178	db2m2c80
《IBM DB2 通用数据库新增内容》	S152-0176	db2q0c80

### 管理信息

此类别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。

此类别的安装目录为 doc/htmlcd/%L/admin。

表 28. 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	db2d1c80

表 28. 管理信息 (续)

书名	书号	PDF 文件名
《IBM DB2 通用数据库管理指南: 实现》	S152-0165	db2d2c80
《IBM DB2 通用数据库管理指南: 性能》	S152-0166	db2d3c80
<i>IBM DB2 Universal Database Administrative API Reference</i>	SC09-4824	db2b0x80
<i>IBM DB2 Universal Database Data Movement Utilities Guide and Reference</i>	SC09-4830	db2dmx80
《IBM DB2 通用数据库数据恢复和高可用性指南与参考大全》	S152-0181	db2hac80
《IBM DB2 通用数据库数据仓库中心管理指南》	S152-0188	db2ddc80
<i>IBM DB2 Universal Database Federated Systems Guide</i>	GC27-1224	db2fpx80
《IBM DB2 通用数据库管理和开发 GUI 工具指南》	S152-0180	db2atc80
<i>IBM DB2 Universal Database Replication Guide and Reference</i>	SC27-1121	db2e0x80
《IBM DB2 安装和管理卫星环境》	G152-0272	db2dsc80
<i>IBM DB2 Universal Database SQL Reference, Volume 1</i>	SC09-4844	db2s1x80
<i>IBM DB2 Universal Database SQL Reference, Volume 2</i>	SC09-4845	db2s2x80
<i>IBM DB2 Universal Database System Monitor Guide and Reference</i>	SC09-4847	db2f0x80

### 应用程序开发信息

此类别中的信息对于应用程序开发者或使用 DB2 的程序员特别有用。将可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（如嵌入式 SQL、ODBC、JDBC、SQLj 和 CLI）访问 DB2 所需的文档。如果您联机查看 HTML 格式的此信息，则还可以访问一组 HTML 格式的 DB2 样本程序。

此类别的安装目录为 doc/htmlcd/%L/ad。

表 29. 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	db2axc80
IBM DB2 Universal Database Application Development Guide: Programming Client Applications	SC09-4826	db2a1x80
IBM DB2 Universal Database Application Development Guide: Programming Server Applications	SC09-4827	db2a2x80
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1	SC09-4849	db2l1x80
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2	SC09-4850	db2l2x80
IBM DB2 Universal Database Data Warehouse Center Application Integration Guide	SC27-1124	db2adx80
IBM DB2 XML Extender Administration and Programming	SC27-1234	db2sxx80

### 商务智能信息

此类别中的信息描述如何使用将增强“DB2 通用数据库”的数据入库功能和分析功能的组件。

此类别的安装目录为 doc/htmlcd/%L/wareh。

表 30. 商务智能信息

书名	书号	PDF 文件名
IBM DB2 Warehouse Manager Information Catalog Center Administration Guide	SC27-1125	db2dix80
《IBM DB2 仓库管理器安装指南》	G152-0187	db2idc80

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”来存取主机或 iSeries 数据。

此类别的安装目录为 doc/htmlcd/%L/conn。

表 31. DB2 Connect 信息

书名	书号	PDF 文件名
APPC, CPI-C, and SNA Sense Codes	无书号	db2apx80
IBM Connectivity Supplement	无书号	db2h1x80
《IBM DB2 Connect 快速入门, DB2 Connect 企业版》	G152-0271	db2c6c80
《IBM DB2 Connect 快速入门, DB2 Connect 个人版》	G152-0171	db2c1c80
《IBM DB2 Connect 用户指南》	S152-0172	db2c0c80

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。

此类别的安装目录为 doc/htmlcd/%L/start。

表 32. 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	db2itc80
《IBM DB2 通用数据库快速入门, DB2 服务器版》	G152-0173	db2isc80
《IBM DB2 通用数据库快速入门, DB2 个人版》	G152-0175	db2i1c80
《IBM DB2 通用数据库安装与配置补遗》	G152-0174	db2iyc80
《IBM DB2 通用数据库快速入门, DB2 Data Links Manager 版》	G152-0169	db2z6c80

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。



此类别的安装目录为 doc/htmlcd/%L/tutr。

表 33. 教程信息

书名	书号	PDF 文件名
《商务智能教程：数据仓库简介》	无书号	db2tuc80
《商务智能教程：数据入库扩展课程》	无书号	db2tac80
<i>Development Center Tutorial for Video Online using Microsoft Visual Basic</i>	无书号	db2tdx80
<i>Information Catalog Center Tutorial</i>	无书号	db2aix80
<i>Video Central for e-business Tutorial</i>	无书号	db2twx80
《Visual Explain 教程》	无书号	db2tvx80

可选组件信息

此类别中的信息描述如何使用可选 DB2 组件。

此类别的安装目录为 doc/htmlcd/%L/opt。

表 34. 可选组件信息

书名	书号	PDF 文件名
<i>IBM DB2 Life Sciences Data Connect Planning, Installation, and Configuration Guide</i>	GC27-1235	db2lsx80
<i>IBM DB2 Spatial Extender User's Guide and Reference</i>	SC27-1226	db2sbx80
<i>IBM DB2 Database Data Links Manager Administration Guide and Reference</i>	SC27-1221	db2z0x80
<i>IBM DB2 Universal Database Net Search Extender Administration and Programming Guide</i>	SH12-6740	N/A
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。它们还提供了并入到每个发行版和修订包中的文档更新的总结。

表 35. 发行说明

书名	书号	PDF 文件名
《DB2 发行说明》	参见“注”。	参见“注”。
《DB2 安装说明》	仅在产品 CD-ROM 上提供。	仅在产品 CD-ROM 上提供。

注: 发行说明的 HTML 版本可从“信息中心”或产品 CD-ROM 上获取。要在基于 UNIX 的平台上查看 ASCII 文件, 参见 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:

- /usr/opt/db2\_08\_01 (在 AIX 上)
- /opt/IBM/db2/V8.1 (在所有其它 UNIX 操作系统上)

### 相关任务:

- 第 318 页的『从 PDF 文件打印 DB2 书籍』
- 第 319 页的『订购打印的 DB2 书籍』
- 第 320 页的『访问联机帮助』
- 第 323 页的『通过从管理工具访问“DB2 信息中心”来查找产品信息』
- 第 324 页的『直接从 DB2 HTML 文档 CD 联机查看技术文档』

---

## 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:

确保具有 Adobe Acrobat Reader。它可从 Adobe Web 站点获得, 网址为 [www.adobe.com](http://www.adobe.com)。

### 过程:

要从 PDF 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, 请参考《快速入门》一书。

2. 启动 Adobe Acrobat Reader。
3. 从下列位置之一打开 PDF 文件:
  - 在 Windows 操作系统上:

`x:\doc\language` 目录, 其中 *x* 表示 CD-ROM 盘符, *language* 表示两个字符的地区代码(它表示您所用的语言), 例如, EN 表示英语。
  - 在 UNIX 操作系统上:

CD-ROM 上的 `/cdrom/doc/%L` 目录, 其中 `/cdrom` 表示 CD-ROM 的安装点而 `%L` 表示期望的语言环境的名称。

#### 相关任务:

- 第 319 页的『订购打印的 DB2 书籍』
- 第 323 页的『通过从管理工具访问“DB2 信息中心”来查找产品信息』
- 第 324 页的『直接从 DB2 HTML 文档 CD 联机查看技术文档』

#### 相关参考:

- 第 311 页的『“DB2 通用数据库”技术信息概述』

---

## 订购打印的 DB2 书籍

### 过程:

#### 要订购打印的书籍:

- 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)。
- 在美国可致电 1-800-879-2755, 在加拿大则可致电 1-800-IBM-4YOU。
- 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)。

还可通过从您的 IBM 分销商订购您的 DB2 产品的文档包来获得印刷的 DB2 手册。文档包是 DB2 库中的手册的一个子集, 它们被选择来帮助您使用您购买的 DB2 产品进行初步的操作。文档包中的手册与 *DB2 PDF 文档 CD* 中以 PDF 格式提供的手册相同, 并包含与 *DB2 HTML 文档 CD* 中提供的文档相同的内容。

#### 相关任务:

- 第 318 页的『从 PDF 文件打印 DB2 书籍』
- 第 321 页的『通过从浏览器访问“DB2 信息中心”来查找主题』
- 第 324 页的『直接从 DB2 HTML 文档 CD 联机查看技术文档』

### 相关参考:

- 第 311 页的『“DB2 通用数据库”技术信息概述』

---

## 访问联机帮助

所有 DB2 组件附带提供的联机帮助有三种类型:

- 窗口和笔记本帮助
- 命令行帮助
- SQL 语句帮助

窗口和笔记本帮助说明可在窗口或笔记本中执行的任务并描述各控件。此帮助有两种类型:

- 可从**帮助**按钮访问的帮助
- 弹出信息

**帮助**按钮让您可以访问概述和先决条件信息。弹出信息描述窗口或笔记本中的各控件。窗口和笔记本帮助可从具有用户界面的 DB2 中心和组件获得。

命令行帮助包括“命令”帮助和“消息”帮助。“命令”帮助说明命令行处理器中命令的语法。“消息”帮助描述产生错误消息的原因并描述为解决错误而应采取的任何操作。

SQL 语句帮助包括 SQL 帮助和 SQLSTATE 帮助。DB2 返回可作为 SQL 语句结果的条件的 SQLSTATE 值。SQLSTATE 帮助说明 SQL 语句 (SQL 语句和类代码) 的语法。

**注:** SQL 帮助对于 UNIX 操作系统不可用。

### 过程:

要访问联机帮助:

- 对于窗口和笔记本帮助, 单击**帮助**或单击该控件, 然后单击 **F1**。如果选择了**工具设置**笔记本常规页上的**自动显示弹出信息**复选框, 则还可以通过将鼠标光标置于特定控件上来查看该控件的弹出信息。
- 对于命令行帮助, 打开命令行处理器并输入:
  - 对于“命令”帮助:

`? 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 显示 CATALOG DATABASE 命令的帮助。

- 对于“消息”帮助:

`? XXXnnnnnn`

其中 `XXXnnnnnn` 表示有效消息标识符。

例如, ? SQL30081 将显示有关 SQL30081 消息的帮助。

- 对于 SQL 语句帮助, 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 321 页的『通过从浏览器访问“DB2 信息中心”来查找主题』
- 第 324 页的『直接从 DB2 HTML 文档 CD 联机查看技术文档』

---

## 通过从浏览器访问“DB2 信息中心”来查找主题

“DB2 信息中心”可从浏览器访问, 从而使您能够访问为充分利用“DB2 通用数据库”和 DB2 Connect 所需的信息。“DB2 信息中心”还记录主要的 DB2 功能部件和组件, 包括复制、数据入库、元数据和 DB2 extender。

从浏览器访问的“DB2 信息中心”包括下列主要元素:

**导航树** 导航树位于浏览器窗口左边的框架中。该树可展开和折叠以显示和隐藏主题、词汇表和“DB2 信息中心”中的主索引。

#### 导航工具栏

导航工具栏位于浏览器窗口的右上边框架中。导航工具栏包含一些使您能够执行下列操作的按钮: 搜索“DB2 信息中心”、隐藏导航树以及查找导航树中当前显示的主题。

#### 内容框架

内容框架位于浏览器窗口的右下边框架中。当单击导航树中的链接、单击搜索结果或访问另一主题或主索引的链接时, 内容框架会显示“DB2 信息中心”的主题。

#### 先决条件:

要从浏览器访问“DB2 信息中心”，必须使用下列浏览器之一：

- Microsoft Explorer，版本 5 或更高版本
- Netscape Navigator，版本 6.1 或更高版本

#### 限制：

“DB2 信息中心”只包含您选择从 *DB2 HTML 文档 CD* 安装的那些主题集。如果您尝试访问指向某个主题的链接时 Web 浏览器返回找不到文件错误，则您必须安装 *DB2 HTML 文档 CD* 中的一个或多个附加的主题集。

#### 过程：

要通过使用关键字进行搜索来查找主题：

1. 在导航工具栏中，单击**搜索**。
2. 在“搜索”窗口最上面的文本输入字段中，输入一个或多个与您感兴趣的领域相关的词条，并单击**搜索**。一个按准确度排列的主题列表将显示在**结果**字段中。每一单项旁的数字等级提供了匹配程度的指示（较大的数字表示较高的匹配程度）。

输入较多的项会提高查询的精度，同时还会减少从查询返回的主题数目。

3. 在**结果**字段中，单击想要阅读的主题的标题。该主题将会显示在内容框架中。

要查找导航树中的主题：

1. 在导航树中，单击与您感兴趣的区域相关的主题类别的书籍图标。一个子类别列表将显示在该图标下面。
2. 继续单击书籍图标，直到找到包含您感兴趣的主题的类别为止。链接至主题的类别在您将光标移到类别标题上时将类别标题显示为带下划线的链接。导航树使用页图标来标识主题。
3. 单击主题链接。该主题会显示在内容框架中。

要查找主索引中的主题或项：

1. 在导航树中，单击“索引”类别。该类别展开，并在导航树中显示按字母顺序排列的链接列表。
2. 在导航树中，单击相应于与感兴趣主题相关的项的第一个字符的链接。具有该首字符的项列表将会显示在内容框架中。具有多个索引条目的项将由一个书籍图标标识。
3. 单击与您感兴趣的项相对应的书籍图标。一个子项和主题列表将显示在您单击的项下面。主题是由页图标标识的，其标题带有下划线。
4. 单击符合需要的主题的标题。该主题会显示在内容框架中。

#### 相关概念：

- 第 329 页的『易使用性』
- 第 331 页的『从浏览器访问的 DB2 信息中心』

#### 相关任务:

- 第 323 页的『通过从管理工具访问“DB2 信息中心”来查找产品信息』
- 第 325 页的『更新安装在机器上的 HTML 文档』
- 第 327 页的『对于使用 Netscape 4.x 搜索 DB2 文档进行故障诊断』
- 第 328 页的『搜索 DB2 文档』

#### 相关参考:

- 第 311 页的『“DB2 通用数据库”技术信息概述』

---

## 通过从管理工具访问“DB2 信息中心”来查找产品信息

“DB2 信息中心”提供了对 DB2 产品信息的快速访问且在可以使用 DB2 管理工具的所有操作系统上可用。

从工具访问的“DB2 信息中心”提供了六种类型的信息。

**任务** 可使用 DB2 执行的关键任务。

**概念** DB2 的关键概念。

**参考** DB2 参考信息，如关键字、命令以及 API。

#### 故障诊断

帮助您解决常见 DB2 问题的错误消息和信息。

**样本** 随 DB2 提供的样本程序的 HTML 列表的链接。

**教程** 用来帮助您了解 DB2 功能部件的指导性辅助。

#### 先决条件:

“DB2 信息中心”中的某些链接指向因特网上的 Web 站点。要显示这些链接的内容，首先必须与因特网连接。

#### 过程:

要通过从工具访问“DB2 信息中心”来查找产品信息:

1. 用下列方法之一启动“DB2 信息中心”:

- 从图形管理工具中，单击工具栏中的**信息中心**图标。还可从**帮助**菜单中选择它。
- 在命令行中输入 **db2ic**。

2. 单击与试图查找的信息相关的信息类型的选项卡。
3. 浏览整个树并单击感兴趣的主体。“信息中心”将启动 Web 浏览器以显示信息。
4. 要查找信息而无须浏览列表，可单击列表右边的**搜索**图标。  
一旦“信息中心”启动了浏览器来显示信息，就可通过单击导航工具栏中的**搜索**图标来执行全文本搜索。

#### 相关概念:

- 第 329 页的『易使用性』
- 第 331 页的『从浏览器访问的 DB2 信息中心』

#### 相关任务:

- 第 321 页的『通过从浏览器访问“DB2 信息中心”来查找主题』
- 第 328 页的『搜索 DB2 文档』

---

## 直接从 DB2 HTML 文档 CD 联机查看技术文档

还可直接从 CD 读取可从 *DB2 HTML 文档 CD* 安装的所有 HTML 主题。因此，可查看文档而不必安装它。

#### 限制:

由于“工具”帮助是从 DB2 产品 CD 而不是从 *DB2 HTML 文档 CD* 安装的，您必须安装 DB2 产品才能查看该帮助。

#### 过程:

1. 插入 *DB2 HTML 文档 CD*。在 UNIX 操作系统上，安装 *DB2 HTML 文档 CD*。有关如何在 UNIX 操作系统上安装 CD 的详细信息，参考《快速入门》一书。
2. 启动 HTML 浏览器并打开适当的文件:

- 对于 Windows 操作系统:

```
e:\program files\IBM\SQLLIB\doc\htmlcd\%L\index.htm
```

其中 *e* 表示 CD-ROM 驱动器，%L 是想要使用的文档的语言环境，例如，**en\_US** 表示英语。

- 对于 UNIX 操作系统:

```
/cdrom/program files/IBM/SQLLIB/doc/htmlcd/%L/index.htm
```

其中 */cdrom/* 表示安装 CD 的地方，%L 是想要使用的文档的语言环境，例如，**en\_US** 表示英语。



### 相关任务:

- 第 321 页的『通过从浏览器访问“DB2 信息中心”来查找主题』
- 第 326 页的『将文件从 DB2 HTML 文档 CD 复制到 Web 服务器』

### 相关参考:

- 第 311 页的『“DB2 通用数据库”技术信息概述』

---

## 更新安装在机器上的 HTML 文档

现在，就有可能在 IBM 进行了更新之后更新从 *DB2 HTML 文档 CD* 安装的 HTML。可用以下两种方法之一来完成：

- 使用“信息中心”（如果安装了 DB2 管理 GUI 工具的话）。
- 通过下载和应用 DB2 HTML 文档修订包。

**注：**这将不会更新 DB2 代码；它只更新从 *DB2 HTML 文档 CD* 安装的 HTML 文档。

### 过程:

要使用“信息中心”来更新本地文档：

1. 用下列方法之一启动“DB2 信息中心”：
  - 从图形管理工具中，单击工具栏中的**信息中心**图标。还可从**帮助**菜单中选择它。
  - 在命令行中输入 **db2ic**。
2. 确保您的机器对外部因特网具有访问权；更新程序将从 IBM 服务器下载最新的文档修订包（如果需要的话）。
3. 从菜单中选择**信息中心** —> **更新本地文档**以启动更新。
4. 提供代理信息（如果需要的话）以连接至外部因特网。

“信息中心”将下载并应用最新的文档修订包（如果有的话）。

要手工下载并应用文档修订包：

1. 确保机器已连接至因特网。
2. 在浏览器中打开 **DB2** 支持页，网址为：  
[www.ibm.com/software/data/db2/udb/winows2unix/support](http://www.ibm.com/software/data/db2/udb/winows2unix/support)。
3. 访问版本 8 的链接并查找“文档修订包”（Documentation FixPaks）链接。
4. 通过将文档修订包级别与已安装的文档级别进行比较来确定本地文档的版本是否已过时。您机器上的此当前文档处于以下级别：**DB2 v8.1 GA**。

5. 如果有更新的文档版本，则下载适用于您的操作系统的修订包。有一个适用于所有 Windows 平台的修订包和一个适用于所有 UNIX 平台的修订包。
6. 应用修订包:
  - 对于 Windows 操作系统：文档修订包是自解压 zip 文件。将下载文档修订包置于一个空目录中并运行它。这将创建一个 **setup** 命令，可运行该命令来安装文档修订包。
  - 对于 UNIX 操作系统：文档修订包是压缩的 tar.Z 文件。解压并解取该文件。这将创建一个带有称为 **installdocfix** 的脚本的名为 `delta_install` 的目录。运行此脚本来安装文档修订包。

#### 相关任务:

- 第 326 页的『将文件从 DB2 HTML 文档 CD 复制到 Web 服务器』

#### 相关参考:

- 第 311 页的『“DB2 通用数据库”技术信息概述』

---

## 将文件从 DB2 HTML 文档 CD 复制到 Web 服务器

在 *DB2 HTML 文档 CD* 上交了整个 DB2 信息库，可将它安装在 Web 服务器上以更便于访问。将想要的语言的文档复制至 Web 服务器即可。

**注：**当您通过低速连接从 Web 服务器访问 HTML 文档时，可能会遇到低性能的情况。

#### 过程:

要将文件从 *DB2 HTML 文档 CD* 复制到 Web 服务器，使用适当的源路径：

- 对于 Windows 操作系统：

```
E:\program files\IBM\SQLLIB\doc\htmlcd\%L\*.*
```

其中 *E* 表示 CD-ROM 驱动器，*%L* 表示语言标识符。

- 对于 UNIX 操作系统：

```
/cdrom/program files/IBM/SQLLIB/doc/htmlcd/%L/*.*
```

其中 *cdrom* 表示 CD-ROM 驱动器的安装点，*%L* 表示语言标识。

#### 相关任务:

- 第 328 页的『搜索 DB2 文档』

#### 相关参考:

- 『受支持的 DB2 界面语言、语言环境和代码页』（《DB2 服务器快速入门》）

- 第 311 页的『“DB2 通用数据库”技术信息概述』

---

## 对于使用 Netscape 4.x 搜索 DB2 文档进行故障诊断

大多数搜索问题都与 web 浏览器提供的 Java 支持有关。此任务描述可能的解决办法。

### 过程:

一个 Netscape 4.x 常见问题是丢失和误置安全性类。尝试下列解决办法，尤其是当您在浏览器 Java 控制台中看到以下行时更应尝试此方法:

找不到类 java/security/InvalidParameterException

- 在 Windows 操作系统上:

从 *DB2 HTML 文档 CD*，将提供的 `x:program files\IBM\SQLLIB\doc\htmlcd\locale\InvalidParameterException.class` 文件复制到相对于 Netscape 浏览器安装的 `java\classes\java\security\` 目录，其中 *x* 表示 CD-ROM 驱动器盘符，*locale* 表示期望的语言环境的名称。

**注:** 可能必须创建 `java\security\` 子目录结构。

- 在 UNIX 操作系统上:

从 *DB2 HTML 文档 CD*，将提供的 `/cdrom/program files/IBM/SQLLIB/doc/htmlcd/locale/InvalidParameterException.class` 文件复制到相对于 Netscape 浏览器安装的 `java/classes/java/security/` 目录，其中 *cdrom* 表示 CD-ROM 的安装点，*locale* 表示期望的语言环境的名称。

**注:** 可能必须创建 `java/security/` 子目录结构。

如果 Netscape 浏览器仍无法显示搜索输入窗口，则尝试下列操作:

- 停止 Netscape 浏览器的所有实例以确保机器上无任何 Netscape 代码运行。然后，打开 Netscape 浏览器的新实例并再次尝试启动搜索。
- 清除浏览器的高速缓存。
- 尝试另一版本的 Netscape 或另一浏览器。

### 相关任务:

- 第 328 页的『搜索 DB2 文档』

---

## 搜索 DB2 文档

可搜索 DB2 文档库来定位所需的信息。单击“DB2 信息中心”（从浏览器访问）导航工具栏中的搜索图标时，将打开一个弹出式搜索窗口。可能需要一分钟来装入搜索，取决于您的计算机和网络的速度。

### 先决条件:

需要 Netscape 6.1 或更高版本或者 Microsoft 的 Internet Explorer 5 或更高版本。确保启用了浏览器的 Java 支持。

### 限制:

使用文档搜索时，将存在下列限制:

- 搜索不是区分大小写的。
- 不支持布尔搜索。
- 不支持通配符搜索和部分搜索。例如，对 *java\**（或 *java*）的搜索将仅查找文字字符串 *java\**（或 *java*），而找不到 *javadoc*。

### 过程:

要搜索 DB2 文档:

1. 在导航工具栏中，单击**搜索**图标。
2. 在“搜索”窗口最上面的文本输入字段中，输入一个或多个与您感兴趣的领域相关的词条（由空格分隔），并单击**搜索**。一个按准确度排列的主题列表将显示在**结果**字段中。每一单项旁的数字等级提供了匹配程度的指示（较大的数字表示较高的匹配程度）。

输入较多的项会提高查询的精度，同时还会减少从查询返回的主题数目。

3. 在**结果**列表中，单击要阅读的主题的标题。主题将显示在“DB2 信息中心”的内容框架中。

**注:** 执行搜索时，第一个（最高级别的）结果自动装入到浏览器框架中。要查看其它搜索结果的内容，单击结果列表中的结果。

### 相关任务:

- 第 327 页的『对于使用 Netscape 4.x 搜索 DB2 文档进行故障诊断』

---

## 联机 DB2 故障诊断信息

在 DB2<sup>®</sup> UDB 版本 8 的发行版中，将不再提供 *Troubleshooting Guide*。曾经包含在此指南中的故障诊断信息都已集成到 DB2 出版物中，从而使我们能向您提供最新信息。要查找有关故障诊断实用程序和 DB2 功能的信息，可从任何工具访问“DB2 信息中心”。

如果您遇到问题且想要获取查找可能原因及解决方案的帮助，请参考 Online Support 站点。该支持站点包含了一个不断更新的大型数据库，数据库的内容涉及 DB2 出版物、技术说明、APAR（产品问题）记录、修订包和其它资源。可使用该支持站点来搜索此知识库并查找问题的可能解决方案。

访问 [www.ibm.com/software/data/db2/udb/winos2unix/support](http://www.ibm.com/software/data/db2/udb/winos2unix/support) 站点（网址为 [www.ibm.com/software/data/db2/udb/winos2unix/support](http://www.ibm.com/software/data/db2/udb/winos2unix/support)），或通过单击“DB2 信息中心”中的 **在线支持** 按钮来访问它。现在，还可从此站点获取经常更改的信息，如内部 DB2 错误代码列表。

### 相关概念:

- 第 331 页的『从浏览器访问的 DB2 信息中心』

### 相关任务:

- 第 323 页的『通过从管理工具访问“DB2 信息中心”来查找产品信息』

---

## 易使用性

易使用性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）的用户成功使用软件产品。以下是“DB2<sup>®</sup> 通用数据库版本 8”中主要的易使用性功能部件:

- 通过键盘即可对所有 DB2 功能部件进行操作，而不必使用鼠标。参见第 330 页的『键盘输入和导航』。
- DB2 允许您定制字体的大小和颜色。参见第 330 页的『界面显示的易使用性』。
- DB2 允许您接收可视或音频警告信号。参见第 330 页的『备用警告信号』。
- DB2 支持使用 Java<sup>™</sup> Accessibility API 的易使用性应用程序。参见第 330 页的『与辅助技术的兼容性』。
- DB2 附带了以易使用的格式提供的文档。参见第 330 页的『可访问文档』。

## 键盘输入和导航

### 键盘输入

只使用键盘就可对“DB2 工具”进行操作。使用键或键组合就可执行使用鼠标完成的大多数操作。

### 键盘焦点

在基于 UNIX 的系统中，键盘焦点的位置是突出显示的，指示窗口的哪个区域处于活动状态且击键对何处会有影响。

## 界面显示的易使用性

“DB2 工具”中的功能部件增强了用户界面，使视力不太好的用户更易使用。这些易使用性方面的增强包括了对可定制字体特性的支持。

### 字体设置

“DB2 工具”允许您通过使用“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。

### 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。

## 备用警告信号

可使用“工具设置”笔记本来指定是否想要通过音频或可视信号接收警告。

## 与辅助技术的兼容性

“DB2 工具”界面支持对屏幕阅读器启用 Java Accessibility API 并支持有某些缺陷的用户使用其它辅助技术。

## 可访问文档

DB2 产品系列的文档提供了 HTML 格式的版本。使您可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性技术。

---

## 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:

必须先从 *DB2 HTML* 文档 CD 中安装教程，才能使用以下的链接来访问这些教程。

如果不想安装这些教程，则可直接从 *DB2 HTML* 文档 CD 查看这些教程的 HTML 版本。还可在 *DB2 PDF* 文档 CD 上获取这些教程的 PDF 版本。

某些教程课程使用了样本数据或代码。有关各个教程特定任务的任何先决条件的描述，参见每个教程的内容。

#### **“DB2 通用数据库”教程:**

如果从 *DB2 HTML* 文档 CD 安装了教程，则可单击下表中的某个教程标题来查看该教程。

《商务智能教程：数据仓库中心简介》

使用“数据仓库中心”来执行介绍性的数据入库任务。

《商务智能教程：数据入库的扩展课程》

使用“数据仓库中心”来执行高级数据入库任务。

*Development Center Tutorial for Video Online using Microsoft® Visual Basic*

使用 Microsoft Visual Basic 的“开发中心加载件”来构建应用程序的各个组件。

*Information Catalog Center Tutorial*

使用“信息目录中心”来创建和管理信息目录以定位并使用元数据。

*Video Central for e-business Tutorial*

使用 WebSphere® 产品来开发和部署高级“DB2 Web 服务”应用程序。

《Visual Explain 教程》

使用 Visual Explain 来分析、优化和调整 SQL 语句以获取更好的性能。

---

## 从浏览器访问的 DB2 信息中心

“DB2® 信息中心”让您访问在您的业务中充分利用 DB2 通用数据库™ 和 DB2 Connect™ 所需的所有信息。“DB2 信息中心”文档还记录主要的 DB2 功能部件和组件，包括复制、数据入库、信息目录中心、Life Sciences Data Connect 和 DB2 extender。

从浏览器访问的“DB2 信息中心”具有以下功能部件（如果是在 Netscape Navigator 6.1 或更高版本或者 Microsoft Internet Explorer 5 或更高版本中查看）。某些功能部件需要您启用对 Java 或 JavaScript 的支持：

### 定期更新的文档

通过下载更新的 HTML，使您的主题保持为最新。

**搜索** 通过单击导航工具栏中的**搜索**来搜索安装在工作站上的所有主题。

### 集成的导航树

从一个导航树中就可找出 DB2 资料库中的任何主题。导航树是按信息类型组织的，如下所示：

- “任务”提供了有关如何完成目标的逐步指示信息。
- “概念”提供了主题的概述。
- “参考”主题提供了有关主题的详细信息，包括语句和命令语法、消息帮助以及需求。

**主索引** 从主索引访问从 *DB2 HTML 文档 CD* 中安装的信息。索引是按索引项以字母顺序组织的。

### 主词汇表

主词汇表定义在“DB2 信息中心”中使用的术语。词汇表是按词汇表术语以字母顺序组织的。

### 相关任务:

- 第 321 页的『通过从浏览器访问“DB2 信息中心”来查找主题』
- 第 323 页的『通过从管理工具访问“DB2 信息中心”来查找产品信息』
- 第 325 页的『更新安装在机器上的 HTML 文档』



---

## 附录 B. 声明

IBM 可能在其它国家或地区不提供本文档中讨论的产品、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 代理咨询。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证。您可以用书面方式将许可证查询寄往：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

有关双字节（DBCS）信息的许可证查询，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

**本条款不适用于联合王国或任何这样的条款与当地法律不一致的国家或地区：**国际商业机器公司以“按现状”的基础提供本出版物，不附有任何形式的（无论是明示的，还是默示的）保证，包括（但不限于）对非侵权性、适销性和适用于某特定用途的默示保证。某些国家或地区在某些交易中不允许免除明示或默示的保证。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。该 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 可以它认为合适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它程序（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。

本文档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 国际程序许可证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，在其它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、其出版说明或其它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、兼容性或任何其它关于非 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。

本资料中可能包含用于日常业务运作的数据和报表的示例。为了尽可能完整地说明问题，这些示例可能包含个人、公司、品牌和产品的名称。所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址有雷同，纯属巧合。

版权许可证：

本资料中可能包含源语言的样本应用程序，它们举例说明了各种操作平台上的编程技术。为了开发、使用、营销或分发符合编写这些样本程序所针对操作平台的应用程序编程接口的应用程序，您可以以任何形式复制、修改和分发这些样本程

序，而不必向 IBM 付款。尚未在所有条件下彻底测试这些示例。因此，IBM 不能保证或默示这些程序的可靠性、适用性或功能。

这些样本程序或任何派生产品的每个副本或任何部分都必须包括如下版权声明：

©（您的公司名）（年份）。本代码的某些部分是从“IBM 公司样本程序”派生的。

© 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 / 或其它国家或地区的商标, 且已在 DB2 UDB 文档库中的至少一份文档中使用。

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extender	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational	SystemView
Database Architecture	Tivoli
DRDA	VisualAge
eServer	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
IBM	WebSphere
IMS	WIN-OS/2
IMS/ESA	z/OS
iSeries	zSeries

下列各项是其它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 / 或其它国家或地区的商标。

Intel 和 Pentium 是 Intel Corporation 在美国和 / 或其它国家或地区的商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其它国家或地区的商标。

UNIX 是 The Open Group 在美国和其它国家或地区的注册商标。

其它公司、产品或服务名称可能是其它公司的商标或服务标记。



# 索引

## [ B ]

### 编译器

- 构建文件 89
- 使用 AIX IBM COBOL 171
- 使用 AIX Micro Focus COBOL 179
- 使用 HP-UX Micro Focus COBOL 212
- 使用 Solaris Micro Focus COBOL 267
- 使用 Windows IBM COBOL 295
- 使用 Windows Micro Focus COBOL 301
- 受支持 AIX 版本 9
- 受支持 HP-UX 版本 10
- 受支持 Linux 版本 11
- 受支持 Solaris 版本 13
- 受支持 Windows 版本 14
- makefile 92
- 标志, SQL 92 和 MVS 一致性 3
- 表函数
  - OLE DB 276
- 不可用 329

## [ C ]

### 操作系统

- 受支持 AIX 版本 9
- 受支持 HP-UX 版本 10
- 受支持 Linux 版本 11
- 受支持 Solaris 版本 13
- 受支持 Windows 版本 14
- 受 DB2 支持 8
- DB2 安装路径 44

### 存储过程

- 构建文件 89
- 构建 C
  - AIX 148
  - HP-UX 193
  - Linux 225

### 存储过程 (续)

- 构建 C (续)
  - Solaris 操作环境 247
  - Windows 289
- 构建 COBOL
  - AIX IBM 175
  - AIX Micro Focus 182
  - HP-UX Micro Focus 216
  - Solaris Micro Focus 271
  - Windows IBM 299
  - Windows Micro Focus 305
- 构建 C++
  - AIX 159
  - HP-UX 204
  - Linux 235
  - Solaris 操作环境 259
  - Windows 289
- 构建 JDBC 107
- 构建 SQLJ 119
- 使用 Visual Basic 进行 OLE 自动化 282
- 使用 Visual C++ 进行 OLE 自动化 285
- AIX 上的 VisualAge C++ 配置文件 169
- COBOL 样本程序文件 75
- C/C++ 样本程序文件 65
- JDBC 样本程序文件 70
- SQLJ 样本程序文件 72

## [ D ]

### 订购 DB2 书刊 319

### 对象链接与嵌入

- 数据库表函数
  - 描述 276
  - 样本文件 85
- 样本程序文件 84
- 自动化
  - 使用 Visual Basic 282
  - 使用 Visual C++ 285

### 对象链接与嵌入技术

- 在 DB2 AD 客户机中的支持 3
- 多线程应用程序
  - 构建文件 89
- 多线程应用程序, 构建
  - 使用 AIX C 153
  - 使用 AIX C++ 164
  - 使用 HP-UX C 199
  - 使用 HP-UX C++ 210
  - 使用 Linux C 230
  - 使用 Linux C++ 241
  - 使用 Solaris C 253
  - 使用 Solaris C++ 265

## [ G ]

### 构建文件

- 对于 AIX 上的 C 例程 148
- 对于 AIX 上的 C 应用程序 144
- 对于 AIX 上的 C++ 例程 159
- 对于 AIX 上的 C++ 应用程序 155
- 对于 AIX C 多线程应用程序 153
- 对于 AIX C++ 多线程应用程序 164
- 对于 AIX IBM COBOL 例程 175
- 对于 AIX Micro Focus COBOL 例程 182
- 对于 AIX Micro Focus COBOL 应用程序 179
- 对于 HP-UX 上的 C 例程 193
- 对于 HP-UX 上的 C 应用程序 189
- 对于 HP-UX 上的 C++ 例程 204
- 对于 HP-UX 上的 C++ 应用程序 200
- 对于 HP-UX C 多线程应用程序 199

## 构建文件 (续)

- 对于 HP-UX C++ 多线程应用程序 210
- 对于 HP-UX Micro Focus COBOL 例程 216
- 对于 HP-UX Micro Focus COBOL 应用程序 213
- 对于 Linux 上的 C 例程 225
- 对于 Linux 上的 C 应用程序 221
- 对于 Linux 上的 C++ 例程 235
- 对于 Linux 上的 C++ 应用程序 232
- 对于 Linux C 多线程应用程序 230
- 对于 Linux C++ 多线程应用程序 241
- 对于 Solaris 上的 C 例程 247
- 对于 Solaris 上的 C 应用程序 243
- 对于 Solaris 上的 C++ 应用程序 255, 259
- 对于 Solaris C 多线程应用程序 253
- 对于 Solaris C++ 多线程应用程序 265
- 对于 Solaris Micro Focus COBOL 例程 271
- 对于 Solaris Micro Focus COBOL 应用程序 268
- 对于 Windows 上的 C/C++ 例程 289
- 对于 Windows 上的 C/C++ 应用程序 285
- 对于 Windows IBM COBOL 例程 299
- 对于 Windows IBM COBOL 应用程序 296
- 对于 Windows Micro Focus COBOL 例程 305
- 对于 Windows Micro Focus COBOL 应用程序 302
- 描述 89
- 用于 AIX IBM COBOL 应用程序 172

## 故障诊断

- 联机信息 329
- DB2 文档搜索 327

## [ J ]

教程 330

## [ K ]

可存取性  
功能部件 329

## [ L ]

### 例程

- 构建文件 89
- 构建 AIX IBM COBOL 175
- 构建 AIX Micro Focus COBOL 182
- 构建 HP-UX Micro Focus COBOL 216
- 构建 JDBC 107
- 构建 Solaris Micro Focus COBOL 271
- 构建 SQLJ 119
- 构建 Windows IBM COBOL 299
- 构建 Windows Micro Focus COBOL 305
- 在 AIX 上构建 C 148
- 在 AIX 上构建 C++ 159
- 在 AIX 上装入 COBOL 共享库 143
- 在 HP-UX 上构建 C 193
- 在 HP-UX 上构建 C++ 204
- 在 Linux 上构建 C 225
- 在 Linux 上构建 C++ 235
- 在 Windows 上构建 C/C++ 289
- 在“Solaris 操作环境”上构建 C 247
- 在“Solaris 操作环境”上构建 C++ 259
- AIX 入口点 142
- AIX 上的 CREATE 语句 142
- COBOL 样本程序文件 75
- C/C++ 样本程序文件 65

## 例程 (续)

- JDBC 样本程序文件 70
- SQL 过程样本程序文件 80
- SQLJ 样本程序文件 72
- 例程的入口点, AIX 142
- 联机  
帮助, 存取 320

## [ M ]

命令行处理器 (CLP)  
样本文件 85  
DB2 AD 客户机 3

## [ P ]

配置文件  
对于 AIX 上的 VisualAge C++ 166  
批处理文件  
描述 89

## [ Q ]

迁移应用程序  
应用程序 44  
嵌入式 SQL  
错误检查实用程序文件 95  
构建文件 89  
在 DB2 AD 客户机中的支持 3  
“Solaris 操作环境”上的多线程 C 应用程序 253  
AIX 上的多线程 C 应用程序 153  
AIX 上的多线程 C++ 应用程序 164  
AIX 上的 VisualAge C++ 配置文件 168  
AIX IBM COBOL 例程 175  
AIX IBM COBOL 应用程序 172  
AIX Micro Focus COBOL 例程 182  
AIX Micro Focus COBOL 应用程序 179  
C 例程, 在 AIX 上构建 148  
C 例程, 在 HP-UX 上构建 193



## 嵌入式 SQL (续)

C 例程, 在 Linux 上构建 225  
C 例程, 在 “Solaris 操作环境”  
上构建 247  
C 应用程序, 在 AIX 上构建  
144  
C 应用程序, 在 HP-UX 上构建  
189  
C 应用程序, 在 Linux 上构建  
221  
C 应用程序, 在 “Solaris 操作环  
境” 上构建 243  
COBOL 样本程序文件 75  
C++ 例程, 在 HP-UX 上构建  
204  
C++ 例程, 在 Linux 上构建 235  
C++ 例程, 在 “Solaris 操作环  
境” 上构建 259  
C++ 例程, 在 AIX 上构建 159  
C++ 应用程序, 在 HP-UX 上构  
建 200  
C++ 应用程序, 在 Linux 上构建  
232  
C++ 应用程序, 在 “Solaris 操作  
环境” 上构建 255  
C++ 应用程序, 在 AIX 上构建  
155  
C/C++ 例程, 在 Windows 上构建  
289  
C/C++ 样本程序文件 65  
C/C++ 应用程序, 在 Windows 上  
构建 285  
HP-UX 上的多线程 C 应用程序  
199  
HP-UX 上的多线程 C++ 应用程  
序 210  
HP-UX Micro Focus COBOL 例程  
216  
HP-UX Micro Focus COBOL 应用  
程序 213  
Linux 上的多线程 C 应用程序  
230  
Linux 上的多线程 C++ 应用程序  
241  
Solaris 操作环境 Micro Focus  
COBOL 例程 271

## 嵌入式 SQL (续)

Solaris 操作环境 Micro Focus  
COBOL 应用程序 268  
Solaris 上的多线程 C++ 应用程序  
265  
Windows IBM COBOL 例程 299  
Windows IBM COBOL 应用程序  
296  
Windows Micro Focus COBOL 例  
程 305  
Windows Micro Focus COBOL 应  
用程序 302

## [ R ]

日志管理用户出口样本文件 88

## [ S ]

### 实例

数据库管理器 5

### 实用程序

错误检查文件 95

## [ Y ]

### 样本

#### 程序

目录 53

设计 58

文件扩展名 53

支持的语言 53

Java 样本目录 101

### 样本数据库

绑定 42

编目 42

创建 39

设置 39

在主机系统上创建 41

印刷书刊, 订购 319

### 用户定义函数 (UDF)

构建文件 89

使用 Visual Basic 进行 OLE 自动  
化 282

使用 Visual C++ 进行 OLE 自动  
化 285

## 用户定义函数 (UDF) (续)

AIX 上的 VisualAge C++ 配置文  
件 170

### C

AIX 148

HP-UX 193

Linux 225

Solaris 操作环境 247

COBOL 样本程序文件 75

### C++

AIX 159

HP-UX 204

Linux 235

Solaris 操作环境 259

### C/C++

样本程序文件 65

Windows 289

JDBC 107

JDBC 样本程序文件 70

SQLj 119

SQLj 样本程序文件 72

远程数据对象样本程序文件 82

## [ Z ]

### 主机系统

创建样本数据库 41

受支持服务器 7

## A

### ActiveX 数据对象

使用 Visual Basic 277

使用 Visual C++ 283

样本程序设计 58

在 DB2 AD 客户机中的支持 3

Visual Basic 样本程序文件 82

Visual C++ 样本程序文件 83

### API

“Solaris 操作环境” 上的 C 应用  
程序 243

AIX 上的 C 应用程序 144

AIX 上的 C++ 应用程序 155

AIX 上的 VisualAge C++ 配置文  
件 167

AIX IBM COBOL 应用程序 172

## API (续)

- AIX Micro Focus COBOL 应用程序 179
- COBOL 样本程序文件 75
- C++ 应用程序 “Solaris 操作环境” 255
- C/C++ 样本程序文件 65
- HP-UX 上的 C 应用程序 189
- HP-UX 上的 C++ 应用程序 200
- HP-UX Micro Focus COBOL 应用程序 213
- Linux 上的 C 应用程序 221
- Linux 上的 C++ 应用程序 232
- Solaris 操作环境 Micro Focus COBOL 应用程序 268
- Windows 上的 C/C++ 应用程序 285
- Windows IBM COBOL 应用程序 296
- Windows Micro Focus COBOL 应用程序 302

## applet

- 构建 JDBC 105
- 构建 SQLJ 111
- 杂项提示 103
- JDBC 样本程序文件 70
- SQLJ 样本程序文件 72

## C

### C

- 错误检查实用程序文件 95
- 多线程应用程序
  - AIX 153
  - HP-UX 199
  - Linux 230
  - Solaris 操作环境 253
- 构建文件 89
- 例程
  - 在 AIX 上构建 148
  - 在 HP-UX 上构建 193
  - 在 Linux 上构建 225
  - 在 Windows 上构建 289
  - 在 “Solaris 操作环境” 上构建 247

## C (续)

- 受支持版本
  - AIX 9
  - HP-UX 10
  - Linux 11
  - Solaris 操作环境 13
  - Windows 14
- 样本程序设计 58
- 样本程序文件 65
- 应用程序
  - 在 AIX 上构建 144
  - 在 HP-UX 上构建 189
  - 在 Linux 上构建 221
  - 在 Windows 上构建 285
  - 在 “Solaris 操作环境” 上构建 243
- makefile 92
- CALL 语句
  - 和存储过程 137
- CLI 样本程序文件 68
- COBOL 语言
  - 错误检查实用程序文件 95
  - 构建文件 89
  - 使用 HP-UX Micro Focus 编译器 212
  - 受支持 AIX 版本 9
  - 受支持 HP-UX 版本 10
  - 受支持 Windows 版本 14
  - 受支持 “Solaris 操作环境” 版本 13
  - 样本程序文件 75
  - 在 AIX 上安装和运行 143
  - AIX IBM 编译器 171
  - AIX IBM 例程 175
  - AIX IBM 应用程序 172
  - AIX Micro Focus 编译器 179
  - AIX Micro Focus 例程 182
  - AIX Micro Focus 应用程序 179
  - HP-UX Micro Focus 例程 216
  - HP-UX Micro Focus 应用程序 213
  - makefile 92
  - Solaris Micro Focus 编译器 267
  - Solaris Micro Focus 例程 271
  - Solaris Micro Focus 应用程序 268

## COBOL 语言 (续)

- Windows IBM 编译器 295
- Windows IBM 例程 299
- Windows IBM 应用程序 296
- Windows Micro Focus 编译器 301
- Windows Micro Focus 例程 305
- Windows Micro Focus 应用程序 302
- CREATE 语句
  - 和 AIX 例程 142
- CREATE 语句的 EXTERNAL NAME 子句 142
- C++
  - 错误检查实用程序文件 95
  - 带 Visual C++ 的 ADO 应用程序 283
  - 多线程应用程序
    - AIX 164
    - HP-UX 210
    - Linux 241
    - Solaris 操作环境 265
  - 构建文件 89
  - 例程
    - 在 AIX 上构建 159
    - 在 HP-UX 上构建 204
    - 在 Linux 上构建 235
    - 在 Windows 上构建 289
    - 在 “Solaris 操作环境” 上构建 259
  - 使用 Visual C++ 进行 OLE 自动化 285
  - 受支持 AIX 版本 9
  - 受支持 HP-UX 版本 10
  - 受支持 Linux 版本 11
  - 受支持 Windows 版本 14
  - 受支持 “Solaris 操作环境” 版本 13
  - 样本程序设计 58
  - 样本程序文件 65
  - 应用程序
    - 在 AIX 上构建 155
    - 在 HP-UX 上构建 200
    - 在 Linux 上构建 232
    - 在 Windows 上构建 285

## C++ (续)

### 应用程序 (续)

在“Solaris 操作环境”上构建  
255

AIX 上的 VisualAge 配置文件  
166

makefile 92

## D

### DB2 的 OLE DB 提供程序

对于带 Visual Basic 的 DB2 277

对于带 Visual Basic 的  
ODBC 277

对于带 Visual C++ 的 DB2 283

DB2 教程 330

### DB2 文档搜索

使用 Netscape 4.x 327

DB2 信息中心 331

DB2 CLI 样本程序文件 68

DB2INSTANCE 环境变量 39

DB2INSTPROF 环境变量 5

DB2PATH 环境变量 5

DB2\_SQLROUTINE\_COMPILE

\_COMMAND 环境变量 21

DB2\_SQLROUTINE\_COMPILER

\_PATH 环境变量 21

## F

### FORTRAN 语言

DB2 支持 8

## G

GET ROUTINE CLP 命令 134

GET\_ROUTINE\_SAR 内置存储过程  
134

## J

### Java

插件样本文件 75

构建

JDBC 例程 107

## Java (续)

### 构建 (续)

JDBC 应用程序 106

JDBC applet 105

SQLj 例程 119

SQLj 应用程序 110, 113

SQLj applet 111

环境设置 20

受支持 AIX JDK 版本 9

受支持 HP-UX JDK 版本 10

样本

程序设计 58

目录 101

支持, DB2 AD 客户机 3

AIX 环境设置 25

applet

使用要点 103

HP-UX 环境设置 26

JDBC

样本程序文件 70

Linux

环境设置 26

受支持 JDK 版本 11

makefile 92

Solaris 操作环境

设置 27

受支持 JDK 版本 13

SQLj (Java 嵌入式 SQL)

样本程序文件 72

UNIX 环境设置 24

WebSphere 样本文件 74

Windows

环境设置 35

受支持 JDK 版本 14

### JDBC

构建例程 107

构建应用程序 106

构建 applet 105

样本程序文件 70

在 DB2 AD 客户机中的支持 3

applet, 使用要点 103

JDK\_PATH, 数据库管理器配置关键字 19

## K

KEEPFENCED, 数据库管理器配置关键字 19

## M

### makefile

描述 92

命令 92

## N

NOCONVERT 选项 275

## O

### OLE DB 提供程序

对于带 Visual C++ 的

ODBC 283

## P

### Perl

DB2 支持 8

PHP, DB2 支持 8

PUT ROUTINE CLP 命令 134

PUT\_ROUTINE\_SAR 存储过程 134

## R

RDO 应用程序, 使用 Visual Basic 构建 280

### REXX 语言

构建 AIX 应用程序 186

构建 Windows 应用程序 308

受支持 AIX 版本 9

受支持 Windows 版本 14

在 DB2 AD 客户机中的支持 3

DB2 支持 8

## S

### SQL 过程

保留中间文件 131

SQL 过程 (续)	Windows Management
备份和复原 133	Instrumentation (WMI)
创建 127	样本 277
定制预编译和绑定选项 132	
分发已编译内容 134	
环境设置 21	
使用 UNIX 客户机应用程序调用	
128	
使用 Windows 客户机应用程序调用	
129	
样本程序文件 80	
CALL 语句 137	
UNIX 环境设置 28	
Windows 环境设置 37	
SQLJ (Java 嵌入式 SQL)	
程序	
构建 110	
构建文件 89	
例程, 构建 119	
样本程序文件 72	
应用程序	
构建 113	
在 DB2 AD 客户机中的支持 3	
applet	
构建 111	
使用 103	

## V

### Visual Basic

构建 ADO 应用程序 277
构建 RDO 应用程序 280
受支持 Windows 版本 14
样本程序设计 58
样本程序文件 82
OLE 自动化 282
Visual C++ 样本程序文件 83

## W

WCHARTYPE CONVERT 预编译选项 275
wchar_t 数据类型
转换预编译选项 275
Windows 上的 Object REXX 308

---

## 与 IBM 联系

在美国，请致电下列其中一个号码以与 IBM 联系：

- 1-800-237-5511，可获取客户服务
- 1-888-426-4343，可了解所提供的服务项目
- 1-800-IBM-4YOU (426-4968)，可获取有关 DB2 市场营销与销售的信息

在加拿大，请致电下列其中一个号码以与 IBM 联系：

- 1-800-IBM-SERV (1-800-426-7378)，可获取客户服务
- 1-800-465-9600，可了解所提供的服务项目
- 1-800-IBM-4YOU (1-800-426-4968)，可获取有关 DB2 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，可查看 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)。

---

## 产品信息

有关“DB2 通用数据库”产品的信息，可打电话获取或通过万维网获取，网址为：[www.ibm.com/software/data/db2/udb](http://www.ibm.com/software/data/db2/udb)。

此站点包含有关技术库、订购书刊、客户机下载、新闻组、修订包、新闻和 Web 资源链接的最新信息。

您如果住在美国，请致电下列其中一个号码：

- 1-800-IBM-CALL (1-800-426-2255)，可订购产品或获取一般信息。
- 1-800-879-2755，可订购出版物。

有关如何在美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: CT17WSC

中国印刷

S152-0168-00



(1P) P/N: CT17WSC



Spine information:



IBM® DB2 通用数据库™

构建和运行应用程序

版本 8