

IBM® DB2® 通用数据库



# 管理指南：实现

版本 7



IBM® DB2® 通用数据库



# 管理指南：实现

版本 7

在使用本资料及其支持的产品之前，请阅读第417页的『附录M. 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。

通过您当地的 IBM 代表或 IBM 分部可订购出版物，或者，通过致电 1-800-879-2755（在美国）或 1-800-IBM-4YOU（在加拿大）来订购出版物。

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

© Copyright International Business Machines Corporation 1993, 2001. All rights reserved.

# 目录

关于本书 . . . . .	ix
谁应使用本书 . . . . .	x
本书的结构 . . . . .	x
“管理指南”的其他卷的简要概述 . . . . .	xi
管理指南: 计划 . . . . .	xi
管理指南: 性能 . . . . .	xii

## 第1部分 使用“控制中心”进行管理 . . . . . 1

<b>第1章 使用 GUI 工具管理 DB2.</b> . . . . .	<b>3</b>
管理工具 . . . . .	4
公共工具功能部件 . . . . .	6
显示 SQL 和显示命令 . . . . .	6
显示相关项 . . . . .	7
生成 DDL . . . . .	7
过滤器 . . . . .	8
帮助 . . . . .	10
控制中心 . . . . .	10
“控制中心”的主要元素 . . . . .	11
在 DB2 OS/390 版中使用“定制的控制中 心” . . . . .	11
可管理的系统 . . . . .	12
可管理的对象 . . . . .	12
在“控制中心”中显示系统 . . . . .	13
管理 DB2 OS/390 版对象 . . . . .	14
添加 DB2 OS/390 版子系统 . . . . .	14
管理网关连接 . . . . .	15
可从“控制中心”执行的功能 . . . . .	15
创建新对象 . . . . .	16
使用现存的对象 . . . . .	16
查找对象 (仅适用于 DB2 OS/390 版) . . . . .	16
卫星管理中心 . . . . .	17
命令中心 . . . . .	18
脚本中心 . . . . .	18
用“脚本中心”处理现存的脚本 . . . . .	19
调度保存的命令脚本运行 . . . . .	19
日志 . . . . .	19
使用作业 . . . . .	20
许可证中心 . . . . .	20
警告中心 . . . . .	21
客户机配置辅助程序 . . . . .	21

性能监视器 . . . . .	22
事件监视器 . . . . .	23
使用监视器工具 . . . . .	23
监视某个时间点的性能 . . . . .	25
预定义监视器 . . . . .	26
当一个对象出现在“警告中心”时需要执行 的操作 . . . . .	28
在一段时间内分析事件 . . . . .	28
事件分析程序 . . . . .	28
分析 SQL 语句 . . . . .	30
改进查询性能 . . . . .	31
分析简单的动态 SQL 语句 . . . . .	32
管理远程数据库 . . . . .	32
管理用户 . . . . .	34
授予和取消权限和特权 . . . . .	34
移动数据 . . . . .	35
管理存储器 . . . . .	37
估计表和索引大小 . . . . .	37
检查表空间中的可用空间 . . . . .	38
向表空间添加更多空间 . . . . .	38
故障诊断 . . . . .	39
复制数据 . . . . .	39
使用轻量级目录访问协议 . . . . .	40
使用 Java 控制中心 . . . . .	41
将“控制中心”作为 Java 小应用程序运行 . . . . .	41
使用 Java 工具进行管理 . . . . .	41

## 第2部分 实现设计 . . . . . 43

<b>第2章 在创建数据库之前.</b> . . . . .	<b>45</b>
创建数据库之前的先决条件 . . . . .	46
启动 DB2 . . . . .	46
在 Windows NT 上启动 DB2 UDB . . . . .	47
使用数据库管理器的多个实例 . . . . .	47
按模式组织和分组对象 . . . . .	48
启用并行性 . . . . .	49
启用数据分区 . . . . .	50
停止 DB2 . . . . .	52
有关创建数据库的详细信息 . . . . .	53
设计逻辑数据库和物理数据库的特征 . . . . .	53
创建实例 . . . . .	54

许可证管理 . . . . .	61	创建函数模板 . . . . .	124
建立环境变量和简要表注册表 . . . . .	61	创建用户定义类型 (UDT) . . . . .	125
创建 DB2 管理服务服务器 (DAS) . . . . .	69	创建用户定义单值类型 . . . . .	125
创建节点配置文件 . . . . .	84	创建用户定义结构化类型 . . . . .	126
创建数据库配置文件 . . . . .	86	创建类型映射 . . . . .	126
使用响应文件复制配置信息 . . . . .	87	创建视图 . . . . .	127
启用 FCM 通信 . . . . .	87	创建类型视图 . . . . .	129
<b>第3章 创建数据库 . . . . .</b>	<b>89</b>	创建摘要表 . . . . .	129
初始节点组的定义 . . . . .	90	创建别名 . . . . .	131
初始表空间的定义 . . . . .	91	创建包装器 . . . . .	132
系统目录表的定义 . . . . .	92	创建服务器 . . . . .	133
数据库目录的定义 . . . . .	93	使用服务器选项来帮助定义数据源并简化认证处理 . . . . .	134
本地数据库目录 . . . . .	93	创建别名 . . . . .	139
系统数据库目录 . . . . .	93	引用别名和数据源对象 . . . . .	140
节点目录 . . . . .	94	使用别名和数据源对象 . . . . .	140
DCE 目录服务 . . . . .	94	标识现存的别名和数据源 . . . . .	140
“轻量级目录访问协议” (LDAP) 的目录服务 . . . . .	95	创建索引、索引扩充或索引规范 . . . . .	141
创建节点组 . . . . .	95	使用索引 . . . . .	144
数据库恢复日志的定义 . . . . .	96	使用 CREATE INDEX 语句 . . . . .	145
将实用程序与数据库绑定 . . . . .	96	创建用户定义扩充索引类型 . . . . .	147
编目数据库 . . . . .	97	有关索引维护的详细信息 . . . . .	147
创建表空间 . . . . .	97	有关索引搜索的详细信息 . . . . .	148
创建系统临时表空间 . . . . .	100	有关索引利用的详细信息 . . . . .	148
创建用户临时表空间 . . . . .	100	定义索引扩充的一个方案 . . . . .	149
在节点组中创建表空间 . . . . .	101		
原始 I/O . . . . .	101	<b>第4章 改变数据库 . . . . .</b>	<b>153</b>
创建模式 . . . . .	103	在改变数据库之前 . . . . .	153
设置模式 . . . . .	104	更改逻辑设计和物理设计特征 . . . . .	153
创建和填充表 . . . . .	104	更改许可证信息 . . . . .	153
大对象 (LOB) 列的注意事项 . . . . .	107	更改实例 . . . . .	153
定义约束 . . . . .	108	更改环境变量和简要表注册表变量 . . . . .	156
在新表上定义生成列 . . . . .	112	更改节点配置文件 . . . . .	157
创建用户定义的临时表 . . . . .	113	更改数据库配置 . . . . .	157
在新表上定义标识列 . . . . .	114	改变数据库 . . . . .	158
创建序列 . . . . .	115	删除数据库 . . . . .	159
比较 IDENTITY 列和序列 . . . . .	117	改变节点组 . . . . .	159
创建类型表 . . . . .	117	改变表空间 . . . . .	159
填充类型表 . . . . .	117	删除模式 . . . . .	164
层次结构表 . . . . .	117	修改表的结构和内容 . . . . .	165
在多个表空间中创建一个表 . . . . .	118	改变用户定义结构化类型 . . . . .	179
在分区数据库中创建表 . . . . .	119	删除和更新类型表的行 . . . . .	179
创建触发器 . . . . .	120	重命名现存表 . . . . .	179
触发器从属性 . . . . .	121	删除表 . . . . .	180
创建用户定义函数 (UDF) 或方法 . . . . .	122	删除用户定义临时表 . . . . .	182
创建函数映射 . . . . .	123	删除触发器 . . . . .	182

删除用户定义函数 (UDF)、类型映射或方法 . . . . .	182
删除用户定义类型 (UDT) 或类型映射 . . . . .	183
改变或删除视图 . . . . .	183
删除摘要表 . . . . .	185
删除包装器 . . . . .	186
改变或删除服务器 . . . . .	186
改变或删除别名 . . . . .	187
删除索引、索引扩充或索引规范 . . . . .	188
更改对象时的语句从属性 . . . . .	189

### 第3部分 数据库安全性 . . . . . 191

#### 第5章 控制数据库访问 . . . . . 193

为安装选择用户标识和组 . . . . .	193
Windows NT 平台注意事项 . . . . .	195
UNIX 平台注意事项 . . . . .	195
一般规则 . . . . .	196
为服务器选择认证方法 . . . . .	196
远程客户机的认证注意事项 . . . . .	200
分区数据库注意事项 . . . . .	201
使用“DCE 安全性服务”来认证用户 . . . . .	201
如何设置 DB2 用户来使用 DCE . . . . .	202
如何设置 DB2 服务器来使用 DCE . . . . .	203
如何设置 DB2 客户机实例来使用 DCE . . . . .	205
使用 DCE 安全性的 DB2 限制 . . . . .	205
联合数据库认证处理 . . . . .	206
认证设置 . . . . .	206
将用户标识和密码传送到数据源 . . . . .	207
联合数据库认证示例 . . . . .	209
特权、权限和授权 . . . . .	211
系统管理权限 (SYSADM) . . . . .	213
系统控制权限 (SYSCTRL) . . . . .	214
系统维护权限 (SYSMAINT) . . . . .	215
数据库管理权限 (DBADM) . . . . .	215
LOAD 权限 . . . . .	216
数据库特权 . . . . .	216
模式特权 . . . . .	218
表空间特权 . . . . .	220
表和视图特权 . . . . .	220
别名特权 . . . . .	222
服务器特权 . . . . .	222
软件包特权 . . . . .	223
索引特权 . . . . .	223
序列特权 . . . . .	224
控制对数据库对象的访问 . . . . .	224

授予特权 . . . . .	224
取消特权 . . . . .	225
通过创建和删除对象来管理隐式授权 . . . . .	226
建立计划或软件包的所有权 . . . . .	227
允许通过软件包授予间接特权 . . . . .	227
允许通过包含别名的软件包授予间接特权 . . . . .	228
使用视图控制对数据的访问 . . . . .	229
使用审计设施监视对数据的访问 . . . . .	231
数据加密 . . . . .	232
任务和必需的授权 . . . . .	233
使用系统目录 . . . . .	234
检索具有授予的特权的授权名 . . . . .	234
检索具有 DBADM 权限的全部名称 . . . . .	235
检索被授权访问表的名称 . . . . .	235
检索授予用户的全部特权 . . . . .	236
保密系统目录视图 . . . . .	236

#### 第6章 审计 DB2 活动 . . . . . 239

审计设施的行为 . . . . .	241
审计设施使用方案 . . . . .	242
审计设施消息 . . . . .	246
审计设施记录条目的布局 . . . . .	246
审计设施提示和技术 . . . . .	261
控制 DB2 审计设施活动 . . . . .	262

### 第4部分 移动数据 . . . . . 267

#### 第7章 数据移动实用程序 . . . . . 269

### 第5部分 恢复 . . . . . 271

#### 第8章 恢复数据库 . . . . . 273

### 第6部分 附录 . . . . . 275

#### 附录A. 命名规则 . . . . . 277

一般命名规则 . . . . .	277
对象命名规则 . . . . .	277
关于模式名的附加信息 . . . . .	278
关于密码的附加信息 . . . . .	280
在对象名中使用定界标识符 . . . . .	280
如何在联合系统中保留区分大小写的值 . . . . .	280

#### 附录B. 使用分布式计算环境 (DCE) 的目录服务 . . . . . 283

创建目录对象 . . . . .	283
------------------	-----

数据库对象 . . . . .	284	使用服务器认证的样本方案: . . . . .	324
数据库定位器对象 . . . . .	285	使用客户机认证和 Windows NT 客户机的样 本方案: . . . . .	324
路由信息对象 . . . . .	286	使用客户机认证和 Windows 95 客户机的样本 方案: . . . . .	325
每个对象类的属性 . . . . .	288	对 DB2 使用备份域控制器 . . . . .	325
有关每个属性的详细信息 . . . . .	289	使用 DB2 Windows NT 版的用户认证 . . . . .	326
目录服务安全性 . . . . .	292	用户名和组名限制 . . . . .	326
配置参数和注册表变量 . . . . .	294	DB2 Windows NT 版安全性服务 . . . . .	326
CATALOG 和 ATTACH 命令以及 CONNECT 语句 . . . . .	295	在备份域控制器上安装 DB2 . . . . .	327
CATALOG GLOBAL DATABASE 命令 . . . . .	295	使用组和域安全性的认证 . . . . .	328
CONNECT 语句 . . . . .	296	<b>附录F. 使用 Windows NT 性能监视器 . . . 331</b>	
ATTACH 命令 . . . . .	296	向 Windows NT 性能监视器注册 DB2 . . . . .	331
客户机如何与数据库连接 . . . . .	296	允许远程访问 DB2 性能信息 . . . . .	332
与相同单元中的数据库连接 . . . . .	298	显示 DB2 和 DB2 Connect 性能值 . . . . .	332
与不同单元中的数据库连接 . . . . .	299	访问远程 DB2 性能信息 . . . . .	333
如何搜索目录 . . . . .	300	复位 DB2 性能值 . . . . .	334
ATTACH 命令 . . . . .	300	<b>附录G. 使用 Windows NT 或 Windows 2000 数据库分区服务器 . . . . . 335</b>	
CONNECT 语句 . . . . .	301	列示实例中的数据库分区服务器 . . . . .	335
临时替换 DCE 目录信息 . . . . .	303	向实例添加数据库分区服务器 . . . . .	335
目录服务任务 . . . . .	303	更改数据库分区 . . . . .	337
DCE 管理员任务 . . . . .	303	从实例中删除数据库分区 . . . . .	338
数据库管理员任务 . . . . .	304	<b>附录H. 配置多个逻辑节点 . . . . . 341</b>	
数据库用户任务 . . . . .	305	<b>附录I. 高速节点间通信 . . . . . 343</b>	
目录服务的限制 . . . . .	306	使用 TCP/IP 的高速互连 . . . . .	344
<b>附录C. 数据库恢复的用户出口 . . . . . 307</b>		使用 IBM Netfinity SP Switch 的先决条件 . . . . .	344
<b>附录D. 向多个数据库分区发出命令 . . . . . 309</b>		使用 VI 的高速互连 . . . . .	345
命令 . . . . .	309	虚拟接口 (VI) 硬件设置 . . . . .	346
命令描述 . . . . .	310	启用 DB2 以使用 VI 运行 . . . . .	353
指定要运行的命令 . . . . .	310	<b>附录J. “轻量级目录访问协议” (LDAP) 的目 录服务 . . . . . 355</b>	
在基于 UNIX 的平台上以并行方式运行命 令 . . . . .	311	支持 LDAP 客户机和服务器配置 . . . . .	355
在基于 UNIX 的平台上监视 rah 进程 . . . . .	312	对 Windows 2000 活动目录的支持 . . . . .	356
附加 rah (运行所有主机) 信息 (仅适用于 Solaris 和 AIX) . . . . .	313	配置 DB2, 以使用活动目录 . . . . .	356
前缀序列 . . . . .	313	在 IBM LDAP 环境中配置 DB2 . . . . .	357
指定机器列表 . . . . .	316	创建 LDAP 用户 . . . . .	357
从机器列表中删除重复项 . . . . .	316	为 DB2 应用程序配置 LDAP 用户 . . . . .	358
控制 rah 命令 . . . . .	317	在安装之后注册 DB2 服务器 . . . . .	359
在基于 UNIX 平台上的 \$RAHDOTFILES . . . . .	318	更新 DB2 服务器的协议信息 . . . . .	360
在 Windows NT 上设置缺省环境简表 . . . . .	319	编目节点别名以便连接 . . . . .	361
在基于 UNIX 的平台上确定 rah 中的问题 . . . . .	319	注销 DB2 服务器 . . . . .	361
<b>附录E. DB2 Windows NT 版如何使用 Windows NT 的安全性 . . . . . 323</b>			



注册数据库 . . . . .	361	MyInstance.java . . . . .	393
连接远程服务器 . . . . .	362	MyDB2.java . . . . .	394
注销数据库 . . . . .	362	MyDatabases.java . . . . .	395
刷新本地数据库和节点目录中的 LDAP 项 搜索 . . . . .	362 363	MySYSPLAN.java . . . . .	395
注册主机数据库 . . . . .	364	MyTable.java . . . . .	396
在用户级设置 DB2 注册表变量 . . . . .	365	MyDBUser.java . . . . .	397
安装完成后启用 LDAP 支持 . . . . .	366	MyToolBarAction.java . . . . .	398
禁用 LDAP 支持 . . . . .	367	MyAlterAction.java . . . . .	398
LDAP 支持和 DB2 Connect . . . . .	367	MyAction.java . . . . .	398
安全性注意事项 . . . . .	367	MyDropAction.java . . . . .	399
Windows 2000 活动目录的安全性注意事项	368	MyCascadeAction.java . . . . .	399
扩展具有 DB2 对象类和属性的目录模式 . . . . .	369	MyCreateAction.java . . . . .	400
扩展“IBM eNetwork 目录”版本 2.1 的目 录模式 . . . . .	369 369	<b>附录L. 使用 DB2 资料库 . . . . .</b>	<b>401</b>
扩展 Windows 2000 活动目录的目录模式	369	DB2 PDF 文件和打印的书籍 . . . . .	401
Windows 2000 活动目录中的 DB2 对象	371	DB2 信息 . . . . .	401
DB2 使用的对象类和属性 . . . . .	371	打印 PDF 书籍 . . . . .	409
<b>附录K. 扩展控制中心 . . . . .</b>	<b>385</b>	订购打印书籍 . . . . .	410
性能注意事项 . . . . .	385	DB2 联机文档 . . . . .	411
封装注意事项 . . . . .	385	访问联机帮助 . . . . .	411
接口描述 . . . . .	385	查看联机信息 . . . . .	412
CCExtension . . . . .	386	使用 DB2 向导 . . . . .	414
CCObject . . . . .	387	设置文档服务器 . . . . .	415
CCMenuAction . . . . .	390	搜索联机信息 . . . . .	416
CCToolBarAction . . . . .	390	<b>附录M. 声明 . . . . .</b>	<b>417</b>
使用方案 . . . . .	391	商标 . . . . .	419
MyExtension.java . . . . .	392	<b>索引 . . . . .</b>	<b>421</b>
MySample.java . . . . .	392	<b>与 IBM 联系 . . . . .</b>	<b>433</b>
MyDatabaseActions.java . . . . .	393	产品信息 . . . . .	433



---

## 关于本书

本管理指南在它的三卷中提供了必要的参考信息，以便使用和管理 2000 年就绪的 DB2\* 关系数据库管理系统 (RDBMS) 产品，包括：

- 关于数据库设计的信息（可在《管理指南：计划》中找到）
- 关于实现和管理数据库的信息（可在《管理指南：实现》中找到）
- 关于配置和调整数据库环境以改进性能的信息（可在《管理指南：性能》中找到）。

本书中描述的许多任务可以使用不同的界面来执行：

- **命令行处理器**，它允许您从图形界面访问和操纵数据库。从此界面，您也可执行 SQL 语句和 DB2 实用程序功能。本书中的大多数示例说明了如何使用此界面。有关使用命令行处理器的更多信息，参见 *Command Reference*。
- **应用程序编程接口**，它允许您在应用程序内执行 DB2 实用程序功能。有关使用应用程序编程接口的更多信息，参见 *Administrative API Reference* 手册。
- **控制中心**，它允许您以图形方式执行管理任务，如配置系统、管理目录、备份和恢复系统、调度作业以及管理媒体。控制中心还包含复制管理，以便以图形方式设置系统之间数据的复制。此外，“控制中心”使您能通过图形用户界面执行 DB2 实用程序功能。根据平台的不同，调用“控制中心”的方法也不同。例如，在命令行上使用 db2cc 命令（在 OS/2 上）、从 DB2 文件夹选择控制中心图标，或在 Windows 平台上使用开始屏面。要获得介绍性帮助，从控制中心窗口的帮助下拉菜单中选择入门。**Visual Explain** 和**性能监视器**工具是从控制中心中调用的。

还可使用其他工具来执行管理任务。它们包括：

- **脚本中心**，用于存储称为脚本的小应用程序。这些脚本可包含 SQL 语句、DB2 命令以及操作系统命令。
- **警告中心**，用于监视其他 DB2 操作产生的消息。
- **工具设置**，用于更改“控制中心”、“警告中心”和“复制”的设置。
- **日志**，用于调度要以无人照管方式运行的作业。
- **数据仓库中心**，用于管理仓库对象。

---

## 谁应使用本书

本书主要面向数据库管理员、系统管理员、安全性管理员和系统操作员，他们需要设计、实施和维护由本地或远程客户机访问的数据库。需要了解 DB2 关系数据库管理系统的管理和操作的程序员和其他用户也可使用本书。

---

## 本书的结构

本书包含关于下列主要主题的信息：

### 使用“控制中心”进行管理

- 第1章 使用 GUI 工具管理 DB2，介绍用来管理数据库的图形用户界面 (GUI) 工具。

### 实现设计

- 第2章 在创建数据库之前，描述在创建数据库和数据库中的对象之前所需的先决条件。
- 第3章 创建数据库，描述与创建数据库和数据库中的对象相关联的任务。
- 第4章 改变数据库，描述与改变和删除数据库以及数据库中的对象相关联的先决条件和任务。

### 数据库安全性

- 第5章 控制数据库访问，描述如何控制对数据库资源的访问。
- 第6章 审计 DB2 活动，描述如何检测和监视对数据的不需要的访问或不期望的访问。

### 移动数据

- 第7章 数据移动实用程序，描述“装入”、“自动装入程序”、“导入”和“导出”实用程序。

### 恢复

- 第8章 恢复数据库，描述当选择数据库和表空间恢复方法时要考虑的因素。恢复任务包括备份和复原数据库或表空间以及使用前滚恢复。

### 附录

- 附录A. 命名规则，介绍命名数据库和对象时需遵循的规则。
- 附录B. 使用分布式计算环境 (DCE) 的目录服务，提供有关如何使用“DCE 目录服务”的信息。
- 附录C. 数据库恢复的用户出口，讨论如何对数据库日志文件使用用户出口程序，并描述一些样本用户出口程序。

- 附录D. 向多个数据库分区发出命令，讨论如何使用 *db2\_all* 和 *rah shell* 脚本将命令发送至一个分区数据库环境中的所有分区。
- 附录E. DB2 Windows NT 版如何使用 Windows NT 的安全性，描述 DB2 如何使用 Windows NT 安全性。
- 附录F. 使用 Windows NT 性能监视器，描述 DB2 Windows NT 版用户可使用的两个性能监视器：“DB2 性能监视器”和“Windows NT 性能监视器”。
- 附录G. 使用 Windows NT 或 Windows 2000 数据库分区服务器，描述 Windows NT 和 Windows 2000 用来使用分区数据库服务器的实用程序。
- 附录H. 配置多个逻辑节点，描述如何在分区数据库环境中配置多个逻辑节点。
- 附录I. 高速节点间通信，描述如何在 Windows NT 环境中启用“虚拟接口体系结构”以用于 DB2 扩充企业版。
- 附录J. “轻量级目录访问协议” (LDAP) 的目录服务，提供关于如何使用“LDAP 目录服务”的信息。
- 附录K. 扩展控制中心，提供有关如何通过添加包含新操作的新工具栏按钮、添加新对象定义及添加新操作定义来扩展控制中心的信息。
- 附录L. 使用 DB2 资料库，提供关于 DB2 库的结构资料，包括向导、联机帮助、消息和书籍。

---

## “管理指南”的其他卷的简要概述

### 管理指南：计划

《管理指南：计划》主要讨论数据库设计。它讨论了逻辑和物理设计问题；分布式事务问题；以及高可用性主题。下面简要描述该卷中的特定章节和附录：

#### “DB2 通用数据库”的世界

- “管理 DB2 通用数据库”提供“DB2 通用数据库”的介绍和概述。

#### 数据库概念

- “基本关系数据库概念”提供数据库对象（包括恢复对象、存储器对象和系统对象）的概述。
- “联合系统”讨论联合系统，联合系统是一个数据库管理系统 (DBMS)，它支持提交特定 SQL 语句的应用程序和用户，这些 SQL 语句在单条语句中引用两个或多个 DBMS 或数据库。
- “并行数据库系统”提供有关 DB2 可使用的并行性类型的介绍。
- “关于数据入库”提供数据入库和数据入库任务的概述。

- “关于 Spatial Extender” 通过说明 Spatial Extender 的用途以及讨论它所处理的数据对 Spatial Extender 作了介绍。

### 数据库设计

- “逻辑数据库设计” 讨论逻辑数据库设计的概念和准则。
- “物理数据库设计” 讨论物理数据库设计的准则，包括与数据存储相关的注意事项。

### 分布式事务处理

- “设计分布式数据库” 讨论如何在单个事务中访问多个数据库。
- “针对事务管理器的设计” 讨论如何在分布式事务处理环境（如 CICS）中使用数据库。

### 高可用性系统

- “介绍高可用性和故障转移支持” 概述了 DB2 提供的高可用性故障转移支持。

### 附录

- “计划数据库迁移” 描述关于将数据库迁移至版本 7 的信息。
- “发行版间的不兼容性” 描述不同发行版（最高到版本 7）之间的不兼容性。
- “本地语言支持” (NLS) 描述 “DB2 本地语言支持”，包括有关国家或地区、语言和代码页的信息。

## 管理指南：性能

《管理指南：性能》主要讨论性能问题，如建立、测试和改进应用程序的性能，以及改进 “DB2 通用数据库” 产品本身的性能。下面简要描述该卷中的特定章节和附录：

### 性能介绍

- “性能元素” 介绍有关管理和改进 DB2 UDB 性能的概念和注意事项。
- “体系结构和处理概述” 介绍下层 “DB2 通用数据库” 体系结构和处理。

### 调整应用程序性能

- “应用程序注意事项” 描述在设计应用程序时用于改善数据库性能的一些技术。
- “环境注意事项” 描述在设置数据库环境时用于改善数据库性能的一些技术。
- “系统目录统计信息” 描述如何收集并使用数据统计信息以确保最优性能。
- “了解 SQL 编译程序” 描述使用 SQL 编译程序编译 SQL 语句的过程。

- “SQL 解释设施”描述“解释”设施，它允许您检查 SQL 编译程序为访问数据所做的选择。

## 调整和配置系统

- “运行性能”概述数据库管理器如何使用内存以及影响运行时性能的其他注意事项。
- “使用控制器”介绍如何使用控制器来控制数据库管理的某些方面。
- “调整配置”描述与增加数据库系统的大小相关的一些注意事项和任务。
- “将数据重新分布到各分区中”讨论在分区数据库环境中将数据重新分布在各分区中所需的任务。
- “基准测试”提供有关基准测试和如何执行基准测试的概述。
- “配置 DB2”讨论数据库管理器、数据库配置文件和配置参数的值。

## 附录

- “DB2 注册表和环境变量”描述简要表的注册表值和环境变量。
- “解释表和定义”描述“DB2 解释”设施使用的表以及如何创建那些表。
- “SQL 解释工具”描述如何使用 DB2 解释工具：db2expln 和 dynexpln。
- “db2exfmt - 解释表格式工具”描述如何使用 DB2 解释工具来格式化解释表数据。





---

## 第1部分 使用“控制中心”进行管理



---

## 第1章 使用 GUI 工具管理 DB2

“DB2 通用数据库”提供了图形用户界面 (GUI) 工具，帮助您从一个称为“控制中心”的中央位置轻松管理本地和远程数据库。

本章概述您可使用的“DB2 通用数据库”管理工具，并说明如何使用它们来简便有效地完成作业。还提供了“Java 控制中心”的摘要，并讲述如何定制“控制中心”来包含您自己的启用 Java 的工具。

本章提供有关下列内容的信息：

- 第4页的『管理工具』
- 第6页的『公共工具功能部件』
- 第10页的『控制中心』
- 第17页的『卫星管理中心』
- 第18页的『命令中心』
- 第18页的『脚本中心』
- 第19页的『日志』
- 第20页的『许可证中心』
- 第21页的『警告中心』
- 第21页的『客户机配置辅助程序』
- 第22页的『性能监视器』
- 第32页的『管理远程数据库』
- 第34页的『管理用户』
- 第35页的『移动数据』
- 第37页的『管理存储器』
- 第39页的『故障诊断』
- 第39页的『复制数据』
- 第40页的『使用轻量级目录访问协议』
- 第41页的『使用 Java 控制中心』
- 第41页的『使用 Java 工具进行管理』

管理 DB2 的工具是“管理客户机”的一部分，“管理客户机”是大多数“DB2 通用数据库”产品的可选组件。“管理客户机”也可在一套 CD-ROM 中获得，它们包括 DB2 适用的所有操作系统平台上的“管理客户机”。它们允许您在任何工作站上安装和使用“管理客户机”：无论您的数据库服务器是本地的还是远程的，无论这些数据库服务器在什么操作系统上运行，都无关紧要。这些工具使您可从“图形用户界面”执行从“命令行处理器”可执行的相同功能。这些功能包括输入 DB2 命令、SQL 语句或系统命令。但是，使用这些工具，您不必记住复杂的语句或命令，并可获得附加的辅助。

**注：**“管理客户机”是一个安装选项。

下列工具可从“控制中心”工具栏找到：

- 控制中心。“控制中心”是管理数据库的主要 DB2 图形工具。从“控制中心”，可获得本地编目的所有系统和数据库对象的清晰概要。
- 卫星管理中心。“卫星管理中心”允许您管理 DB2 卫星服务器。
- 命令中心。“命令中心”允许您发出 DB2 数据库命令、SQL 语句和操作系统命令；调用先前的命令；以及卷动 SQL 查询的访问计划。
- 脚本中心。“脚本中心”允许您创建、运行和调度操作系统命令、DB2 命令脚本和 SQL 语句脚本。
- 警告中心。“警告中心”通知何时超过了您设置的阈值，或多节点环境中的一个节点何时不再响应。
- 日志。“日志”允许您查看作业的状态、重新调度作业，以及查看恢复历史日志和消息日志。
- 信息中心。“信息中心”令您可快速访问 DB2 产品手册中的信息和样本程序，并提供对 Web 上其他 DB2 信息源的访问。
- 许可证中心。“许可证中心”显示您的许可证的状态，并允许您配置系统以进行正确的许可证监视。

对于可使用 GUI 工具执行的某些功能，提供了使用“向导”的选项。可从“控制中心”中的弹出菜单调用向导。它们提供更高级别的帮助，逐步提示您如何填充执行的任务所需的信息，甚至根据您提供的信息进行计算并提出建议。若您新的数据库管理员或只是偶尔管理数据库，则向导是非常有用的。

在“DB2 通用数据库”中，存在下列“向导”：

- 备份数据库。它询问您关于数据库中的数据、数据库的可用性以及可恢复性要求等基本问题。然后它建议一个备份计划、创建作业脚本并调度它。要调用

“备份数据库向导”，选择代表您要备份的数据库的图标，单击鼠标右键，然后选择**备份** → **使用向导备份数据库**。

- 创建数据库。此“向导”允许您创建数据库、分配存储器以及选择基本性能选项。要调用“创建数据库向导”，选择“数据库”图标，单击鼠标右键，并选择**创建** → **使用向导创建数据库**。
- 创建表。此“向导”帮助您使用预定义的列模板来设计列，创建表的主键并将一个或多个表空间分配给表。要调用“向导”，选择“表”图标，单击鼠标右键，并选择**创建** → **使用向导创建表**。
- 创建表空间。此“向导”允许您创建一个新的表空间并设置基本存储器和性能选项。要调用它，选择“表空间”图标，单击鼠标右键，并选择**创建** → **使用向导创建表空间**。
- 创建索引。使用此“向导”确定对给定的一组 SQL 语句，要创建或删除哪些索引。根据您指定的工作负荷提出建议。要调用“创建索引向导”，选择“索引”文件夹，单击鼠标右键，并选择**创建** → **使用向导创建索引**。
- 性能配置。此“向导”帮助您请求有关数据库、它的数据以及系统目的等信息来调整数据库。然后它对数据库和实例建议新的配置参数，并在需要时自动应用它们。要调用此“向导”，选择数据库的图标，单击鼠标右键，并选择**使用向导配置**。
- 复原数据库。此“向导”带您经历恢复数据库的全过程。要调用该“向导”，选择数据库的图标，单击鼠标右键，并选择**复原** → **使用向导复原数据库**。
- 配置多站点更新。此“向导”允许您配置数据库以便使应用程序可同时更新多个站点。当所有站点的数据必须一致时，执行此操作就非常重要。要调用此“向导”，选择实例，单击鼠标右键，并选择**多站点更新** → **使用向导配置多站点更新**。

**注：**DB2 OS/390 版子系统不存在“向导”。

除了可从“控制中心”工具栏调用的图形工具外，还有一些不是从“控制中心”工具栏直接调用的附加的 GUI 工具。

- 性能监视器。“性能监视器”是一个监视 DB2 对象（如实例、数据库、表、表空间和连接）的工具。使用此工具可检测性能问题并调整数据库以达到最佳性能。“性能监视器”是作为“控制中心”中弹出菜单上的一个选项来调用的。
- 事件监视器。“事件监视器”是一个工具，它通过记录在特定事件发生时数据库的状态来分析资源的使用情况。从 DB2 命令行输入 db2emcrt 可创建“事件监视器”。
- 事件分析程序。“事件分析程序”是分析“事件监视器”收集的数据的工具。从 DB2 命令行输入 db2evmon 可调用“事件分析程序”。

- **Visual Explain。** Visual explain 功能允许您将 SQL 语句的访问计划作为一个图来查看，这样可调整您的 SQL 查询以获得更好的性能。在版本 6 之前，使用 Visual Explain 工具查看访问计划。现在，Visual Explain 不再是一个单独的工具；此功能可从“控制中心”的不同数据库对象的弹出菜单获得，也可从“命令中心”获得。

除这些工具之外，管理数据库还有一个有用的工具，它不是“控制中心”的一部分，而是“客户机配置辅助”。“客户机配置辅助”是一个工具，包含帮助用户设置客户机以便与远程服务器通信的向导。

稍后将详细说明所有这些工具。以下章节将概述这些工具的功能部件。

---

## 公共工具功能部件

下列功能部件可在几个工具中找到：

- 显示 SQL 和显示命令
- 显示相关项
- 生成 DDL
- 过滤器
- 帮助

### 显示 SQL 和显示命令

若一个工具生成了 SQL 语句，那么在该工具界面上将有**显示 SQL** 按钮可用。类似地，生成 DB2 命令的工具将有**显示命令**按钮可用。单击这两个按钮之一，可以：

- 查看该工具根据您在图形界面中所做选择而生成的 SQL 语句或 DB2 命令。此信息帮助您了解该界面是如何工作的。
- 将语句或命令另存为脚本以备将来使用。若您希望再次运行相同的语句或命令，此性能使您不必重新输入 SQL 语句或 DB2 命令。一旦 SQL 语句或 DB2 命令已保存到脚本中，您可调度此脚本，或编辑此脚本进行更改，或创建类似的脚本而不必重新输入语句或命令。

要显示 SQL 语句或 DB2 命令：

1. 从“控制中心”转至一个窗口或一个笔记本以使用生成 SQL 语句或 DB2 命令的工具。**显示 SQL** 按钮或**显示命令**按钮显示为可用。
2. 单击**显示 SQL** 按钮或**显示命令**按钮。一个适当的窗口打开。

若 SQL 语句或 DB2 命令比较复杂，则保存 SQL 语句和 DB2 命令特别有用。

当使用**显示命令**或**显示 SQL** 按钮时，可创建新脚本，以后可编辑它；也可以关闭对话框以返回到初始对话进行更改。若您单击“创建脚本”按钮，则出现“新建命令脚本”窗口。可在该窗口中编辑 SQL 语句或 DB2 命令，然后保存脚本。

## 显示相关项

**显示相关项**显示表、索引、视图、别名、触发器、表空间、“用户定义函数”以及“用户定义类型”之间的直接关系。例如，若您选择一个表并选择显示相关视图，将只能看到直接基于此表的那些视图。您看不到基于相关视图的任何视图，因为那些视图不是从该表直接创建的。

显示相关对象帮助您：

- 了解数据库的结构。
- 确定表已存在哪些索引。
- 确定表空间中存储了什么对象。
- 了解哪些其他对象与一个对象相关，因此受可能执行的任何操作的影响。例如，若您希望删除有从属视图的表，**显示相关项**告诉您哪些视图将不起作用。

要使用“显示相关项”功能部件：

- 从“控制中心”的“内容”窗格中选择一个对象，然后单击鼠标右键。
- 选择**显示相关项**。
- 单击选项卡，打开您需要的相关对象的那一页。取决于选择的选项卡，将列出不同的相关对象。只显示与选择的对象直接相关的那些对象。

可右键单击所选页上的相关对象，然后从弹出菜单中选择“显示相关项”。所选页更改为显示与最新选择相关的对象。也可单击所选对象旁的向下箭头，显示先前选择的对象列表以显示关系。

- 单击**关闭**以关闭“显示相关项”笔记本并返回到“控制中心”。

## 生成 DDL

**生成 DDL** 功能允许您在脚本文件中重新创建并保存下列对象的 DDL、SQL 语句和统计信息：

- 数据库对象
- 授权语句
- 表空间、节点组和缓冲池
- 数据库统计信息

这允许您：

- 保存 DDL 以便在另一个数据库中创建定义完全相同的表、数据库和索引，例如，用于数据库仓库应用程序
- 使用 DDL 将数据库从测试环境复制到生产环境，或从一个系统复制到另一个系统
- 编辑 DDL 以创建相似的对象

单击**生成 DDL**按钮，出现“显示命令”窗口，该窗口内显示由 **db2look** 实用程序生成的语句。从“显示命令”窗口，可单击**保存脚本**按钮来保存这些语句。将这些语句存放到一个脚本中。若您单击**生成**按钮，“运行脚本”窗口打开。

**注：**使用系统 390 的“控制中心”时，生成 DDL 语句的方法不同。有关那些差异的特殊性，参见帮助信息。

可选择是希望对所选模式还是数据库内的所有模式生成 DDL 语句。若希望在生产环境中使用此脚本之前进行更改，可以编辑此脚本。要使用生成的 DDL 语句创建完全相同的数据库，只需使用您生成的脚本并在新环境中运行它。

要生成 DDL 语句：

1. 突出显示您希望对其生成 DDL 语句的对象，然后单击鼠标右键。
2. 选择**生成 DDL**。出现“运行脚本”窗口。
3. 输入用户标识和密码，单击**确定**。着就创建了一个带有 **db2look** 命令内容的作业。出现一个 DB2 消息窗口，它显示新作业的作业标识。
4. 单击**确定**以关闭此消息窗口。
5. 使用“日志”笔记本的“作业历史”页，查看作业的结果，并查看与此作业相关的已保存脚本的内容。
6. 选择作业并单击鼠标右键。从弹出菜单选择**显示结果**。“作业结果”窗口打开。**db2look** 命令的输出显示在“作业输出”窗格中。
7. 选择**创建脚本**以创建结果的脚本。出现“新建命令脚本”窗口。
8. 若希望再次使用它，则保存此新脚本。

## 过滤器

在“控制中心”中，可过滤在“内容”窗格中显示的信息，或者可过滤从一个表检索的信息以作为实际结果集。通过为一个或多个对象创建过滤器，可限制显示的对象数或返回的对象数。一旦设置了过滤器，若希望再次在树中显示所有的对象，需要清除或删除此过滤器。

### 过滤显示

要减少“内容”窗格中出现的对象数以便管理：



1. 从位于“控制中心”底部的“内容”窗格工具栏选择“过滤器”图标，或从“视图”菜单条选择“过滤器”。
2. 选择要使用的标准来减少对象数。
3. 选择“启用过滤器”复选框来激活过滤器。

当稍后选择一个对象来查看其内容时，与该对象相关的过滤器将根据您先前设置的标准限制查看的内容。

### 过滤检索的数据

要减少在查询中返回的行数并缩短响应时间，可定义选择一个对象时显示在“内容”窗格中的输出或结果集。

1. 从对象树选择一个文件夹对象并单击鼠标右键。
2. 从弹出菜单选择**过滤器**。“过滤器”窗口打开。
3. 使用“过滤器”功能定义一组标准以检索属于该对象的行。

### 定义检索特定数据集的过滤器

要定义一个过滤器来检索特定的数据集：

1. 从“控制中心”，展开“数据库”或“子系统”文件夹，这取决于您所用的平台。
2. 选择您希望定义过滤器的对象。右键单击该对象。
3. 从弹出菜单选择**过滤器**。这将打开“过滤器”笔记本。
4. 在“查找”页上，指定所选对象的名称或其他描述性过滤器标准。过滤的结果是与“控制中心”的“内容”窗格中显示的所选对象相关的结果集。
5. 在“查找”页上，选择一个单选按钮指定是要满足在“查找”页上的字段中选择的所有条件，还是至少满足一个条件。
6. 在“过滤器”笔记本的“高级”页上，可使用附加的标准，编辑显示的文本以进一步限制返回的行数。
7. 单击**确定**，以使用您定义的过滤器标准。

要根据行数自动调用此过滤器笔记本，从菜单条选择“工具”，然后从弹出菜单选择“工具设置”。**行数超过时选择过滤**复选框允许您预定义从任何选择返回的行数的阈值。当达到此阈值时，“过滤器”笔记本出现，这样您可以根据定义的标准限制当前的检索。当一个表的增长超出预计而先前未过滤它时，此功能特别有用。取决于您的平台和数据，您可能正在尝试返回几百万行，而您只需要它的一个子集。

## 帮助

这些管理工具都提供了各种帮助信息。在所有窗口和笔记本以及菜单工具栏上都有帮助按钮。您可获取一般帮助，也可获取关于如何填充字段和执行任务的帮助。从帮助菜单，还可访问术语的索引或参考信息以及产品手册中提供的信息。

## 控制中心

使用“控制中心”作为管理的中心点来管理系统、DB2 实例、数据库、数据库对象（如表、视图和用户组）。也可使用“控制中心”访问 DB2 OS/390 版子系统。所有 DB2 数据库都必须编目，它们才会出现在“控制中心”中。图1显示“控制中心”的主要功能部件由于操作系统的差别，出现在您系统上的“控制中心”可能与图示有所不同。

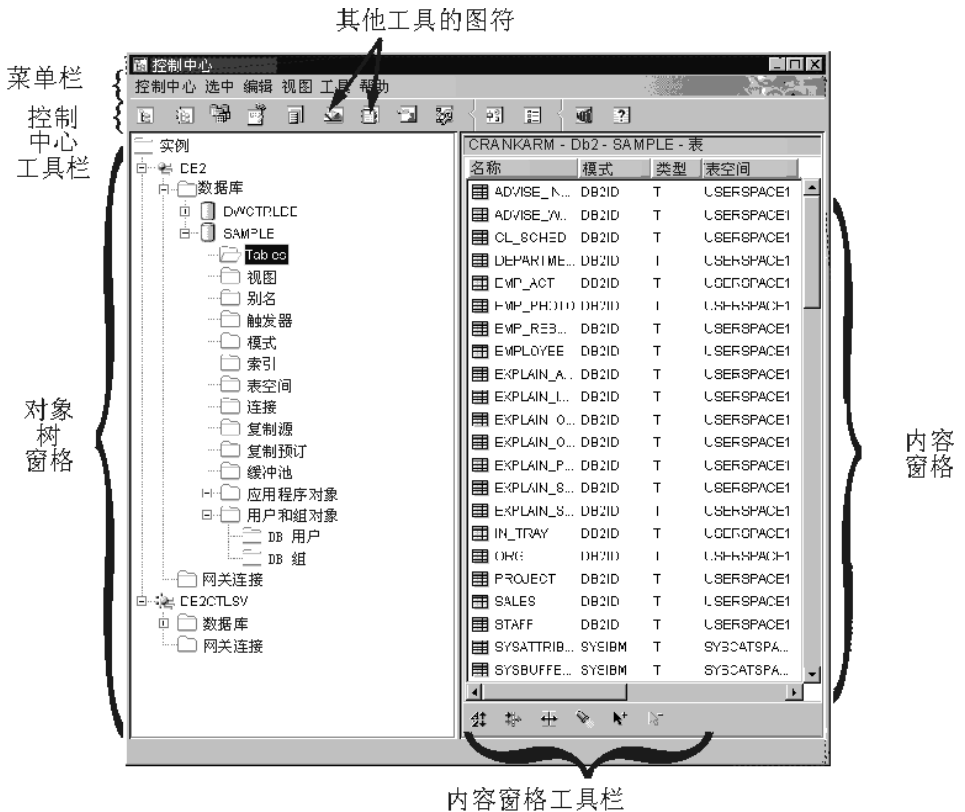


图1. “控制中心”功能部件

## “控制中心”的主要元素

“控制中心”的主要元素是:

- 菜单条。菜单条在屏幕的顶部。从菜单条选择菜单允许您执行许多功能，如关闭 DB2 工具、访问图形工具以及访问联机帮助和产品信息。可通过单击菜单条上的每一项来熟悉这些功能。
- “控制中心”工具栏。“控制中心”和其他工具的图标位于“控制中心”工具栏上。当把光标置于放到图标上时，悬浮帮助将标识每个图标。  
可从“控制中心”工具栏选择“工具设置”图标，来更改这些工具的设置。
- 对象树。对象树位于屏幕的左窗格中。它显示可从“控制中心”管理的所有数据库服务器和对象的图标。必须首先编目一个远程数据库服务器，它才会出现在“对象树”窗格中。“对象树”窗格中的某些对象包含其他对象。对象左面的加号 (+) 指示该对象是折叠的。可单击加号展开它。当展开对象后，在它的左面会出现一个减号 (-)。要折叠此对象，单击减号。
- 内容窗格。“内容”窗格位于屏幕的右窗格中。此窗格显示在“对象树”窗格中所选对象包含的所有对象，例如，若您在“对象树”窗格中选择表文件夹，所有的表都将出现在“内容”窗格中。若您选择数据库文件夹，“内容”窗格会变为显示所有数据库。可通过单击“内容”窗格工具栏中的“过滤器”图标并指定必需的信息来过滤出现在“内容”窗格中的列，或通过选择工具栏上的工具，然后选择工具设置来过滤对象。必须确保在“内容窗格”过滤器对话框中选择启用过滤器复选框。
- 内容窗格工具栏。此工具栏出现在“内容”窗格的底部。它允许您裁剪“内容”窗格中的信息。此工具栏是公共控件，它出现在产品中所有最详细的视图的底部或侧面。

在“控制中心”中工作时，可见到一些字段突出显示为红色粗边界。此边界指示这是必要的字段，要求您输入。一旦您选择了值或输入值，则红色边界将不出现。

## 在 DB2 OS/390 版中使用“定制的控制中心”

在 DB2 OS/390 版平台上使用“定制的控制中心”作为您自己定义的管理点，来管理子系统、数据库或数据库对象（如表、视图和数据库用户）。可使用此“定制的控制中心”来访问您定义的任何 DB2 OS/390 版对象。

“定制的控制中心”的主要元素与为缺省“控制中心”列出的那些相同。“定制的控制中心”允许您指定希望在个人化“控制中心”中包括的对象。可保存和调用此用户定义的树来管理 DB2 对象。它并不替换对所有用户而言的缺省“控制中

心”树，但是，若您希望在每次调用“控制中心”时以相同方式访问一组对象，则它很有用。您可根据需要创建许多定制的树，每一个树可包含不同的对象集，它们可以按照您选择的任何方式排序。

使用定制的树可减少在固定的 DB2 对象层次结构中的导航工作，并提供一种将相关对象分组的方法。例如，可定义一个树，它仅包含具有工资单信息的表。

## 可管理的系统

从“控制中心”，可管理用于 OS/2、Windows 和 UNIX 平台的“DB2 通用数据库”系列产品的数据库对象。参考《快速入门》一书以获取特定平台安装和设置信息。

也可将数据从 DB2 AS/400 版、DB2 VSE 和 VM 系统版和 DB2 OS/390 版复制到产品的“DB2 通用数据库”系列。参考 *Replication Guide and Reference* 手册以获取有关在产品之间复制的信息。

## 可管理的对象

若希望从“控制中心”管理对象，必须将它们添加至对象树。若要除去一个数据库或在“控制中心”外取消它的目录，并希望使用“控制中心”对它执行任务，必须将它添加至对象树。

可从“控制中心”管理的“DB2 通用数据库”对象有：

- 系统
- 实例
- 表
- 视图
- 索引
- 触发器
- 用户定义类型
- 用户定义函数
- 软件包
- 别名
- 复制对象
- 用户和组

可从“控制中心”管理的 DB2 OS/390 版的版本 5 对象有：

- 缓冲池

- 视图
- 目录表
- 存储器组
- 别名
- 同义词
- DB2 用户
- 位置
- 应用程序对象（集合、软件包、计划、过程）
- 数据库
- 表
- 表空间
- 索引
- 复制源
- 复制预订

对于 DB2 OS/390 版版本 6，可从“控制中心”管理的对象是版本 5 提到的所有对象加上：

- 模式
- 触发器
- 用户定义函数
- 单值类型

要查看可对这些对象执行什么操作，在“对象”窗格中选择对象并单击鼠标右键。出现一个列出功能的弹出窗口。

## 在“控制中心”中显示系统

要显示在您系统上编目的并安装了 DB2 的所有系统：

1. 单击系统旁的加号 (+)，展开“对象树”。显示代表本地机器和任何远程机器的图标。您的本地系统由“本地”图标表示。仅当本地机器是 DB2 服务器时此图标才出现。若右键单击“本地”图标，则弹出菜单中出现一个选项，称为**与管理服务器连接**。“管理服务器”允许您利用诸如性能监视和调度等功能。DB2 管理工具使用它来满足 DB2 服务请求，而且它是自动创建和启动的。“DB2 管理服务器”的缺省名随平台的不同而有所变化。例如，在 Windows 和 OS/2 平台上，使用“DB2DAS00”，在 AIX 上，使用“db2as”。
2. 展开“本地”图标。在本地机器上的 DB2 的实例以树结构显示。

在 OS/2、Windows 和受支持的 DB2 UNIX 系统上，可将数据库管理器代码的每个副本当作一个单独的实例，该实例存储在您机器上的一个目录中。在 DB2 OS/390 版上，实例被称为子系统。当您安装 DB2 时会创建一个缺省的本地实例。在单个系统上可有几个实例。可使用这些实例将开发环境与生产环境分开，或限制敏感信息只能由特定的一组人访问。还可以对特定的环境调整实例。

3. 展开实例图标。对每个存在的数据库显示一个图标和名称。

## 管理 DB2 OS/390 版对象

使用“控制中心”，您可执行现存的 DB2 OS/390 版的版本 5 和 DB2 UDB OS/390 版的版本 6 产品的许多功能，如创建、改变和删除对象，还有运行重组数据或装入数据的实用程序。但是，在从“控制中心”管理一个 DB2 OS/390 版子系统之前，必须首先配置与它的连接以将它添加至对象树。

## 添加 DB2 OS/390 版子系统

若您已安装“客户机配置辅助程序”，可使用它轻松配置与 DB2 OS/390 版系统的连接。若尚未安装“客户机配置辅助程序”，则必须使用“命令行处理器”（CLP）手工配置与 DB2 OS/390 版系统的连接。

可使用“客户机配置辅助程序”搜索网络，找到 LAN 上您的客户机可用的所有 DB2 OS/390 版系统。若希望添加一个 DB2 OS/390 版子系统，可使用添加数据库向导来添加子系统，使用简要表导入连接，或者手工添加连接。

若选择搜索此网络，网络上需要有一个 DB2 Connect 产品，它具有为该系统定义的连接。若选择使用访问简要表，则需要从简要表中选择表示该系统的 DB2 Connect 服务器连接。若选择手工配置该连接，您需要知道系统名、通信协议和通信协议参数（如主机名、TCP/IP 的端口号码或 SNA 的“符号目的地名”）。在添加 DB2 OS/390 版系统之后，DB2 Connect 服务器连接的对象将出现在“控制中心”的本地系统中。

当添加 DB2 OS/390 版的版本 5 或更新版本的系统时，它出现在它自己的“控制中心”对象树部分中。要查看驻留在特定系统中的 DB2 OS/390 版和其他数据库对象，从表示 DB2 OS/390 版系统的 DB2 OS/390 版系统图标展开对象树。

要查看可对特定对象执行的操作的列表，当此对象出现在对象树中时，选择它并单击鼠标右键。出现一个弹出菜单，显示可对该对象执行的可用操作。例如，可创建、改变或删除视图，也可以查看其内容，修改它的特权，显示与它相关的其他对象的列表。有关可执行哪些功能的更多信息，参见 DB2 OS/390 版对象的联机帮助。

## 管理网关连接

当编目 DB2 Connect 服务器时，一个“网关连接”文件夹显示在本地系统的实例对象之下的“控制中心”对象树中。“网关连接”文件夹包含对象的一个层次结构，用于管理与主机和本地编目的 AS/400 数据库的连接。可使用与这些连接管理对象相关的操作，来列出、强制并监视主机和 AS/400 数据库连接。

“网关连接”文件夹下的对象树可用于管理与主机和 AS/400 数据库的连接，但是不能用于执行数据库管理任务。不过，若您需要在本机系统上添加、更改或删除主机或 AS/400 数据库，则使用“客户机配置辅助程序”。

## 可从“控制中心”执行的功能

从“控制中心”，您可：

- 管理数据库对象。可以创建、改变和删除数据库、表空间、表、视图、索引、触发器和模式。也可以管理用户。
- 管理数据。可以装入、导入、导出、重组数据，并搜集统计信息。
- 调度作业。作业可以是暂挂的、正在运行的或已完成执行的脚本。可调度作业在特定的时间开始。
- 通过备份和复原数据库或表空间，执行预防性维护。
- 监视性能并执行故障诊断。
- 复制数据。
- 配置并调整实例和数据库。
- 管理数据库连接，如 DB2 Connect 服务器和子系统。管理应用程序。
- 使用“解释型 SQL”分析查询，查看访问计划。
- 更改“控制中心”中用于显示“菜单”和“文本”的字体。可更改为可用字体之一，还可更改字体大小和显示的颜色。要使更改生效，必须重新启动“控制中心”。
- 启动其他工具。例如，可启动“卫星管理中心”或“命令中心”。

要查看可对一个对象执行的所有操作，只须从“对象树”窗格或“内容”窗格中选择此对象并单击鼠标右键。出现一个弹出菜单，显示可对该类对象执行的所有功能；例如，若选择表文件夹，则可创建一个新表（利用或不用“向导”的帮助），监视表的性能，过滤出现在“内容”窗格中的表等等。根据选择的对象的不同，可执行的功能会有所不同。

右键单击“内容”窗格中的对象，以便对特定对象执行附加功能。例如，若在“内容”窗格中选择一个表并单击鼠标右键，则一个弹出窗口显示可对该表使用的功能。

## 创建新对象

要创建新对象:

1. 展开数据库文件夹。对象类型显示为文件夹图标。
2. 右键单击一个对象的文件夹图标，例如，单击表图标。显示弹出菜单。对于某些对象，可有两个选项来执行一个功能。一个选项是使用“向导”。并不是您可执行的所有功能都有“向导”。
3. 选择**创建**。因为有用来创建表的“向导”，所以您得到两个选项，其中之一是使用向导来创建表。若选择了“向导”选项，将提示您输入信息，并对应该采用的选项给出建议。对于新用户或不经常创建数据库对象的人来说，“向导”特别有用。

## 使用现存的对象

当在“对象树”窗格中单击一个对象如表文件夹时，已经存在的所有表都出现在“内容”窗格中。然后可选择您想使用的表并单击鼠标右键，调用希望对该特定表执行的任何功能。

有关使用“控制中心”的更多信息，参考它的联机帮助，该联机帮助可从**帮助**菜单获得，或在“控制中心”中的任何地方按 F1 键获得。

## 查找对象（仅适用于 DB2 OS/390 版）

使用“查找”笔记本可以较容易地搜索数据库或子系统对象。这允许您:

- 不必浏览“控制中心”的树结构来查找对象。此对象可能在数据库、子系统或表空间中，或跨越数据库、表和支持的对象。
- 在一个子系统内的多个数据库中查找对象（表空间、表和索引）。

使用“查找”笔记本的“查找”页来指定搜索条件。使用“查找”笔记本的“高级”页来进一步定制搜索。在“高级”页上编辑提供的文本，并添加或修改搜索条件。

要查找在数据库或 DB2 OS/390 版子系统内定义的对象:

1. 在“控制中心”中，右键单击一个对象。从弹出菜单选择**查找**。“查找”笔记本打开。
2. 从“对象类型”字段，选择要搜索的数据库对象的类型。取决于您开始搜索的对象，可用的目标对象的列表有所不同。
3. 在“查找”页上，填写搜索条件。必须输入至少一个搜索条件，而且在搜索中可使用通配符。除非您使用有效的定界符来特别标记小写字符或扩充字符集，否则要将字符转换为大写字体。



4. 在“查找”页上，选择一个单选按钮，指定是要满足在“查找”页上的字段中选择的所有条件还是至少满足一个条件。
5. 单击**确定**，以使用此搜索条件。搜索的结果显示在“查找结果”窗口中。输出表的格式取决于您搜索的对象类型。
6. 要使用相同或不同的条件重复搜索，单击**应用**。
7. 可选择在“查找结果”窗口中出现的一行，右键单击该行，可看到一个弹出菜单，显示您可执行的附加操作。

---

## 卫星管理中心

“卫星管理中心”是可从“DB2 控制中心”使用的一组工具。它们允许从一个中心设置和管理 DB2 服务器集合。属于一个组的每个 DB2 服务器称为一个卫星。从中心管理卫星意味着可隐藏 DB2 而不让使用 DB2 卫星的任何人看到，这样可防止他们了解数据库管理。

用组来组织具有共享特征（如在服务器上运行的应用程序，或支持应用程序的数据库配置）的 DB2 服务器。这些 DB2 服务器在数据库配置、使用和用途方面是相似的。

通过将 DB2 服务器分组，可管理若干组 DB2 服务器，而不必个别管理每个 DB2 服务器。若获得附加的 DB2 服务器，它们提供的功能与一组现存的 DB2 服务器提供的功能相同，可使用“卫星管理中心”将它们添加至该组。

从“卫星管理中心”，可创建组、卫星、应用程序版本、批处理和认证凭证。也可定义成功代码集并执行与卫星环境的管理相关的其他功能。有关卫星环境的信息存储在称为卫星控制数据库的中央数据库中。此数据库还记录其他信息，如哪些卫星在该环境中、每一卫星属于哪一个组以及卫星正在运行的最终用户应用程序的版本。此数据库位于称为 DB2 控制服务器的 DB2 服务器上。

在可以启用“卫星管理中心”的功能之前，必须首先在“控制中心”上编目卫星控制数据库 (SATCTLDDB)。当启用之后，可使用“卫星控制中心”来设置和维护卫星、组以及当卫星与它们的应用程序版本同步时要执行的批处理。

要设置和维护数据库配置，每个卫星要与卫星控制数据库连接，以下载与最终用户应用程序的版本对应的批处理。卫星在本地执行这些批处理，然后将结果报告回卫星控制数据库。下载批处理、执行它们，然后报告批处理执行的结果的过程称为同步。卫星同步即是维护与其他卫星的一致性，这些卫星属于同一个组，它们运行最终用户应用程序的同一个版本。

---

## 命令中心

可单击“控制中心”工具栏上的“命令中心”图标来启动“命令中心”。

“命令中心”允许您:

- 在结果窗口中查看一个或多个 SQL 语句和 DB2 命令的结果输出。可滚动结果并生成报告。
- 创建命令脚本，并将它们保存到“脚本中心”。可编辑命令脚本以创建新的脚本。然后可从“脚本中心”调度此命令脚本，在您指定的任何时间作为一个作业运行。
- 运行 SQL 语句、DB2 命令和操作系统命令。当从“命令中心”运行 DB2 命令时，不必在该命令前加上 **DB2**。可以用任何受支持的操作系统脚本语言（如 REXX），并在命令前加一个惊叹号 (!) 来运行操作系统命令。使用“命令中心”运行命令和语句，可以一次发出许多命令，而不必个别输入和运行每个命令。
- 从主工具栏快速访问 DB2 管理工具，如“控制中心”。
- 在执行 SQL 语句之前查看与该语句相关的访问计划和统计信息。

---

## 脚本中心

可从“控制中心”工具栏选择“脚本中心”的图标来启动“脚本中心”。“脚本中心”是一个工具，它允许您通过编写一组命令和语句来创建脚本，您可调度它在需要时运行。可导入先前创建的脚本或在“命令中心”中保存的脚本。可从保存的脚本集合中选择脚本，而且可编辑现存的脚本来创建新脚本、复制脚本或删除脚本。

可在“脚本中心”内编辑脚本，或在“脚本中心”外使用自己的编辑器编辑。若从“脚本中心”内运行脚本，好处是可将结果记录在“日志”中。

要从“脚本中心”内的一个脚本中运行操作系统命令:

1. 选择**脚本** → **新建**。“新建命令脚本”窗口打开。
2. 对于**脚本类型**，选择 **OS 命令** 单选按钮。
3. 输入脚本名、描述和工作目录。
4. 输入命令。
5. 单击**确定**。

从“脚本中心”，可查看信息，如系统已知的所有命令脚本的描述和脚本类型，而且可执行下列任务:

- 创建包含 DB2 和操作系统命令的命令脚本
- 立即运行保存的命令脚本

- 调度脚本在以后或定期运行；例如，您可能希望创建一个脚本来收集几个表的统计信息。然后可以调度作业夜间运行。通过指定您希望作业运行的小时数、天数、周数、月数、一周多次或一月多次，可调度作业在预定的时间间隔无人照管地运行。无论何时调度脚本或立刻运行脚本，就会创建一个作业。
- 从工具栏访问“日志”，查看使用特定脚本的作业，并查看所有调度作业的状态
- 编辑已保存的命令脚本

## 用“脚本中心”处理现存的脚本

要使用“脚本中心”处理不是从“脚本中心”创建的预先存在的脚本：

1. 从**控制中心**工具栏，单击**脚本中心**图标。“脚本中心”打开。
2. 选择**脚本** → **导入**。“文件浏览器”窗口打开。
3. 选择一个现存的脚本文件并单击**确定**。“新建命令脚本”窗口打开。此脚本显示在窗口的下半部分，该窗口是一个脚本编辑器。完成**实例**、**脚本名**、**脚本描述**和**工作目录**字段，然后选择**脚本类型**。
4. 单击**确定**。将在“脚本中心”创建此脚本。

## 调度保存的命令脚本运行

要调度脚本：

1. 单击“控制中心”工具栏上的**脚本中心**图标。“脚本中心”打开。
2. 右键单击您想调度运行的脚本，然后从弹出菜单选择**调度**。“调度程序”窗口打开。
3. 选择作业的频率和一个完成操作，如完成消息或要启动的另一个命令脚本。
4. 单击**确定**。这会启动一个暂挂作业，您可在“日志”中跟踪它。

---

## 日志

可从“控制中心”工具栏选择“日志”的图标来启动“日志”。“日志”允许您监视作业并复查结果。也可从“日志”显示恢复历史和 DB2 消息。“日志”允许您：

- 监视暂挂作业、正在运行的作业和作业历史记录
- 复查结果
- 显示恢复历史记录和警告消息
- 显示 DB2 消息的日志

## 使用作业

可使用“日志”来处理作业。要打开“日志”：

1. 从“脚本中心”工具栏单击**日志**图标。“日志”打开。
2. 要查看预定在稍晚时间运行的作业，单击**暂挂作业**按钮。在暂挂作业列表中可看到您的作业。也可看到有关这些作业的所有信息。可对暂挂作业执行操作，如重新调度它、显示与它相关的脚本或立即运行它。当修改保存的脚本时，与它相关的所有作业都会继承新的修改过的行为。

也可从“日志”查看当前正在运行的作业和作业历史。

“日志”窗口中的其他页为：

- “恢复”页。此页显示恢复历史（备份操作、复原操作和装入操作的详细信息）并允许您复原恢复日志。
- “警告”页。此页显示所有警告。
- “消息”页。此页显示通过 DB2 管理工具发出的所有消息。

“日志”的联机帮助提供使用作业和日志的详细步骤。

---

## 许可证中心

可使用“许可证中心”显示在您系统上安装的 DB2 产品的许可证状态和使用信息。还可使用“许可证中心”来配置系统以进行正确的许可证监视。“许可证中心”允许您：

- 添加新的许可证。
- 从产品的试用许可证升级到永久许可证。
- 查看许可证的详细信息。

若您查看许可证信息的详细信息，可看到如下内容：

- 产品名称
- 版本信息
- 到期日期
- 注册的用户
- 并行的用户
- 授权的用户数
- 并行的用户数
- 实施策略

- 处理器数（对于“DB2 通用数据库企业版”和“DB2 通用数据库扩充企业版”）。

---

## 警告中心

“警告中心”是监视系统以向您警告有潜在问题的工具。可将“警告中心”设置为自动打开，显示已超过其阈值并因此处于警报或警告状态的任何受监视对象。使用可从“控制中心”调用的“性能监视器”设置这些阈值。图标的颜色指示警告的严重性。红色图标指示警报信号。黄色图标指示警告。

---

## 客户机配置辅助程序

“客户机配置辅助程序” (CCA) 主要是一个包含向导的工具，这些向导用来帮助对本地或远程 DB2 服务器设置客户机。然而，此工具还可用来容易地帮助配置 DB2 Connect 服务器。

“客户机配置辅助程序”允许您维护应用程序可以连接的数据库的列表。它编目节点和数据库，使您不必执行这些本来相当复杂的任务。

在“客户机配置辅助程序”中，您可以执行下列任务：

- 添加、修改与删除数据库连接项。
- 测试与所选数据库的连接。
- 配置数据库管理器配置参数。
- 配置 CLI/ODBC 设置。
- 将 DB2 实用程序和其他应用程序绑定至选择的数据库。
- 导入和导出配置信息。这使您能够使用先前配置的机器上的现存配置来配置新机器。
- 更改用来连接所选数据库的用户标识的密码。

“客户机配置辅助程序”提供了下列方法来帮助您添加新的数据库连接项：

- 使用简要表。可以从先前配置的机器导出简要表，并将使用它来配置新机器。可以从“控制中心”导出服务器简要表，并可以从 CCA 导出客户机或服务器简要表。
- 搜索网络。CCA 可以在网络中搜索正在运行管理服务器的 DB2 系统。提供了“搜索发现”和“已知（或直接）发现”方式。“搜索发现”方式受网络配置限制。（通常，网络路由器不允许传送“搜索发现”请求。）“已知发现”只需要很少的信息便可找到期望的服务器系统。还可以找到先前在网关上定义的主机或 AS/400 系统。

- 手工配置与数据库的连接。必须提供所有信息，但启动了一个向导，有助于简化任务。

---

## 性能监视器

“性能监视器”提供有关“DB2 通用数据库”的状态和它控制的数据的信息。它是一个可为您的数据库环境定制的图形实用程序。可定义阈值或范围，以便当“性能监视器”收集的值不在可接受的范围之内时触发警告或警报信号。

通过在“对象树”窗格或在“内容”窗格中选择对象并单击鼠标右键，可监视 DB2 对象，如实例、数据库、表、表空间和连接。从那里您可选择启动监视活动。

当监视对象时，图标的颜色显示为绿色、黄色或红色以指示监视的状态。这些颜色指示问题的严重性，这由您设置的阈值所定义。绿色表示监视器在运行而且一切正常。黄色是警告，表示被监视的对象正在接近阈值。红色指示警报，被监视的对象已达到阈值。可使用 DB2 所带的预定义监视器，也可创建自己的监视器。

要查看“性能监视器”收集什么信息，右键单击对象并在弹出窗口中选择**显示监视活动**。

使用“性能监视器”的信息可：

- 检测性能问题
- 调整数据库以获得最优性能
- 分析性能趋势
- 分析数据库应用程序的性能
- 防止发生问题

“性能监视器”允许您将数据库信息转换为一种直观的表达以便分析趋势，这些信息包括磁盘活动、缓冲池使用情况、预取量、锁定使用情况以及在特定时间间隔的记录分块。

当需要监视现有问题或希望观察系统的性能时，可使用此工具。它可使您获得在某个时间点的数据库活动和性能数据的快照。可使用这些快照来比较不同时间的性能。图形上的每一点表示一个数据值。在第25页的『监视某个时间点的性能』中提供了获得快照的步骤。此信息可帮助您标识和分析潜在问题，或根据您设置的阈值标识异常情况。若需要知道数据库管理器及其数据库应用程序在单个时间点的性能并查看性能随时间变化的趋势，可使用此性能工具。使用它还可获得哪些元素处于警报状态的直观表示。这有助于标识哪些参数可能需要调整。然后可仔细查看已对该元素设置的参数，并更改它以改进性能。

## 事件监视器

与获得时间点的快照相反，事件监视器收集一段时间的数据库活动信息。收集的信息为特定的数据库事件活动（例如，数据库连接或 SQL 语句）提供了一个有用的摘要。事件监视记录在特定事件发生时数据库的状态。可获得有关该数据库活动的跟踪记录。在捕捉到数据之后，首先存储事件监视器记录，然后进行分析。若需要知道一个事务花了多少时间，例如一条 SQL 语句占用了 CPU 多少时间，使用事件监视器。然后可使用“事件分析程序”读取从事件监视器记录的数据。

对于每个数据库连接，都会生成一条连接事件记录。对于在该连接中运行的每条语句，都会生成一条语句记录。每一连接事件记录映射为“事件分析程序”的“连接视图”窗口中的一行。此窗口显示在监视周期期间连接的每个应用程序的信息，包括：

- 应用程序名
- 执行标识
- 连接时间
- 总计 CPU 时间
- 锁定等待时间
- 总计排序时间
- 死锁
- 断开连接时间
- 应用程序标识

每一语句事件记录映射为“事件分析程序”的“语句视图”窗口中的一行。

## 使用监视器工具

“性能监视器”和“事件分析程序”具有下列优点：

- 广泛、灵活的数据收集。支持 200 个以上的性能变量，包括缓冲池和 I/O、锁定和死锁、排序、通信、代理程序以及记录信息。显示数据库管理器、数据库、表空间、表、缓冲池、连接、事务和 SQL 语句的数据。
- 易于使用、直观的查看。可使用表达清晰的图形或组织为逻辑组的文本视图来实时查看数据。提供了详细信息和摘要视图，这样可访问更详细的信息。
- 强大的警告功能。要获得任何性能测定值，可指定阈值来定义异常情况。通过将特定区域内的性能测定值描绘在性能图上，可使用阈值直观标识何时性能测定值达到或超过阈值。当达到阈值时，可指定希望执行的下列操作之一或所有操作。
  - 通过“警告中心”通知您。
  - 接收一个声音警报。

- 运行一个程序。
- 显示一条消息。

或者，您可以决定不应给出通知。

图2举例说明这些监视器是如何一起工作的

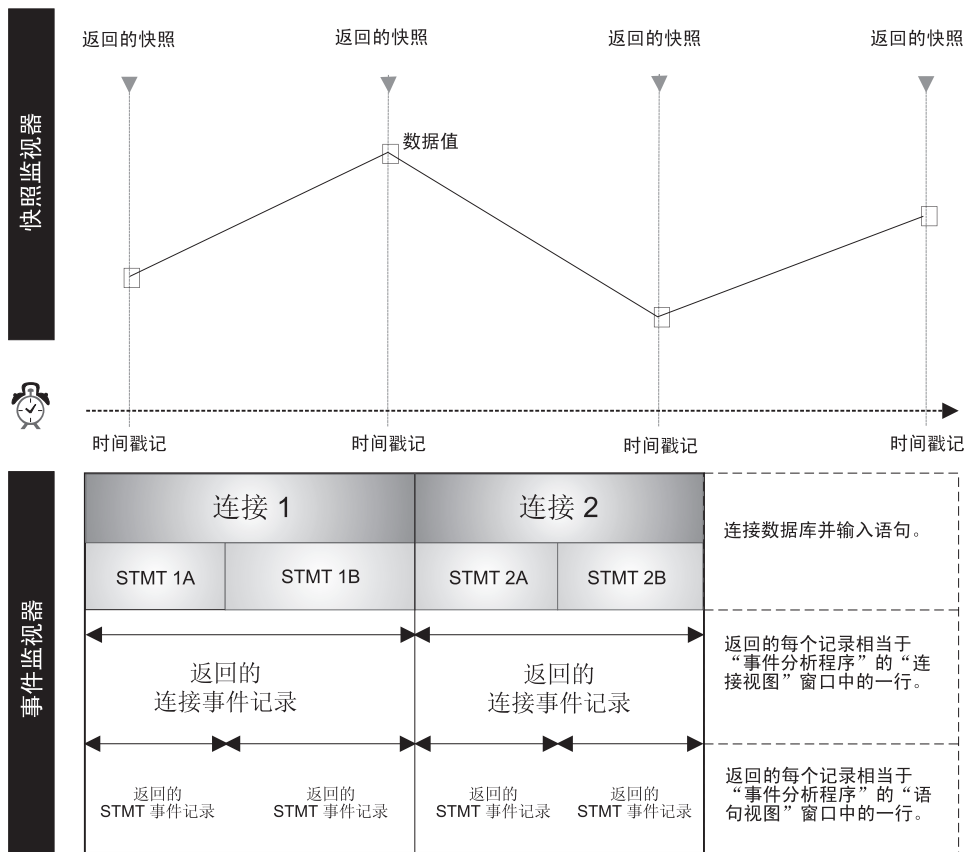


图 2. 比较：获得快照和监视事件。（事件监视器、事件分析程序）

### 监视和调整数据库的注意事项

在开始监视和调整数据库之前，应当执行以下操作：

- 定义您的目标。例如，可能希望了解应用程序如何在特定时间点实例级使用资源，以便执行一些操作，例如，可以检查当启动特定应用程序时数据库并行



性是否降低。或者希望了解当应用程序运行时发生哪些实例级事件，例如，当特定应用程序运行时是否整体性能较差。

- 确定您将分析什么信息。例如，要查看瓶颈是否与硬件相关，您可能希望获得快照来监视数据库连接活动或表空间、缓冲池和 I/O 活动。要查看瓶颈是否与环境相关，可使用“事件分析程序”来监视是否有以下现象：

- 在高峰期间调用了太多的数据库任务
- 用户连接数目过高
- 数据库分区（硬件负荷平衡）没有最优化
- 服务器不只是作为数据库服务器使用

例如，一些明显的效果：

- 查询 / 响应很慢
  - 调度的任务未按时完成
  - 应用程序超时
- 决定是否使用 DB2 提供的预定义监视器，或是否创建您自己的监视器。

下一节描述如何获得快照以及如何使用“警告中心”来跟踪任何与性能相关的问题。

## 监视某个时间点的性能

若您希望进行复杂的数据收集并分析数据以查找潜在的问题，应使用“性能监视器”获得系统的快照并观察性能数据随时间的变化。

该工具允许您：

- 以图形显示性能信息
- 设置性能快照的捕捉频率
- 查看性能计算的结果
- 定义阈值和阈值操作
- 生成并存储警告
- 查看摘要信息（例如，所有数据库）

捕捉下列类型的信息：

- 有关长期活动的信息（如一个应用程序所花时间过长以致无法完成时的数据库活动）。
- 跟踪有关当前级别的活动信息的计数器（如对一个数据库打开游标的次数）。
- 有关数据库活动的累积信息（如数据库实例活动时进行连接的最大数目，或针对特定数据库执行的 SQL 语句的总数）。

在预定义的时间间隔获得快照，可提供在数据库管理器及其应用程序中活动的当前状态。此信息可用于：

- 检测性能问题
- 分析性能趋势
- 调整数据库管理器和数据库配置参数
- 分析数据库应用程序的性能

可获得下列数据库对象的性能信息：

- 实例
- 数据库
- 表
- 表空间
- 数据库连接

对于每一项，可监视其各种性能变量。可从任何“快照监视器”窗口的帮助菜单获得“性能变量引用帮助”，该帮助提供所有性能变量的描述。这些变量分为不同的类别。存在下列类别：

- 实例：代理程序、连接、排序
- 数据库：锁定和死锁、缓冲池和 I/O、连接、排序、SQL 语句活动
- 表：表
- 表空间：缓冲池和 I/O
- 数据库连接：缓冲池和 I/O、锁定和死锁、排序、SQL 语句活动

有关如何生成快照的详细信息，参见联机帮助。

## 预定义监视器

“DB2 性能监视器”包含一组预定义监视器，可按原样使用它们，或者可复制并修改它们以满足您的要求。它们提供了一个完备的性能计算集合。不能更改 IBM 提供的性能监视器的名称、等式或文本描述；但可更改阈值和警告操作。使用预定义监视器来了解性能监视，并创建自己的监视器：复制一个预定义监视器，然后从该副本中添加或删除性能变量。

与 DB2 一起提供的“预定义监视器”有：

- 监视容量。使用此监视器来获取关于系统容量的信息。可定期检查此监视器，来查看经过一段时间后系统的整体使用情况。
- 排序。使用此监视器来确保排序堆和排序堆阈值参数的设置正确。当第一次启动系统时、在活动的高峰期或应用程序改变时，应运行此监视器。

- 锁定。使用此监视器来确定系统中的锁定情况，以及是否正确设置了锁定列表参数。
- 高速缓存。使用此监视器来优化高速缓存的使用。在高峰期监视这些值，可确定是否需要增加该高速缓存的大小。
- 缓冲池。对小表使用此监视器来确定它们是否需要自己的缓冲池。
- 死锁。使用此监视器来确定应用程序是否进入死锁状态。
- 快速通信管理器。使用此监视器来查看在节点间传送信息所用的内存的百分比。
- 预取程序。使用此监视器来确定是否为系统定义了足够的预取程序。
- 磁盘性能。使用此监视器来监视输入和输出。此监视器包含在数据库级和表空间级控制磁盘性能的性能变量。
- 全局内存。使用此监视器来监视应用程序的内存使用情况。
- 长时间运行的内存。使用此监视器来帮助确定为何一个查询花很长时间才完成。
- 网关连接。使用此监视器来监视 DB2 Connect 服务器连接。

有关如何使用预定义监视器的示例，参阅为性能监视提供的联机帮助。

要查看可用监视器的列表，从“控制中心”右键单击**系统**文件夹，然后从弹出菜单中选择**列出监视器**。“列出监视器”窗口打开。它列出存储在您当前连接的 JDBC 服务器上的监视器。对于每个监视器，可看到该监视器的名称、描述、状态、它是否是缺省监视器，以及谁创建了该监视器。“监视器的状态”指示本地系统上的监视器的状态，而不是 JDBC 服务器上的那些监视器的状态。“缺省”级别指示缺省监视器是位于实例级、数据库级、表级、表空间级还是连接级。对于预定义监视器，“创建者”列包含 **NULLID**。该窗口右边包含一些按钮，用来对监视器执行各种任务。有关 JDBC 服务器的更多信息，参见第41页的『将“控制中心”作为 Java 小应用程序运行』。

可选择将哪个监视器作为一个对象的缺省监视器启动。

一旦启动了一个性能监视器，可单击工具栏上的“警告中心”按钮来查看正监视的任何对象的状态，以及哪些对象由于已达到它们的任何一个阈值而处于警告状态。这些信息仅在超过阈值期间出现。

若要密切监视所监视的对象，可将“警告中心”一直打开，或者可将摘要页上的**显示监视信息**窗口一直打开，来查找任何红色或黄色的项目。也可修改“控制中心”设置，以便在添加新的警告或警报时“警告中心”自动打开。从“警告中心”，也可临时暂停那些警告，继续进行监视。

## 当一个对象出现在“警告中心”时需要执行的操作

可将“警告中心”设置为自动打开，显示任何处于警报或警告状态（即已超过其阈值）的受监视对象。可从“工具设置”窗口更改此缺省值。

若在“警告中心”看到一个对象，右键单击该对象，选择**性能监视器** → **显示监视信息**来查看该数据库对象的性能详细信息。

## 在一段时间内分析事件

“事件分析程序”是另一个 DB2 性能工具。当需要已发生的一个事件的诊断信息时，使用此工具。“事件分析程序”可以与事件监视器一起使用。例如，在一个数据库活动时，可使用事件监视器来跟踪数据库活动，如连接、事务、语句和死锁。事件监视器也可记录当一个应用程序与数据库断开时所记录的累积性能数据。当事件监视器创建了事件监视文件时，可使用“事件分析程序”查看性能信息。

事件监视器工具可用来执行下列操作：

- 创建事件监视器以监视您感兴趣的数据库事件类型。
- 激活事件监视器以开始收集事件数据。该数据存储在文件中。
- 让事件监视器停止收集事件数据。
- 查看由事件监视器生成的跟踪类型摘要信息。
- 当不再需要事件监视器时，除去它。有一个选项，可用来清除跟踪文件。
- 显示与该数据库相关的事件监视器的列表。
- 查看事件监视器的定义。

“事件分析程序”允许查看事件监视器为下列事件类型生成的数据：

- 数据库连接活动（在连接开始和断开之间的那段时间）
- 事务（工作单元）
- SQL 语句执行情况
- 死锁活动的检测

## 事件分析程序

可为下列事件类型创建一个事件监视器，然后使用“事件分析程序”来查看收集的信息：使用 db2evmon 可执行文件（在 *Command Reference* 和 *System Monitor Guide and Reference* 中有描述）来查看为下列类型生成的数据：

- 死锁
- 数据库活动
- 表空间活动

- 表活动
- 语句活动

要使用事件监视器和事件分析程序分析事件数据，执行下面的步骤。它们只代表如何为连接和语句事件创建事件监视器的一个示例。要创建事件监视器：

1. 从“命令中心”的命令行，输入 **db2emcrt**。“事件监视器”窗口打开。
2. 单击**事件监视器**，然后从菜单中选择“创建”。“创建事件监视器”窗口打开。
3. 在该字段中，指定正创建的事件监视器的名称。这个新的事件监视器不能与任何现存的监视器有相同的名称。名称中不允许空格。
4. 仅在“DB2 通用数据库扩充企业版”产品中，从“在节点上”下拉列表中选择事件监视器文件将驻留的一个节点。
5. 仅在“DB2 通用数据库扩充企业版”产品中，为该事件监视器选择一个作用域。缺省情况下，作用域是“全局”。
6. 选择一个或多个复选框，以指示要监视的事件类型。注意“死锁”事件类型是缺省选项。
7. 指示希望何时启动此监视器。注意“立即启动”是缺省选择。
8. 为连接、语句或事务定义一个或多个条件，这些条件将在这些级别控制监视。
9. 标识监视器写入事件数据文件所用的路径（目录名）。
10. 单击**选项**，为**指定事件监视器文件**选项打开一个窗口。这些选项确定如何处理监视器输出，以及这些输出如何影响事件监视器的性能。
11. 单击“确定”以创建该监视器，或单击“取消”以退出而不创建监视器。
12. 右键单击一个事件监视器，然后从弹出菜单中选择**停止事件监视**，来关闭事件监视。

这强制事件监视器编写跟踪文件。若未关闭该监视器，当缓冲区满时或所有连接结束时只将信息写入磁盘。从“事件监视器”窗口，右键单击创建的事件监视器，从弹出菜单中选择**查看事件监视器文件**，来查看生成的事件数据。“监视周期视图”窗口打开。

要从“事件分析程序”访问事件数据：

1. 从“命令中心”的命令行输入 **db2eva** 来启动“事件分析程序”。“事件分析程序”窗口打开。
2. 在“路径”字段中，标识存储数据文件的路径（目录名）。若不曾移动这些文件，它将是创建事件监视器时指定的路径。若移动过这些文件，则指定该目录。可单击 ... 来列出现存的目录。

注：若数据文件存储在距离较远的地方，必须将这些文件 FTP 到本地机器以便查看它们。取决于文件大小，此传送可能要花一定的时间。可将文件传送到任何本地路径。不需要选择创建文件时使用的相同路径。

3. 单击**确定**以访问目录中包含的数据文件，或单击**取消退出**。“监视周期视图”窗口打开。
4. 右键单击一个监视周期，然后从弹出菜单中选择**打开为 → 连接**。“连接视图”窗口打开。它显示在事件监视会话期间建立的连接的列表。（可能会列出多个连接。您感兴趣的连接可能不是列表中的第一个。）
5. 右键单击一个连接，然后从弹出菜单中选择**打开为 → 语句**。“SQL 语句视图”窗口打开。它显示用于所选连接的所有语句。为每条语句提供了信息列，包括：
  - 操作
  - 软件包名
  - 创建程序
  - 起始时间
  - 经过时间
  - 总计 **CPU** 时间
  - 文本

事件监视器和“事件分析程序”的联机帮助提供了有关创建事件监视器并查看生成的事件数据的详细说明。

---

## 分析 SQL 语句

可将解释的 SQL 语句的访问计划作为一个图来查看，并使用此信息来调整 SQL 查询以获得更好的性能。

访问计划图显示以下详细信息：

- 表（及其相关的列）和索引
- 运算符（如表扫描、排序以及连接）
- 表空间和函数

在版本 6 之前，可使用 Visual Explain 工具查看访问计划。现在，不能再从命令行把 Visual Explain 当作一个单独的工具调用；但是，仍可从“控制中心”的不同数据库对象或从“命令中心”调用 Visual Explain 功能。在本节中，术语 *visual explain* 功能即表示此功能。

可使用 visual explain 功能：

- 查看优化时使用的统计信息。然后将这些统计信息与当前目录统计信息比较，以帮助确定重新绑定软件包是否可以改进性能。
- 确定是否使用了索引来访问表。若未使用索引，则 `visual explain` 功能可帮助您确定若创建了索引哪些列可能受益。
- 通过比较一个查询的访问计划图调整之前和之后的版本，查看执行各种调整技术的效果。
- 获取访问计划中关于每个操作的信息，包括总估计成本和检索行数（基数）。

## 改进查询性能

可使用 `visual explain` 功能来分析和帮助调整 SQL 语句。它将解释的 SQL 语句的访问计划显示为一个图形视图。表和索引以及对它们执行的每个操作表示为节点，数据流由这些节点之间的链路表示。可使用此图中的信息来找到调整 SQL 查询以获取更好性能的方法。

`visual explain` 功能捕捉 SQL 语句的编译信息。此信息使您可了解 SQL 语句的计划和潜在的执行性能。

此信息可帮助您：

- 设计应用程序。
- 设计数据库。
- 了解两个表是如何连接的：使用的连接方法、连接表的次序、排序实例和排序类型。
- 确定改进 SQL 语句性能的方法（例如，创建新索引）。
- 查看优化时使用的统计信息。然后将这些统计信息与当前目录统计信息比较，以帮助确定重新绑定软件包是否可以改进性能。它也帮助您确定收集统计信息是否可改进性能。
- 确定是否使用了索引来访问表。若未使用索引，则 `visual explain` 功能可帮助您确定哪些列可包括在索引中以有助于改善查询性能。
- 比较一个查询的访问计划图调整之前和之后的版本，可看到为获得更优性能而执行各种调整技术的效果。
- 获取访问计划中关于每个操作的信息，包括总估计成本和检索行数。

在使用 `visual explain` 功能来理解说明 SQL 语句的访问计划之后，您可以确定索引可改进该查询的性能。您应使用“索引向导”来接收对该查询建议的索引；或者，也可以使用 `RECOMMENDED_INDEXES EXPLAIN` 方式。有关“索引向导”的更多信息，转至“控制中心”并进入“信息中心”。

有关 RECOMMENDED\_INDEXES EXPLAIN 方式的更多信息，参考《管理指南：性能》。

## 分析简单的动态 SQL 语句

本节提供有关如何分析动态 SQL 查询的一个简单示例。

1. 从“控制中心”，右键单击 SAMPLE 数据库，然后从弹出菜单中选择**解释 SQL**。“解释 SQL 语句”窗口打开。
2. 在 **SQL 文本** 字段中，输入以下 SQL 语句：

```
select * from staff order by name
```
3. 单击**确定**。“访问计划图”窗口打开。该图表示优化器为提供查询的结果选择的最有效路径。
4. 可选的：双击任何一个节点（例如，RETURN 运算符节点）。“运算符详细信息”窗口打开，显示该运算符的详细信息。

自动保存解释的 SQL 语句。若以后要查看它：

1. 从“控制中心”，右键单击 SAMPLE 数据库，然后从弹出菜单中选择**显示解释语句历史**。“解释语句历史”窗口打开。
2. 找到想要的项。可查看 **SQL 文本** 列来查看先前已解释的 SQL 语句。
3. 右键单击该项，然后从弹出菜单中选择**显示访问计划**。“访问计划图”窗口打开。

Visual explain 的联机帮助（可从**帮助**菜单访问）提供有关如何解释“访问计划图”窗口以改进 SQL 语句性能的详细信息。该联机帮助还包含详细的示例，可帮助您学习如何使用 Visual Explain。

---

## 管理远程数据库

下面内容介绍如何：

- 添加远程系统
- 添加要对该系统使用的实例
- 添加要在该实例下使用的数据库

DB2 首先检查节点目录（它对一个数据库客户机可以连接的所有服务器和该连接中使用的通信协议都包含一项）以查看远程系统是否是已知的。若远程系统是未知的，且有系统、实例或数据库在远程系统上，则需要将您自己设置为远程系统的客户机。



当安装 DB2 后，可使用“客户机配置辅助程序”在网络中搜索系统、实例和数据库，并为它们配置通信。然后编目远程系统以添加它。这将在节点目录中为系统创建一项，以便可使它的实例和数据库成为已知的。然后，必须通过编目来添加系统的实例和数据库，即分别在节点目录和数据库目录中为它们创建一项。这将在节点目录和数据库目录中为它们创建一项。当配置完成时，那些远程系统会显示在“控制中心”中，这样可使用它们。

要添加远程系统:

1. 从“控制中心”，右键单击**系统**对象，然后从弹出菜单中选择**添加**。“添加系统”窗口打开。
2. 在**系统名**字段中输入系统名。  
若将实例的 **Discover** 配置参数设置为 **search**，且 **discover comm** 配置参数不是空白，可选择**刷新**来获取远程系统的列表。然后可从**系统名**字段下的列表中选择其中一个系统。
3. 在**远程实例名**字段中输入远程实例名。
4. 从**操作系统**列表中选择远程系统的操作系统类型。
5. 选择要用于与远程位置通信的协议。对于本地系统，会自动选择**本地**，但它不是有效的协议。对于远程系统，可能的协议有：
  - APPC
  - IPX/SPX
  - NetBIOS
  - TCP/IP
  - 命名管道（仅适用于 Windows NT 和 Windows 9x 操作系统）

只有当前为计算机设置的那些协议出现在列表框中。

6. 输入恰当的协议参数。
7. 输入与系统相关的一个注释。
8. 单击**应用**以将系统添加到节点目录。

然后添加要在该系统上使用的实例:

1. 从“控制中心”，右键单击属于您刚添加的系统的**实例**对象。
2. 从弹出菜单选择**添加**。“添加实例”窗口打开。
3. 在字段中输入必需值。
4. 单击**刷新**按钮以显示现存实例的列表。
5. 选择要使用的实例。
6. 单击**应用**按钮，然后单击**关闭**按钮。

最后，在该实例下添加要使用的数据库：

1. 从“控制中心”，右键单击**数据库**对象。
2. 从弹出菜单选择**添加**。“添加数据库”窗口打开。
3. 输入数据库名和通信协议的类型，可任选地输入别名。在本例中别名是用于标识数据库的替代名称。
4. 单击**刷新**按钮以显示该实例的现存数据库列表。
5. 选择数据库。
6. 单击**应用**按钮，然后单击**关闭**按钮。

---

## 管理用户

作为数据库管理员，您可能需要控制用户对数据访问的类型，或限制他们查看数据。以下信息告诉您如何使用管理工具来管理数据库权限和数据库对象的特权。

数据库权限涉及到对作为一个整体的数据库所执行的操作。当创建一个数据库时，会自动将某些权限授予访问该数据库的任何人。例如，将 **CONNECT**、**CREATETAB**、**BINDADD** 和 **IMPLICIT\_SCHEMA** 权限授予所有用户。数据库特权涉及到对数据库内的特定对象执行的操作。当创建一个数据库时，会自动将某些特权授予访问该数据库的任何人。例如，授予对目录视图的 **SELECT** 特权，将每个成功绑定的实用程序的 **EXECUTE** 和 **BIND** 特权授予所有用户。

特权和权限一起控制对一个实例及其数据库对象的访问权。用户只能访问那些他们具有适当授权（即必需的特权或权限）的对象。

### 授予和取消权限和特权

可使用 **DB2** 管理工具来为数据库、表空间、表、视图和模式的用户和组授予和取消特权。

1. 从“控制中心”，右键单击要授予或取消特权的数据库、表、视图、模式或索引。从弹出菜单选择**权限**或**特权**。“权限”窗口或“特权”窗口打开。
2. 选择**用户**页来使用用户权限或特权，或选择**组**页来使用组权限或特权。
3. 选择一个或多个用户或组。要向列表添加用户或组，单击**添加用户**或**添加组**按钮。
4. 在窗口底部，对每个单独的权限或特权选择**是**、**否**或**授权**。仅对适用的对象显示**授予**。
5. 完成时，单击**应用**按钮。

若要复查或更改特定用户被授权使用的对象，可选择一个用户，单击鼠标右键，然后添加或更改一个对象的授权或除去授权。

---

## 移动数据

DB2 提供“导入”、“导出”和“装入”实用程序来帮助您从现存源将数据移入一个表。本节中提供的信息是关于移动数据的简短概述。有关移动数据的更详细的信息，应参考 *Data Movement Utilities Guide and Reference* 手册。

“导入”实用程序从输入文件中提取数据，然后将该数据插入一个表或视图。在这种情况下，输入文件包含从现存数据源如 Lotus 1-2-3 文件或 ASCII 文件抽取的数据。也可使用导入实用程序重新创建用“导出”实用程序保存的表或视图。下列信息讲述如何导入数据。

一旦有一个格式受支持的输入文件，可使用“导入”笔记本将文件中的数据插入现存的表中。若此表已包含数据，可用该文件中的数据替换现存的数据或追加至现存数据之后。

也可使用“导入”笔记本来创建一个新表，可用输入文件填充；或在所选表中删除现存的行，并使用输入文件中的数据重新填充它。

要将一个文件导入现存的表：

1. 打开“导入”笔记本的“文件”页。
2. 可选的。指定“导入”笔记本。
3. 可选的。检索“大对象”。
4. 可选的。指定列导入选项。
5. 单击**确定**

要打开“导入”笔记本的“文件”页：

1. 从“控制中心”，展开该对象树，直到找到**表**文件夹。
2. 单击**表**文件夹。任何现存的表都显示在内容窗格中。
3. 右键单击内容窗格中的一个表，然后从弹出菜单中选择**导入**。“导入”笔记本打开，并显示“文件”页。

要指定文件选项：

1. 在“文件”页的**导入文件**字段中，输入包含要导入的数据的文件名。
2. 选择下列其中一项，以指定要导入的文件类型
  - 非定界的 ASCII 格式 (ASC)  
非定界的 ASCII 数据是在列中对齐的数据。

- 定界的 ASCII 格式 (DEL)

定界的 ASCII 数据是存储数据的常见方式，它使用用户定义的定界字符如逗号来分隔列值。

- 工作表格式 (WSF)
- 集成交换格式 (IXF)

PC/IXF 是数据库表或视图的结构化描述。用 PC/IXF 格式导出的数据可导入或装入到另一个“DB2 通用数据库”产品数据库中。

参阅支持的特定产品和发行版的联机帮助。

3. 可选的：单击对应的**选项**按钮，以指定文件类型修饰符。该格式的“选项”窗口打开。
4. 选择**导入方式**。可用的导入方式随选择的文件类型的不同而改变。
5. 可选的：在**提交记录数**字段中，输入在提交更改前要导入的记录数。
6. 可选的：在**重新启动**字段中，输入在开始导入操作前在文件中要跳过的记录数。
7. 可选的：在**组合**字段中，输入一个数以指定在一个可执行块中将执行多少条 SQL 语句。
8. 可选的：选择在**十进制数据中插入隐含的十进制小数点 (IMPLIEDECIMALPOINT)** 复选框。
9. 在**信息文件**字段中，输入将包含在导入期间出现的警告和出错信息的文件的名称。

要从单独的文件中检索大对象，使用“导入”笔记本的“大对象”页从存储大对象 (LOB) 文件的一个或多个路径中检索 LOB:

1. 单击在**单独的文件中检索大对象 (LOB) (LOBSINFILE)**复选框，以启用“大对象”页上的选项。
2. 单击**添加**按钮，以便在 LOB 路径列表框中指定单独的 LOB 文件的位置。搜索这些路径（按它们出现在 **LOB 路径**列表框中的次序），查找在输入文件的 LOB 列中指定的 LOB 文件。
3. 单击**确定**，以接受在其他笔记本页上的缺省值并开始导入过程。

指定列导入选项。使用**导入**笔记本的**列**页，以指定列导入选项:

1. 在**包括列方法**框中单击其中一个单选按钮，以指定把数据文件列导入该表所用的列方法。可用的方法随在“文件”页上选择的文件类型和方式的不同而改变。
2. 可选的：单击**更改**按钮，以指定或更改导入文件列属性。  
若选择了**缺省值 (方法 D)**单选按钮，则此选项不可用。

作为数据库管理员，您需要估计表和索引的大小，检查表空间中可用的空间量，并在现有的表空间满时向它添加更多空间。

本节描述如何：

- 估计表和索引的大小
- 检查表空间中可用的空间量
- 当一个现存的表空间开始变满时，向其添加更多空间

### 估计表和索引大小

可调用“估计大小”对话框来估计新的或现存的表或索引的存储空间容量。选择个别表和索引，右键单击它们，以调用此对话框；或从“创建表”和“创建索引”窗口中选择**估计大小**。根据特定表及其从属索引的定义来估计大小。该估计值是在表具有给定行数时要使用的预定存储空间容量。也根据变长字段的最小大小和最大大小来估计最小和最大空间。当在一个表或索引上调用时，**估计大小**对话框预先填有该表的说明，并包含与该表及其所有从属索引相关的数。当单击**刷新**按钮时，根据您在**新的总行数**和**新的平均行长度**字段中输入的数来更新估计大小、最小大小和最大大小。

估计表或索引的大小对下列情况有帮助：

- 创建新表，且想知道可以使表空间增长到多大。
- 根据现存表的大小估计创建新表。
- 想知道系统用完存储空间时，在一个表空间中不同的表和索引对象使用了多大的空间。
- 估计在装入数据前一个表的预定大小。

**注：**当在“DB2 通用数据库扩充企业版”产品上使用“估计大小”时，确保根据表中数据的逻辑大小来估计大小。

若一段时间内未更新表的统计信息，可单击**运行统计信息**按钮来更新所选表的统计信息。若选择一个索引并按下**运行统计**按钮，则对相关的表进行统计。

要估计一个表的大小：

- 打开“估计大小”窗口。
- 为**新的总行数**选择不同的值，或接受缺省值。
- 单击**刷新**以查看新值的大小估计。
- 为**新的平均行长度**选择不同的值，或接受缺省值。
- 单击**刷新**以查看新值的大小估计。

## 检查表空间中的可用空间

要检查一个 DMS 表空间中可用的空间量:

1. 从“控制中心”，双击**表空间**图标。所有表空间的列表显示在内容窗格中。
2. 滚动至**分配的大小**、**使用的大小**和**使用的百分比**这些列，查看与一个表空间中可用的空间量相关的更多信息。空间按页计，一页是 4 KB。

可使用“内容”窗格底部的“定制列”图标来定制列的次序并确定显示哪些列。

要检查一个 SMS 表空间中的可用空间，使用操作系统提供的设施来监视空间使用情况，并确保该表空间的目录中可用的空间未用完。

## 向表空间添加更多空间

一个 DMS 表空间的容量是分配给该表空间的容器的总计大小。当一个 DMS 表空间达到其容量（取决于表空间的使用情况，90% 是可能的阈值）时，应向其添加更多的空间。数据库管理器将自动用所有可用容器来重新平衡 DMS 表空间中的表。在重新平衡期间，该表空间中的数据仍是可访问的。

对于已达到其容量的 DMS 表空间，可添加另一个容器:

1. 从“控制中心”，右键单击“内容”窗格中要添加容器的表空间，然后从弹出菜单中选择**改变**。“改变表空间”窗口打开。
2. 单击**添加**。“添加容器”窗口打开。
3. 选择**文件**或**原始设备**单选按钮，并完成那些字段。参阅联机帮助以获得辅助。
4. 单击**确定**。

一般情况下，扩展 SMS 表空间的大小并不容易，因为 SMS 容量取决于文件系统中可用的空间以及操作系统支持的文件的最大大小。但可能可以使用操作系统设施增加文件系统的大小，这取决于您的操作系统。对于 UNIX 系统上的 SMS 表空间，可使用适当的 UNIX 系统命令来增加表空间的大小。参阅运行的 UNIX 系统的有关文档。若包含 SMS 表空间的文件系统也包含非 DB2 文件，可能可以将这些文件移动到另一个文件系统，从而使该文件系统中有更多的空间可供 DB2 使用。也可执行重定向复原，以将一个表空间复原为相对于备份时的大小而言，有更多数目的容器。可从“复原数据库”笔记本执行重定向复原：选择要复原的数据库，从弹出菜单中选择**复原** → **数据库**。

---

## 故障诊断

DB2 提供了一本故障诊断手册，供 DB2 服务器和客户机的技术支持代表参考。它帮助您：

- 以一种简明的方式标识问题或错误
- 根据其症状解决问题
- 使用可用的诊断工具
- 为日常执行的 DB2 操作制定故障诊断策略

*Troubleshooting Guide* 描述下列基本的故障诊断主题：

- 故障诊断常用方法
- 服务器故障诊断
- 客户机故障诊断
- 主机通信故障诊断
- 应用程序故障诊断
- 故障诊断和问题确定。

*Troubleshooting Guide* 描述下列高级故障诊断主题：

- DB2 进程模型
- 使用记录的信息
- 采取跟踪
- UNIX、OS/2 、Microsoft Windows 操作系统所用的诊断工具。

最新简报和技术文档可从万维网上的 <http://www.software.ibm.com/data/db2/library/> 处获得。

参阅本书末尾的那一节，以获取如何与 IBM 联系的详细信息。

---

## 复制数据

复制是从源服务器上的数据库日志中提取更改，然后将它们应用于目标服务器的过程。可以使用复制来定义、同步化、自动化和管理整个企业数据的复制操作。可自动将数据从主机系统传送到目标站点。例如，可将数据和应用程序复制到分部、零售点，甚至销售代表的膝上型计算机。

复制中使用的两个操作组件是 Capture 和 Apply。Capture 组件通过读取数据库日志，来捕捉对要复制的源表数据所做的更改。Apply 组件读取先前捕捉的并存储在更改数据表中的已更改数据，并将它应用于目标表。

通过“控制中心”，可使用定义为复制源和定义预订操作来执行复制必需的设置。复制组件 Capture 和 Apply 在 DB2 管理工具外运行。

复制管理员可从“控制中心”执行下列操作：

- 定义复制源
- 定义复制预订
- 指定 SQL 以便在 Apply 过程期间增强数据

复制数据的高级步骤如下所示。有关详细信息，参考 *Replication Guide and Reference*。

1. 设计复制计划（映射源表和目标表）。
2. 定义复制源（这与 Capture 操作相关）。

要定义复制源：

1. 指定要捕捉的源列。
2. 选择复制选项。
3. 定义复制预订（这与 Apply 操作相关）。
4. 使用“数据捕捉更改”选项改变源表。
5. 启动 Capture 以读取并存储数据更改。
6. 启动 Apply 以将更改复制到目标表。

要定义复制预订：

1. 命名该预订集。
2. 指定数据库和目标表。
3. 指定目标列。
4. 指定行选择。
5. 指定用于运行时处理的 SQL。
6. 设置预订定时。

---

## 使用轻量级目录访问协议

可使用“客户机配置辅助程序”(CCA) 来在 LDAP 服务器上添加和删除项。所有在 LDAP 服务器上注册的数据库实例都将在客户机上自动编目（高速缓存）。在“控制中心”中，它们作为正规节点显示在导航器树中。可按您机器上已编目的其他数据库的相同管理方式来管理这些数据库（唯一不同的是在此发行版中 ADD DATABASE 和 DROP DATABASE 选项尚不可用）。



要管理 LDAP 数据库，选择该数据库并单击鼠标右键。一个弹出窗口列出可执行的功能。有关 LDAP 的更多信息，参见第355页的『附录J. “轻量级目录访问协议” (LDAP) 的目录服务』。

---

## 使用 Java 控制中心

可将该“控制中心”作为一个 Java 应用程序来运行，或通过 web 服务器将它作为 Java 小应用程序来运行。在这两种情况下，需要在机器上安装一个受支持的“Java 虚拟机” (JVM) 来运行“控制中心”。要将该“控制中心”作为一个 Java 应用程序运行，还必须安装正确的“Java 运行时环境” (JRE)。“Java 虚拟机”可以是可运行应用程序的“Java 运行时环境” (JRE)，或是可运行小应用程序的支持 Java 的浏览器。

只要安装了正确的 JRE，Java 应用程序的运行与机器上其他应用程序的运行相似。

### 将“控制中心”作为 Java 小应用程序运行

Java 小应用程序是在支持 Java 的浏览器内运行的程序。“控制中心”小应用程序代码可驻留在远程机器上，并可通过 Web 服务器供客户机的浏览器调用。若将“控制中心”作为 Java 小应用程序运行，必须使用在 Windows 32 位或 OS/2 操作系统上运行的、受支持的启用 Java 的浏览器。目前，没有受 UNIX 操作系统支持的浏览器。

“控制中心 JDBC 小应用程序服务器”必须用在该小应用程序服务器驻留的机器上具有管理员权限的用户帐户来启动。可将“控制中心 JDBC 小应用程序服务器”设置为启动时自动启动。

要将“控制中心”作为 Java 小应用程序运行，必须在包含“控制中心”小应用程序代码和“控制中心 JDBC 小应用程序服务器”的机器上设置 Web 服务器。该 Web 服务器必须允许访问 sqllib 目录。若选择使用一个虚拟目录，则用此目录替换主目录。例如，若将虚拟目录命名为 temp，则应使用 sqllib/temp。DB2 不支持在 OS/2 的 FAT 驱动器上安装“控制中心”，因为 OS/2 FAT 驱动器不支持 Java 需要的长文件名。有关将“控制中心”作为 Java 应用程序或 Java 小应用程序来安装和配置的更多信息，参考您的平台的《快速入门》手册。

---

## 使用 Java 工具进行管理

DB2 包括一组 Java 接口，这些接口可用来扩展“控制中心”的能力。这些 Java 接口允许您：

- 在使用对象时向菜单列表添加附加的项目。
- 向“控制中心”工具栏添加按钮。

要利用此能力，必须安装级别正确的 Java 软件。有关使用此功能的更多信息，参见第385页的『附录K. 扩展控制中心』。

---

## 第2部分 实现设计



---

## 第2章 在创建数据库之前

在确定了数据库的设计之后，必须创建数据库和其中的对象。这些对象包括模式、节点组、表空间、表、视图、包装器、服务器、别名、类型映射、函数映射、别名、用户定义类型 (UDT)、用户定义函数 (UDF)、触发器、约束、索引和软件包。可以在命令行处理器中使用 SQL 语句、从控制中心或通过应用程序中的 API 来创建这些对象。

有关 SQL 语句的信息，参考 *SQL Reference* 手册。有关命令行处理器命令的信息，参考 *Command Reference*。有关 API 的信息，参考 *Administrative API Reference* 手册。

**注：**您的平台也许支持可用来创建数据库对象的用户界面。此界面可以用来代替 SQL 语句、命令行处理器命令或 API。检查为您的平台配备的《快速入门》手册，以确定您是否具有此能力。

在本章中，使用“控制中心”完成任务的方法放在框中，予以强调。后面立即跟有使用命令行的方法，供您比较，有时还带有示例。在有些情况下，任务只显示了一种方法。当使用“控制中心”时，请您记住，您可以使用那里的帮助，它可以提供比此处的概述信息更多的详细信息。

本章主要讨论您在创建数据库及其所有对象之前应该了解的信息。在创建数据库之前，您必须了解一些先决条件概念和主题，并执行几项任务。

下一章包含各种对象的简要讨论，那些对象可能会成为您的数据库设计实现的一部分。

此部分的最后一章先是讨论在改变数据库之前必须考虑的主题，然后说明了如何改变或删除数据库对象。

对于“DB2 通用数据库”与操作系统相互影响的领域，本章及下一章中的某些主题可能会讨论操作系统特定的差别。您也许能够利用本机操作系统的能力或差别，而不是 DB2 UDB 提供的那些能力或差别。您应该参考适当的《快速入门》手册和特定的操作系统文档，以了解准确的差别。

例如，Windows NT\*\* 支持称为“服务”的应用程序类型。DB2 Windows NT 版可以将一个 DB2 实例定义为一个服务。一个服务可以在系统引导时自动启动，这可由用户通过“服务”控制面板小应用程序来启动，或通过基于 Win32 的一个应

用程序来启动，该应用程序使用包括在 Microsoft\*\* Win32\*\* 应用程序编程接口 (API) 中的服务功能。甚至当没有用户登录至该系统时，服务也可以执行。

---

## 创建数据库之前的先决条件

在实现数据库前，应了解下列先决条件任务：

- 『启动 DB2』
- 第47页的『在 Windows NT 上启动 DB2 UDB』
- 第47页的『使用数据库管理器的多个实例』
- 第48页的『按模式组织和分组对象』
- 第49页的『启用并行性』
- 第50页的『启用数据分区』
- 第52页的『停止 DB2』

## 启动 DB2

在正常的商业操作期间，可能需要启动或停止 DB2；例如，在可以执行下列任务前必须启动一个实例：

- 在该实例中与一个数据库连接
- 预编译应用程序
- 将软件包与数据库绑定
- 访问主机数据库。

要在系统上启动一个 DB2 实例：

1. 用在该实例中具有 SYSADM、SYSCTRL 或 SYSMAINT 权限的用户标识或名称登录；或者作为实例所有者登录。
2. 在 UNIX 操作系统上，运行启动脚本，如下所示：

```
. INSTHOME/sql1lib/db2profile      (对于 Bourne 或 Korn shell)
source INSTHOME/sql1lib/db2cshrc   (对于 C shell)
```

其中，INSTHOME 是您想要使用的实例的主目录。

3. 使用这两种方法之一来启动实例：
  - a. 使用“控制中心”来启动实例：

- 1) 展开对象树，直到您看到**实例**文件夹为止。
- 2) 右键单击您想要启动的实例，并从弹出菜单中选择**启动**。

- b. 使用命令行来启动实例，输入：

db2start

注: **db2start** 命令根据第60页的『设置当前实例』中的规则启动实例。

## 在 Windows NT 上启动 DB2 UDB

db2start 命令将 DB2 作为 NT 服务来启动。通过在调用 DB2START 时指定 "/D" 开关, 仍可以将 Windows NT 上的 DB2 作为进程来运行。使用“控制面板”或“NET START”命令也可以将 DB2 作为服务启动。

为了成功地从 DB2START 将 DB2 作为一个服务来启动, 用户帐户必须具有 Windows NT 操作系统定义的正确特权, 才可启动 NT 服务。用户帐户可以是“管理员”、“服务器操作员”或“高级用户”组的一个成员。

当在分区数据库环境中运行时, 每个数据库分区服务器都作为 NT 服务来启动。

## 使用数据库管理器的多个实例

可在单个服务器上创建数据库管理器的多个实例。这意味着可以在一台物理机器上创建同一个产品的几个实例, 并使它们并行运行。这提供了在设置环境方面的灵活性。

您可能希望有多个实例来创建下列环境:

- 将开发环境与生产环境分离。
- 针对实例要服务的特定应用程序单独调整每一个实例。
- 保护敏感信息, 使管理员访问不到它。例如, 也许需要将工资单数据库保护在它自己的实例中, 以使其他实例的所有者不能查看工资单数据。

DB2 程序文件实际存储在特定机器上的一个位置中。创建的每个实例都指向此位置, 这样不会为创建的每个实例复制程序文件。几个相关的数据库可以位于单个实例内。

在节点目录中将实例编目为本地的或远程的。缺省实例由 DB2INSTANCE 环境变量来定义。可以与其他实例连接, 来执行只能在实例级执行的维护和实用程序任务, 如创建数据库、强制断开应用程序、监视数据库或更新数据库管理器配置。当试图与不在缺省实例中的实例连接时, 将使用该节点目录来确定如何与该实例通信。

*Command Reference* 提供了有关执行每个命令所需的连接类型的信息。

DB2 对多个实例的支持因操作系统的不同而不同。有关在一台机器上定义多个 DB2 实例的信息, 参考适合您平台的《快速入门》指南。

要与另一个可能是远程的实例连接，使用 ATTACH 命令，如 *Command Reference* 手册中所述。

使用“控制中心”：

1. 展开对象树，直到您看到**实例**文件夹为止。
2. 单击您想要连接的实例。
3. 右键单击选择的实例名。
4. 在“连接 DB2”窗口中，输入用户标识和密码，并单击**确定**。

要使用命令行来与实例连接，输入：

```
db2 attach to <instance name>
```

例如，要与节点目录中先前编目的称为 testdb2 的实例连接：

```
db2 attach to testdb2
```

在对 testdb2 实例执行了维护活动之后，可执行以下命令来与该实例拆离：

```
db2 detach
```

## 按模式组织和分组对象

数据库对象名可由单个标识符组成，也可以是由两个标识符组成的模式限定对象。模式限定对象的模式或高位部分提供了一种将数据库中的对象分类或分组的方法。当创建象表、视图、别名、单值类型、函数、索引、软件包或触发器这样的对象时，会给它分配一个模式。此指定可显式或隐式地执行。

当在一条语句中引用对象时若使用了两部分对象名的高位部分，则显式使用了该模式。例如，用户 A 在模式 C 中发出 CREATE TABLE 语句，如下所示：

```
CREATE TABLE C.X (COL1 INT)
```

当不使用两部分对象名的高位部分时，即是隐式使用该模式。当发生这种情况时，CURRENT SCHEMA 专用寄存器用于标识完成对象名的高位部分所用的模式名。CURRENT SCHEMA 的初始值是当前会话用户的授权标识。若希望在当前会话期间更改它，可使用 SET SCHEMA 语句来将该专用寄存器设置为另一个模式名。有关更多信息，参考 *SQL Reference*。

正如第92页的『系统目录表的定义』中描述的，当创建数据库时会在特定的模式内创建一些对象。



在动态 SQL 语句中，模式限定对象名隐式地使用 CURRENT SCHEMA 专用寄存器值来作为非限定对象名引用的限定符。在静态 SQL 语句中，QUALIFIER 预编译 / 绑定选项隐式地指定非限定数据库对象名的限定符。

在创建自己的对象之前，需要考虑是按自己的模式创建，还是通过使用将对象按逻辑分组的另一种模式来创建。若正在创建将共享的对象，则使用不同的模式名将会非常有用。有关如何显式创建模式的更多信息，参见第103页的『创建模式』。

## 启用并行性

必须修改配置参数，以利用数据库分区内或非分区数据库内的并行性。例如，可以使用分区内并行性来利用对称多处理机 (SMP) 机器上的多个处理器。

### 启用分区内并行性

可使用“控制中心”来了解或修改特定数据库中或数据库管理器配置文件中的个别项的值。

您还可以使用 GET DATABASE CONFIGURATION 和 GET DATABASE MANAGER CONFIGURATION 命令，以了解特定数据库中或数据库管理器配置文件中的个别项的值。要修改特定数据库或数据库管理器配置文件中的个别项，可分别使用 UPDATE DATABASE CONFIGURATION 和 UPDATE DATABASE MANAGER CONFIGURATION 命令。

影响分区内并行性的配置参数包括 *max\_querydegree* 和 *intra\_parallel* 数据库管理器参数，以及 *dft\_degree* 数据库参数。有关配置参数的更多信息，参考《管理指南：性能》。

### 启用分区内查询并行性

为使分区内查询并行性可用，必须修改数据库配置参数和数据库管理器配置参数。

#### **INTRA\_PARALLEL**

数据库管理器配置参数。有关此参数的更多信息，参考《管理指南：性能》。

#### **DFT\_DEGREE**

数据库配置参数。提供 DEGREE 绑定选项和 CURRENT DEGREE 专用寄存器的缺省值。有关此参数的更多信息，参考《管理指南：性能》。

#### **DEGREE**

静态 SQL 的预编译或绑定选项。有关更多信息，参考 *Command Reference*。

## CURRENT DEGREE

动态 SQL 的专用寄存器。有关更多信息，参考 *SQL Reference*。

有关配置参数设置和如何使应用程序以并行方式处理的更多信息，参考《管理指南：性能》中的“配置 DB2”。

### 启用分区间查询并行性

根据数据库分区的数目和数据在这些分区上的分布，分区间并行性自动生效。

### 启用实用程序并行性

本节概述如何为下列实用程序启用分区内并行性：

- 装入
- 创建索引
- 备份数据库 / 表空间
- 复原数据库 / 表空间

根据数据库分区的数目，实用程序的分区间并行性自动生效。

**装入：** “装入”实用程序自动利用并行性，也可在 LOAD 命令上使用下列参数：

- CPU\_PARALLELISM
- DISK\_PARALLELISM

有关 LOAD 命令的信息，参考 *Data Movement Utilities Guide and Reference*。

**自动装入程序：** 可以对 autoloader.cfg 文件中的 LOAD 说明指定 MODIFIED BY ANYORDER 参数，以对“自动装入程序”启用多个单独的进程。有关更多信息，参考 *Data Movement Utilities Guide and Reference*。

**创建索引：** 要在创建索引时启用并行性：

- INTRA\_PARALLEL 数据库管理器配置参数必须设置为 ON
- 表必须足够大，以便能从并行性受益
- 在 SMP 机器上必须启用多个处理器。

有关 CREATE INDEX 语句的信息，参考 *SQL Reference*。

## 启用数据分区

当在多分区数据库环境中运行时，可以使用 CREATE DATABASE 命令或 sqlcrea() 应用程序编程接口 (API)，从任何存在于 db2nodes.cfg 文件中的节点创建数据库。有关信息，参考 *Command Reference* 和 *Administrative API Reference* 手册。

在创建一个分区数据库之前，必须确定您将是作为本地客户机还是作为远程客户机连接至将在其中创建数据库的实例。第二，您必须与该实例连接。还必须选择哪个数据库分区将是该数据库的目录节点。您要连接并对其执行 `CREATE DATABASE` 命令的数据库分区成为该特定数据库的目录节点。

目录节点是用于存储所有系统目录表的数据库分区。对系统表的所有访问都必须通过此数据库分区进行。所有联合数据库对象（包装器、服务器、别名等）都存储在此节点的系统目录表中。

若可能的话，应该在单独的实例中创建每个数据库。若不可能做到此点（即，必须在每个实例中创建多个数据库），应该将目录节点分布至可用的数据库分区中。这样做可以减少在单个数据库分区中对目录信息的争用。

**注：**应该定期执行目录节点的备份，并避免将用户数据置于该节点上（无论何时都有可能），因为其他数据会增加备份所需的时间。

当创建数据库时，会自动在 `db2nodes.cfg` 文件中定义的所有数据库分区上创建它。

当创建了系统中的第一个数据库时，就会形成一个系统数据库目录。将有关您创建的任何其他数据库的信息追加至它后面。系统数据库目录是 `sqldbdir`，它位于主目录下的 `sqllib` 目录中。此目录必须驻留在共享文件系统（例如，UNIX 平台上的 NFS）上，因为对于组成分区数据库的所有数据库分区只存在一个系统数据库目录。

驻留在 `sqldbdir` 目录中的还有系统意向文件。它称为 `sqldbins`，用于确保数据库分区保持同步。该文件也必须驻留在共享文件系统上，因为所有数据库分区中只有一个目录。该文件由组成数据库的所有分区共享。

必须修改配置参数，才能利用数据分区。使用 `GET DATABASE CONFIGURATION` 和 `GET DATABASE MANAGER CONFIGURATION` 命令，以了解特定数据库中或数据库管理器配置文件中的个别项的值。要修改特定数据库中或数据库管理器配置文件中的个别项，可分别使用 `UPDATE DATABASE CONFIGURATION` 和 `UPDATE DATABASE MANAGER CONFIGURATION` 命令。

影响分区数据库的数据库管理器配置参数包括 `conn_elapse`、`fcm_num_anchors`、`fcm_num_buffers`、`fcm_num_connect`、`fcm_num_rqb`、`max_connretries`、`max_coordagents`、`max_time_diff`、`num_poolagents` 和 `stop_start_time`。

有关配置参数的更多信息，参考《管理指南：性能》。

## 备份数据库 / 表空间

要在备份数据库或表空间时启用 I/O 并行性:

- 使用多个目标媒体。
- 为并行 I/O 配置表空间。
- 在 BACKUP 命令上使用 PARALLELISM 参数以指定并行度。
- 在 BACKUP 命令上使用 WITH num-buffers BUFFERS 参数, 以保证提供足够的缓冲区来满足该并行度。缓冲区数应比已有的目标媒体数与选择的并行度之和略大。

同时使用满足以下条件的备份缓冲区大小:

- 尽可能大。4 MB 或 8 MB (1024 或 2048 页) 比较合适。
- 至少等于要备份的表空间中 (块大小 \* 容器数之积) 的最大者。

有关 BACKUP DATABASE 命令的信息, 参考 *Command Reference*。

## 复原数据库 / 表空间

要在复原数据库或表空间时启用 I/O 并行性:

- 使用多个源媒体。
- 为并行 I/O 配置表空间。
- 在 RESTORE 命令上使用 PARALLELISM 参数以指定并行度。
- 在 RESTORE 命令上使用 WITH num-buffers BUFFERS 参数, 以保证提供足够的缓冲区来满足该并行度。缓冲区数应比已有的目标媒体数与选择的并行度之和略大。

同时使用满足以下条件的复原缓冲区大小:

- 尽可能大。4 MB 或 8 MB (1024 或 2048 页) 比较合适。
- 至少等于要复原的表空间中 (块大小 \* 容器数之积) 的最大者。
- 等于备份缓冲区大小或是其偶数倍。

有关 RESTORE DATABASE 命令的信息, 参考 *Command Reference*。

## 停止 DB2

**db2stop** 命令只能在服务器上运行。当运行此命令时, 不允许有任何数据库连接; 但是, 若有任何实例连接, 则在停止 DB2 之前要强制将它们断开。

**注:** 若命令行处理器会话与一个实例连接, 则在运行 **db2stop** 命令前必须运行 **terminate** 命令来结束每个会话。**db2stop** 命令停止由 DB2INSTANCE 环境变量定义的实例。

要在系统上停止一个 DB2 实例, 必须执行下列操作:

1. 用在该实例中具有 SYSADM、SYSCTRL 或 SYSMAINT 权限的用户标识或名称登录或与实例连接；或者作为实例所有者登录。
2. 显示与要停止的特定数据库连接的所有应用程序和用户。要确保没有关键性的或极重要的应用程序在运行，列出应用程序。为此，需要 SYSADM、SYSCTRL 或 SYSMAINT 权限。
3. 强制所有应用程序和用户与该数据库断开。需要 SYSADM 或 SYSCTRL 权限来强制用户。
4. 在 UNIX 操作系统上，运行启动脚本，如下所示：

```
. INSTHOME/sql1lib/db2profile      (对于 Bourne 或 Korn shell)
source INSTHOME/sql1lib/db2cshrc   (对于 C shell)
```

其中，INSTHOME 是您想要使用的实例的主目录。

5. 使用以下一个方法停止实例：

- a. 展开对象树，直到您找到实例文件夹为止。
- b. 单击您想要停止的每个实例。
- c. 右键单击选择的任何实例，并从弹出菜单中选择**停止**。
- d. 在“确认停止”窗口上，单击**确定**。

使用命令行来停止实例，输入：

```
db2stop
```

---

## 有关创建数据库的详细信息

在创建数据库之前，应该考虑或执行下列任务：

- 设计逻辑数据库和物理数据库的特征
- 创建实例
- 建立环境变量和简要表注册表
- 创建 DB2 管理服务器 (DAS)
- 创建节点配置文件
- 创建数据库配置文件
- 使用响应文件复制配置信息
- 启用 FCM 通信

### 设计逻辑数据库和物理数据库的特征

在创建数据库之前，必须决定如何设计逻辑数据库和物理数据库。要对逻辑和物理数据库设计有更多的了解，参考《管理指南：计划》。

## 创建实例

实例是一个逻辑数据库管理器环境，可在其中编目数据库并设置配置参数。根据需要，可创建多个实例。可使用多个实例执行以下操作：

- 将一个实例用作开发环境，将另一个实例用作生产环境。
- 调整一个实例以用作特定的环境。
- 限制对敏感信息的访问。
- 控制每个实例中对 `SYSADM`、`SYSCTRL` 和 `SYSMAINT` 权限的指定。
- 优化每个实例的数据库管理器配置。
- 限制实例失效所带来的影响。若一个实例失效，则只影响一个实例。其他实例可继续正常运行。

应注意多个实例存在一些小缺点：

- 每个实例都需要额外的系统资源（虚拟内存和磁盘空间）。
- 由于要管理附加实例，因此增加了管理工作量。

实例目录存储着与一个数据库实例相关的所有信息。实例目录一旦创建，就不能更改其位置。该目录包含：

- 数据库管理器配置文件
- 系统数据库目录
- 节点目录
- DB2 诊断文件 (`db2diag.log`)
- 节点配置文件 (`db2nodes.cfg`)
- 包含调试信息（如异常情况 / 寄存器转储或用于完成 DB2 进程的调用堆栈）的任何其他文件。

在 UNIX 操作系统上，该实例目录位于 `INSTHOME/sql1ib` 目录中，其中 `INSTHOME` 是实例所有者的主目录。

在分区数据库系统中，该实例目录是由属于该实例的所有数据库分区服务器共享的。因此，必须在该实例中的所有机器可以访问的一个网络共享驱动器上创建实例目录。

在安装期间，要创建一个 DB2 的初始实例，称为 "DB2"。在 UNIX 上，可以随意命名该初始实例，只要符合命名规则。实例名用于设置目录结构。

要想立即使用此实例，在安装期间设置下列各项：

- 将环境变量 `DB2INSTANCE` 设置为 "DB2"。
- 将 DB2 注册表变量 `DB2INSTDEF` 设置为 "DB2"。

在 UNIX 上，可以随意命名该缺省实例，只要符合命名规则。

这些设置将“DB2”建立为缺省实例。可以更改在缺省情况下使用的实例，但首先必须创建一个附加实例。

在使用 DB2 前，必须更新每个用户的数据库环境，以便该环境可访问实例并运行 DB2 程序。这适用于所有用户（包括管理类用户）。

在 UNIX 操作系统上，提供了样本脚本文件来帮助您设置数据库环境。这些文件有：用于 Bourne 或 Korn shell 的 db2profile，以及用于 C shell 的 db2cshrc。这些脚本位于实例所有者主目录下的 sqllib 子目录中。实例所有者或属于该实例的 SYSADM 组的任何用户可为该实例的所有用户定制脚本。或者，可为每个用户复制并定制该脚本。

样本脚本包含用于执行以下操作的语句：

- 将以下目录添加到现有的搜索路径中以更新用户的 PATH：在实例所有者主目录的 sqllib 子目录下的 bin、adm 和 misc 子目录。
- 将 DB2INSTANCE 环境变量设置为实例名。

### 自动设置 DB2 环境

**注：**本讨论只适用于 UNIX 操作系统环境。

在缺省情况下，仅在当前会话期间脚本才影响用户环境。可更改 .profile 文件，以便当用户使用 Bourne 或 Korn shell 登录时可自动运行 db2profile 脚本。对于 C shell 的用户，可更改 .login 文件以使它可运行 db2shrc 脚本文件。

将下列其中一条语句添加到 .profile 或 .login 脚本文件：

- 对于共享该脚本的一个版本的用户，添加：

```
. INSTHOME/sqllib/db2profile    (对于 Bourne 或 Korn shell)
source INSTHOME/sqllib/db2cshrc (对于 C shell)
```

其中，INSTHOME 是希望使用的实例的主目录。

- 对于在其主目录中存在该脚本的定制版本的用户，添加：

```
. USERHOME/db2profile        (对于 Bourne 或 Korn shell)
source USERHOME/db2cshrc     (在 C shell 中)
```

其中，USERHOME 是用户的主目录。

### 手工设置 DB2 环境

**注：**本讨论只适用于 UNIX 操作系统环境。

要选择您要使用的实例，在命令提示符处输入下列其中一条语句。句点 (.) 和空格是必需的。

- 对于共享该脚本的一个版本的用户，添加：

```
. INSTHOME/sql1lib/db2profile    (对于 Bourne 或 Korn shell)
source INSTHOME/sql1lib/db2cshrc  (对于 C shell)
```

其中，INSTHOME 是希望使用的实例的主目录。

- 对于在其主目录中存在该脚本的定制版本的用户，添加：

```
. USERHOME/db2profile           (对于 Bourne 或 Korn shell)
source USERHOME/db2cshrc        (在 C shell 中)
```

其中，USERHOME是用户的主目录。

若要同时使用多个实例，则在单独的窗口中对要使用的每个实例运行该脚本。例如，假设有两个实例 test 和 prod，它们的主目录是 /u/test 和 /u/prod。

在窗口 1 中：

- 在 Bourne 或 Korn shell 中，输入：

```
. /u/test/sql1lib/db2profile
```
- 在 C shell 中，输入：

```
source /u/test/sql1lib/db2cshrc
```

在窗口 2 中：

- 在 Bourne 或 Korn shell 中，输入：

```
. /u/prod/sql1lib/db2profile
```
- 在 C shell 中，输入：

```
source /u/prod/sql1lib/db2cshrc
```

用窗口 1 实现 test 实例，用窗口 2 实现 prod 实例。

**注：**输入 **which db2** 命令，以保证搜索路径已正确设置。此命令返回 DB2 CLP 可执行文件的绝对路径。验证它位于该实例的 sql1lib 目录下。

### 系统上有多个实例

一个系统上可能有多个实例。然而，每次只能在 DB2 的一个实例内工作。

实例所有者和“系统管理”(SYSADM)组与每个实例相关。实例所有者和SYSADM组是在创建实例期间指定的。一个用户标识或用户名只能用于唯一一个实例。该用户标识或用户名也称为实例所有者。



每个实例所有者必须有一个唯一的主目录。运行实例需要的所有文件都在该实例所有者的用户标识或用户名的主目录中创建。若需要从系统中除去实例所有者的用户标识或用户名，可能会丢失与该实例相关的文件并失去对存储在此实例中的数据的访问权。由于这个原因，建议将一个实例所有者的用户标识或用户名专门用于运行 DB2。

实例所有者的主组也很重要。此主组自动成为该实例的系统管理组，并获得该实例的 **SYSADM** 权限。作为该实例所有者主组成员的其他用户标识或用户名也获得此级别的权限。因此，可能需要将该实例所有者的用户标识或用户名指定给为管理实例而保留的一个主组。（还要确保将一个主组分配给该实例所有者的用户标识或用户名；否则，使用系统的缺省主组。）

若已经有一个组并希望使它成为该实例的系统管理组，当创建实例所有者的用户标识或用户名时可将此组指定为主组。要赋予其他用户对该实例的管理权限，将他们添加到指定作为系统管理组的那一组。

为了将不同实例的 **SYSADM** 权限区别开，确保每个实例所有者的用户标识或用户名使用不同的主组。但是，若选择对多个实例具有公共的 **SYSADM** 权限，则可对这多个实例使用同一个主组。

### 添加实例

若在 OS/2 上具有管理权限，或者在 Windows NT 上属于管理组，或者在 UNIX 平台上具有超级权限，则可添加附加的 DB2 实例。添加实例的机器成为“拥有实例的机器”（节点零）。确保在“管理服务器”驻留的机器上添加实例。

要添加另一实例，执行下列步骤：

1. 在具有“管理”权限或属于本地“管理员”组的用户标识或名称下登录。
2. 要添加实例，使用下列其中一种方法：

使用“控制中心”：

- a. 展开对象树，直到找到您想要的系统的实例文件夹为止。
- b. 右键单击实例文件夹，并从弹出菜单中选择添加。
- c. 填写信息，并单击应用。

要使用命令行添加实例，输入：

```
db2icrt <instance_name>
```

3. 创建“管理服务器”。

当使用 **db2icrt** 命令来添加 DB2 的另一实例时，应该提供实例所有者的登录名，并可选择是否指定该实例的认证类型。该认证类型适用于在该实例下创建的所有

数据库。认证类型是对将在何处进行用户认证的说明。有关认证的更多信息，参见第193页的『第5章 控制数据库访问』。

**注：**可以选择使用 **db2iupdt** 命令来更新实例配置。

可在 **DB2PATH** 中使用 **DB2INSTPROF** 环境变量更改实例目录的位置。需要该实例目录的写访问权。若想要在非 **DB2PATH** 的路径中创建目录，在输入 **db2icrt** 命令前必须设置 **DB2INSTPROF**。

**添加实例时的 DB2 扩充企业版详细信息：** 当使用“DB2 通用数据库扩充企业版”时，还需要声明正添加的新实例是一个分区数据库系统。这可在命令行上使用 **-s eee** 来完成。

**创建实例时的 UNIX 详细信息：** 当使用 UNIX 操作系统时，**db2icrt** 命令具有下列可选参数：

- **-h** 或 **-?**

此参数用于显示此命令的帮助菜单。

- **-d**

此参数设置在确定问题期间要使用的调试方式。

- **-a AuthType**

此参数指定该实例的认证类型。有效的认证类型是 **SERVER**、**CLIENT**、**DCS** 或 **DCE**。若未指定此参数，且若已安装了 **DB2** 服务器，则缺省值为 **SERVER**。否则，将它设置为 **CLIENT**。

**注：**

1. 该实例的认证类型适用于实例拥有的所有数据库。
2. 在 UNIX 操作系统上，认证类型 **DCE** 不是有效的选项。

- **-u FencedID**

此参数是受防护的用户定义函数 (UDF) 和存储过程执行期间所归属的用户。若安装了 **DB2** 客户机或“**DB2** 应用程序开发客户机”，则不需要此参数。对于其他 **DB2** 产品，它是必需的参数。

**注：** **FencedID** 不能是 **"root"** 或 **"bin"**。

- **-p PortName**

此参数指定要使用的 **TCP/IP** 服务名或端口号码。对于实例中的每个数据库，将在实例的数据库配置文件中设置此值。

- **-s InstType**

允许添加不同类型的实例。有效的实例类型是：**ee**、**eee** 和 **client**。

示例:

- 要为 DB2 服务器添加实例, 您可以使用如下命令:

```
db2icrt -u db2fenc1 db2inst1
```

- 若只安装了“DB2 Connect 企业版”, 也可将该实例名用作 Fenced ID:

```
db2icrt -u db2inst1 db2inst1
```

- 要为 DB2 客户机添加实例, 您可以使用如下命令:

```
db2icrt db2inst1 -s client -u fencedID
```

当要让一个工作站与其他数据库服务器连接, 且该工作站上不需要本地数据库时, 便创建 DB2 客户机实例。

**创建实例时的 Windows NT 详细信息:** 当使用 Windows NT 操作系统时, **db2icrt** 命令具有下列可选参数:

- `-s InstType`

允许创建不同类型的实例。有效的实例类型是: `ee`、`eee` 和 `client`。

- `/p:InstProf_Path`

这是一个可选参数, 用来指定另一实例简要表路径。若不指定该路径, 将在 `SQLLIB` 目录下创建实例目录, 并将共享名称 `DB2` 与实例名并置后的名称作为其名称。自动将读写许可权授予域中的每个人。可以更改许可权, 以限制该目录的访问权。

若指定另一个实例简要表路径, 必须创建一个共享的驱动器或目录。除非更改了许可权, 否则这将允许域中的每个用户有机会访问实例目录。

- `/u:username,password`

当创建分区数据库环境时, 必须声明 `DB2` 服务的登录、帐户名和密码。

- `/r:base_port,end_port`

这是一个可选参数, 用于指定“快速通信管理器”(FCM) 的 `TCP/IP` 端口范围。若指定 `TCP/IP` 端口范围, 则必须确保该端口范围在分区数据库系统中的所有机器上都可用。

例如, 在“DB2 Windows NT 版扩充企业版”上, 可以参考以下示例:

```
db2icrt inst1 -s eee
/p:\machineA\db2mpp
/u:yourname,yourpwd /r:9010,9015
```

**注:** **db2icrt** 命令授予用来创建实例的用户名如下权限:

- 作为操作系统的一部分操作
- 创建令牌对象

- 增加份额
- 作为服务登录
- 替换进程级令牌

该实例需要这些用户权利来访问共享驱动器、认证用户帐户和作为 Windows NT 服务运行 DB2。

## 列出实例

要使用“控制中心”获得系统上可用的所有实例的列表:

1. 展开对象树，直到找到系统的实例文件夹为止。
2. 右键单击“实例”文件夹，并从弹出菜单中选择添加。
3. 在“添加数据库”窗口上，单击刷新。
4. 单击下拉箭头来查看数据库实例的列表。
5. 单击取消以退出该窗口。

要使用命令行列出系统上可用的所有实例，输入:

```
db2ilist
```

要确定哪一个实例适用于当前会话，输入:

```
set db2instance
```

**注:** 在 UNIX 操作系统上，输入:

```
db2 get instance
```

## 设置当前实例

当运行命令来启动或停止实例的数据库管理器时，DB2 将该命令应用于当前实例。

DB2 按如下所示确定当前实例:

- 若为当前会话设置 DB2INSTANCE 环境变量，则其值为当前实例。要设置 DB2INSTANCE 环境变量，输入:

```
set db2instance=<new_instance_name>
```

- 若没有为当前会话设置 DB2INSTANCE 环境变量，则 DB2 使用系统环境变量中 DB2INSTANCE 环境变量的设置。在 Windows NT 上，在“系统环境”中设置系统环境变量。在 Windows 95 上，在 autoexec.bat 文件中设置它们。在 OS/2 上，在 config.sys 文件中设置它们。
- 若根本没有设置 DB2INSTANCE 环境变量，则 DB2 使用注册表变量 DB2INSTDEF。

要在注册表的全局级设置 DB2INSTDEF 注册表变量，输入:

```
db2set db2instdef=<new_instance_name> -g
```

### 自动启动实例

在 UNIX 操作系统上，要允许一个实例在每次系统重新启动后自动启动，输入以下命令：

```
db2iauto -on InstName
```

其中 InstName 是实例的登录名。

在 UNIX 操作系统上，要阻止一个实例在每次系统重新启动后自动启动，输入以下命令：

```
db2iauto -off InstName
```

其中 InstName 是实例的登录名。

### 同时运行多个实例

可以启动多个 DB2 实例，只要它们使用同一级别的代码。

要使用“控制中心”并行地运行多个实例：

1. 展开对象树，直到您找到数据库文件夹为止。
2. 右键单击一个实例，并从弹出菜单中选择启动。
3. 重复步骤 2，直到启动了要并行运行的所有实例为止。

要使用命令行并行地运行多个实例：

1. 输入以下命令，将 DB2INSTANCE 变量设置为要启动的另一个实例的名称：

```
set db2instance=<another_instName>
```

2. 输入 **db2start** 命令以启动该实例。

## 许可证管理

DB2 产品的许可证管理主要是通过该产品的联机界面即“控制中心”内的“许可证中心”来进行的。从“许可证中心”，可针对每个安装的产品检查许可证信息、统计信息、注册的用户和当前用户。

## 建立环境变量和简要表注册表

环境变量和注册表变量控制数据库环境。

在引入 DB2 简要表注册表之前，例如在 Windows 或 OS/2 工作站上更改环境变量要求您在更改环境变量之后重新引导。现在，除少数例外情况，您的环境均受到存储在 DB2 简要表注册表中的注册表变量的控制。具有给定实例的系统管理

(SYSADM) 权限的用户可更新该实例的注册表值。使用 **db2set** 命令来更新注册表变量，而不需重新引导；此信息会立即存储在简要表注册表中。该 DB2 注册表将更新的信息应用于在进行更改后启动的 DB2 服务器实例和 DB2 应用程序。

当更新注册表时，更改不会影响当前正在运行的 DB2 应用程序或用户。在更新后启动的应用程序将使用新值。

**注：** DB2 环境变量 DB2INSTANCE、DB2NODE、DB2PATH 和 DB2INSTPROF 不能存储在 DB2 简要表注册表中，这取决于操作系统。为了更新这些环境变量，必须使用 **set** 命令。在下次重新引导系统之后，这些更改不再有效。在 UNIX 平台上，可以使用 **export** 命令来代替 **set** 命令。

使用简要表注册表允许集中控制环境变量。《管理指南：性能》中的“DB2 注册表和环境变量”列出了许多环境变量和注册表变量。现在通过不同的环境简要表提供了不同级别的支持。当使用“DB2 管理服务器”时，还提供了对环境变量的远程管理。

有四个简要表注册表：

- DB2 实例级简要表注册表。大多数 DB2 环境变量都位于此注册表中。特定实例的环境变量设置保存在此注册表中。在此级别定义的值将覆盖在全局级的对应设置。
- DB2 全局级简要表注册表。若未对特定的实例设置环境变量，则使用此注册表。此注册表具有在当前机器范围内有效的环境变量设置。在 DB2 UDB EEE 中，每台机器上都有一个全局级简要表。
- DB2 实例节点级简要表注册表。在数据库分布于不同的数据库分区的系统中，此注册表驻留在每个节点（即，机器）上，且包含在该节点上存储数据的所有实例的环境变量设置。在此级别定义的值将覆盖在实例级和全局级的对应设置。
- DB2 实例简要表注册表。此注册表包含此系统可识别的所有实例名的列表。

用户可以使用 **set** 命令（或 UNIX 平台上的导出命令）来更改会话环境变量设置，以覆盖用于其会话的“DB2 实例简要表注册表”环境变量设置。

DB2 按下列次序检查注册表值和环境变量并确定它们的值以配置操作环境：

1. 使用 **set** 命令设置的环境变量。（或 UNIX 平台上的 **export** 命令。）
2. 使用实例节点级简要表设置的注册表值（使用带有节点号的 **db2set -i** 命令，如下所示）。
3. 使用实例简要表设置的注册表值（使用 **db2set -i** 命令，如下所示）。
4. 使用全局简要表设置的注册表值（使用 **db2set -g** 命令，如下所示）。

## 使用 db2set 命令

**db2set** 命令支持本地声明注册表变量（和环境变量）。

要显示该命令的帮助信息，使用：

```
db2set ?
```

要列出所有受支持的注册表变量的完整集合，使用：

```
db2set -lr
```

要列出当前实例或缺省实例的所有已定义的注册表变量，使用：

```
db2set
```

要列出简要表注册表中所有定义的注册表变量，使用：

```
db2set -all
```

要显示一个注册表变量在当前实例或缺省实例中的值，使用：

```
db2set registry_variable_name
```

要显示一个注册表变量在所有级别的值，使用：

```
db2set registry_variable_name -all
```

要删除一个变量在特定级别的值，可使用相同的命令语法来设置该变量但不对该变量值指定任何内容。例如，要删除该变量在节点级的设置，输入：

```
db2set registry_variable_name= -i instance_name  
node_number
```

要删除一个变量的值并限制其使用，若它是在更高的简要表级别定义的，输入：

```
db2set registry_variable_name= -null instance_name
```

此命令将删除指定的参数的设置，并限制高级别的简要表更改此变量的值（在本示例中为 DB2 全局级简要表）。但指定的变量仍可能会被低级别简要表（在本示例中为 DB2 节点级简要表）设置。

要在当前实例或缺省实例中更改一个注册表变量，使用：

```
db2set registry_variable_name=new_value
```

要更改该实例中所有数据库的注册表变量缺省值，使用：

```
db2set registry_variable_name=new_value  
-i instance_name
```

要更改系统中所有实例的注册表变量缺省值，使用：

```
db2set registry_variable_name=new_value -g
```

要设置用户级的注册表变量，使用：

```
db2set -u1
```

要为特定用户设置用户级的注册表变量，使用：

```
db2set -u1 user_name
```

**注：**

1. 在同一个命令中不能同时使用参数 "-i"、"-g" 和 "-ul"。
2. 某些参数将始终缺省为全局级简要表。不能在实例级或节点级简要表如 db2system 和 db2instdef 设置它们。
3. 在 UNIX 上，必须具有系统管理 (SYSADM) 权限，才能更改实例的注册表值。只有具有超级权限的用户才能更改全局级注册表的参数。

当在 LDAP 环境中运行时，有可能在 LDAP 中将 DB2 注册表变量值设置为作用域适用于属于某目录分区或 Windows NT 域的所有机器和所有用户。目前，可以在 LDAP 全局级设置的 DB2 注册表变量只有 DB2LDAP\_SEARCH\_SCOPE。

要在 LDAP 全局级设置此变量，使用 db2set 命令的 -gl 选项。

**注：**这与用来在机器全局级设置 DB2 注册表变量的 -g 选项不同。-gl 专门用于 LDAP 全局级。并且，在 LDAP 中设置此 DB2注册表变量只有在 Windows 平台上才受支持。

要在 LDAP 中的全局级设置搜索范围值，使用：

```
db2set -gl db2ldap_search_scope = value
```

其中，value 可以是 "local"、"domain"或 "global"。

要更改实例中特定节点的注册表变量缺省值，使用：

```
db2set registry_variable_name=new_value  
-i instance_name node_number
```

要将实例的注册表变量复位为在“全局简要表注册表”中找到的缺省值，使用：

```
db2set -r registry_variable_name
```

要将实例中节点的注册表变量复位为在“全局简要表注册表”中找到的缺省值，使用：

```
db2set -r registry_variable_name node_number
```



## 在 OS/2 上设置环境变量

极力建议针对 DB2 的所有注册表变量都在 DB2 简要表注册表中定义。若 DB2 变量是在注册表外设置的，则不可能对这些变量进行远程管理，要使变量值生效必须重新启动该工作站。

在 OS/2 上，不应在 config.sys 中定义除 DB2PATH 和 DB2INSTPROF 之外的任何环境变量。除了那些保持为真的环境变量外，所有其他变量都应使用 **db2set** 命令在简要表注册表中定义。

DB2INSTANCE 也是真的环境变量，但是若使用 DB2INSTDEF 注册表变量，则不需要它。若未设置 DB2INSTANCE，则此注册表变量定义要使用的缺省实例名。

DB2INSTANCE 和 DB2PATH 是在安装 DB2 时设置的；而 DB2INSTPROF 可以在安装后设置。必须设置环境变量 DB2PATH；此环境变量是在安装期间设置的，且不应修改它。设置 DB2INSTANCE 和 DB2INSTPROF 环境变量是可选的。

要确定环境变量的设置，输入：

```
set variable
```

要更改环境变量的设置，输入以下命令：

```
set variable=value
```

要设置系统环境变量，执行下列操作：编辑 config.sys 文件，然后重新引导系统以使更改生效。

根据下列说明，查找不同的简要表注册表：

- “DB2 实例级简要表注册表” 文件位于：  
    %DB2INSTPROF%\instance\_name\PROFILE.ENV
- “DB2 全局级简要表注册表” 位于：  
    %DB2INSTPROF%\DEFAULT.ENV
- “DB2 实例简要表注册表” 位于：  
    %DB2INSTPROF%\PROFILES.REG

## 在 Windows NT 和 Windows 95 上设置环境变量

极力建议针对 DB2 的所有注册表变量都在 DB2 简要表注册表中定义。若 DB2 变量是在注册表外设置的，则不可能对这些变量进行远程管理，要使变量值生效必须重新启动该工作站。

Windows 32 位操作系统有一个系统环境变量 DB2INSTANCE，可以在简要表注册表外设置它；但不是必须设置 DB2INSTANCE。可以在全局级简要表设置 DB2 简要表注册表变量 DB2INSTDEF，以指定要使用的实例名（若未定义 DB2INSTANCE）。

Windows NT 上的“DB2 扩充企业版”服务器有两个系统环境变量 DB2INSTANCE 和 DB2NODE，只能在简要表注册表外设置它们。不要求您设置 DB2INSTANCE。可以在全局级简要表设置 DB2 简要表注册表变量 DB2INSTDEF，以指定要使用的实例名（若未定义 DB2INSTANCE）。

DB2NODE 环境变量用于将请求按路径发送到一台机器内的目标逻辑节点。必须在发出该应用程序或命令的会话中，而不在 DB2 简要表注册表中，设置此环境变量。若未设置此变量，则目标逻辑节点缺省为在该机器上用端口 0 定义的那个逻辑节点。

要确定环境变量的设置，使用 **echo** 命令。例如，要检查 DB2PATH 环境变量的值，输入：

```
echo %db2path%
```

要设置系统环境变量，执行下列操作：

**在 Windows 95 和 Windows 98 上：**编辑 *autoexec.bat* 文件，然后重新引导系统以使更改生效。

**在 Windows NT 4.x 上：**可设置 DB2 环境变量 DB2INSTANCE、DB2PATH 和 DB2INSTPROF，如下所示：

- 选择开始，设置，控制面板。
- 双击系统图标。
- 在“系统控制面板”的“系统环境变量”部分，执行下列操作：
  1. 若 DB2INSTANCE 变量不存在：
    - a. 选择任何系统环境变量。
    - b. 将变量字段中的名称更改为 DB2INSTANCE。
    - c. 将值字段更改为实例名，例如 db2inst。
  2. 若 DB2INSTANCE 变量已经存在，则追加新的值：
    - a. 选择 DB2INSTANCE 环境变量。
    - b. 将值字段更改为实例名，例如 db2inst。
  3. 选择“设置”。
  4. 选择“确定”。
  5. 重新引导系统，以使这些更改生效。

**注:** 也可在会话（进程）级设置环境变量 DB2INSTANCE。例如，若想启动第二个 DB2 实例 TEST，在命令窗口中发出下列命令：

```
set db2instance=TEST
db2start
```

简要表注册表位于如下所示的位置：

- “DB2 实例级简要表注册表” 位于 Windows NT 操作系统注册表中，其路径为：

```
\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\PROFILES\instance_name
```

**注:** *instance\_name* 特指您正在使用的数据库分区。

- “DB2 全局级简要表注册表” 位于 Windows NT 注册表中，其路径为：

```
\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\GLOBAL_PROFILE
```

- “DB2 实例节点级简要表注册表” 位于 Windows NT 注册表中，其路径为：

```
...\SOFTWARE\IBM\DB2\PROFILES\instance_name\NODES\node_number
```

**注:** *instance\_name* 和 *node\_number* 特指您正在使用的数据库分区。

DB2 UDB 提供在远程机器上访问实例级的 DB2 UDB 注册表变量的能力。当前，DB2 UDB 注册表变量以三种不同的级别存储：机器或全局级、实例级和节点级。对于以实例级（包括节点级）存储的注册表变量，可使用 DB2REMOTEPREG 将其重定向到其他机器。当设置了 DB2REMOTEPREG 后，DB2 UDB 将从 DB2REMOTEPREG 指向的机器访问 DB2 UDB 注册表变量。例如，

```
db2set DB2REMOTEPREG=rmtwkstn
```

其中，*rmtwkstn* 是远程工作站名。

**注:** 应该小心设置此选项，因为所有的 DB2 实例简要表和实例列表都将位于这个指定的远程机器名上。

此功能可与设置 DBINSTPROF 结合起来使用，以指向同一台机器上包含该注册表的远程 LAN 驱动器。

### 在 UNIX 系统上设置环境变量

极力建议针对 DB2 的所有注册表变量都在 DB2 简要表注册表中定义。若 DB2 变量是在注册表外设置的，则不可能对这些变量进行远程管理。

在 UNIX 操作系统上，必须设置系统环境变量 DB2INSTANCE。

提供脚本 **db2profile**（对于 Korn shell）和 **db2cshrc**（对于 Bourne shell 或 C shell）作为示例，以帮助您设置数据库环境。可以在 `insthome/sqllib` 中找到这些文件，其中 `insthome` 是实例所有者的主目录。

这些脚本包括对下列各项的说明：

- 使用下列目录来更新用户的路径：
  - `insthome/sqllib/bin`
  - `insthome/sqllib/adm`
  - `insthome/sqllib/misc`
- 将 `DB2INSTANCE` 设置为用于执行的缺省本地 `instance_name`。

**注：**除 `PATH` 和 `DB2INSTANCE` 外，所有其他 DB2 支持的变量都必须在 DB2 简要表注册表中设置。要设置 DB2 不支持的变量，在脚本文件 **db2profile** 和 **db2cshrc** 中定义它们。

实例所有者或 `SYSADM` 用户可以为一个实例的所有用户定制这些脚本。或者，用户可以复制和定制脚本，然后直接调用脚本或将它添加至它们的 `.profile` 或 `.login` 文件。

要更改当前会话的环境变量，发出类似于以下的命令：

- 对于 Korn shell：

```
db2instance=inst1
export db2instance
```
- 对于 Bourne shell 或 C shell：

```
set db2instance inst1
```

为了正确地管理 DB2 简要表注册表，在 UNIX 操作系统上必须遵循下列文件所有权规则。（有关“DB2 管理服务器”（DAS）的信息，参见第69页的『创建 DB2 管理服务器（DAS）』。）

- “DB2 实例级简要表注册表”文件位于：

```
INSTHOME/sqllib/profile.env
```

此文件的访问许可权和所有权应该是：

```
-rw-r--r-- Instance_Owner DAS_Instance_Group profile.env
```

`INSTHOME` 是实例所有者的主路径。

- “DB2 全局级简要表注册表”位于：
  - 对于 AIX、Solaris、SINIX 和 NUMA-Q(Sequent) 操作系统，位于 `/var/db2/<version_id>/default.env`（其中，`<version_id>` 是当前版本）。

- 对于 HP-UX 操作系统，位于 `/var/opt/db2/<version_id>/default.env`（其中，`<version_id>` 是当前版本）。

此文件的访问许可权和所有权应该是：

```
-rw-r--r-- DAS_Instance_Owner DAS_Instance_Group default.env
```

为了修改全局注册表变量，用户必须作为 `root` 用户或 `DAS` 实例所有者登录。有关“DB2 管理服务器”的更多信息，参见『创建 DB2 管理服务器 (DAS)』。

- “DB2 实例节点级简要表注册表”位于：

```
INSTHOME/sql1lib/nodes/node_number.env
```

该目录和此文件的访问许可权和所有权应该是：

```
drwxrwxr-x Instance_Owner DAS_Instance_Group nodes
```

```
-rw-r--r-- Instance_Owner DAS_Instance_Group node_number.env
```

**注：**`Instance_Owner` 和 `DAS_Instance_Owner` 应该都是 `DAS_Instance_Group` 的成员。

`INSTHOME` 是实例所有者的主路径。

- “DB2 实例简要表注册表”位于：
  - 对于 AIX、Solaris、SINIX 和 NUMA-Q(Sequent) 操作系统，位于 `/var/db2/<version_id>/profiles.reg`（其中，`<version_id>` 是当前版本）。
  - 对于 HP-UX 操作系统，位于 `/var/opt/db2/<version_id>/profiles.reg`（其中，`<version_id>` 是当前版本）。

此文件的访问许可权和所有权应该是：

```
-rw-r--r-- root system profiles.reg
```

## 创建 DB2 管理服务器 (DAS)

“DB2 管理服务器” (DAS) 是一个特殊的 DB2 管理控制点，它仅用于辅助其他 DB2 服务器上的管理任务。若要使用“客户机配置辅助程序”或“控制中心”，则必须要有一个正在运行的 DAS。当执行下列管理任务时，DAS 会辅助“控制中心”和“客户机配置辅助程序”。

- 启用 DB2 服务器的远程管理。
- 为作业管理提供设施，包括调度 DB2 及操作系统命令脚本执行的能力。这些命令脚本是用户定义的。“控制中心”用于定义作业的调度、查看已完成作业的结果，并对 DAS 的远程或本地作业执行其他管理任务。

- 与 DB2 Discovery 实用程序一起提供一种查找方法，以查找关于 DB2 实例、数据库及其他“DB2 管理服务器”的配置信息。“客户机配置辅助程序”和“控制中心”使用此信息，来简化和自动执行至 DB2 数据库的客户机连接的配置。

在一台机器上只能有一个 DAS。在安装期间，将 DAS 配置为在引导操作系统时启动。

DAS 用于在主机系统上代表来自“控制中心”或“客户机配置辅助程序”的客户机请求执行远程任务。对 DAS 的授权访问要求客户机具有 SYSADM 权限。所有客户机都可以是 SYSADM\_GROUP 配置参数的一部分。

某些请求的任务可能需要特定的权限才可运行。DAS 在特定用户的标识符下运行。授予该用户的特权只能用来执行那些与管理员将要执行的任务或操作相关联的命令。通常，必需的任务或操作包括：

- 查询操作系统 (OS) 配置信息。
- 查询用户和组信息的 OS。
- 对其他 DB2 实例执行操作，以启动或停止它们。
- 执行调度作业。
- 收集“连通性和协议配置”的信息。

有关设置 DAS 通信的更多信息，参见适合您平台的《快速入门》。

## 创建 DAS

通常，设置程序在安装 DB2 期间会在拥有实例的机器上创建一个 DAS。但是，若设置程序创建它失败，可手工创建 DAS。

当安装过程与 DAS 相关时，作为在安装过程期间发生的情况的一个概述，考虑下列事项：

- 在 OS/2 或 Windows NT 平台上：

使用具有本地“管理员”权限的帐户，登录到要创建 DAS 的机器上。若要标识特定的用户，创建具有本地“管理员”权限的用户。输入 `db2admin create`。若期望使用一个特定的用户帐户，在发出 `db2admin create` 时必须使用 `"/USER:"` 和 `"/PASSWORD:"`。

当创建 DAS 时，可以选择是否提供用户帐户名和用户密码。若用户帐户名和密码有效，它们将标识该 DAS 的所有者。不要使用为 DAS 创建的用户标识或帐户名作为“用户帐户”。将该帐户名的密码设置为“密码永远不到期”。在创建了 DAS 之后，就可以使用 `db2admin setid` 命令提供一个用户帐户名和用户密码，来建立或修改它的所有权。有关此命令的更多信息，参考 *Command Reference*。

在“DB2 UDB Windows NT 版扩充企业版”上，若要使用“客户机配置辅助程序”或“控制中心”来自动配置与 DB2 服务器的连接，则与 DAS 在同一机器上的数据库分区服务器将是协调程序节点。这表示从客户机到数据库的所有物理连接在按路径发送到其他数据库分区服务器之前，将定向到拥有实例的机器上的数据库分区服务器。

在“DB2 UDB Windows NT 版扩充企业版”上，在其他机器上创建附加的“管理服务器”可以允许“客户机配置辅助程序”或“控制中心”使用 DB2 Discovery 将其他系统配置为协调程序节点。为此，执行下列操作：

1. 使用具有本地“管理员”权限的帐户登录到机器上。
2. 创建一个具有本地“管理员”权限的 Windows NT 帐户，以供 DAS 使用。确保该帐户的用户名符合 DB2 命名约定。当为 DAS 创建该帐户时，注意下列事项：
  - 不要使用该 DAS 的帐户作为“用户帐户”。
  - 将该帐户的密码设置为密码永远不到期。
3. 运行以下命令：

```
db2admin create /user:username  
/password:passwd
```

其中，username 和 passwd 是 DAS 的用户名和密码。

• 在 UNIX 平台上：

1. 确保您具有超级权限。
2. 在命令提示符处，从 DB2 通用数据库实例路径的 instance 子目录中发出以下命令：

```
dasicrt ASName
```

- 在 AIX 上：

```
/usr/lpp/db2_nn_00&/instance/  
dasicrt ASName
```

- 在 HP-UX、NUMA-Q(Sequent) 或 Solaris 上：

```
/opt/IBMDB2/<version_id>/instance/  
dasicrt ASName
```

- 在 Linux 上：

```
/usr/IBMDB2/<version_id>/instance/  
dasicrt ASName
```

其中，ASName 是“管理服务器”的实例名，而 db2\_nn\_00& 或 <version\_id> 是当前版本标识符。

注：若是在运行 NIS 和 NIS+，则需要按以下方式设置用户名和组名：

- DAS 的主组必须在所有实例的次级组中。
- DAS 的次级组必须包含所有实例的主组。

仅当 NIS 和 NIS+ 不在系统上运行时，才自动修改次级组列表。

因为一个用户标识只能拥有一个实例，因此必须有一个单独的用户标识来拥有您创建的每个“DB2 管理服务器”(DAS)。

一旦创建了“管理服务器”，则应使用它来建立目录结构和访问许可权。

### 启动和停止 DAS

要手工启动或停止 DAS，必须首先使用具有本地管理员权限的帐户或用户标识登录到机器上。

当在 DB2 OS/2 版或 DB2 Windows NT 版环境中工作时，必须执行以下操作：

- 要启动 DAS，输入 `db2admin start`
- 要停止 DAS，输入 `db2admin stop`

**注：**对于 Windows NT 下的这两种情况，使用这些命令的人必须具有 `SYSADM`、`SYSCTRL` 或 `SYSMAINT` 权限。

当在 DB2 的任何一个 UNIX 操作系统版环境中工作时，必须执行以下操作：

- 要启动 DAS：
  1. 作为 DAS 所有者登录。
  2. 使用下列其中一个命令运行启动脚本：

```
. INSTHOME/sqlllib/db2profile    (对于 Bourne 或 Korn shell)
source INSTHOME/sqlllib/db2cshrc (对于 C shell)
```

其中，INSTHOME是该实例的主目录。

3. 要启动 DAS，请使用 **db2admin** 命令：

```
db2admin start
```

**注：**在每次系统重新启动后自动启动 DAS。

- 要停止 DAS：
  1. 作为 DAS 所有者登录。
  2. 使用下列其中一个命令运行启动脚本：

```
. INSTHOME/sqlllib/db2profile    (对于 Bourne 或 Korn shell)
source INSTHOME/sqlllib/db2cshrc (对于 C shell)
```

其中，INSTHOME是该实例的主目录。

3. 使用 **db2admin** 命令停止 DAS，如下所示：



```
db2admin stop
```

**注：**对于在 UNIX 下的这两种情况，使用这些命令的人必须先使用 DAS 所有者的授权标识登录。

### 列出 DAS

要获得您机器上 DAS 实例的名称，输入：

```
db2admin
```

还可使用此命令来启动或停止 DAS、创建新用户和密码、删除 DAS 实例以及建立或修改与该 DAS 实例相关联的用户帐户。

### 配置 DAS

要查看与 DAS 相关的那些管理配置参数的当前值，输入：

```
db2 get admin cfg
```

它显示在安装该产品期间指定为缺省值的当前值，或在先前更新配置参数期间指定的那些值。

要更新与 DAS 相关的数据库管理器配置文件中的个别项，输入：

```
db2 update admin cfg using ...
```

有关可修改哪些数据库管理器配置参数的更多信息，参考 *Command Reference*。

要将配置参数复位为建议的数据库管理器缺省值，输入：

```
db2 reset admin cfg
```

对数据库管理器配置文件的更改仅在把更改装入内存后才生效（即，在执行 db2admin stop 之后，再执行 db2admin start 时生效；或对于 Windows NT 平台，停止并再启动该服务后生效）。

要设置 DAS 的通信协议，参考适合您平台的《快速入门》。

### DAS 的安全性注意事项

必须首先使用具有本地“管理员”权限的帐户或用户标识登录到机器上。

**注：**在 Windows NT 上，不应使用控制面板中的服务实用程序来更改 DAS 的登录帐户，因为不会为该登录帐户设置某些必需的访问权。始终使用 **db2admin** 命令来设置或更改“DB2 管理服务器”(DAS) 的登录帐户。

当创建 DAS 后，可使用 **db2admin** 命令设置或更改登录帐户：

```
db2admin setid username password
```

其中, *username* 和 *password* 是具有本地“管理员”权限的帐户的用户名和密码。

建议在该环境内的每个服务器上该用户标识或用户名都具有 SYSADM 权限, 以便在必要时它可以启动或停止其他实例。

### 更新 DAS

在 UNIX 操作系统上, 若通过安装“程序临时修订”(PTF) 或代码补丁来更新 DB2, 则应更新所有“DB2 管理服务器”(DAS) 以及所有现存实例。要更新 DAS, 使用 **dasiupdt** 命令, 可在已安装的 DB2 版本和发行版所在的特定子目录下的 instance 子目录中找到该命令。

必须首先作为“root 用户”(在 UNIX 上)、使用具有本地管理权限的帐户或用户标识登录到机器上。

按如下所示使用该命令:

```
dasiupdt InstName
```

InstName 是实例所有者的登录名。此命令也有一些可选参数, 可设置在 InstName 前, 中间用空格分开:

- -h 或 -?  
显示此命令的帮助菜单。
- -d  
设置调试方式, 用于分析问题。

### 除去 DAS

必须首先作为“root 用户”(在 UNIX 上)、使用具有本地管理权限的帐户或用户标识登录到机器上。

要除去 DAS:

- 在 OS/2 或 Windows NT 操作系统上:
  1. 使用 db2admin stop 来停止 DAS。
  2. 备份(若需要的话) sql1lib 子目录下 db2das00 子目录中的所有文件。由 DB2INSTPROF 注册表变量指示此实例目录。

**注:** 本示例假设 db2das00 是要除去的 DAS 的名称。

3. 使用 db2admin drop 来删除 DAS。

**注:** 在 Windows NT 下, 使用此命令的人必须具有 SYSADM、SYSCTRL 或 SYSMMAINT 权限。

- 在 UNIX 操作系统上:

1. 作为 DAS 所有者登录。
2. 使用下列其中一个命令运行启动脚本:

```
. INSTHOME/sql1lib/db2profile    (对于 Bourne 或 Korn shell)
source INSTHOME/sql1lib/db2cshrc (对于 C shell)
```

其中, INSTHOME是该实例的主目录。

3. 使用 **db2admin** 命令停止 DAS, 如下所示:

```
db2admin stop
```

4. 备份(若需要的话) DAS 主目录下 sql1lib 子目录中的所有文件。此实例目录由 *DB2INSTPROF* 注册表变量指示。
5. 注销。
6. 作为 root 用户登录, 并使用 **dasidrop** 命令除去 DAS, 如下所示:

```
dasidrop ASName
```

其中, ASName 是“管理服务器”的实例名。可在已安装的 DB2 版本和发行版所在的特定子目录下的 instance 子目录中找到此命令。

**注:** **dasidrop** 命令除去“DB2 管理服务器”(DAS) 主目录下的 sql1lib 目录。

### 在 EEE 系统中设置 DAS

下列信息显示使用“控制中心”(CC) 配置 DB2 EEE 服务器 (Solaris、NT、Sequent、HP-UX 和 AIX) 以便远程管理的必要步骤。

在安装期间, 设置程序会在拥有实例的机器上创建一个 DAS。您可能想要在其他机器上创建附加的 DAS, 来允许“控制中心”或“客户机配置辅助程序”访问其他协调程序节点。然后可将协调程序节点的工作负荷分布到一个实例中的多个节点上。

要分布协调程序功能:

1. 在该分区数据库系统所选的附加机器上创建一个新的 DAS。
2. 在“控制中心”或“客户机配置辅助程序”中将每个 DAS 编目为单个系统。
3. 在每个新的系统下编目相同的实例, 每次指定相同的机器名来编目该 DAS。

配置有两个方面: “DB2 管理服务器”(DAS) 所必需的配置, 以及建议受管理的目标 DB2 实例使用的配置。在下面的三节中, 有一节专门阐述这两个配置主题。在每个配置主题之前, 都有一段描述假设的环境。

**环境示例:**

产品 / 版本:

DB2 UDB EEE V7.1

安装路径:

install\_path

**TCP services** 文件:

tcp\_services\_file

**DB2** 实例:

名称: db2inst

所有者标识:

db2inst

实例路径:

instance\_path

节点: 3 个节点, db2nodes.cfg:

- 0 hostA 0 hostA0switch
- 1 hostA 1 hostA1switch
- 2 hostB 0 hostBswitch

**DB** 名称:

db2instDB

**DAS:**

名称: db2as

所有者 / 用户标识:

db2as

实例路径:

das\_path

安装 / 运行主机:

hostA

节点间通信端口:

16000 (hostA 和 hostB 的未使用端口)

**注:** 请用具体值替换上面的字段。例如, 下表包含每个受支持的 EEE 平台的路径名示例:

表 1. 受支持的 EEE 平台的示例路径名

路径	DB2 UDB EEE AIX 版	DB2 UDB EEE Solaris 版	DB2 UDB EEE Windows NT 版
<i>install_path</i>	/usr/lpp/<v_r_ID>	/opt/IBMdb2/<v_r_ID>	C:\sqllib
<i>instance_path</i>	/home/db2inst/sqllib	/home/db2inst/sqllib	C:\profiles\db2inst
<i>das_path</i>	/home/db2as/sqllib	/home/db2as/sqllib	C:\profiles\db2as
<i>tcp_services_file</i>	/etc/services	/etc/services	C:\winnt\system32\drivers\etc\services

在表中，<v\_r\_ID> 是特定于平台的版本和发行版标识符。例如，在 DB2 UDB EEE AIX 版的版本 5.2 中，<v\_r\_ID> 是 db2\_05\_00。

当安装 DB2 UDB EEE 时，设置程序会在拥有实例的机器上创建一个 DAS。数据库分区服务器驻留在 DAS 所在的机器上，并是该实例的连接点。即此数据库分区服务器是从“控制中心”或“客户机配置辅助程序”发出给该实例的请求的协调程序节点。

**DAS 配置:** DAS 是一个管理控制点，它代表“控制中心”执行特定的任务。每台物理机器最多只能有一个 DAS。对于由几台机器组成的一个 EEE 实例，至少有一台机器必须运行 DAS，以便“控制中心”可以管理该 EEE 实例。此 DAS (db2as) “表示”在“控制中心”导航器树中作为目标 DB2 实例 (db2inst) 的父级的系统。

例如，db2inst 由分布在两个物理机器或主机上的三个节点组成。通过在 hostA 或 hostB 上运行 db2das，可以满足最低需求。

**注:**

1. 在 hostA 上存在的分区数对可在该主机上运行的 DAS 数没有任何影响。不管 hostA 的多个逻辑节点 (MLN) 配置如何，您只能在该主机上运行 db2as 的一个副本。
2. 不必在所有主机上创建 DAS 标识 (即 db2as)。它只需要存在于它运行所在的主机上。同样不需要将 DAS 标识的主目录安装在所有主机上。尤其是在此示例中，标识 db2as 必须存在于 hostA 上，不必存在于 hostB 上，且 db2as 的主目录不必安装在 hostB 上。

**“控制中心”与 DAS 的通信: 服务端口:** “控制中心”使用 TCP 服务端口 523 来与 DAS 通信。因为此端口是专门为 DB2 UDB 使用保留的，因此不需要向 *tcp\_services\_file* 插入新的项目。

**节点间管理通信: 服务端口:** 对于一些管理任务, DAS 必须建立与所有节点的通信。为此, 必须在 `tcp_services_file` 中为参与此实例的每个主机定义一个命名的 TCP 端口。

**注:** Windows NT EEE 将试图把 TCP 端口项添加至 `tcp_services_file` 中。

例如, 定义的 `db2inst` 包括两个主机: `hostA` 和 `hostB`。正如第75页的『环境示例』中所指定, 在两个主机上不使用端口 16000。因此, 必须将如下行插入到 `hostA` 和 `hostB` 的 `tcp_services_file` 中。

```
db2ccmsrv 16000/tcp
```

`db2ccmsrv` 端口名必须存在, 且必须完全按上面显示的样子拼写, 而且在所有主机上必须使用同一个所选端口号。

**节点间管理通信: UNIX DB2 EEE 服务器:** 一旦把 TCP 端口行插入 `hostA` 和 `hostB` 上的 `tcp_services_file` 后, 需要在参与该实例的所有主机上启动一个管理侦听器进程或守护程序 `db2cclst`。可从命令行手工调用, 或者可配置系统以便在每次系统引导时自动调用 `db2cclst`:

**手工:** 使用您希望管理的实例的标识 (即 `db2inst`), 从 `hostA` 或 `hostB` 调用如下命令:

```
rah '<install_path>/bin/db2cclst'
```

例如, 在 AIX 上, 此命令的调用格式将是:

```
rah '/usr/lpp/<v_r_ID>/bin/db2cclst'
```

可在版本和发行版子目录中的 `instance` 子目录中找到 `rah` 命令。版本和发行版子目录的精确名称随操作系统的不同而有所变化。 `instance` 是您希望使用的实例的主目录。

在此情况下, `<v_r_ID>` 是特定于平台的版本和发行版标识符。例如, 在 DB2 UDB EEE AIX 版的版本 5.2 中, `<v_r_ID>` 是 `db2_05_00`。

**自动:** 作为 root 用户, 将适当的条目添加至

```
/etc/inittab
```

文件。例如, 在 AIX 上, 此命令调用应在 主机 A 和主机 B 上运行:

```
mkitab "db2cclst::once:su - db2inst -c  
/usr/lpp/<v_r_ID>/bin/db2cclst"
```

每次机器引导时, 都调用 `db2cclst` 而无需用户干预。

在表中，<v\_r\_ID> 是特定于平台的版本和发行版标识符。例如，在 DB2 UDB EEE AIX 版的版本 5.2 中，<v\_r\_ID> 是 db2\_05\_00。

要验证侦听器的守护程序在每个主机上是活动的，可从该实例标识 db2inst 调用如下命令：

```
rah 'ps -ef | grep db2cclst'
```

若没有找到在每台主机上运行的 db2cclst 进程，可把如下行添加到每台主机上的 /etc/syslog.conf 中，以获得附加的诊断信息：

```
*.info /tmp/db2/user.info
```

其中文件 /tmp/db2/user.info 可由一个更恰当的文件来替换。

**注：**该文件必须存在，且在做了更改后必须请 SYSLOG 守护程序重读其配置文件：

```
kill -1 <syslogd PID>
```

其中 syslogd PID 可通过执行下列命令得到：

```
ps -ef | grep syslogd
```

然后，如以上所述在手工调用侦听器之后，可以查看失败的主机上的 syslog 文件 /tmp/db2/user.info，以获得由 db2cclst 生成的出错信息。

**节点间管理通信：** *Windows NT DB2 EEE 服务器：* “DB2 远程命令服务” (db2rcmd.exe) 自动处理节点间管理通信。若发生故障，则 Windows NT 注册表将包含诊断信息。

**安全性：** 为使 DAS 对实例执行某些管理任务，它必须拥有足够的权限。尤其是，DAS 必须是受管理的目标实例的“系统管理员” (SYSADM)。

对于 DAS 要管理的所有 DB2 实例，需要授予 DAS 这种权限。候选实例是那些与 DAS 安装在同一台机器上的实例。对于 DB2 EEE 实例，至少有一个数据库分区服务器必须与 DAS 存在于同一台机器上，这样它才可成为上述的人选实例。

例如，在 UNIX 上，可以授予 db2as 必需的权限来管理 db2inst 的一种方法是，确保 db2inst 和 db2as 的主组相同。另一种方法也足以达到该目的，使 db2inst 的主组成为 db2as 次组，而使 db2as 的主组成为 db2inst 的次组。最后，另一个选项是把 db2inst 的 SYSADM\_GROUP 数据库管理配置参数设置为 db2as 的主组。

在 Windows NT 上，db2as 必须是 hostA 和 hostB 上的“本地管理员”组的一个成员。除了创建 db2as 标识并将它添加到两个主机上的“本地管理员”组的选项之外，您还可以为 db2as 创建一个域标识，并将此域标识添加到每台主机上的“本地管理员”组中。

**环境：** 安装 DAS 时应配置完成适当操作所需要的特定注册表变量。要验证这些变量的当前值，使用 DB2 实例标识 db2inst 或 DAS 标识 db2das 执行如下命令：

```
db2set -g
```

至少必须将下列参数定义为下列值：

```
DB2SYSTEM=hostA  
DB2ADMINSERVER=db2as
```

而且，为了从“控制中心”与 DAS 通信，还须确保将 DB2COMM 注册表变量设置为“TCPIP”。要验证此设置，使用 DAS 标识 db2as 执行如下命令，并在全局级 (-g) 和实例级 (-i) 进行检查（只需设置一个）：

```
db2set -a11
```

在同样这些行上，验证已将 DB2 实例的 DB2COMM 参数设置为“TCPIP”，以启用在“控制中心”和 db2inst 之间的通信，验证方法是用 db2inst 标识发出如下命令：

```
db2set -a11
```

若修改 DAS 的这个参数，必须重新启动 DAS，才可使该更改生效。若修改了 DB2 实例的这个参数，也需要重新启动 DB2 实例。对于 db2inst，要发出 *db2stop*，接着发出 *db2start*，而对于 DAS，则应发出 *db2admin stop* 和 *db2admin start*。

**查找管理服务器、实例和数据库：** Known Discovery 允许在系统上查找客户机知道的实例和数据库，并添加新系统，以便可找到它们的实例和数据库。Search Discovery 提供 Known Discovery 的所有设施，并添加选项以允许搜索本地网络找到其他 DB2 服务器。

要让服务器支持 Known Discovery，在 DAS 配置文件中将 *discover* 参数设置为 KNOWN。要让它支持 Search Discovery，将此参数设置为 SEARCH。要阻止查找某个服务器及其所有实例和数据库，将此参数设置为 DISABLE。

**注：** 由 Search Discovery 返回给客户机的 TCP/IP 主机名与输入 **hostname** 命令时由 DB2 服务器系统返回的主机名相同。在该客户机上，此主机名所映射成的 IP 地址由客户机上配置的 TCP/IP 域名服务器 (DNS)（若尚未配置 DNS）或客户机的 hosts 文件中的一个映射项确定。若在 DB2 服务器系统上配置了多个适配卡，必须确保在服务器上配置 TCP/IP 以返回正确的主机名，还必须确保 DNS 或本地客户机的 hosts 文件将主机名映射成期望的 IP 地址。



在客户机上，也可使用 *discover* 参数启用 Discovery；但在这种情况下，要在客户机实例（或充当客户机的服务器）中设置 *discover* 参数，如下所示：

- **KNOWN**

允许“客户机配置辅助程序”刷新已知列表中的系统，并可使用**添加系统**按钮将新系统添加到该列表中。当将 *discover* 参数设置为 KNOWN 时，“客户机配置辅助程序”将无法搜索网络。

- **SEARCH**

启用 Known Discovery 的所有设施，并启用网络搜索。

仅当选择了此选项时，“其他系统（搜索网络）”图标才会出现。这是缺省设置。

- **DISABLE**

禁用 Discovery。在这种情况下，在“添加数据库向导”中没有**搜索网络**选项。

**注：***discover* 参数在所有客户机和服务器实例上缺省为 SEARCH。*discover* 参数在所有“DB2 管理服务器”（DAS）上缺省为 SEARCH，但在“UNIX 扩充企业版”环境中安装的 DAS 除外，在这种环境中，*discover* 缺省为 KNOWN。

**Search Discovery 的附加设置：** Search Discovery 要求在服务器（在“DB2 管理服务器”的配置文件中）和客户机（在数据库管理器配置文件中）上设置 *discover\_comm* 参数。

*discover\_comm* 参数用于控制服务器在监听来自客户机的搜索请求时以及客户机发送出搜索请求时所用的通信协议。可将 *discover\_comm* 参数设置为 TCP/IP 或 NetBIOS。当前只支持这些协议。

在 DAS 上，为 *discover\_comm* 指定的值必须等于为 DB2COMM 设置的值或作为其子集。

**注：**要避免“控制中心”和“客户机配置辅助程序”出现问题，确保在 DB2 注册表中**使用 db2set 命令设置 DB2COMM 注册表变量**。不建议使用任何其他方法设置 DB2COMM 注册表变量。

在服务器上，在 DAS 配置文件中设置 *discover\_comm* 参数。在客户机（或充当客户机的服务器）上，在数据库管理器配置文件中设置 *discover\_comm*。

**注：**当使用 Search Discovery 时，在客户机上由 *discover\_comm* 参数指定的至少一个协议必须与 DAS 上由 *discover\_comm* 参数指定的那些协议匹配。若没有匹配的协议，服务器不会应答客户机的请求。

要检查 DB2COMM 注册表变量的设置，输入：

```
db2set db2comm
```

此外，有两个 DB2 简要表注册表变量可用来在客户机上通过 NetBIOS 调整 Search Discovery: DB2DISCOVERYTIME 和 DB2NBDISCOVERYRECVBUFS。在大多数情况下，这些注册表变量的缺省值应适于使用。

在客户机实例（或充当客户机的服务器）上设置 DB2DISCOVERYTIME 和 DB2NBDISCOVERRCVBUFS 简要表注册表变量。按如下所示设置这两个注册表变量：

- 要将 DB2DISCOVERYTIME 注册表值设置为 60 秒，输入以下命令：

```
db2set db2discoverytime=60
```

它指定 Search Discovery 应等待 60 秒来获得服务器的响应。

- 要将 DB2NBDISCOVERRCVBUFS 注册表值设置为 20，输入：

```
db2set db2nbdiscoverrcvbufs=20
```

它指定将为当前响应信息分配的 NetBIOS 缓冲区的数目，这些信息来自找到的服务器。

**隐藏服务器实例和数据库以免找到：** 可能在一个服务器上有多个实例，而在这些实例中有多个数据库。您可能想隐藏其中某些实例，以免 Discovery 进程找到。

要允许客户机发现一个系统上的服务器实例，在该系统上的每个服务器实例中将 *discover\_inst* 数据库管理器配置参数设置为 ENABLE（这是缺省值）。将此参数设置为 DISABLE，可隐藏此实例及其数据库以免被 Discovery 发现。

要允许从客户机查找一个数据库，将 *discover\_db* 数据库配置参数设置为 ENABLE（这是缺省值）。将此参数设置为 DISABLE，可隐藏该数据库以免被发现。

**设置 Discovery 参数：** 在服务器系统上的 DAS 配置文件中以及客户机上的数据库管理器配置文件中设置 *discover* 和 *discover\_comm* 参数。按如下所示设置这两个参数：

- 在 DAS 上：

使用以下命令进程更新 DAS 配置文件：

```
update admin cfg using discover [ DISABLE | KNOWN |  
SEARCH ]  
update admin cfg using discover_comm [ NETBIOS | TCPIP ]
```

使用以下命令停止并重新启动 DAS：

```
db2admin stop  
db2admin start
```

注: Search Discovery 将只基于 NetBIOS 和 TCP/IP 运行。

- 使用“控制中心”:

1. 启动“客户机配置辅助程序”。
2. 单击**客户机设置**按钮。
3. 选择**通信**选项卡。
4. 从**参数**窗口选择要修改的参数。
5. 从**值框**中为要修改的参数选择一个值。
6. 单击**确定**按钮以关闭**客户机设置**窗口。一个 DB2 消息窗口打开。
7. 单击**确定**按钮, 重新启动应用程序以使更改生效。

注: 若 *discover\_comm* 包括 NETBIOS, 必须确保为客户机和 DAS 设置“工作站名”(*nname*) 参数。还必须确保将 DB2NBADAPTERS 注册表变量设置为要使用的“适配器号”。

使用“控制中心”设置 *discover\_inst* 和 *discover\_db* 参数:

1. 展开对象树, 直到您找到**实例**文件夹为止。
2. 右键单击实例, 并从弹出菜单中选择**配置**。
3. 在“环境”页上, 选择 *discover\_inst* 参数。
4. 为了能够从客户机发现服务器实例, 选择**启用**并单击**确定**。
5. 右键单击对象树中的数据库, 并从弹出菜单中选择**配置**。
6. 在“环境”页上, 选择 *discover\_db* 参数。
7. 为了能够从客户机发现数据库, 选择**启用**并单击**确定**。

### 设置 DAS 以使用“客户机配置辅助程序”和“控制中心”

必须配置 DB2 Discovery 来检索网络上有关系统的信息。DB2 Discovery 是“客户机配置辅助程序”和“控制中心”使用的一个功能部件。配置此功能部件时, 可能要求更新实例列表和“DB2 管理服务器”(DAS) 配置, 以确保 DB2 Discovery 检索正确的信息。

**更新实例列表:** 一个“DB2 管理服务器”(DAS) 可能无法认识一个分区数据库系统中的所有实例, 因为最初创建一个实例时只有该拥有实例的机器上的 DAS 认识该实例。

若在没有 DAS 的机器上创建了一个实例, 则可在该机器上创建一个 DAS 使该实例成为已知的。

若创建了多个 DAS，且希望每个 DAS 认识数据库分区系统中的所有实例，执行下列步骤：

### 1. 对于每个 DAS

在“管理服务器”机器上运行 **db2ilist** 命令，以显示此 DAS 所认识的实例的列表。

注：若实例列表是完整的，则不需要执行其余步骤，但可继续下一节。

### 2. 对于在前一步骤中从实例列表中丢失的每个实例

在该拥有实例的机器上，运行 **db2nlist** 命令以查看是否存在一项表示含有该 DAS 的机器。若不存在，必须运行 **db2ncrt** 命令以将此机器添加到该实例中。

注：该实例的网络共享驱动器必须在 DAS 机器上可用。

## 更新 DAS 配置

缺省情况下，设置程序将 DB2SYSTEM 注册表变量设置为 Windows NT 计算机名称。由 Discovery 检索到的系统名是“DB2 管理服务器” (DAS) 所驻留的系统。当建立连接时，Discovery 将这些系统用作协调程序节点。

更新 DAS 配置有两个方法：

- 若想从 DB2 系统列表选择一个协调程序节点，在每个“DB2 管理服务器”的配置文件中设置 DISCOVER=SEARCH（它是缺省值）。

当存在多个 DAS 时，同一个实例可能出现在“客户机配置辅助程序”或“控制中心”界面上的多个系统中；但每个系统将有不同的通信访问路径来访问实例。用户可选择不同的 DB2 系统作为通信的协调程序节点，因此可重新分布工作负荷。

- 若不希望用户选择协调程序节点，则在所有 DAS 上设置 DISCOVER=KNOWN，只在一个 DAS 上的 DAS 配置中设置 DISCOVER=SEARCH。Discovery 使用数据库分区服务器，当建立连接时 DAS 作为协调程序节点驻留在其中。

## 创建节点配置文件

若数据库要在分区数据库环境中运行，必须创建一个称为 db2nodes.cfg 的节点配置文件。在可以使用多个分区的并行能力来启动数据库管理器之前，此文件必须位于该实例的主目录 sqllib 子目录中。该文件包含一个实例中所有数据库分区的配置信息，并且它由该实例的所有数据库分区共享。

**Windows NT 注意事项：**若是在 Windows NT 上使用 DB2 扩充企业版，则节点配置文件是在创建实例时创建的。您不应尝试手工修改节点配置文件。

**注:** 不应该在不是 DB2 创建的 `sqllib` 子目录下创建文件或目录, 以防止在删除实例时数据丢失。但有两个例外情况。若系统支持存储过程, 则将该存储过程应用程序置于 `sqllib` 子目录下的 `function` 子目录中。(有关存储过程的信息, 参考《管理指南: 性能》中的“存储过程”。) 另一个例外情况是已创建了用户定义的相异函数 (UDF) 时。允许 UDF 可执行程序位于同一个目录中。

对于属于一个实例的每个数据库分区该文件都包含一行。每行的格式如下:

```
nodenum hostname [logical-port [netname]]
```

记号由空格定界。这些变量是:

### **nodenum**

该节点号可以在 0 至 999 之间, 它唯一地定义一个节点。节点号必须以升序排列。该顺序中可以有间隙。

一旦指定了节点号, 就不能更改它。(否则, 指定将数据如何分区的分区映象中的信息将会被泄露。)

若删除了一个节点, 则它的节点号可以再次用于添加的任何新节点。

节点号用于在数据库目录中生成节点名。它的格式为:

```
NODEnnnn
```

*nnnn* 是节点号, 不足四位的, 其左边以零填充。CREATE DATABASE 和 DROP DATABASE 命令也使用此节点号。

### **hostname**

用作分区间通信的 IP 地址的主机名。(当指定了网络名时, 存在一个例外情况。在此情况下, 网络名用于大多数通信, 而主机名只用于 DB2START、DB2STOP 和 db2\_all。)

### **logical-port**

此参数是可选的, 它指定该节点的逻辑端口号。此号码与数据库管理器实例名一起用来标识 `etc/services` 文件中的 TCP/IP 服务名项。

IP 地址和逻辑端口的组合作为公认地址, 且在所有支持节点间通信连接的应用程序中必须是唯一的。

对于每个主机名, 一个逻辑端口必须为 0 (零) 或空白 (缺省为 0)。与此逻辑端口相关的节点是与客户机连接的主机上的缺省节点。可以使用 `db2profile` 脚本中的 `DB2NODE` 环境变量或 `sqlsetc()` API 来覆盖它。

若在同一主机上有多个节点 (即, 对于一个主机, 有多个 *nodenum*), 则应该以从 0 开始且无间隙的升序为逻辑节点指定 *logical-port* 号。

## netname

此参数是可选的，且用于支持有多个活动的 TCP/IP 接口的主机，每个接口有它自己的主机名。

下列示例显示一个 RS/6000 SP 系统的一个可能的节点配置文件，在该系统上，SP2EN1 有多个 TCP/IP 接口、两个逻辑节点且使用 SP2SW1 作为 DB2 通用数据库接口。它还显示了由 1 开始（而不是由 0 开始）的节点号，以及节点号序列中的一个间隙：

<i>nodenum</i>	<i>hostname</i>	<i>logical-port</i>	<i>netname</i>
1	SP2EN1	0	SP2SW1
2	SP2EN1	1	SP2SW1
4	SP2EN2	0	
5	SP2EN3		

可以使用选择的编辑器更新 `db2nodes.cfg` 文件。（例外是：不应在 Windows NT 上使用编辑器。）然而，必须小心保护文件中信息的完整性，因为数据分区要求不更改节点号。该节点配置文件在您发出 `DB2START` 时被锁定，而在 `DB2STOP` 结束数据库管理器时解锁。当文件被锁定时，若有必要的话，`DB2START` 命令可以更新该文件。例如，可以发出带有 `RESTART` 选项或 `ADDNODE` 选项的 `DB2START`。

**注：**若 `DB2STOP` 命令不成功而又未解锁该节点配置文件，则发出 `DB2STOP FORCE` 来将它解锁。

## 创建数据库配置文件

还要为每个数据库创建数据库配置文件。此文件的创建已完成。此文件包含影响数据库使用的各种配置参数的值，如：

- 当创建数据库时指定的和 / 或使用的参数（例如，数据库代码页、整理顺序、DB2 发行版级别）
- 指示数据库当前状态的参数（例如，备份暂挂标志、数据库一致性标志、前滚暂挂标志）
- 定义数据库操作可以使用的系统资源的数量的参数（例如，缓冲池大小、数据库记录、排序内存大小）。

这些参数在《管理指南：性能》中的“配置 DB2”中有详细的描述。

不应手工更改配置文件中的参数。只应使用受支持的接口。

**性能提示：**许多配置参数都带有缺省值，但是为达到数据库的最优性能，可能需要更新它们。

对于多分区： 当有一个分布在多个分区上的数据库时，该配置文件在所有数据库分区上应是相同的。一致性是必需的，因为 SQL 编译程序根据本地节点配置文件中的信息来编译分布的 SQL 语句，并创建一个访问计划以满足 SQL 语句的需要。维护数据库分区上的不同配置文件可能产生不同的访问计划，这取决于准备该语句所在的数据库分区。使用 **db2\_all** 来保持所有数据库分区上的配置文件同步。

## 使用响应文件复制配置信息

可以使用一个响应文件生成器实用程序 *db2rspgn* 来创建响应文件，当重新安装系统时，或当希望将当前系统的注册表变量、数据库管理器配置参数和数据库管理配置参数复制到相同的系统中时，可使用该响应文件。

当在一个系统中安装了一个或多个 DB2 产品后，且在调整该环境的参数后，可使用 *db2rspgn* 在响应文件中生成必需的值。然后可使用该响应文件重新创建完全相同的系统。

命令行语法说明响应文件和任何支持的文件的目的地目录。此外，可以选择指定要复制的实例；而且，可以选择禁用管理实例和 / 或 DataLinks 服务器实例。有关大量部署问题的更多信息，参见 《卫星管理指南和参考》。

参考恰当的《快速入门》，以了解此实用程序的语法详细信息和有关如何使用生成的响应文件的讨论。

## 启用 FCM 通信

在分区数据库环境中，数据库分区之间的大多数通信都是由“快速通信管理器”(FCM) 来处理。要在一个数据库分区上启用 FCM 并允许与其他数据库分区通信，必须在 `etc` 目录的 `services` 文件创建一个服务项，如下所示。FCM 使用指定的端口来通信。若已在同一个主机上定义了多个分区，则必须定义一个端口范围，如下所示。

### Windows NT 注意事项

若在 Windows NT 环境中使用 DB2 扩充企业版，则会通过下列程序自动把 TCP/IP 端口范围添加至 `services` 文件：

- 创建实例或添加新节点时的安装程序
- 创建新实例时的 `db2icrt` 实用程序
- 在机器上添加第一个节点时的 `db2ncrt` 实用程序

有关其他信息，参考 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings*。

服务项的语法如下所示：

```
DB2_instance port/tcp #comment
```

### **DB2\_instance**

*instance* 的值是数据库管理器实例的名称。该名称的所有字符必须为小写。假定实例名为 `db2puser`，则应指定 `DB2_db2puser`。

### **port/tcp**

要为该数据库分区保留的 TCP/IP 端口。

### **#comment**

想要与该项目关联的任何注释。注释之前必须加 `#` 符号。

若 `/etc/services` 文件是共享的，必须确保在该文件中分配的端口的数量大于或等于该实例中多个数据库分区的最大数量。当分配端口时，还要确保考虑了可以用作备份的任何处理器。

若 `/etc/services` 文件不是共享的，则注意事项基本相同，但有一个附加注意事项：必须确保为 DB2 实例定义的项在所有 `/etc/services` 文件中都是相同的（而不适用于分区数据库的其他项不必相同）。

若在一个实例中的相同主机上有多个数据库分区，则必须定义多个端口以供 FCM 使用。为此，在 `etc/services` 文件中包括两行，以指示要分配的端口的范围。第一行指定第一个端口，而第二行指示端口块的结束。在下列示例中，为实例 `sales` 分配了五个端口。这意味着该实例中不会有处理器具有多于五个的数据库分区。

```
DB2_sales      9000/tcp
DB2_sales_END  9004/tcp
```

**注：**只能用大写字母来指定 `END`。还必须确保包括了两个下划线 (`_`) 字符。



---

## 第3章 创建数据库

本章包含各种对象的简要讨论，那些对象可能会成为您的数据库设计实现的一部分。

前一章主要讨论您在创建数据库之前需要了解的信息。该章还包括必须在创建数据库之前执行的几个主题和任务。

此部分的最后一章讨论您在改变数据库之前必须要考虑的事项。此外，该章还说明了如何改变或删除数据库对象。

创建数据库时，为您完成了下列所有任务：

- 设置数据库所需的所有系统目录表
- 分配数据库恢复日志
- 创建数据库配置文件，设置缺省值
- 将数据库实用程序与数据库绑定

下列数据库特权被自动授予 `PUBLIC`：对系统目录视图的 `CREATETAB`、`BINDADD`、`CONNECT`、`IMPLICIT_SCHEMA` 和 `SELECT` 特权。

要使用“控制中心”创建数据库：

1. 展开对象树，直到您找到**数据库**文件夹为止。
2. 右键单击**数据库**文件夹，并从弹出菜单中选择**创建** → **使用向导创建数据库**。
3. 遵循步骤来完成此任务。

下列命令行处理器命令在缺省位置创建称为 `person1` 的一个数据库，并带有相关注解：“Personnel DB for BSchiefer Co”。

```
create database person1
with "Personnel DB for BSchiefer Co"
```

若希望在另一个可能是远程的数据库管理器实例中创建数据库，参见第47页的『使用数据库管理器的多个实例』。此主题还介绍若要对非缺省实例的实例（包括远程实例）执行任何实例级管理需要使用的命令。

**注：**有关缺省数据库位置和使用 `CREATE DATABASE` 命令指定不同位置的信息，参考 *Command Reference*。

下面几节讨论创建数据库时您执行的任务或数据库管理器为您执行的任务:

- 『初始节点组的定义』
- 第91页的『初始表空间的定义』
- 第92页的『系统目录表的定义』
- 第93页的『数据库目录的定义』
- 第94页的『DCE 目录服务』
- 第95页的『“轻量级目录访问协议”(LDAP) 的目录服务』
- 第96页的『数据库恢复日志的定义』
- 第96页的『将实用程序与数据库绑定』
- 第97页的『编目数据库』
- 第95页的『创建节点组』
- 第97页的『创建表空间』
- 第103页的『创建模式』
- 第104页的『创建和填充表』
- 第120页的『创建触发器』
- 第122页的『创建用户定义函数 (UDF) 或方法』
- 第125页的『创建用户定义类型 (UDT)』
- 第127页的『创建视图』
- 第129页的『创建摘要表』
- 第131页的『创建别名』
- 第132页的『创建包装器』
- 第133页的『创建服务器』
- 第139页的『创建别名』
- 第141页的『创建索引、索引扩充或索引规范』

有关数据库的物理实现的其他信息, 参考《管理指南: 计划》。

---

## 初始节点组的定义

当最初创建数据库时, 会为所有在 `db2nodes.cfg` 文件中指定的分区创建数据库分区。其他分区可以使用 `ADD NODE` 和 `DROP NODE` 命令来添加或删除。

定义了三个节点组:

- 用于容纳 `SYSCATSPACE` 表空间的 `IBMCATGROUP`, 它保存系统目录表

- 用于容纳 TEMPSPACE1 表空间的 IBMTEMPGROUP，它保存数据库处理期间创建的临时表
- 用于容纳 USERSPACE1 表空间的 IBMDEFAULTGROUP，缺省情况下它保存用户表和索引。

---

## 初始表空间的定义

当创建一个数据库时，要定义三个表空间：

- 用于保存系统目录表的 SYSCATSPACE（参见第92页的『系统目录表的定义』）
- 用于保存数据库处理期间创建的系统临时表的 TEMPSPACE1
- 用于保存用户定义的表和索引的 USERSPACE1

**注：**当第一次创建一个数据库时，不创建用户临时表空间。

若未使用 CREATE DATABASE 命令指定任何表空间参数，则数据库管理器使用系统管理的存储器 (SMS) 目录容器创建这些表空间。这些目录容器将在为该数据库创建的子目录中创建（有关数据库物理目录的更多信息，参考《管理指南：计划》）。这些表空间的数据块大小被设置为缺省值。

要使用“控制中心”来定义初始表空间：

1. 展开对象树，直到您看到**数据库**文件夹为止。
2. 右键单击**数据库**文件夹，并从弹出菜单中选择**创建** → **使用向导创建数据库**。
3. 遵循步骤来完成此任务。

要使用命令行来定义初始表空间，输入：

```
CREATE DATABASE <name>
  CATALOG TABLESPACE
    MANAGED BY SYSTEM USING ('<path>')
    EXTENTSIZE <value> PREFETCHSIZE <value>
  USER TABLESPACE
    MANAGED BY DATABASE USING (FILE'<path>' 5000,
                                FILE'<path>' 5000)
    EXTENTSIZE <value> PREFETCHSIZE <value>
  TEMPORARY TABLESPACE
    MANAGED BY SYSTEM USING ('<path>')
  WITH "<comment>"
```

若不希望使用这些表空间的缺省定义，可以在 CREATE DATABASE 命令上指定它们的特征。例如，可能使用下列命令在 OS/2 上创建数据库：

```

CREATE DATABASE PERSONL
  CATALOG TABLESPACE
    MANAGED BY SYSTEM USING ('d:\pcatalog','e:\pcatalog')
    EXTENTSIZE 16 PREFETCHSIZE 32
  USER TABLESPACE
    MANAGED BY DATABASE USING (FILE'd:\db2data\person1' 5000,
                                FILE'd:\db2data\person1' 5000)
    EXTENTSIZE 32 PREFETCHSIZE 64
  TEMPORARY TABLESPACE
    MANAGED BY SYSTEM USING ('f:\db2temp\person1')
  WITH "Personnel DB for BSchiefer Co"

```

在此示例中，每个初始表空间的定义是显式提供的。只需要为不希望使用缺省定义的那些表空间指定表空间定义。

在 CREATE DATABASE 命令上的 MANAGED BY 短语的编码与 CREATE TABLESPACE 命令上的 MANAGED BY 短语遵循相同的格式。有关其他示例，参见第97页的『创建表空间』。

在创建数据库之前，参考《管理指南：计划》手册和有关设计和选择表空间的资料。

---

## 系统目录表的定义

对于每个数据库，都会创建和维护一组系统目录表。这些表包含有关数据库对象（例如，表、视图、索引和软件包）的定义的信息，以及用户对这些对象所拥有的访问类型的安全性信息。这些表存储在 SYSCATSPACE 表空间中。

在一个数据库的操作期间，例如，创建一个表时，要更新这些表。不能显式地创建或删除这些表，但是可以查询和查看它们的内容。当创建该数据库时，除系统目录表对象外，在系统目录中还定义下列数据库对象：

- 在 SYSFUN 模式中创建一组用户定义函数 (UDF)。有关这些系统创建的函数的更多信息，参考 *SQL Reference*。
- 在 SYSCAT 模式中创建一组系统目录表的只读视图。有关这些视图的信息，参考 *SQL Reference* 中的 "Catalog Views"。
- 在 SYSSTAT 模式中创建一组可更新的目录视图。这些可更新的视图允许您更新特定的统计信息来调查假设数据库的性能，也可不使用 RUNSTATS 实用程序来更新统计信息。有关这些视图的信息，参考 *SQL Reference* 中的 "Updatable Catalog Views"。

在创建了数据库之后，您也许希望限制对系统目录视图的访问，如第236页的『保密系统目录视图』中所述。

---

## 数据库目录的定义

当建立或设置新数据库时，会使用三个目录。

- 本地数据库目录
- 系统数据库目录
- 节点目录

### 本地数据库目录

本地数据库目录文件存在于定义了数据库的每个路径（在一些操作系统上，称为“驱动器”）中。对于可从该位置访问的每个数据库此目录都包含一项。每一项包含：

- CREATE DATABASE 命令提供的数据库名
- 数据库别名（若未指定别名，它与数据库名相同）
- 描述该数据库的注解，是使用 CREATE DATABASE 命令提供的
- 该数据库的根目录的名称
- 其他系统信息。

要查看特定数据库的此文件的内容，发出下列命令，其中 *location* 指定该数据库的位置：

```
LIST DATABASE DIRECTORY ON location
```

### 系统数据库目录

对于数据库管理器的每个实例都存在一个系统数据库目录文件，该文件对于为此实例编目的每个数据库都包含一项。当发出 CREATE DATABASE 命令时将隐式地编目数据库，也可以使用 CATALOG DATABASE 命令显式地编目它。有关编目数据库的信息，参见第97页的『编目数据库』。

对于创建的每个数据库，都要将包含下列信息的一项添加至该目录：

- CREATE DATABASE 命令提供的数据库名
- 数据库别名（若未指定别名，它与数据库名相同）
- 使用 CREATE DATABASE 命令提供的数据库注解
- 本地数据库目录的位置
- 指示该数据库是间接的指示符，表示它与系统数据库目录文件驻留在相同的机器上
- 其他系统信息。

要查看此文件的内容，发出 LIST DATABASE DIRECTORY 命令，而不用指定该数据库目录文件的位置。

在分区数据库环境中，必须确保所有的数据库分区始终访问该实例主目录的 sqlbdir 子目录中的同一个系统数据库目录文件 sqlbdir。若在同一 sqlbdir 子目录中的系统数据库目录或系统意向文件 sqlbins 是通向共享文件系统中另一个文件的符号链路，可能发生不可预测的错误。第50页的『启用数据分区』中描述了这些文件。

## 节点目录

数据库管理器在编目第一个数据库分区时会创建节点目录。要编目数据库分区，使用 CATALOG NODE 命令。要列出本地节点目录的内容，使用 LIST NODE DIRECTORY 命令。在每个数据库客户机上创建并维护节点目录。对于具有客户机可以访问的一个或多个数据库的每个远程工作站，该目录都包含一项。无论何时请求数据库连接或实例连接，DB2 客户机都使用该节点目录中的通信端点信息。

该目录中的项还包含客户机与远程数据库分区通信要使用的通信协议的类型信息。编目一个本地数据库分区会为驻留在同一台机器上的实例创建一个别名。当用户的客户机要访问的同一个工作站上存在多个实例时，应该编目一个本地节点。

---

## DCE 目录服务

DCE 是一种“开放式系统基础\*\*” (OSF\*\*) 体系结构，它提供工具和服务以支持在分布式异类计算环境中创建、使用和维护应用程序。它是操作系统、网络和分布式应用程序之间的一层，允许客户机应用程序访问远程服务器。

使用本地目录时，目标数据库的物理位置单独存储在每个客户机工作站上的数据库目录和节点目录中。因此，数据库管理员可能要花费大量的时间来更新和更改这些目录。DCE 目录服务提供替代本地目录的中心目录。它允许将有关数据库或数据库管理器实例的信息一次性记录在中心目录中，且任何更改或更新都在该位置进行。

DCE 不是运行 DB2 的先决条件，但是若是在 DCE 环境中操作，可参见第283页的『附录B. 使用分布式计算环境 (DCE) 的目录服务』以获取更多信息。

---

## “轻量级目录访问协议” (LDAP) 的目录服务

“轻量级目录访问协议” (LDAP) 是一个业界标准的访问目录服务的方法。目录服务是一个关于分布式环境中的多个系统和资源的资源信息的储存库；它提供对这些资源的客户机和服务器访问。每个数据库服务器实例都将它的存在情况发布给 LDAP 服务器，并在创建数据库时向 LDAP 目录提供数据库信息。当客户机与数据库连接后，可从 LDAP 目录检索服务器的目录信息。不再要求每个客户机将目录信息以本地方式存储在每台机器上。客户机应用程序搜索 LDAP 目录，查找连接数据库所需的信息。

LDAP 不是运行 DB2 的先决条件，但是若在 LDAP 环境中操作，可参见第355页的『附录J. “轻量级目录访问协议” (LDAP) 的目录服务』以获取更多信息。

---

## 创建节点组

可使用 CREATE NODEGROUP 语句创建节点组。此语句指定表空间容器和表数据将驻留的一组节点。此语句还可以：

- 为节点组创建分区映象。有关分区映象的详细信息，参考《管理指南：计划》。
- 生成分区映象标识。
- 将记录插入下列目录表：
  - SYSCAT.NODEGROUPS
  - SYSCAT.PARTITIONMAPS
  - SYSCAT.NODEGROUPDEF

要使用“控制中心”创建节点组：

1. 展开对象树，直到您看到**节点组**文件夹为止。
2. 右键单击**节点组**文件夹，并从弹出菜单中选择**创建**。
3. 在“创建节点组”窗口上，填写信息，使用箭头来将节点从**可用节点**框移至**选择的节点**框，并单击**确定**。

要使用命令行来创建节点组，输入：

```
CREATE NODEGROUP <name> ON NODES (<value>,<value>)
```

假定要在数据库中一个数据库分区子集上装入一些表。应使用以下命令创建一个节点组，它包含至少由三个（0 至 2）节点组成的数据库中的两个节点（1 和 2）：

```
CREATE NODEGROUP mixng12 ON NODES (1,2)
```

有关创建节点组的更多信息，参考 *SQL Reference* 手册。

CREATE DATABASE 命令或 `sqlcrea()` API 也创建缺省系统节点组：IBMDEFAULTGROUP、IBMCATGROUP 和 IBMTEMPGROUP。（有关节点组的更多信息，参考《管理指南：计划》。）

---

## 数据库恢复日志的定义

数据库恢复日志保存对一个数据库所做的所有更改（包括新表的添加或对现存表的更新）的记录。此日志由大量日志块组成，每一个包含在称为日志文件的一个单独的文件中。

数据库恢复日志可以用于确保故障（例如，系统断电或应用程序出错）不会使数据库处于不一致的状态。若发生故障，则回滚已进行而未提交的更改，重新执行可能未实际写入磁盘的所有已提交事务。这些操作确保了数据库的完整性。

有关更多信息，参见 *Data Movement Utilities Guide and Reference*。

---

## 将实用程序与数据库绑定

当创建一个数据库时，数据库管理器试图将 `db2ubind.lst` 中的实用程序与该数据库绑定。此文件存储在 `sqllib` 目录的 `bnd` 子目录中。

绑定一个实用程序将创建一个软件包，软件包是这样对象，它包括处理单个源文件中特定 SQL 语句所需的所有信息。

**注：**若希望从客户机使用这些实用程序，必须显式地绑定它们。有关信息，参考适合您平台的快速入门手册。

若由于某种原因需要将实用程序与数据库绑定或重新绑定，则使用命令行处理器发出下列命令：

```
connect to sample
bind @db2ubind.lst
```

**注：**必须位于这些文件所驻留的目录中，才能在 `sample` 数据库中创建软件包。在 `SQLLIB` 目录的 `BND` 子目录中可找到绑定文件。在此示例中，`sample` 是该数据库的名称。



---

## 编目数据库

当创建一个新数据库时，会在系统数据库目录文件中自动将它编目。也可以使用 `CATALOG DATABASE` 命令在系统数据库目录文件中显式地编目数据库。`CATALOG DATABASE` 命令允许您使用不同的别名来编目数据库，或编目先前使用 `UNCATALOG DATABASE` 命令删除的数据库项。

下列命令行处理器命令将 `person1` 数据库编目为 `humanres`：

```
catalog database person1 as humanres
with "Human Resources Database"
```

此处，系统数据库目录项将使 `humanres` 作为数据库别名，以便与数据库名 (`person1`) 区分。

还可以在非缺省的实例上编目数据库。在下列示例中，至数据库 B 的连接也连接 `INSTANCE_C`。

```
catalog database b as b at node instance_c
```

**注：**在客户机节点上也使用 `CATALOG DATABASE` 命令，以编目驻留在数据库服务器上的数据库。有关更多信息，参考适合您平台的《快速入门》手册。

有关“分布式计算环境” (DCE) 单元目录的信息，参见第94页的『DCE 目录服务』和第283页的『附录B. 使用分布式计算环境 (DCE) 的目录服务』。

**注：**要改善性能，可在内存中高速缓存目录文件，包括数据库目录。（有关启用目录高速缓存的信息，参考《管理指南：性能》中的“目录高速缓存支持 (`dir_cache`)”。）当启用目录高速缓存时，另一个应用程序对目录所做的更改（例如，使用 `CATALOG DATABASE` 或 `UNCATALOG DATABASE` 命令）可能要在重新启动应用程序后才会生效。要刷新命令行处理器会话所用的目录高速缓存，发出 `db2 terminate` 命令。

除应用程序级高速缓存外，数据库管理器级高速缓存也用于内部的数据库管理器查找。要刷新此“共享的”高速缓存，发出 `db2stop` 和 `db2start` 命令。

有关目录高速缓存的更多信息，参考《管理指南：性能》中的“目录高速缓存支持 (`dir_cache`)”。

---

## 创建表空间

在一个数据库内创建表空间，会将容器分配到表空间，并在数据库系统目录中记录它的定义和属性。然后就可以在此表空间内创建表。

有关表空间的设计信息，参考《管理指南：计划》。

CREATE TABLESPACE 语句的语法在 *SQL Reference* 中进行了详细地讨论。有关 SMS 和 DMS 表空间的信息，参考 《管理指南：计划》。

要使用“控制中心”来创建表空间：

1. 展开对象树，直到您看到表空间文件夹为止。
2. 右键单击表空间文件夹，并从弹出菜单中选择创建 → 使用向导创建表空间。
3. 遵循向导中的步骤来完成该任务。

要使用命令行来创建 SMS 表空间，输入：

```
CREATE TABLESPACE <NAME>
  MANAGED BY SYSTEM
  USING ('<path>')
```

要使用命令行来创建 DMS 表空间，输入：

```
CREATE TABLESPACE <NAME>
  MANAGED BY DATABASE
  USING (FILE'<path>' <size>)
```

通过使用三个不同的驱动器上的三个目录，下列 SQL 语句在 OS/2 或 Windows NT 上创建了一个 SMS 表空间：

```
CREATE TABLESPACE RESOURCE
  MANAGED BY SYSTEM
  USING ('d:\acc_tbsp', 'e:\acc_tbsp', 'f:\acc_tbsp')
```

通过使用各自有 5,000 页的两个文件容器，下列 SQL 语句在 OS/2 上创建了一个 DMS 表空间：

```
CREATE TABLESPACE RESOURCE
  MANAGED BY DATABASE
  USING (FILE'd:\db2data\acc_tbsp' 5000,
        FILE'e:\db2data\acc_tbsp' 5000)
```

在上述两个示例中，为容器提供了显式的名称。然而，若指定相对容器名，则将在为该数据库创建的子目录中创建容器（有关数据库物理目录的更多信息，参考 《管理指南：计划》）。

另外，若指定的路径名的一部分不存在，数据库管理器将会创建它。若数据库管理器创建了一个子目录，当删除该表空间时数据库管理器也可能将它删除。

在上述示例中假定这些表空间不与特定的节点组相关。当该语句中未指定下列参数时，使用缺省节点组 IBMDEFAULTGROUP：

```
IN nodegroup
```

通过使用各有 10 000 页的三个逻辑卷，下列 SQL 语句在基于 UNIX 的系统上创建了一个 DMS 表空间，并指定它们的 I/O 特征：

```
CREATE TABLESPACE RESOURCE
  MANAGED BY DATABASE
  USING (DEVICE '/dev/rdb1v6' 10000,
        DEVICE '/dev/rdb1v7' 10000,
        DEVICE '/dev/rdb1v8' 10000)
  OVERHEAD 24.1
  TRANSFERRATE 0.9
```

在此 SQL 语句中提到的 UNIX 设备必须已经存在，且实例所有者和 SYSADM 组必须能够写入它们。

下列示例在一个 UNIX 分区数据库中称为 ODDNODEGROUP 的节点组上创建了一个 DMS 表空间。ODDNODEGROUP 必须是先前使用 CREATE NODEGROUP 语句创建的。在此示例中，假设 ODDNODEGROUP 节点组由编号为 1、3 和 5 的数据库分区组成。在所有数据库分区上，都使用具有 10 000 个 4 KB 页的 /dev/hdisk0 设备。另外，还为每个数据库分区说明了一个包含 40 000 个 4 KB 大小的页的设备。

```
CREATE TABLESPACE PLANS
  MANAGED BY DATABASE
  USING (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n1hd01' 40000) ON NODE 1
        (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n3hd03' 40000) ON NODE 3
        (DEVICE '/dev/HDISK0' 10000, DEVICE '/dev/n5hd05' 40000) ON NODE 5
```

UNIX 设备分为两类：字符串行设备和块结构化设备。对于所有文件系统设备，正常情况是每个块设备（或已处理的设备）都有一个对应的字符串行设备（或原始设备）。块结构化设备通常由类似于“hd0”或“fd0”的名称来指定。字符串行设备通常由类似于“rhd0”、“rfd0”或“rmt0”的名称来指定。这些字符串行设备访问速度比块设备快。在 CREATE TABLESPACE 命令上应使用字符串行设备名，而不应使用块设备名。

额外开销和传送速率有助于确定在编译 SQL 语句时要使用的最佳访问路径。有关 OVERHEAD 和 TRANSFERRATE 参数的信息，参考《管理指南：性能》中的“调整应用程序性能”。

DB2 可使用顺序预取设施（它使用并行 I/O）来大大改善顺序 I/O 的性能。有关此设施的详细信息，参考《管理指南：性能》中的“了解顺序预取”。

您还可以创建一个表空间，它使用的页大小比缺省的 4 KB 大小更大。下列 SQL 语句在基于 UNIX 的系统上创建一个具有 8 KB 页大小的 SMS 表空间。

```
CREATE TABLESPACE SMS8K
  PAGESIZE 8192
  MANAGED BY SYSTEM
  USING ('FMS_8K_1')
  BUFFERPOOL BUFFPOOL8K
```

注意相关的缓冲池也必须具有相同的 8 KB 页大小。

创建的表空间只有在它所引用的缓冲池被激活后才能使用。

ALTER TABLESPACE SQL 语句可以用于将容器添加至 DMS 表空间，并修改表空间的 PREFETCHSIZE、OVERHEAD 和 TRANSFERRATE 设置。应尽可能提交发出表空间语句的事务，以防止系统目录的争用。

**注：**PREFETCHSIZE 应该是 EXTENTSIZE 的倍数。例如，若 EXTENTSIZE 是 10，则 PREFETCHSIZE 应是 20 或 30。有关更多信息，参考《管理指南：性能》中的“了解顺序预取”。

## 创建系统临时表空间

系统临时表空间用来存储系统临时表。创建数据库时，定义的三个缺省表空间的其中之一便是名为“TEMPSPACE1”的系统临时表空间。

**注：**因为系统临时表只能存储在系统临时表空间中，所以数据库必须始终有至少一个这样的表空间。

可使用 CREATE TABLESPACE 语句来创建另一个系统临时表空间。例如，

```
CREATE SYSTEM TEMPORARY TABLESPACE tmp_tbsp
  MANAGED BY SYSTEM
  USING ('d:\tmp_tbsp','e:\tmp_tbsp')
```

创建系统临时表空间时，只能指定 IBMTEMPGROUP 节点组。

## 创建用户临时表空间

用户临时表空间用来存储已说明临时表。

可使用 CREATE TABLESPACE 语句来创建用户临时表空间：

```
CREATE USER TEMPORARY TABLESPACE usr_tbsp
  MANAGED BY DATABASE
  USING (FILE 'd:\db2data\user_tbsp' 5000,
        FILE 'e:\db2data\user_tbsp' 5000)
```

与常规表空间一样，用户临时表空间可以在除 IBMTEMPGROUP 之外的任何节点组中创建。创建用户临时表空间时使用的缺省节点组是 IBMDEFAULTGROUP。

DECLARE GLOBAL TEMPORARY TABLE 语句定义供在用户临时表空间中使用的已说明临时表。

## 在节点组中创建表空间

通过将一个表空间放置在一个多数据库分区节点组中，就将该表空间内的所有表划分到或分区到该节点组的各个数据库分区中。由此该表空间被创建到了一个节点组中。一旦位于一个节点组中，该表空间就必须保留在该处；而不能更改到另一个节点组中。CREATE TABLESPACE 语句用于将一个表空间与一个节点组关联。

## 原始 I/O

“DB2 通用数据库”支持直接磁盘访问（原始 I/O）。这允许您将直接访问磁盘（原始）设备连接至任何“DB2 通用数据库”系统。（例外情况只有 Windows 95 和 Windows 98 操作系统。）以下列表说明标识此类设备的物理方法和逻辑方法：

- 在 Windows 上，要指定物理硬盘驱动器，使用以下语法：

```
\\.\PhysicalDriveN
```

其中，N 表示系统中的一个物理驱动器。在这种情况下，N 可以替换为 0、1、2 或任何其他正整数：

```
\\.\PhysicalDisk5
```

- 在 Windows 上，要指定逻辑原始分区（即，未格式化的分区），使用以下语法：

```
\\.\N:
```

其中 N: 表示系统中的一个逻辑驱动器盘符。例如，N: 可被 E: 或任何其他盘符替换。

- **注意：**您必须拥有安装了服务包 3 的 Windows NT 版本 4.0，才能够将日志写入设备。
- 在基于 UNIX 的平台上，使用字符串行设备名；例如，/dev/rhd0

## 在 Linux 上使用原始 I/O

Linux 具有一个原始设备节点池，必须将该池绑定到块设备上，然后才能对该池执行原始 I/O。有一个原始设备控制器，它充当原始设备至块设备绑定信息的中心库。使用名为 raw 的实用程序执行绑定，该程序通常由 Linux 发行人提供。

在 Linux 上设置原始 I/O 之前，需要下列条件：

- 一个或多个自由 IDE 或 SCSI 磁盘分区

- Linux 内核 2.4.0 或更高版本（然而，某些 Linux 分发在 2.2 内核上提供原始 I/O。）
- 名为 `/dev/rawctl` 或 `/dev/raw` 的原始设备控制器。若没有该设备，则创建一个符号链路：  

```
# ln -s /dev/your_raw_dev_ctrl /dev/rawctl
```
- 原始实用程序，通常通过 Linux 分发提供该实用程序
- DB2 版本 7.1 修订包 3 或更高版本

**注：**对于当前支持原始 I/O 的分发，原始设备节点的命名各不相同：

分发	原始设备节点	原始设备控制器
RedHat 6.2	<code>/dev/raw/raw1</code> 至 255	<code>/dev/rawctl</code>
SuSE 7.0	<code>/dev/raw1</code> 至 63	<code>/dev/raw</code>

DB2 支持上述原始设备控制器中的任何一种，并且支持原始设备节点的大多数其它名称。在 Linux/390 上 DB2 不支持原始设备。

要在 Linux 上配置原始 I/O：

在本示例中，要使用的原始分区是 `/dev/sda5`。它应该不包含任何有用的数据。

步骤 1. 计算本分区中的页数，每个页面的字节数为 4096 个字节，必要时四舍五入。例如：

```
# fdisk /dev/sda
Command (m for help): p

Disk /dev/sda: 255 heads, 63 sectors, 1106 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot   Start    End    Blocks  Id System
/dev/sda1             1     523    420097  83 Linux
/dev/sda2             524   1106   4682947+  5 Extended
/dev/sda5             524   1106   4682947  83 Linux

Command (m for help): q
#
```

在 `/dev/sda5` 中的页数为

```
num_pages = floor( ((1106-524+1)*16065*512)/4096 )
num_pages = 11170736
```

步骤 2. 将未使用的原始设备节点绑定至此分区。每次重新引导机器时，需要执行此操作，且需要 root 访问。使用 `raw -a` 来查看已经在使用哪些原始设备：

```
# raw /dev/raw/raw1 /dev/sda5
/dev/raw/raw1: bound to major 8, minor 5
```

步骤 3. 设置对原始设备控制器和磁盘分区的全局读许可权。设置对原始设备的全局读和写许可权。

```
# chmod a+r /dev/rawctl
# chmod a+r /dev/sdb1
# chmod a+rw /dev/raw/raw1
```

步骤 4. 在 DB2 中创建表空间，指定原始设备，而不是磁盘分区。例如：

```
CREATE TABLESPACE dms1
    MANAGED BY DATABASE
    USING (DEVICE '/dev/raw/raw1' 11170736)
```

DB2 所支持的所有其它页大小也支持原始设备上的表空间。

---

## 创建模式

在将数据组织成表时，它可能还有助于将表（和其他相关的对象）编组在一起。为此，须使用 `CREATE SCHEMA` 语句来定义一个模式。有关该模式的信息保存在连接的数据库的系统目录表中。当创建其他对象时，就可以将它们置于此模式内。

`CREATE SCHEMA` 语句的语法在 *SQL Reference* 中有详细的描述。系统目录中不能已经存在该新模式名，且该名称不能以 "SYS" 开头。

若用户具有 `SYSADM` 或 `DBADM` 权限，则用户可以使用任何有效的名称来创建模式。当创建数据库时，会将 `IMPLICIT_SCHEMA` 权限授予 `PUBLIC`（即，授予所有用户）。

作为 `CREATE SCHEMA` 语句的一部分创建的任何对象的定义者是模式所有者。此所有者可以 `GRANT` 和 `REVOKE` 其他用户的模式特权。

此语句必须由具有 `DBADM` 权限的用户发出。

模式也可能是在用户具有 `IMPLICIT_SCHEMA` 权限时隐式创建的。使用此权限，无论何时用户使用未存在的模式名创建对象，都会隐式创建一个模式。

若用户不具有 `IMPLICIT_SCHEMA` 权限，则他们可以创建的唯一模式是具有与他们自己的授权标识同名的模式。

由于使用模式来强制数据库中的唯一性，所以不允许直接访问模式中的对象。当考虑两个用户有可能使用同一名称创建两个表（或其它对象）时，这一点变得很清楚。没有模式来强制唯一性时，若第三个用户试图查询该表，将会造成不明确。没有进一步的限时时，则不可能确定要使用哪一个表。

要允许另一用户来访问表而不必输入模式名作为表名的限定部分，需要为该用户建立视图。视图定义将定义包括用户模式的全限定表名；用户将只需要使用视图名进行查询。将通过作为视图定义一部分的用户模式对视图进行全限定。

要使用“控制中心”创建模式：

1. 展开对象树，直到您看到**模式**文件夹为止。
2. 右键单击**模式**文件夹，并从弹出菜单中选择**创建**。
3. 填写新模式的信息，并单击**确定**。

要使用命令行来创建模式，输入：

```
CREATE SCHEMA <name> AUTHORIZATION <name>
```

以下是 CREATE SCHEMA 语句的一个示例，它为具有授权标识“joe”的个别用户创建一个模式：

```
CREATE SCHEMA joeschma AUTHORIZATION joe
```

## 设置模式

您也许想建立一个缺省模式，用于在特定的 DB2 连接期间内发出的动态 SQL 语句中非限定的对象引用。为此，将专用寄存器 CURRENT SCHEMA 设置为希望用作缺省模式的模式。任何用户都可以设置此专用寄存器：不需任何授权。

SET SCHEMA 语句的语法在 *SQL Reference* 手册中有详细的描述。

以下是如何设置 CURRENT SCHEMA 专用寄存器的一个示例：

```
SET CURRENT SCHEMA = 'SCHEMA01'
```

此语句可以在应用程序内使用或以交互方式发出。CURRENT SCHEMA 专用寄存器的值一旦被设置后，就用作动态 SQL 语句中非限定对象引用的限定符（模式），但存在对数据库对象的非限定引用的 CREATE SCHEMA 语句除外。

CURRENT SCHEMA 专用寄存器的初始值等于当前会话用户的授权标识。

---

## 创建和填充表

在确定如何将数据组织成表之后，下一步就是使用 CREATE TABLE 语句来创建那些表。表描述存储在连接的数据库的系统目录中。

CREATE TABLE 语句的语法在 *SQL Reference* 中有详细的描述。有关创建摘要表的信息，参见第129页的『创建摘要表』。有关命名表、列和其他数据库对象的信息，参见第277页的『附录A. 命名规则』。



CREATE TABLE 语句给予该表一个名称（该名称可以是限定的或非限定的标识符），并给它的每个列一个定义。可以将每个表存储在单独的表空间中，以使一个表空间只包含一个表。若一个表经常被删除和创建，一个更有效的方法是将其存储在单独的表空间中，然后删除该表空间而不是表。也可以将多个表存储在单个表空间中。在分区数据库环境中，选择的表空间也定义用于存储表数据的节点组和数据库分区。

最初，该表不包含任何数据。要将数据行添加至该表，使用下列其中一项：

- INSERT 语句，在 *SQL Reference* 中有描述
- LOAD 或 IMPORT 命令，在 *Command Reference* 中有描述
- 自动装入程序实用程序（若在如 *Data Movement Utilities Guide and Reference* 中所描述的分区数据库环境中工作）。

有关将数据移入和移出表的详细信息，可参阅 *Data Movement Utilities Guide and Reference*。

可以在不记录更改的情况下向表添加数据。CREATE TABLE 语句上的 NOT LOGGED INITIALLY 子句阻止记录对表的更改。在创建该表的同一个工作单元中由 INSERT、DELETE、UPDATE、CREATE INDEX、DROP INDEX 或 ALTER TABLE 操作对该表所做的任何更改都不会被记录。记录在后续的工作单元中开始。

一个表由一个或多个列定义组成。可为一个表定义最多 500 列。列表示实体的属性。任何列中的值都是相同类型的信息。有关更多信息，参考 *SQL Reference*。

**注：**当使用 4 KB 页大小时，最多可有 500 列。当使用 8 KB、16 KB 或 32 KB 页大小时，最多可有 1012 列。

列定义包括列名、数据类型和任何需要的空属性或缺省值（由用户可选）。

列名描述列中包含的信息，它应是易于识别的。列名在表内必须是唯一的；但是，在其他表中可以使用与它相同的名称。有关命名规则的信息，参见第277页的『附录A. 命名规则』。

列的数据类型指示列值的长度和适合该列的数据类型。数据库管理器使用字符串、数字、日期、时间和大对象数据类型。图形字符串数据类型只可用于使用多字节字符集的数据库环境。另外，可使用用户定义单值类型来定义列，这在第125页的『创建用户定义类型 (UDT)』中做了讨论。

缺省属性规范指示在未提供值的情况下，将使用什么值。可以指定缺省值，也可以使用系统定义的缺省值。可以为带或不带空值属性规范的列指定缺省值。

空值属性规范指示列是否可以包含空值。

要使用“控制中心”来创建表:

1. 展开对象树, 直到您看到表文件夹为止。
2. 右键单击表文件夹, 并从弹出菜单中选择**创建** → **使用向导创建表**。
3. 遵循向导中的步骤来完成该任务。

要使用命令行来创建表, 输入:

```
CREATE TABLE <NAME>
  (<column_name> <data_type> <null_attribute>)
  IN <TABLE_SPACE_NAME>
```

以下是使用 CREATE TABLE 语句在 RESOURCE 表空间中创建 EMPLOYEE 表的一个示例。此表是在样本数据库中定义的:

```
CREATE TABLE EMPLOYEE
  (EMPNO      CHAR(      ) NOT NULL PRIMARY KEY,
   FIRSTNME  VARCHAR(12) NOT NULL,
   MIDINIT   CHAR(1)     NOT NULL WITH DEFAULT,
   LASTNAME  VARCHAR(15) NOT NULL,
   WORKDEPT  CHAR(3),
   PHONENO   CHAR(4),
   PHOTO     BLOB(10M)  NOT NULL)
  IN RESOURCE
```

当创建一个表时, 可根据结构化类型的属性选择该表的某些列。这种表称为“类型表”。

可定义一个类型表, 以便从另一个类型表继承其某些列。这种表称为“子表”, 而被继承的表称为其“超表”。一个类型表及其所有子表的组合称为“表层次结构”。该表层次结构中最上层的表(没有超表的表)称为该层次结构的“根表”。

下面几节以上一个示例为基础讨论应该考虑的其他选项:

- 第107页的『大对象 (LOB) 列的注意事项』
- 第108页的『定义唯一约束』
- 第112页的『在新表上定义生成列』
- 第113页的『创建用户定义的临时表』
- 第114页的『在新表上定义标识列』
- 第115页的『创建序列』
- 第117页的『比较 IDENTITY 列和序列』
- 第117页的『创建类型表』
- 第117页的『填充类型表』

- 第117页的『层次结构表』
- 第118页的『在多个表空间中创建一个表』
- 第119页的『在分区数据库中创建表』

也可以创建根据查询结果定义的表。这种类型的表称为摘要表。有关更多信息，参见第129页的『创建摘要表』。

## 大对象 (LOB) 列的注意事项

在创建包含大对象列的表之前，需要作出下列决策：

### 1. 要记录 LOB 列的更改吗？

若不想记录这些更改，当创建表时必须通过指定 NOT LOGGED 子句来将记录关闭：

```
CREATE TABLE EMPLOYEE
  (EMPNO      CHAR(4)      NOT NULL PRIMARY KEY,
   FIRSTNAME  VARCHAR(12)  NOT NULL,
   MIDINIT    CHAR(1)      NOT NULL WITH DEFAULT,
   LASTNAME   VARCHAR(15)  NOT NULL,
   WORKDEPT   CHAR(3),
   PHONENO    CHAR(4),
   PHOTO      BLOB(10M)    NOT NULL NOT LOGGED)
IN RESOURCE
```

若该 LOB 列大于 1 GB，则必须将记录关闭。（根据经验，可能不希望记录大于 10 MB 的 LOB 列。）与列定义中指定的其他选项一样，更改记录选项的唯一方法是重新创建该表。

即使选择不记录更改，也会对 LOB 列建立影子，以允许回滚更改，不管该回滚是系统生成的错误导致的还是应用程序请求的结果。建立影子是一种恢复技术，使用该技术从不会覆盖当前存储器页的内容。即，旧的未修改的页保留为“影子”副本。当不再需要这些副本来支持事务的回滚时，就将它们废弃。

**注：**当使用 RESTORE 和 ROLLFORWARD 命令恢复数据库时，自上次备份以来“未记录的”和已写入的 LOB 数据会被二进制零替换。

### 2. 要最小化 LOB 列所需的空间吗？

可以在 CREATE TABLE 语句中使用 COMPACT 子句来使 LOB 列尽可能地小。例如：

```
CREATE TABLE EMPLOYEE
  (EMPNO      CHAR(4)      NOT NULL PRIMARY KEY,
   FIRSTNAME  VARCHAR(12)  NOT NULL,
   MIDINIT    CHAR(1)      NOT NULL WITH DEFAULT,
   LASTNAME   VARCHAR(15)  NOT NULL,
```

```

        WORKDEPT CHAR(3),
        PHONENO CHAR(4),
        PHOTO BLOB(10M) NOT NULL NOT LOGGED COMPACT)
IN RESOURCE

```

当追加至含有压缩的 LOB 列的表时，会存在性能成本，尤其是在 LOB 值的大小增加的情况下（因为必须调整存储器）。

在 OS/2 这种不支持稀疏文件分配且将 LOB 置于 SMS 表空间中的平台上，应考虑使用 COMPACT 子句。稀疏文件分配涉及到操作系统如何使用物理磁盘空间。支持稀疏文件分配的操作系统存储 LOB 所用的物理磁盘空间，没有不支持稀疏文件分配的操作系统所用的空间大。无论是否支持稀疏文件分配，COMPACT 选项允许“节省”甚至更大的物理磁盘空间。因为使用 COMPACT 可以节省一些物理磁盘空间，所以若操作系统不支持稀疏文件分配，应该考虑使用 COMPACT。

**注：**DB2 系统目录使用 LOB 列，且可能会占用比先前版本中更多的空间。

- 希望 LOB 列（包括在 DB2 系统目录中的那些 LOB 列）有更好的性能吗？目录表中存在大对象 (LOB) 列。LOB 数据不与其他数据一起保留在缓冲池中，而是每次需要时从磁盘中读取。从磁盘中读取会降低涉及目录的 LOB 列的 DB2 的性能。因为一个文件系统常常有它自己存储（或高速缓存）数据的地方，所以使用 SMS 表空间或在文件容器上构建的 DMS 表空间，使得在先前引用 LOB 时避免 I/O 成为可能。

## 定义约束

本节讨论如何定义约束：

- 『定义唯一约束』
- 第109页的『定义参考约束』
- 第112页的『定义表检查约束』。

有关约束的更多信息，参考《管理指南：计划》中有关为约束强制作计划的章节，并参考 *SQL Reference*。

### 定义唯一约束

唯一约束确保在指定键中的每个值都是唯一的。一个表可以有任意多个唯一约束，且至多将一个唯一约束定义为一个主键。

可在 CREATE TABLE 或 ALTER TABLE 语句中使用 UNIQUE 子句来定义唯一约束。唯一键可以由多个列组成。在一个表上允许多个唯一约束。但可能无法在子表上定义唯一约束。

一旦建立了该约束，当 `INSERT` 或 `UPDATE` 语句修改表中的数据时，数据库管理器会自动实现该唯一约束。唯一约束通过唯一索引来实现。

当在 `ALTER TABLE` 语句中定义唯一约束且在该唯一键的同一组列上存在一个索引时，该索引就成为唯一索引且被该约束使用。

可以提取任何一个唯一约束，并将它用作主键。主键可以用作参考约束（以及其他唯一约束）中的父键。每个表只能有一个主键。可在 `CREATE TABLE` 或 `ALTER TABLE` 语句中使用 `PRIMARY KEY` 子句来定义主键。主键可以由多个列组成。

主索引强制该主键的值为唯一的。当使用主键创建表时，数据库管理器会在该键上创建一个主索引。

用作唯一约束的索引的某些性能提示包括：

- 当初次装入带索引的空表时，`LOAD` 将提供比 `IMPORT` 更好的性能。不管是使用 `LOAD` 的 `INSERT` 方式还是 `REPLACE` 方式，这种情况都一样。
- 当把大量数据追加到一个带索引的现存表中（使用 `IMPORT INSERT` 或 `LOAD INSERT`）时，`LOAD` 的性能只比 `IMPORT` 的稍好一点。
- 若要使用 `IMPORT` 命令来进行大量数据的初始装入，则要在导入或装入数据之后创建唯一键。这样避免了当装入该表时维护索引的额外开销。它还使索引使用最少量的存储器。
- 若正在以 `REPLACE` 方式使用装入实用程序，则在装入数据之前创建唯一键。在这种情况下，在装入期间创建索引比在装入之后使用 `CREATE INDEX` 语句更有效。

### 定义参考约束

参考完整性是通过将参考约束添加到表和列定义来实现的。参考约束是使用 `CREATE TABLE` 或 `ALTER TABLE` 语句中的 `FOREIGN KEY` 子句和 `REFERENCES` 子句建立的。有关参考约束对类型表以及对作为类型表的父表的影响的更多信息，参考 *SQL Reference*。

外键的标识在一个表的行内或两个表的行之间的值上施加约束。数据库管理器检查在表定义中指定的约束，并根据它来维持关系。目标是无论何时一个数据库对象引用另一个数据库对象都要维持完整性。

例如，主键和外键各有一个部门号列。对于 `EMPLOYEE` 表，该列名为 `WORKDEPT`，而对于 `DEPARTMENT` 表，该列名为 `DEPTNO`。这两个表之间的关系由下列约束定义：

- 对于 EMPLOYEE 表中的每个雇员只有一个部门号，且该编号存在于 DEPARTMENT 表中。
- EMPLOYEE 表中的每一行都只与 DEPARTMENT 表中的一行相关。这两个表之间存在唯一的关系。
- 在 EMPLOYEE 表中具有 WORKDEPT 的非空值的每一行只与 DEPARTMENT 表的 DEPTNO 列中的一行相关。
- DEPARTMENT 表是父表，而 EMPLOYEE 表是从属表。

定义父表 DEPARTMENT 的 SQL 语句如下所示：

```
CREATE TABLE DEPARTMENT
  (DEPTNO   CHAR(3)      NOT NULL,
   DEPTNAME VARCHAR(29) NOT NULL,
   MGRNO    CHAR(6),
   ADMRDEPT CHAR(3)     NOT NULL,
   LOCATION CHAR(16),
           PRIMARY KEY (DEPTNO))
IN RESOURCE
```

定义从属表 EMPLOYEE 的 SQL 语句如下所示：

```
CREATE TABLE EMPLOYEE
  (EMPNO    CHAR(4)      NOT NULL PRIMARY KEY,
   FIRSTNME VARCHAR(12) NOT NULL,
   LASTNAME VARCHAR(15) NOT NULL,
   WORKDEPT CHAR(3),
   PHONENO  CHAR(4),
   PHOTO    BLOB(10m)   NOT NULL,
           FOREIGN KEY DEPT (WORKDEPT)
           REFERENCES DEPARTMENT ON DELETE NO ACTION)
IN RESOURCE
```

通过将 DEPTNO 列指定为 DEPARTMENT 表的主键，而将 WORKDEPT 指定为 EMPLOYEE 表的外键，就对 WORKDEPT 值定义了参考约束。此约束实现这两个表的值之间的参考完整性。在这种情况下，添加至 EMPLOYEE 表的任何雇员必须具有一个可以在 DEPARTMENT 表中找到的部门号。

雇员表中的参考约束的删除规则为 NO ACTION，这表示若一个部门中有任何雇员，则不能将该部门从 DEPARTMENT 表中删除。

虽然先前的示例使用 CREATE TABLE 语句来添加参考约束，但是也可以使用 ALTER TABLE 语句。参见第165页的『修改表的结构和内容』。

另一个示例：使用与先前示例所用的相同表定义。另外，在 EMPLOYEE 表之前创建 DEPARTMENT 表。每个部门有一个经理，且该经理在 EMPLOYEE 表中列出。DEPARTMENT 表的 MGRNO 实际是 EMPLOYEE 表的外键。因为此参考循环，此约束存在一个小小的问题。可能会稍后添加外键（参见第168页的『添加

主键和外键』)。也可能使用 CREATE SCHEMA 语句来同时创建 EMPLOYEE 表和 DEPARTMENT 表（参见 *SQL Reference* 中的示例）。

**FOREIGN KEY 子句：** 外键在同一个或另一个表中引用主键或唯一键。外键的指定指示将根据指定的参考约束来维持该参考完整性。可在 CREATE TABLE 或 ALTER TABLE 语句中使用 FOREIGN KEY 子句来定义外键。

外键中的列数必须等于父表的对应主约束或唯一约束（称为父键）中的列数。另外，键列定义的对应部分必须具有相同的数据类型和长度。可以赋予外键一个约束名。若未赋予一个名称，则会自动赋予一个。为便于使用，建议赋予一个约束名，而不要使用系统生成的名称。

若一个组合的外键每一列的值等于父键对应列的值，则该外键的值与该父键的值匹配。包含空值的外键不能与父键的值匹配，因为定义的父键不能有空值。但是，一个空的外键值始终是有效的，无论它的任何一个非空部分的值如何。

下列规则适用于外键定义：

- 一个表可以有多个外键
- 若任何部分都是可空的，则该外键为可空的
- 若任何部分都是空值，则该外键值为空值。

**REFERENCES 子句：** REFERENCES 子句标识一个关系中的父表，并定义需要的约束。可以将它包括在列定义中，或作为一个单独的子句与 FOREIGN KEY 子句一起包括在 CREATE TABLE 或 ALTER TABLE 语句中。

若将 REFERENCES 子句指定为列约束，则隐式列表由列出的一个或多个列名组成。记住，多个列可以有分离的 REFERENCES 子句，而单个列可以有多个 REFERENCES 子句。

REFERENCES 子句中包括的是删除规则。在我们的示例中，使用 ON DELETE NO ACTION 规则，它指示若有雇员分配给部门，则不能删除该部门。其他删除规则包括 ON DELETE CASCADE、ON DELETE SET NULL 和 ON DELETE RESTRICT。有关实现参考完整性时的 DELETE 规则的更多信息，参考《管理指南：计划》。

**实用程序操作的意义：** 装入实用程序将关闭对自引用和从属表的约束检查，并将这些表置于检查暂挂状态。在装入实用程序完成之后，将对需要关闭了其约束检查的所有表打开约束检查。例如，若 DEPARTMENT 和 EMPLOYEE 表是唯一处于检查暂挂状态的表，则可执行下列命令：

```
SET INTEGRITY FOR DEPARTMENT, EMPLOYEE IMMEDIATE CHECKED
```

参考约束以下列方式影响导入实用程序:

- 若该对象表有该表以外的其他对象从属于它, 则不允许 `REPLACE` 和 `REPLACE CREATE` 函数。  
要使用这些函数, 首先要删除该表为父表的所有外键。当导入完成时, 使用 `ALTER TABLE` 语句重新创建这些外键。
- 导入带自参考约束的表的成功与否取决于这些行导入的次序。

### 定义表检查约束

表检查约束指定搜索条件, 对于定义了表检查约束的表的每一行都实现该搜索条件。在创建或改变表时, 通过将检查约束定义与表关联来对该表创建表检查约束。当 `INSERT` 或 `UPDATE` 语句修改该表中的数据时, 就自动激活此约束。表检查约束对 `DELETE` 或 `SELECT` 语句没有影响。检查约束不能与类型表相关。

约束名不能与在同一个 `CREATE TABLE` 语句内指定的任何其他约束相同。若不指定约束名, 系统会为该约束生成 18 个字符的唯一标识符。

表检查约束用于实现键唯一性或参考完整性约束所未涵盖的数据完整性规则。在某些情况中, 表检查约束可以用于实现域检查。在 `CREATE TABLE` 语句上发出的下列约束确保每个活动的开始日期不在同一个活动的结束日期之后:

```
CREATE TABLE EMP_ACT
  (EMPNO      CHAR(6)      NOT NULL,
   PROJNO     CHAR(6)      NOT NULL,
   ACTNO      SMALLINT     NOT NULL,
   EMPTIME    DECIMAL(5,2),
   EMSTDATE   DATE,
   EMENDATE   DATE,
   CONSTRAINT ACTDATES CHECK(EMSTDATE <= EMENDATE) )
IN RESOURCE
```

虽然上一个示例使用 `CREATE TABLE` 语句来添加表检查约束, 但是也可以使用 `ALTER TABLE` 语句。参见第165页的『修改表的结构和内容』。

### 在新表上定义生成列

生成列在基本表中定义, 在这些列中, 存储的值是使用表达式计算得出的, 而不是通过插入或更新操作指定。当创建已知始终要使用特定表达式或谓词的表时, 可对该表添加一个或多个生成列。通过使用生成列, 有机会在查询表数据时改进性能。

例如, 当性能很重要时, 以下两种表达式求值方式成本很高:

1. 必须在查询期间进行许多次表达式求值。
2. 计算很复杂。



为了改进查询的性能，可定义一个附加列，它将包含该表达式的结果。然后，当发出包括同一表达式的查询时，可直接使用生成列，或者，优化器的查询重写组件可用生成列替换该表达式。

也有可能对生成列创建不唯一的索引。

当查询涉及连接两个或多个表中的数据时，添加生成列允许优化器选择可能更好的连接策略。

以下是在 CREATE TABLE 语句上定义生成列的一个示例：

```
CREATE TABLE t1 (c1 INT,  
                 c2 DOUBLE,  
                 c3 DOUBLE GENERATED ALWAYS AS (c1 + c2)  
                 c4 GENERATED ALWAYS AS  
                 (CASE WHEN c1 > c2 THEN 1 ELSE NULL END))
```

在创建此表之后，可以使用生成列来创建索引。例如，

```
CREATE INDEX i1 ON t1(c4)
```

查询可以利用生成列。例如，

```
SELECT COUNT(*) FROM t1 WHERE c1 > c2
```

可以写成

```
SELECT COUNT(*) FROM t1 WHERE c4 IS NOT NULL
```

另一个示例：

```
SELECT c1 + c2 FROM t1 WHERE (c1 + c2) * c1 > 100
```

可以写成

```
SELECT c3 FROM t1 WHERE c3 * c1 > 100
```

将使用生成列来改进查询的性能。结果是，可能在创建和填充表之后添加生成列。有关更多信息，参见第104页的『创建和填充表』。

## 创建用户定义的临时表

可使用 DECLARE GLOBAL TEMPORARY TABLE 语句来定义临时表。此语句需在应用程序中使用。只有在应用程序与数据库断开连接之前，用户定义临时表才是持续的。

此表的描述并不出现在系统目录中，使其对于其他应用程序而言不是持久的，也不能与其他应用程序共享此表。

当使用此表的应用程序终止或与数据库断开连接时，此表中的数据被删除，此表被隐式删除。

下面是定义临时表的一个示例:

```
DECLARE GLOBAL TEMPORARY TABLE gbl_temp
  LIKE emp1tab1
  ON COMMIT DELETE ROWS
  NOT LOGGED
  IN usr_tbsp
```

此语句创建一个名为 `gbl_temp` 的用户临时表。定义此用户临时表所使用的列的名称和描述与 `emp1tab1` 的列的名称和描述完全相同。隐式定义只包括列名、数据类型、可为空特性和列缺省值属性。未定义所有其他列属性，包括唯一约束、外键约束、触发器和索引。执行 `COMMIT` 操作时，若未对该表打开 `WITH HOLD` 游标，则该表中的所有数据都被删除。不记录对用户临时表所作的更改。用户临时表被放在指定的用户临时表空间中。此表空间必须存在，否则此表的声明将失败。

有关 `DECLARE GLOBAL TEMPORARY TABLE` 语句的其他信息，参考 *SQL Reference*。

**注：** 用户定义临时表不支持:

- LOB 类型的列（或基于 LOB 的单值类型列）
- 用户定义类型列
- LONG VARCHAR 列
- DATALINK 列

## 在新表上定义标识列

标识列向 DB2 提供了一种为要添加至表的每一行自动生成保证唯一数字值的方法。当创建一个表时，若您知道需要唯一标识将添加至该表的每一行，则可向该表添加一个标识列。

在创建之后，便不能将表描述改变为包括标识列。

`CREATE TABLE` 语句上的 `AS IDENTITY` 子句允许说明标识列。

以下是在 `CREATE TABLE` 语句上定义标识列的一个示例:

```
CREATE TABLE table (col1 INT,
                    col2 DOUBLE,
                    col3 INT NOT NULL GENERATED ALWAYS AS IDENTITY
                        (START WITH 100, INCREMENT BY 5))
```

在此示例中，第三个列是标识列。还可以指定该列中用来在添加行时唯一标识每一行的值。这里，输入的第一行将值 "100" 放在该列中；后续添加至该表的每一行都有一个关联的值，此值每次增加 5。

使用标识列的一些其他示例包括：订单号码、员工号码、股票代码或事件号码。标识列的值可以由 DB2: ALWAYS 或 BY DEFAULT 生成。

定义为 GENERATED ALWAYS 的标识列保证是唯一的。使用的值始终由 DB2 生成。不允许应用程序提供显式的值。定义成 GENERATED BY DEFAULT 的标识列使应用程序能够显式地为标识列提供值。若应用程序不提供值，则 DB2 将生成一个值。因为由应用程序控制该值，所以 DB2 不能保证该值的唯一性。GENERATED BY DEFAULT 子句的用途是用于目的为复制现存表内容的数据填充；或用于卸装和重新装入表。

**注：**标识列当前在分区数据库环境中不受支持。

若将行插入指定了显式一致列值的表中，则不更新下一个内部生成的值，且可能会与表中现有的值冲突。重复值将会生成出错信息。

有关在新表上定义标识列的其他信息，参考 *SQL Reference*。

## 创建序列

序列是一个数据库对象，它允许自动生成值。序列特别适合于生成唯一键值这一任务。应用程序可以使用序列来避免由于在数据库外生成唯一计数器而引起可能的并发和性能问题。

不象标识列属性，未使序列与特定表列相关，也未将它绑定至唯一表列，只是仅可通过该表列访问它。

可以创建或改变序列，以便序列以下列其中一种方式来生成值：

- 没有约束单调地递增或递减
- 单调地递增或递减至用户定义的限制并停止
- 单调地递增或递减至用户定义的限制并循环回至起点，然后再开始。

以下是创建序列对象的一个示例：

```
CREATE SEQUENCE order_seq
  START WITH 1
  INCREMENT BY 1
  NOMAXVALUE
  NOCYCLE
  CACHE 24
```

在此示例中，序列称为 `order_seq`。它将从 1 开始，并以 1 为增量增加，且没有上限。没有原因循环回至起点并从 1 重新开始，因为没有指定上限。与 `CACHE` 参数关联的数指定了序列值的最大数目，数据库管理器预分配此数目并将它保存在内存中。

生成的序列号具有下列特性：

- 值可以是小数位为零的任何精确数字数据类型。这样的数据类型包括：`SMALLINT`、`BIGINT`、`INTEGER` 和 `DECIMAL`。
- 连续值之间可以有任何指定的整数增量。缺省递增值是 1。
- 计数器值是可恢复的。当需要恢复时，从日志中重建计数器值。
- 可以高速缓存值以改善性能。在高速缓存中预分配并存储值，可以在为序列生成值时减少对日志的同步 I/O。万一发生系统故障，则永远不会使用尚未提交的所有高速缓存值并认为它们已丢失。为 `CACHE` 指定的值是可能丢失的序列值的最大数目。

若将包含一个或多个序列的数据库恢复至前一时间点，则可能会导致对某些序列生成重复值。要避免可能的重复值，不应该将具有序列的数据库恢复至前一时间点。

序列仅在单节点数据库中受支持。

有两种表达式与序列一起使用。

对于当前会话中的先前语句，`PREVVAL` 表达式对指定序列返回最新生成的值。

`NEXTVAL` 表达式对指定序列返回下一个值。当 `NEXTVAL` 表达式指定序列的名称时，生成一个新的序列号。但是，若在查询中存在 `NEXTVAL` 表达式的多个实例指定同一序列名称，则对于每行结果，序列计数器仅递增一次。

通过对第一个表使用 `NEXTVAL` 表达式引用序列号，而对任何附加的表使用 `PREVVAL` 表达式引用序列号，相同序列号可以在两个单独的表中用作唯一键值。

例如：

```
INSERT INTO order (orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1)
```

可在下列位置使用 `NEXTVAL` 或 `PREVVAL` 表达式：

- `INSERT` 语句、`VALUES` 子句
- `SELECT` 语句、`SELECT` 列表

- SET 赋值语句
- UPDATE 语句、SET 子句
- VALUES 或 VALUES INTO 语句

## 比较 IDENTITY 列和序列

虽然在 IDENTITY 列和序列之间存在相似之处，当也存在差别。当设计数据库和应用程序时可以使用其各自的特征。

标识列具有下列特征：

- 仅当创建了表时，才可以将标识列定义为表的一部分。一旦创建了表，您就不能改变它来添加一个标识列。（然而，可以改变现有的标识列特征。）
- 标识列自动为单个表生成值。
- 当将标识列定义为 GENERATED ALWAYS 时，始终由数据库管理器生成所用的值。在修改表的内容期间，不允许应用程序来提供它们自己的值。

序列对象具有下列特征：

- 序列对象是未与任何一个表关联的数据库对象。
- 序列对象生成可在任何 SQL 语句中使用的顺序值。
- 由于任何应用程序可以使用序列对象，所以有两种表达式可用来控制如何检索指定序列中的下一个值和正在执行的语句之前生成的值。对于当前会话中的先前语句，PREVVAL 表达式对指定序列返回最新生成的值。NEXTVAL 表达式对指定序列返回下一个值。使用表达式允许在几个表中的几个 SQL 语句中使用同一值。

虽然这些并非这两项的所有特征，但是这些特征将有助于您根据数据库设计和使用数据库的应用程序来确定使用哪一项。

## 创建类型表

可以使用 CREATE TABLE 语句的一个变体来创建类型表。有关您所需的类型表的所有信息，参考 *Application Development Guide*。

## 填充类型表

在创建结构化类型并接着创建相对应的表和子表之后，可以填充类型表。有关您所需的类型表的所有信息，参考 *Application Development Guide*。

## 层次结构表

层次结构表是与类型表层次结构的实现相关的一个表。它与层次结构的根表同时创建。有关您所需的所有层次结构表信息，参考 *Application Development Guide*。

## 在多个表空间中创建一个表

表数据、表的索引以及与该表相关联的任何长整数列数据可以存储在同一表空间中。也可以将索引放在一个单独的表空间中，并将任何长整数列数据放在一个单独的表空间中，以便与用于存放其余表数据的表空间分隔开。在运行 `CREATE TABLE` 语句之前，所有表空间都必须存在。只能使用 DMS 表空间来分隔表的各个部分。

要使用“控制中心”来在多个表空间中创建一个表：

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击表文件夹，并从弹出菜单中选择**创建** → **使用向导创建表**。
3. 输入表名并单击下一步。
4. 为该表选择列。
5. 在表空间页上，单击**使用单独的索引空间**和**使用单独的长空间**，指定信息，并单击**完成**。

要使用命令行来在多个表空间中创建一个表，输入：

```
CREATE TABLE <name>
  (<column_name> <data_type> <null_attribute>)
  IN <table_space_name>
  INDEX IN <index_space_name>
  LONG IN <long_space_name>
```

以下示例显示如何创建 EMP\_PHOTO 表来将该表的不同部分存储在不同的表空间中：

```
CREATE TABLE EMP_PHOTO
  (EMPNO      CHAR(6)      NOT NULL,
   PHOTO_FORMAT VARCHAR(10) NOT NULL,
   PICTURE    BLOB(100K) )
  IN RESOURCE
  INDEX IN RESOURCE_INDEXES
  LONG IN RESOURCE_PHOTO
```

此示例将使 EMP\_PHOTO 数据按如下所示存储：

- 为 EMP\_PHOTO 表创建的索引将存储在 RESOURCE\_INDEXES 表空间中
- PICTURE 列的数据将存储在 RESOURCE\_PHOTO 表空间中
- EMPNO 和 PHOTO\_FORMAT 列的数据将存储在 RESOURCE 表空间中。

有关对单个表使用多个 DMS 表空间的其他注意事项，参考《管理指南：计划》。

有关 CREATE TABLE 语句的更多信息，参考 *SQL Reference*。

## 在分区数据库中创建表

在创建将以物理方式划分或分区的一个表之前，需要考虑下列事宜：

- 表空间可以横跨多个数据库分区。它们横跨的分区数取决于一个节点组中的分区数。
- 可以通过如下方法来配置表：将表置于同一个表空间中，或置于另一个表空间中，该表空间与第一个表空间一起与相同的节点组相关。有关更多信息，参考《管理指南：计划》。

当在分区数据库环境中创建表时，有一个附加选项：分区键。分区键是作为一个表定义的一部分的键。它确定用于存储每行数据的分区。

选择适当的分区键很重要，因为以后再也不能更改它。再者，必须将任何唯一索引（因此也是唯一键或主键）定义为分区键的一个超集。即，若定义了分区键，则唯一键和主键必须包括所有与分区键相同的列（它们可能有多列）。

若不显式指定分区键，会使用下列缺省值。确保缺省分区键是适当的。

- 若在 CREATE TABLE 语句中指定了主键，则该主键的第一列会用作分区键。
- 若不存在主键，则使用非长整数字段的第一列。
- 若没有列满足缺省分区键的需求，则会不带关键字创建该表（这只在单分区节点组中允许）。

以下是一个示例：

```
CREATE TABLE MIXREC (MIX_CNTL INTEGER NOT NULL,  
                     MIX_DESC CHAR(20) NOT NULL,  
                     MIX_CHR CHAR(9) NOT NULL,  
                     MIX_INT INTEGER NOT NULL,  
                     MIX_INTS SMALLINT NOT NULL,  
                     MIX_DEC DECIMAL NOT NULL,  
                     MIX_FLT FLOAT NOT NULL,  
                     MIX_DATE DATE NOT NULL,  
                     MIX_TIME TIME NOT NULL,  
                     MIX_TMSTMP TIMESTAMP NOT NULL)  
IN MIXTS12  
PARTITIONING KEY (MIX_INT) USING HASHING
```

在上一个示例中，表空间是 MIXTS12，而分区键是 MIX\_INT。若未显式指定分区键，则它是 MIX\_CNTL。（若未指定主键且未定义分区键，则分区键是该列表中的第一个非长整数字段的列。）

表的一行和有关该行的所有信息始终驻留在同一个数据库分区上。

表的一个分区大小不能超过 64 GB 和可用的磁盘空间中较小的那一个。（假设表空间具有 4 KB 的页大小。）该表的最大大小可以是 64 GB（或可用磁盘空间）乘

以数据库分区数之积。若该表空间的页大小为 8 KB，则该表最大的大小可以为 128 GB（或可用的磁盘空间）乘以数据库分区数之积。若该表空间的页大小为 16 KB，则该表的最大大小可为 256 GB（或可用的磁盘空间）乘以数据库分区数之积。若该表空间的页大小为 32 KB，则该表的最大大小可为 512 GB（或可用的磁盘空间）乘以数据库分区数之积。

---

## 创建触发器

触发器定义一组操作，这组操作与用于指定基表或类型表的 INSERT、UPDATE 或 DELETE 子句一起执行或由这些子句触发。触发器的某些用途如下：

- 验证输入数据
- 为新插入的行生成值
- 为交叉引用而从其他表中进行读取
- 为审计跟踪而向其他表写入

不能使用具有别名的触发器。

可使用触发器支持一般形式的完整性或商业规则。例如，在接受订单或更新摘要数据表之前，触发器可以检查客户的信用额度。

使用触发器的优点有：

- 更快地开发应用程序：因为触发器存储在数据库中，所以不必为它在每个应用程序中执行的操作进行编码。
- 更容易维护：一旦定义了一个触发器，则当访问创建它所基于的表时，会自动调用它。
- 商业规则的全局实现：若商业策略改变，只需更改触发器而不必更改每个应用程序。

要使用“控制中心”来创建触发器：

1. 展开对象树，直到您看到**触发器**文件夹为止。
2. 右键单击**触发器**文件夹，并从弹出菜单中选择**创建**。
3. 指定触发器的信息。
4. 指定要让触发器调用的操作，并单击**确定**。

要使用命令行来创建触发器，输入：



```
CREATE TRIGGER <name>
  <action> ON <table_name>
  <operation>
  <triggered_action>
```

下列 SQL 语句创建一个触发器，它在每次雇佣新人时会增加雇员数，方法为每次向 EMPLOYEE 表添加一行时就在 COMPANY\_STATS 表的雇员数 (NBEMP) 列中加 1。

```
CREATE TRIGGER NEW_HIRED
  AFTER INSERT ON EMPLOYEE
  FOR EACH ROW MODE DB2SQL
  UPDATE COMPANY_STATS SET NBEMP = NBEMP+1;
```

触发器主体可以包括下列 SQL 语句中的一个或多个：INSERT、搜索式 UPDATE、搜索式 DELETE、全查询、SET 转换变量和 SIGNAL SQLSTATE。可以在触发器引用的 INSERT、UPDATE 或 DELETE 语句之前或之后激活触发器。有关 CREATE TRIGGER 语句的完整语法信息，参考 *SQL Reference*。有关创建和使用触发器的信息，参考 *Application Development Guide*。

**注：**若触发器是一个 BEFORE 触发器，则由触发操作指定的列名可能不是生成列而是标识列。即，生成的标识值对 BEFORE 触发器可见。

当创建原子触发器时，必须认真对待语句结束字符。缺省情况下，数据库管理器将 ";" 当作是语句结束标记。您应该在脚本中手工编辑语句结束字符来创建原子触发器，以便使用一个非 ";" 的字符。例如，可以用其它特殊字符（如 "#"）替换 ";"。

然后，您必须：

- 使用在“命令中心”中所选择的脚本选项卡，通过工具—>工具设置菜单更改定界符，然后运行脚本；或者，
- 从“命令行处理器”，使用：

```
db2 -td <delimiter> -vf <script>
```

其中，`delimiter` 是替代的语句结束字符，而 `<script>` 是已修改的在其中使用新定界符的脚本。

## 触发器从属性

触发器对某个其他对象的所有从属性都记录在 SYSCAT.TRIGDEP 目录中。一个触发器可从属于许多个对象。这些对象和从属触发器在 *SQL Reference* 关于 DROP 语句的讨论中有更详尽的描述。

若删除这些对象中的一个，则该触发器就会不起作用，但是它的定义仍保留在目录中。要重新激活此触发器，必须从目录中检索它的定义并提交新的 CREATE TRIGGER 语句。

若删除触发器，则它的描述会从 SYSCAT.TRIGGERS 目录视图中被删除，且它所有的从属也从 SYSCAT.TRIGDEP 目录视图中被删除。所有与该触发器有 UPDATE、INSERT 或 DELETE 关系的软件包都会被停用。

若从属对象是视图且已使它不起作用，则也会将该触发器标记为不起作用。任何从属于已标记为不起作用的触发器的软件包都会被停用。（有关更多信息，参见第 189 页的『更改对象时的语句从属性』。）

---

## 创建用户定义函数 (UDF) 或方法

用户定义函数 UDF 扩展并增强了 SQL 的内置函数提供的支持，且可在可使用内置函数的任何地方使用。可按下列两种方式中任何一种创建 UDF：

- 外部函数，它是用一种编程语言编写的
- 有源函数，它的实现是从另一个现存函数继承来的

有三种类型的 UDF：

**标量** 每次调用它时，都返回一个单值答案。例如，内置函数 SUBSTR() 是一个标量函数。标量 UDF 可以是外部函数或有源函数。

**列** 从一组相似的值（一列）中返回单值答案。在 DB2 中，它有时也称为聚集函数。列函数的一个示例为内置函数 AVG()。不能给 DB2 定义外部列 UDF，但是可以定义源于一个内部列函数的列 UDF。这对于单值类型是有用的。

例如，若存在用基本类型 INTEGER 定义的单值类型 SHOESIZE，则应可以定义源于内置函数 AVG(INTEGER) 的 UDF AVG(SHOESIZE)，且它应该是一个列函数。

**表** 将一个表返回至引用它的 SQL 语句。只能在 SELECT 语句的 FROM 子句中引用表函数。这种函数可以用于将 SQL 语言的处理功能应用于非 DB2 数据的数据，或将此类数据转换成 DB2 表。

例如，表函数可以提取一个文件并将它转换成表，将来自万维网的样本数据制成表，或访问 Lotus Notes 数据库，并返回象邮件消息的日期、发送方和文本这样的信息。此信息可以与该数据库中的其他表连接。

表函数只能是外部函数。它不能是有源函数。

有关现存 UDF 的信息记录在 SYSCAT.FUNCTIONS 和 SYSCAT.FUNCPARMS 目录视图中。系统目录不包含 UDF 的可执行代码（因此，当创建备份和恢复计划时，应考虑如何管理 UDF 可执行程序。）

当编译 SQL 语句时，关于 UDF 性能的统计信息很重要。有关如何在系统目录中更新 UDF 统计信息的资料，参考《管理指南：性能》中的“更新用户定义函数的统计信息”。

有关使用 CREATE FUNCTION 语句来编写适合特定应用程序的 UDF 的详细信息，参考 *Application Development Guide*。有关 UDF 语法的详细信息，参考 *SQL Reference*。

## 创建函数映射

在一个联合数据库中，当需要在一个或多个数据源映射本地函数或具有函数的本地函数模板（在第124页的『创建函数模板』中有描述）时，要创建函数映射。为许多数据源函数提供了缺省函数映射。

在下列情况下，函数映射很有用：

- 在一个数据源中出现了一个新的内置函数。
- 需要将一个数据源中的某个用户定义函数映射为本地函数。
- 应用程序需要与缺省映射提供的缺省行为不同的行为。

用 CREATE FUNCTION MAPPING 语句定义的函数映射存储在联合数据库中。

函数（或函数模板）必须与数据源函数具有相同数目的输入参数。此外，联合端的输入参数数据类型应与数据源端的输入参数数据类型兼容。这些要求也适用于返回的值。

使用 CREATE FUNCTION MAPPING 语句创建函数映射。例如，要在一个 Oracle AVGNEW 函数和服务端 ORACLE1 上一个等效的 DB2 函数之间创建函数映射：

```
CREATE FUNCTION MAPPING ORAVGNEW FOR SYSIBM.AVG(INT) SERVER ORACLE1
OPTIONS (REMOTE_NAME 'AVGNEW')
```

必须对联合数据库具有 SYSADM 或 DBADM 权限，才可使用此语句。函数映射属性存储在 SYSCAT.FUNCMAPPINGS 中。

联合服务器将不绑定输入主变量，或检索 LOB、LONG VARCHAR/VARGRAPHIC、DATALINK、单值类型和结构化类型的结果。若输入参数或返回的值包括其中一个类型，则不能创建函数映射。

有关使用和创建函数映射的其他附加详细信息，参考 *Application Development Guide*。有关 CREATE FUNCTION MAPPING 语法的详细信息，参考 *SQL Reference*。

## 创建函数模板

在联合系统中，函数模板为函数映射提供了“锚点”。若对应的 DB2 函数在联合服务器中不存在，可使用函数模板来启用数据源函数的映射。函数映射需要一个函数模板或一个相似的现存 DB2 函数。

该模板只是一个函数外壳：名称、输入参数和返回值。该函数没有本地可执行文件。

因为没有该函数的本地可执行文件，因此即使可在数据源中使用该函数，调用该函数模板仍可能失败。例如，考虑如下查询：

```
SELECT myfunc(C1)
   FROM nick1
  WHERE C2 < 'A'
```

若 DB2 和包含 nick1 引用的对象的数据源没有相同的整理顺序，查询将失败，因为当该函数位于数据源时必须在 DB2 执行比较。若整理顺序相同，可在具有 myfunc 引用的基本函数的数据源中执行比较操作。

函数（或函数模板）必须与数据源函数具有相同数目的输入参数。联合端的输入参数数据类型应与数据源端的输入参数数据类型兼容。这些要求也适用于返回的值。

使用具有 AS TEMPLATE 关键字的 CREATE FUNCTION 语句创建函数模板。当创建该模板后，可使用 CREATE FUNCTION MAPPING 语句将模板映射为数据源。

例如，要为服务器 S1 上的函数 MYS1FUNC 创建一个函数模板和函数映射：

```
CREATE FUNCTION MYFUNC(INT) RETURNS INT AS TEMPLATE

CREATE FUNCTION MAPPING S1_MYFUNC FOR MYFUNC(INT) SERVER S1 OPTIONS
(REMOTE_NAME 'MYS1FUNC')
```

有关使用和创建函数模板的详细信息，参考 *Application Development Guide*。有关 CREATE FUNCTION 语法的详细信息，参考 *SQL Reference*。

---

## 创建用户定义类型 (UDT)

用户定义类型 (UDT) 是由用户在数据库中创建的命名的数据类型。UDT 可以是单值类型，它与内部数据类型或结构化类型共享一个公共的表示法，结构化类型具有一个命名属性序列，其中每个属性都有一个类型。结构化类型可以是另一个定义类型层次结构的结构化类型（称为超类型）的子类型。

UDT 支持强类型转换，这表示即使它们与其他类型共享相同的表示，但一个给定 UDT 的值会被视为只与同一个类型层次结构中相同的 UDT 的值兼容。

SYSCAT.DATATYPES 目录视图允许查看已为数据库定义的 UDT。此目录视图还显示当创建该数据库时由数据库管理器定义的数据类型。有关所有数据类型的完整列表，参考 *SQL Reference*。

UDT 不能用作大多数系统提供的函数或内置函数的自变量。必须提供用户定义函数来启用这些操作和其他操作。

仅在下列情况下，才可以删除 UDT:

- 现存表的列定义中未使用它。
- 未把它用作现存的类型表或类型视图的类型。
- 在不能删除的 UDF 函数中未使用它。若视图、触发器、表检查约束或另一个 UDF 从属于某个 UDF，则不能删除该 UDF。

当删除一个 UDT 时，从属于它的任何函数也会被删除。

### 创建用户定义单值类型

用户定义单值类型是从现存类型（如整数、小数或字符类型）派生的数据类型。可使用 `CREATE DISTINCT TYPE` 语句创建单值类型。

如下 SQL 语句将单值类型 `t_educ` 创建为 `smallint`:

```
CREATE DISTINCT TYPE T_EDUC AS SMALLINT WITH COMPARISONS
```

若在 `CREATE DISTINCT TYPE` 语句中指定 `WITH COMPARISONS` 子句（如本示例），则具有相同单值类型的实例可以相互比较。若源数据类型是大对象、`DATALINK`、`LONG VARCHAR` 或 `LONG VARGRAPHIC` 类型，则不能指定 `WITH COMPARISONS` 子句。

具有单值类型的实例不能用作在源类型上定义的函数的自变量或操作的操作数。类似地，源类型不能用于为使用单值类型定义的自变量或操作数。

一旦创建了单值类型，就可以在 `CREATE TABLE` 语句中使用它来定义列:

```

CREATE TABLE EMPLOYEE
  (EMPNO          CHAR(6)          NOT NULL,
   FIRSTNAME     VARCHAR(12)     NOT NULL,
   LASTNAME      VARCHAR(15)     NOT NULL,
   WORKDEPT     CHAR(3),
   PHONENO      CHAR(4),
   PHOTO        BLOB(10M)       NOT NULL,
   EDLEVEL      T_EDUC)
IN RESOURCE

```

创建单值类型还会生成对在单值类型和源类型之间进行类型转换的支持。因此，类型为 T\_EDUC 的值可转换为 SMALLINT 值，而 SMALLINT 值可转换为 T\_EDUC 值。

有关 CREATE DISTINCT TYPE 语句的完整语法信息，参考 *SQL Reference*。有关创建和使用单值类型的信息，参考 *Application Development Guide*。

可通过转换将 UDT 转换为基本数据类型，或将基本数据类型转换为 UDT。通过 CREATE TRANSFORM 语句创建转换函数。

也可通过 CREATE METHOD 语句以及 CREATE FUNCTION 语句的扩充语句来获得转换支持。有关此支持的详细信息，参考 *SQL Reference*。

## 创建用户定义结构化类型

结构化类型是包含一个或多个属性的用户定义类型，每个属性有它自己的名称和数据类型。结构化类型可用作表的类型，在这种情况下，表中的每一列可从该结构化类型的一个属性中获取名称和数据类型。有关结构化类型上所需的所有信息，参考 *Application Development Guide*。

## 创建类型映射

在联合系统中，类型映射允许将数据源表和视图中的特定数据类型映射为 DB2 相异数据类型。一个类型映射可以适用于一个数据源或一个范围内（某个类型、版本）的数据源。

为内部数据源类型和内部 DB2 类型提供了缺省数据类型映射。将在 SYSCAT.TYPEMAPPINGS 视图中列出您创建的新的数据类型映射。

用 CREATE TYPE MAPPING 语句创建类型映射。必须对联合数据库具有 SYSADM 或 DBADM 权限，才可使用此语句。

类型映射语句的一个示例是：

```

CREATE TYPE MAPPING MY_ORACLE_DEC FROM SYSIBM.DECIMAL(10,2)
TO SERVER ORACLE1 TYPE NUMBER([10..38],2)

```

不能为 LOB、LONG VARCHAR/VARGRAPHIC、DATALINK、结构化类型或单值类型创建类型映射。

有关使用和创建类型映射的详细信息，参考 *Application Development Guide*。有关 CREATE TYPE MAPPING 语法的详细信息，参考 *SQL Reference*。

---

## 创建视图

视图可从一个或多个基表、别名或视图中派生，且可以在检索数据时与基表互换使用。当对视图中显示的数据进行更改时，该数据会在表中自行更改。

可以创建视图来限制对敏感数据的访问，同时又允许对其他数据进行更多的一般访问。

当插入到一个视图中，而该视图的视图定义的 SELECT 列表直接或间接地包括基表的标识列的名称时，同一规则适用，就象 INSERT 语句直接引用基表的标识列一样。有关 INSERT 语句的更多信息，参考 *SQL Reference*。

除按上述方式使用视图外，视图还可以用于：

- 改变表而不影响应用程序。这可通过创建一个基于基础表的视图来完成。使用基础表的应用程序不会因新视图的创建而受影响。新的应用程序可将创建的视图用于不同的目的，而那些使用基础表的应用程序则不能。
- 对一系列中的值求和，选择最大值，或计算平均值。
- 访问一个或多个数据源中的信息。可在 CREATE VIEW 语句内引用别名，并可创建多个位置 / 全局视图（该视图可以连接位于不同系统中多个数据源的信息）。

当使用标准的 CREATE VIEW 语法创建一个引用别名的视图时，将看到一个警告，它警告您视图用户的认证标识而不是视图创建者的认证标识将用于访问数据源处的基本对象。使用 FEDERATED 关键字阻止此警告。

创建视图的一个替代方法是使用嵌套的或公共的表表达式，以减少目录查找，并提高性能。有关公共表表达式的更多信息，参考 *SQL Reference*。

要使用“控制中心”来创建视图：

1. 展开对象树，直到您看到**视图**文件夹为止。
2. 右键单击**视图**文件夹，并从弹出菜单中选择**创建**。
3. 填写信息，并单击**确定**。

要使用命令行来创建视图，输入：

```
CREATE VIEW <name> (<column>, <column>, <column>)  
  SELECT <column_names> FROM <table_name>  
  WITH CHECK OPTION
```

例如，EMPLOYEE 表中可能有工资信息，它不对每个人都是可用的。但是，雇员的电话号码应是一般人都可访问的。在此情况中，可以仅根据 LASTNAME 和 PHONENO 列创建一个视图。可将该视图的访问权授予 PUBLIC，而将整个 EMPLOYEE 表的访问权限制在具有查看工资信息授权的那些人范围内。有关只读视图的信息，参考 *SQL Reference* 手册。

使用视图，可以使表数据的子集可用于一个应用程序，并验证要插入或更新的数据。视图可以有与原始表中对应列的名称不同的列名。

使用视图使程序和最终用户查询可以灵活地查看表数据。

下列 SQL 语句创建基于 EMPLOYEE 表的一个视图，它列出了部门 A00 的所有雇员及其雇员姓名和电话号码：

```
CREATE VIEW EMP_VIEW (DA00NAME, DA00NUM, PHONENO)  
  AS SELECT LASTNAME, EMPNO, PHONENO FROM EMPLOYEE  
  WHERE WORKDEPT = 'A00'  
  WITH CHECK OPTION
```

此语句的第一行对该视图命名并定义它的列。名称 EMP\_VIEW 在 SYSCAT.TABLES 中它的模式内必须是唯一的。尽管不包含数据，视图名看上去仍象一个表名。该视图将有称为 DA00NAME、DA00NUM 和 PHONENO 的三列，它们与 EMPLOYEE 表中的列 LASTNAME、EMPNO 和 PHONENO 相对应。列出的列名按一一对应的关系应用于 SELECT 语句的选择列表。若不指定列名，则视图使用与 SELECT 语句的结果表的列相同的名称。

第二行是描述要从数据库选择哪些值的 SELECT 语句。它可以包括子句：ALL、DISTINCT、FROM、WHERE、GROUP BY 和 HAVING。为视图提供列的数据对象的一个或多个名称必须跟在 FROM 子句后面。

WITH CHECK OPTION 子句指示必须根据该视图定义检查该视图的任何更新的行或插入的行，若它不符合，则拒绝它。这增强了数据完整性，但是需要附加的处理。若将此子句省略，则不会根据视图定义检查插入和更新。

以下 SQL 语句使用 SELECT AS 子句创建基于 EMPLOYEE 表的相同视图：

```
CREATE VIEW EMP_VIEW  
  SELECT LASTNAME AS DA00NAME,  
         EMPNO AS DA00NUM,  
         PHONENO  
  FROM EMPLOYEE  
  WHERE WORKDEPT = 'A00'  
  WITH CHECK OPTION
```



可以创建在其定义中使用 UDF 的视图。但是，要更新此视图以使它包含最新的函数，必须删除它，然后重新创建它。若视图从属于 UDF，则不能删除该函数。

下列 SQL 语句创建一个在其定义中带有函数的视图：

```
CREATE VIEW EMPLOYEE_PENSION (NAME, PENSION)
AS SELECT NAME, PENSION(HIREDATE, BIRTHDATE, SALARY, BONUS)
FROM EMPLOYEE
```

UDF 函数 PENSION 根据涉及 HIREDATE、BIRTHDATE、SALARY 和 BONUS 的一个公式来计算一个雇员应当得到的当前退休金。

## 创建类型视图

可使用 CREATE VIEW 语句来创建类型视图。有关您所需的所有类型视图信息，参考 *Application Development Guide*。

---

## 创建摘要表

摘要表是以查询结果为基础所定义的一种表。因此，摘要表通常包含预先计算的结果，这些结果是根据表定义中引用的一个或多个表中的现存数据计算而得。若 SQL 编译程序确定一个查询对摘要表运行时比对基表运行时效率更高，将对摘要表执行该查询，并会更快地获得结果。

使用复制选项创建摘要表，这种方法可用于在分区数据库环境中的所有节点上复制表。它们称为“复制的摘要表”。有关这样的表的更多概述信息，参考《*管理指南：计划*》。

**注：**不对摘要表使用静态 SQL 或别名。

通常，若一个摘要表或复制的摘要表的隔离级别高于或等于一个查询的隔离级别，则使用该摘要表或复制的摘要表来优化该查询。例如，若一个查询在游标稳定性 (CS) 隔离级别下运行，则只使用在 CS 或更高的隔离级别下定义的摘要表和复制的摘要表进行优化。

要创建摘要表，须使用 CREATE SUMMARY TABLE 语句，并带上 AS *fullselect* 子句和 IMMEDIATE 或 REFRESH DEFERRED 选项。

您可以选择唯一地标识摘要表的列名。列名列表所包含的名称数必须与全查询的结果表中的列数相等。若全查询的结果表带有重复的列名或带有未命名的列，则必须给出列名列表。未命名的列是从使用选择列表的 AS 子句时未命名的常量、函数、表达式或设置操作生成的。若不指定列名列表，则表列继承全查询的结果集的列名。

当创建摘要表时，有一个选项可以指定当更改基表时是否自动刷新摘要表，或是否使用 `REFRESH TABLE` 语句刷新它。要在更改基表时自动刷新摘要表，指定 `REFRESH IMMEDIATE` 关键字。在下列情况下，最好进行立即刷新：

- 当对基表运行查询时，所花时间很长
- 基表不经常更改
- 刷新成本不高。

在这种情况下，摘要表可提供预先计算的结果。若想要延迟摘要表的刷新，指定 `REFRESH DEFERRED` 关键字。使用 `REFRESH DEFERRED` 指定的摘要表将不反映对基础基表的更改。若没有这个要求，应使用摘要表。例如，若运行 `DSS` 查询，应使用摘要表来包含旧数据。

在下列情况下，使用 `REFRESH DEFERRED` 定义的摘要表可用来代替查询：

- 除下列情况外，符合立即刷新摘要表的全查询的限制：
  - 不需要 `SELECT` 列表来包括 `COUNT(*)` 或 `COUNT_BIG(*)`
  - `SELECT` 列表可包括 `MAX` 和 `MIN` 列函数
  - 允许 `HAVING` 子句。

将 `SQL` 专用寄存器 `CURRENT REFRESH AGE SQL` 设置为 `ANY` 或具有值 `9999999999999999`。这串由 `9` 组成的数是此专用寄存器中允许的最大值，该值是数据类型为 `DECIMAL(20,6)` 的时间戳记持续时间值。

**注：**用 `REFRESH DEFERRED` 定义的摘要表不用于优化静态 `SQL`。

使用 `CURRENT REFRESH AGE` 专用寄存器指定有延时刷新功能的摘要表在必须刷新之前可用于动态查询的时间。要设置 `CURRENT REFRESH AGE` 专用寄存器的值，可使用 `SET CURRENT REFRESH AGE` 语句。有关 `CURRENT REFRESH AGE` 专用寄存器和 `SET CURRENT REFRESH AGE` 语句的更多信息，参考 *SQL Reference*。

用 `REFRESH IMMEDIATE` 定义的摘要表适用于静态和动态查询，且不需要使用 `CURRENT REFRESH AGE` 专用寄存器。

**注：**设置 `CURRENT REFRESH AGE` 专用寄存器为一个非零的值时应小心。若允许使用不能表示基础基表值的摘要表来优化查询的处理，则查询的结果就不能准确地表示基础表中的数据。当知道基础数据未更改时，或者当根据对数据的了解愿意接受结果中的错误程度时，这可能是合理的。

由于活动影响源数据，因此经过一段时间的摘要表将不再包含准确的数据。将需要使用 `REFRESH TABLE` 语句。有关更多信息，参考 *SQL Reference*。

若要以任何有效的全查询为基础创建一个新基表，则在创建该表时指定 `DEFINITION ONLY` 关键字。当创建表操作完成时，不将新表作为摘要表而是作为基表来处理。例如，可以创建用于 `LOAD` 和 `SET INTEGRITY` 的异常表，如下所示：

```
CREATE TABLE XT AS
  (SELECT T.*, CURRENT_TIMESTAMP AS TIMESTAMP,CLOB(" ,32K)
  AS MSG FROM T) DEFINITION ONLY
```

以下是关于摘要表的一些关键限制：

1. 不能改变摘要表。
2. 若基表具有摘要表，则不能改变基表的列长度。
3. 不能将数据导入摘要表。
4. 不能在摘要表上创建唯一索引。
5. 不能根据引用一个或多个别名的查询的结果创建摘要表。

有关摘要表限制的完整说明，参考 *SQL Reference*。

---

## 创建别名

别名是引用表、别名或视图的间接方法，这样 `SQL` 语句可与该表或视图的限定名无关。仅当表名或视图名更改的情况下，才必须更改别名定义。可以在一个别名上创建另一个别名。别名可以在视图或触发器定义以及任何 `SQL` 语句中使用，但表检查约束定义除外，在该定义中可以引用现存的表名或视图名。

别名可以在任何可使用现存表名的地方使用，且在别名链中不存在循环引用或重复引用的情况下，可以引用另一个别名。

别名不能与现存的表、视图或别名同名，而只能引用同一个数据库中的一个表。在 `CREATE TABLE` 或 `CREATE VIEW` 语句中使用的表或视图的名称不能与相同模式中的别名相同。

除非别名所处的模式不是您当前的授权标识所拥有的模式（它需要 `DBADM` 权限），否则，创建别名不需要特殊权限。

可以为定义时不存在的表、视图或别名定义别名。但是，当编译包含该别名的 `SQL` 语句时，它必须存在。

当删除一个别名或别名引用的对象时，从属于该别名的所有软件包就会标记为无效，而从属于该别名的所有视图和触发器则标记为不起作用。

要使用“控制中心”来创建别名:

1. 展开对象树, 直到您看到**别名**文件夹为止。
2. 右键单击**别名**文件夹, 并从弹出菜单中选择**创建**。
3. 填写信息, 并单击**确定**。

要使用命令行来创建别名, 输入:

```
CREATE ALIAS <alias_name> FOR <table_name>
```

在编译语句时, 别名被表名或视图名替换。若别名或别名链不能被解析为表名或视图名, 则将导致错误。例如, 若 **WORKERS** 是 **EMPLOYEE** 的一个别名, 则在编译时:

```
SELECT * FROM WORKERS
```

就会使以下语句生效

```
SELECT * FROM EMPLOYEE
```

下列 SQL 语句为 **EMPLOYEE** 表创建别名 **WORKERS**:

```
CREATE ALIAS WORKERS FOR EMPLOYEE
```

**注:** DB2 MVS/ESA 版使用两种不同概念的别名: **ALIAS** 和 **SYNONYM**。这两种概念在 DB2 通用数据库中是有区别的, 如下所示:

- DB2 MVS/ESA 版中的 **ALIAS**:
  - 要求它们的创建者具有特殊的权限或特权
  - 不能引用其他别名。
- DB2 MVS/ESA 版中的 **SYNONYM**:
  - 只能被它们的创建者使用
  - 始终是非限定的
  - 删除引用的表时被删除
  - 不与表或视图共享名称空间。

---

## 创建包装器

在联合数据库中, **CREATE WRAPPER** 语句注册一个包装器。该语句定义联合服务器与特定类别的数据源交互作用的原理。

特定的库必须用于特定的数据源类型、版本、通信协议和操作系统。例如，对于在使用 APPC 通信的 Windows NT 操作系统上运行的联合数据库，可使用 libdrda.dll 库访问 AS/400 和 DB2 OS/390 版数据源。

必须对联合数据库具有 SYSADM 或 DBADM 权限，才可使用 CREATE WRAPPER 语句。

从“控制中心”或从命令行处理器创建包装器都将对联合数据库注册该包装器。

要使用“控制中心”来创建包装器：

1. 展开对象树，直到您看到**联合数据库对象**文件夹为止。
2. 右键单击**联合数据库对象**文件夹，并从弹出菜单中选择**创建包装器**。
3. 填写信息，并单击**确定**。

要使用命令行来创建包装器，输入：

```
CREATE WRAPPER <wrapper_name> LIBRARY '<library_name>'
```

以下 SQL 语句在一个 Windows NT 操作系统上注册包装器 ORACLE8：

```
CREATE WRAPPER ORACLE8 LIBRARY 'libnet8.dll'
```

有关使用 CREATE WRAPPER 语句的详细信息，参考 *SQL Reference*。

---

## 创建服务器

在联合数据库中，创建服务器以便向 DB2 定义数据源并描述它们的特征：名称、包装器、类型、版本、位置和选项。此信息用于将别名映射为特定的数据管理系统，并向 DB2 优化器提供信息。服务器信息位于 SYSCAT.SERVERS 和 SYSCAT.SERVEROPTIONS 目录视图中。

**注：**在本节中，服务器表示数据源，而不是 DRDA 服务器或 DB2 服务器。要访问其他数据源（例如，Oracle），需要 DB2 Connect。

仅当创建了包装器之后，才可创建服务器对象。

必须对联合数据库具有 SYSADM 或 DBADM 权限，才可使用此语句。

可在 DB2 和数据源服务器之间创建用户映射，来管理认证处理中的差异。在第207页的『用户映射』中详细讨论了用户映射。

当删除一个服务器时，也会删除所有依赖于该服务器的对象（如用户映射、别名、函数映射、类型映射和计划）。

创建服务器时要提供服务器选项。这些选项包含有关该服务器的必需详细信息（如节点名）。服务器选项也可设置特定的性能和安全性值。

可从“控制中心”或命令行处理器创建服务器。

要使用“控制中心”创建服务器：

1. 展开对象树，直到您在**联合数据库对象**文件夹下面看到**服务器**文件夹为止。
2. 右键单击**服务器**文件夹，并从弹出菜单中选择**创建服务器**。
3. 填写信息，并单击**确定**。

要使用命令行来创建服务器，输入：

```
CREATE SERVER <server_name> TYPE <server_type>
      VERSION <server_version> WRAPPER <wrapper_name>
      OPTIONS (<server_option_name> <string_constant>)
```

下面的样本 SQL 语句创建 Oracle 服务器 ORA8：

```
CREATE SERVER ORA8 TYPE ORACLE VERSION 8 WRAPPER ORACLE8 OPTIONS
(NODE 'ONODE')
```

下面的样本 SQL 语句创建 DB2 服务器 DB2TEST：

```
CREATE SERVER DB2TEST TYPE DB2 VERSION 6.1 WRAPPER DB2UDB OPTIONS
(NODE 'DB2TEST', DBNAME 'TEST1')
```

在 SERVER SQL 语句中，NODE 的定义随数据源的不同而改变。若数据源是 DB2 DBMS，则该值是指有一个或多个数据库的 DB2 实例。在上一个示例中，注意 DBNAME 选项指定数据库名。若数据源是 DB2 OS/390 版 DBMS，则该值是指 DB2 OS/390 版系统的 LOCATION 名。若数据源是 Oracle DBMS，则不需要 DBNAME 选项，因为 Oracle 实例只包含一个数据库。

关于 CREATE SERVER 语句语法的附加详细信息，参考 *SQL Reference*。有关使用 CREATE SERVER 语句的附加详细信息，参考《*安装和配置补遗*》。

## 使用服务器选项来帮助定义数据源并简化认证处理

可给称为服务器选项的变量设置值，来影响联合服务器访问数据源的方式。本节：

- 说明服务器选项的用途
- 描述可使用哪些 SQL 语句来指定服务器选项
- 显示服务器选项及其设置

## 服务器选项的用途

一般情况下，可使用服务器选项：

- 提供和更新有关数据源的信息。服务器引用包括有关数据源的基本信息 - 例如，它的名称 - 以及可能随时间更改的信息。某些变化的信息通过赋予服务器选项的值来传送。例如，赋予 `cpu_ratio` 选项的值指示该数据源的 CPU 是比 DB2 系统 CPU 快还是慢。若 DB2 系统进行了一次或多次处理器升级，此值应更改。
- 简化认证。可设置某些服务器选项，以确保将用户标识和密码以正确的字体发送到数据源。例如，可设置 `fold_id` 选项，以便在联合服务器将用户标识发送到数据源之前，联合服务器将该名称转换为数据源需要的字体（大写或小写）。或者，若用需要的字体向联合服务器定义了用户标识，可设置 `fold_id` 选项以阻止服务器尝试更改字体并在该过程中消耗额外开销。
- 优化查询。某些服务器选项和它们的值可简化优化过程。例如：在 `CREATE SERVER` 语句中，可将特定的性能统计信息指定为选项值。具体地说，可将 `cpu_ratio` 选项设置为指示数据源 CPU 和联合服务器 CPU 的相对速度的值。而且，可将 `io_ratio` 选项设置成指示数据源和联合服务器的 I/O 设备的相对速率的值。当运行 `CREATE SERVER` 时，会将这些统计信息添加到目录视图 `SYSCAT.SERVEROPTIONS` 中，优化器在制定数据源的访问计划时将参考这些信息。若统计信息更改（例如，若升级了数据源 CPU，则这种情况可能会发生），可使用 `ALTER SERVER` 语句用这个更改更新 `SYSCAT.SERVEROPTIONS`。这样，优化器在制定该数据源的下一个访问计划时可使用这些更新。

## 用于服务器选项的 SQL

可在三条 SQL 语句中给服务器选项指定值：`CREATE SERVER`、`ALTER SERVER` 和 `SET SERVER OPTION`。

使用 `CREATE SERVER` 语句将一个选项设置为一个值，以无限期保持与一个数据源的多个连接。使用此语句，可将选项设置为与缺省值不同的值，或者，若选项没有缺省值，可将它设置为初始值。

若在使用 `CREATE SERVER` 语句将一个服务器选项设置为一个值后，要将其设置为另一个值以保持多个连接，则使用 `ALTER SERVER` 语句。

在与某个数据库的单个连接期间，可使用 `SET SERVER OPTION` 语句临时更改服务器选项的值。`SET SERVER OPTION` 语句必须在与数据源连接后的第一个工作单元内首先发出。

例如，要临时允许对 Oracle 服务器 `ORASEB1` 使用计划提示，可发出以下语句：

```
SET SERVER OPTION plan_hints TO 'Y' FOR SERVER ORASEB1
```

## 服务器选项及其设置

下面的表描述了服务器选项和可设置成的值。除非另有声明，否则所有服务器选项的值必须用单引号引起来。

表 2. 服务器选项及其设置

选项	有效设置	缺省设置
collating_sequence	<p>指定数据源是否使用与联合数据库相同的缺省整理顺序，这基于代码集和国家或地区信息。若数据源的整理顺序与 DB2 的整理顺序不同，则不能在数据源处对取决于 DB2 整理顺序的大多数运算进行远程求值。例如，在一个具有不同整理顺序的数据源处对别名字符列执行 MAX 列函数。因为若是在远程数据源对 MAX 函数求值结果可能会不同，因此 DB2 将在本地执行聚集运算和 MAX 函数。</p> <p>若查询包含等号，即使整理顺序不同（设置为 'N'），也可能将那部分查询下推。例如，可能将谓词 C1 = 'A' 下推至某个数据源。当然，当数据源处的整理顺序不区分大小写时，不能下推这类查询。当数据源不区分大小写时，C1= 'A' 和 C1 = 'a' 的结果是相同的，这在区分大小写的环境 (DB2) 中是无法接受的。</p> <p>管理员可创建具有特定整理顺序的联合数据库，以使该顺序与数据源整理顺序匹配。若所有数据源使用相同整理顺序，或者若大多数或所有列函数都应用于使用相同整理顺序的数据源，则此方法可提高性能。</p> <p>'Y' 数据源的整理顺序与联合数据库的整理顺序相同。</p> <p>'N' 数据源的整理顺序与联合数据库的整理顺序不同。</p> <p>'I' 数据源的整理顺序与联合数据库的整理顺序不同，且不区分大小写（例如，会认为 'TOLLESON' 和 'ToLLESon' 是相同的）。</p>	'N'
comm_rate	指定联合服务器及其相关数据源之间的通信速率。以每秒兆字节表示。	'2.0'
connectstring	指定与 OLE DB 提供者连接所需的初始化特性。有关连接字符串完整的语法和语义，参阅 <i>Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK, Microsoft Press, 1998</i> 中的“OLE DB 核心组件的 DataLinks API”。	无
cpu_ratio	指示数据源的 CPU 比联合服务器的 CPU 快多少或慢多少。值 '1.0' 表示数据源的 CPU 和联合服务器的 CPU 之间的速度相等。值 <1.0 表示数据源的 CPU 较慢。值 >1.0 表示数据源的 CPU 较快。	'1.0'



表 2. 服务器选项及其设置 (续)

选项	有效设置	缺省设置
dbname	您希望联合服务器访问的数据源数据库的名称。此参数是 DB2 系列数据源所必需的；它不适用于 Oracle** 数据源。	无。
fold_id (参见此表末尾的注释 1 和 4。)	<p>应用于用户标识，联合服务器将它们发送到数据源以便认证。有效的值为：</p> <p>'U' 联合服务器在将用户标识发送到数据源之前要将它转换为大写。这是 DB2 系列和 Oracle** 数据源的逻辑选项（参见此表末尾的注释 2）。</p> <p>'N' 联合服务器在将用户标识发送到数据源之前不对它执行任何操作。（参见此表末尾的注释 2）。</p> <p>'L' 联合服务器在将用户标识发送到数据源之前要将它转换为小写。</p> <p>若未使用这些设置，联合服务器会尝试将用户标识以大写形式发送到数据源。若该用户标识失败，服务器再尝试以小写形式发送它。</p>	无。
fold_pw (参见此表末尾的注释 1、3 和 4。)	<p>应用于密码，联合服务器将它们发送到数据源以便认证。有效的值为：</p> <p>'U' 联合服务器在将密码发送到数据源之前要将它转换为大写。这是 DB2 系列和 Oracle** 数据源的逻辑选项。</p> <p>'N' 联合服务器在将密码发送到数据源之前不对它执行任何操作。</p> <p>'L' 联合服务器在将密码发送到数据源之前要将它转换为小写。</p> <p>若未使用这些设置，联合服务器会尝试将密码以大写形式发送到数据源。若该密码失败，服务器再尝试以小写形式发送它。</p>	无。
io_ratio	指示数据源的 I/O 系统比联合服务器的 I/O 系统快多少或慢多少。值 "1.0" 表示数据源的 CPU 和联合服务器的 CPU 之间的速度相等。值 <1.0 表示数据源的 CPU 较慢。值 >1.0 表示数据源的 CPU 较快。	'1.0'

表 2. 服务器选项及其设置 (续)

选项	有效设置	缺省设置
node	<p>将数据源定义为 RDBMS 的实例所用的名称。它是所有数据源必需的参数。</p> <p>对于 DB2 系列数据源, 此名称是在联合数据库的 DB2 节点目录中指定的节点。要查看此目录, 发出 <b>db2 list node directory</b> 命令。</p> <p>对于 Oracle** 数据源, 此名称是在 Oracle** tnsnames.ora 文件中指定的服务器名称。要在 Windows NT 平台上访问此名称, 指定 Oracle** SQL Net Easy 配置工具的<b>查看配置信息</b>选项。</p>	无。
password	<p>指定是否将密码发送至数据源。</p> <p>'Y' 总是将密码发送至数据源, 并验证它们。这是缺省值。</p> <p>'N' 不将密码发送至数据源 (无论是否存在任何用户映射), 也不验证密码。</p> <p>'ENCRYPTION' 总是以加密形式将密码发送至数据源, 并验证它们。只对支持加密密码的 DB2 系列数据源有效。</p>	'Y'
plan_hints	<p>指定是否启用计划提示。计划提示是为数据源优化器提供额外信息的语句段。对于特定查询类型, 此信息可改进查询性能。计划提示可帮助数据源优化器决定是否使用索引、使用哪个索引或使用哪种表连接顺序。</p> <p>'Y' 若数据源支持计划提示, 则在数据源处启用计划提示。</p> <p>'N' 在数据源处不启用计划提示。</p>	'N'
pushdown	<p>'Y' DB2 将考虑允许数据源求值运算。</p> <p>'N' DB2 只检索远程数据源中的列, 而且不允许数据源对其他运算 (如连接) 求值。</p>	'Y'
varchar_no_trailing_blanks	<p>指定此数据源是否使用非空格填充 VARCHAR 比较语义。对于不包含尾部空格的变长字符串, 某些 DBMS 的非空格填充比较语义学会返回与 DB2 的比较语义学相同的结果。若确认在一个数据源中的所有 VARCHAR 表 / 视图列都不包含尾部空格, 考虑对数据源将此服务器选项设置为 'Y'。此选项经常用于 Oracle** 数据源。确保考虑可能会有别名的所有对象 (包括视图)。</p> <p>'Y' 此数据源的非空格填充比较语义学与 DB2 的类似。</p> <p>'N' 此数据源的非空格填充比较语义学与 DB2 的不同。</p>	'N'

有关此表的注释:

1. 无论为认证指定了什么值，都要应用此字段。
2. 因为 DB2 以大写形式存储用户标识，因此值 ‘N’ 和 ‘U’ 在逻辑上是等效的。
3. 当密码的设置为 ‘N’ 时，fold\_pw 的设置就没有作用。因为没有发送密码，因此大小写不会带来影响。
4. 避免其中任何一个选项的设置是空值。空设置看起来好象很有用，因为 DB2 将进行多次尝试来分辨用户标识和密码；但性能可能会降低（DB2 可能将用户标识和密码发送四次，然后才成功地通过数据源认证）。

### 使用与服务器的联通会话

联通会话允许应用程序使用服务器的本机客户机访问方法和本地 SQL 语言，来直接与服务器通信。

在下列情况下，联通会话很有用:

- 应用程序必须在数据源创建对象或执行 INSERT、UPDATE 或 DELETE 操作
- DB2 不支持唯一的数据源操作

当在联通会话中引用对象时，使用该对象的真实名称（不是别名）。

使用 SET PASSTHRU 语句来启动联通会话并直接访问服务器。必须动态发出此语句。此语句的一个示例为:

```
SET PASSTHRU BACKEND
```

它打开与数据源 BACKEND 的一个联通会话。

有关 SET PASSTHRU 和联通会话中 SQL 处理的更多信息，参阅 *SQL Reference*。

---

## 创建别名

在联合数据库中，别名是数据源表、别名和视图的标识符。分布式请求一般引用别名，而不是数据源表或视图。

别名是 DB2 提供位置透明性所用的方法之一。别名依靠服务器定义中的数据源位置信息来查找并有效地访问数据源。例如，ALTER SERVER 语句可透明地更新所有用户和应用程序的服务器性能数据和版本信息，而不需要新别名或更改应用程序代码。

可在“控制中心”中或从命令行处理器创建别名。可为相同的数据源表或视图定义多个别名。

不能在静态 SQL 语句中使用别名。

在创建别名前，应在数据源运行 RUNSTATS 命令的等效命令，并更新数据源对象的统计信息。当在联合数据库目录中创建并存储别名时，会从数据源收集统计信息。此目录数据包括表和列的定义，以及索引定义和统计信息（若有的话）。

以下 SQL 语句创建别名 CUSTOMER:

```
CREATE NICKNAME CUSTOMER for OS390A.SHAWNB.CUSTLIST
```

必须对联合数据库具有 SYSADM 或 DBADM 权限，或必须具有数据库特权 IMPLICIT\_SCHEMA 或模式特权 CREATEIN（对当前模式），才可使用此语句。

有关使用 CREATE NICKNAME 语句的其他附加详细信息，参考 *SQL Reference*。

## 引用别名和数据源对象

引用数据源对象时一般使用定义的别名。唯一的例外是联通会话中的引用（有关更多信息，参见第139页的『使用与服务器的联通会话』）。例如，若为数据源表 DB2MVS1.PERSON.DEPT 定义别名 DEPT，则允许语句 SELECT \* FROM DEPT；不允许语句 SELECT \* FROM DB2MVS1.PERSON.DEPT。

## 使用别名和数据源对象

大多数实用程序命令

（LOAD、IMPORT、EXPORT、REORGCHK、REORGANIZE TABLE）都不支持别名。

支持 COMMENT ON；它更新联合数据库中的系统目录。

不支持对别名执行插入、更新和删除操作。

## 标识现存的别名和数据源

当创建了几个别名后，应使用以下信息来标识给定别名对应于哪个数据源，或标识给定数据源的所有别名。

### 标识别名及其数据源

本示例假设您知道别名 (PAYROLL)，也知道谁创建了它 (ACCTG)，但需要有关该数据源的附加信息。使用以下 SQL 语句来首先获得 PAYROLL 在其数据源 (SERVER) 所用的名称信息。

```
select option, setting
       from syscat.taboptions
       where tablename = 'PAYROLL'
          and tabschema = 'ACCTG'
          and option in ('SERVER','REMOTE_SCHEMA','REMOTE_TABLE');
```

此语句的回答集为 DB2\_MVS、FINANCE、DEPTJ35\_PAYROLL。现在知道 *PAYROLL* 是在服务器 DB2\_MVS 中由 FINANCE 拥有的表 DEPTJ35\_PAYROLL 的别名。可在后续的 SELECT 语句中使用此信息：

```
select option,setting
       from syscat.serveroptions
       where servername = 'DB2_MVS'
         and option in ('NODE', 'DBNAME');
```

此语句的回答集为 REGIONW 和 DB2MVSDB3。现在知道表 DEPTJ35\_PAYROLL 位于 REGIONW 节点上的 DB2MVSDB3 数据库中。

借助于此信息，可使用 LIST NODE DIRECTORY 命令来获得有关 REGIONW 节点的信息，如使用的通信协议和安全性类型。若该节点一直表示数据源而非 DB2 系列，则需要检查该数据源的配置文件以查找相似信息。例如，若该节点一直是 Oracle 数据源，可从 Oracle tnsnames.ora 文件获得相似的信息。

有关系统目录视图的详细信息，参考 *SQL Reference*。

### 标识 DB2 认识的所有别名

以下 SQL 语句提供联合数据库认识的所有别名的列表，包括每个别名的模式名和远程服务器。

```
select tabname,tabschema, setting as remote_server
       from syscat.taboptions
       where option = 'SERVER';
```

---

## 创建索引、索引扩充或索引规范

索引是行位置的列表，按一个或多个指定列的内容来排序。索引通常用于加速对表的访问。但是，它们还可以为逻辑数据设计服务。例如，唯一索引不允许列中存在重复的值，从而保证了一个表中不会有两行相同。也可以创建索引，以指定一个列中值的升序或降序。

索引扩充是一个索引对象，它配合带有结构化类型或相异类型列的索引使用。

索引规范是一个元数据结构。它告诉优化器别名所引用的数据源对象（表或视图）是否存在索引。索引规范不包含行位置的列表 – 它只是索引的描述。优化器使用索引规范来改进对由别名表示的对象的访问。当第一次创建一个别名时，若数据源中的基表有索引且该索引使用 DB2 可识别的一种格式，则生成索引规范。

**注：**需要的话，根据表别名或视图别名（对于视图基于一个表的情况）创建索引规范。

在下列情况下手工创建索引或索引规范：

- 它将改进性能。例如，若要鼓励优化器使用特定的表或别名作为嵌套循环连接的内部表，且不存在索引的话，则基于该连接列创建索引规范。有关何时需要索引或索引规范的更多信息，参考《管理指南：性能》。
- 在创建基表的别名后，添加了该表的索引。

当基表上不存在任何索引时，可创建索引规范（当发出 `CREATE INDEX` 语句时，DB2 将不会检查是否存在远程索引）。即使指定了 `UNIQUE` 关键字，索引规范也不会强制行的唯一性。

“DB2 索引顾问”是一个向导，它辅助您选择一组最优的索引。可通过“控制中心”访问此向导。一个类似的实用程序称为 *db2advis*。

索引是由基表中的列定义的。它可以由表的创建者或知道某些列需要直接访问的用户来定义。除非已经存在用户定义索引，否则会根据主键来自动创建主索引键。

可在特定的基表上定义任意多个索引，且这些索引可对查询性能产生良好的影响。但是，存在的索引越多，在更新、删除和插入操作期间数据库管理器必须修改的索引就越多。为接收很多更新的一个表创建大量索引，可能减慢对请求的处理。因此，仅当频繁访问有明显有利之处时，才使用索引。

任何作为索引键一部分的列都限长 255 字节。

**注：** `DB2_INDEX_2BYTEVARLEN` 注册表变量可用来允许将长度大于 255 字节的列指定为索引键的一部分。

索引中的最大列数是 16。若是对类型表建立索引，则最大列数是 15。索引键的最大长度是 1024 字节。如前所述，表上的索引键太多会减慢请求的处理速度。同样，大型索引键也会减慢处理请求的速度。

索引键是定义了索引的一个列或一些列的集合，它决定索引的有用程度。虽然构成一个索引键的列的次序不会给索引键的创建带来影响，但是当它决定是否使用索引时就可能影响优化器。

若要编制索引的表是空的，则仍创建索引，但是在未装入该表或插入行之前，不会建立任何索引项。若该表不是空的，则数据库管理器在处理 `CREATE INDEX` 语句时会建立索引项。

对于群集索引，将新行实际插入具有相似键值的现存行附近。这可改善查询性能，因为它导致数据页的访问模式更线性化，且产生更有效的预取。

若要使主键索引成为群集索引，不应在 `CREATE TABLE` 中指定主键。一旦创建了主键，就不能修改相关的索引。而是执行不带主键子句的 `CREATE TABLE`。然

后，发出 `CREATE INDEX` 语句，并指定群集属性。最后，使用 `ALTER TABLE` 语句添加与刚创建的索引对应的主键。将把此索引用作主键索引。

通常，若群集索引是唯一的，则群集维护起来就更有效率。

不是唯一索引键的一部分但是要在该索引中存储 / 维护的列数据，称为包含列。只能为唯一索引指定包含列。当用包含列创建索引时，仅对唯一键列排序并考虑其唯一性。当涉及到索引访问时，包含列的使用可提高数据检索的性能。

数据库管理器使用 B+ 树结构来存储索引，该结构的底层由叶节点组成。叶节点或叶页是存储实际的索引键值之处。当创建索引时，可允许联机合并或重组那些索引叶页。联机索引重组用于防止以下情况：在进行大量删除和更新活动后，某个索引的大部分叶页只剩下少许索引键。在这种情况下，若不进行联机重组，那么这些空间只能通过数据和索引的脱机重组收回。当确定是否创建有能力联机重组索引页的索引时，您应考虑此问题：在每次删除键时，因为检查要合并的空间而增加的性能成本以及完成合并（若有足够的空间的话）的实际成本是否大于更好的索引空间利用所带来的好处，且小于执行脱机重组以收回空间这一减少了的需要？

**注：**在一次联机重组合并后释放的页只可用于同一个表中的其他索引。对于全面的重组，释放的那些页可用于其他对象（当使用“数据库管理的存储器”时）或磁盘空间（当使用“系统管理的存储器”时）。此外，联机重组将不释放索引的任何非叶页，而全面重组将使索引尽可能地小，以减少非叶页和叶页以及索引层数。

有关如何实现将联机重组的索引的更多信息，参见第145页的『使用 `CREATE INDEX` 语句』。

使用相同的 `CREATE INDEX` 语句来构建分区数据库中表的索引。根据该表的分区键来将它们分区。一个表上的索引由节点组中每个节点上该表的本地索引组成。注意，在一个多分区环境中定义的唯一索引必须是分区键的超集。

**性能提示：**若要执行以下一系列任务：

1. 创建表
2. 装入表
3. 创建索引
4. 执行 `RUNSTATS`

应该考虑按下列次序执行任务：

1. 创建表

2. 创建索引
3. 装入请求了 `statistics yes` 选项的表。

有关 `LOAD` 性能改进的更多信息，参考 *Data Movement Utilities Guide and Reference*。

在创建索引之后，维护它们。随后，当应用程序使用一个键值来随机访问和处理表中的行时，就可以使用基于该键值的索引来直接访问行。这很重要，因为一个基表中各行的物理存储器是无序的。当插入一行时，除非定义了一个群集索引，否则，将它置于可以容纳它的最方便的存储位置中。当搜索满足特定选择条件的表行且该表没有索引时，就会扫描整个表。索引优化了数据检索，而无需执行长时间的顺序搜索。

索引数据可以与表数据存储相同的表空间中，或存储在包含索引数据的单独表空间中。用于存储索引数据的表空间是在创建该表时确定的（参见第118页的『在多个表空间中创建一个表』）。

要使用“控制中心”来创建索引：

1. 展开对象树，直到您看到索引文件夹为止。
2. 右键单击索引文件夹，并从弹出菜单中选择 **创建** → **使用向导创建索引**。
3. 遵循向导中的步骤来完成该任务。

要使用命令行来创建索引，输入：

```
CREATE INDEX <name> ON <table_name> (<column_name>)
```

下面两节『使用索引』和第145页的『使用 `CREATE INDEX` 语句』提供有关创建索引的更多信息。

## 使用索引

应用程序永远不会直接使用索引。决定是否使用索引和要使用哪些潜在可用的索引是优化器的责任。

表上最好的索引具有下列特点：

- 使用高速磁盘
- 是高度群集的
- 由少数窄列组成

有关索引如何带来方便的详细讨论，参考《管理指南：性能》中的“索引扫描概念”。



## 使用 CREATE INDEX 语句

可以创建一个索引，它将允许重复值（非唯一索引）以便可按非主键的列来执行有效检索，也允许在构成索引的一列或多列中存在重复值。

下列 SQL 语句根据 EMPLOYEE 表中的 LASTNAME 列创建称为 LNAME 且按升序排序的非唯一索引：

```
CREATE INDEX LNAME ON EMPLOYEE (LASTNAME ASC)
```

以下 SQL 语句基于电话号码列创建唯一的索引：

```
CREATE UNIQUE INDEX PH ON EMPLOYEE (PHONENO DESC)
```

唯一索引确保在构成索引的一列或多列中不存在重复的值。在更新行或插入新行的 SQL 语句的结尾，实现该约束。若一个或多个列组成的集合已经有重复的值，则不能创建此类索引。

关键字 ASC 按列的升序放置这些索引项，而 DESC 按列的降序放置它们。缺省值为升序。

可以在两个列上创建一个唯一索引，两列其中之一是一个包含列。在非包含列上定义主键。两列都在目录中作为同一表上的主键显示。通常每个表只能有一个主键。

INCLUDE 子句指定将附加列附加到一组索引键列。使用此子句所包含的任何列未用来强制唯一性。包含的列可能会通过仅访问索引而改进某些查询的性能。这些列必须与用来强制唯一性的列区分开（否则将会接收到出错信息 SQLSTATE 42711）。对于列数和总长度属性的限制适用于唯一键和索引中的所有列。

执行检查以确定现有的索引是否与主键定义匹配（忽略索引中的任何 INCLUDE 列）。若索引定义标识同一组列，则索引定义匹配，而不考虑列的顺序或方向（升序或降序）规范。若找到了匹配的索引定义，则更改索引的描述以表示它是主索引（这是系统所要求的），并且若它是非唯一索引，则将其更改为是唯一的（在确保唯一性之后）。

这就是如目录中所表示的同一个表中可能有多个主键的原因。

当使用结构化类型时，可能需要创建用户定义的索引类型。这需要定义索引维护、索引搜索和索引利用功能。有关创建索引类型的要求信息，参考 *SQL Reference*。

以下 SQL 语句在 EMPLOYEE 表的 LASTNAME 列上创建一个聚类索引，称为 INDEX1：

```
CREATE INDEX INDEX1 ON EMPLOYEE (LASTNAME) CLUSTER
```

要更有效地使用数据库的内部存储器，通过与 ALTER TABLE 语句相关的 PCTFREE 参数来使用群集索引，以便可将新数据插入在正确的页面上。当将数据插入在正确的页面上时，群集次序保持不变。通常，表上的 INSERT 活动越多，为维护群集所需的（该表上的）PCTFREE 值也就越大。因为此索引确定数据在物理页上放置的次序，因此对任何特定的表只能定义一个群集索引。

另一方面，例如，若这些新行的索引键值总是新的大键值，则表的群集属性将尝试把它们放置在表的尾部。其他页上有空闲空间对保持群集没有什么作用。在这种情况下，将表设置为追加方式可能优于使用群集索引，改变表以拥有一个大的 PCTFREE 值。可发出如下命令来将表设置为追加方式：ALTER TABLE APPEND ON。有关 ALTER TABLE 的其他概述信息，参见第176页的『更改表属性』。有关 ALTER TABLE 的附加详细信息，参考 *SQL Reference*。

以上讨论也适用于增加行大小的 UPDATE 所导致的新的“溢出”行。

CREATE INDEX 语句的 MINPCTUSED 子句指定在索引叶页上最小已用空间量的阈值。若使用此子句，可对此索引启用联机索引重组。一旦被启用，可参照下列注意事项来确定是否执行联机重组：当从此索引的一个叶页中删除一个键后，且该页上已用空间的百分比小于指定的阈值，则检查相邻的索引叶页以确定是否可将两个叶页上的键合并到单个索引叶页中。

例如，以下 SQL 语句创建启用联机索引重组的索引：

```
CREATE INDEX LASTN ON EMPLOYEE (LASTNAME) MINPCTUSED=20
```

当从此索引删除一个键时，若该索引页上的其余键占用索引页上百分之二十或更小的空间，可尝试将此索引页的键与相邻索引页的键合并来删除该索引页。若组合的键可全部位于一页上，则执行此合并并删除其中一个索引页。

CREATE INDEX 语句的 PCTFREE 子句指定创建索引时每个索引页中要留作空闲空间的百分比。在索引页上保留更多的空闲空间将导致更少的页分割。这将减少为重新获得顺序索引页而重组表的需要，从而将增加预取。而预取是一个可提高性能的重要部分。再一次，若总是存在大键值，那么要考虑降低 CREATE INDEX 语句的 PCTFREE 子句的值。这样，每个索引页上保留的浪费空间会受到限制。

若有一个复制的摘要表，则其基表必须具有唯一索引，且该索引键列必须用在定义复制的摘要表的查询中。有关复制的摘要表的更多信息，参考《管理指南：计划》。

对于分区内并行性，通过使用多个处理器来处理创建索引期间执行的数据扫描和排序，以提高创建索引的性能。通过将 *intra\_parallel* 设置为 YES(1) 或 ANY(-1)，来允许使用多个处理器。在索引创建期间使用的处理器的数量由系统确定，不受

配置参数 *dft\_degree* 或 *max\_querydegree*、应用程序运行时等级或 SQL 语句的编译等级影响。若数据库配置参数 *indexsort* 是 NO，则索引创建将不使用多个处理器。

在多分区数据库中，必须将唯一索引定义为分区键的超集。

---

## 创建用户定义扩充索引类型

为了支持用户定义索引类型，“DB2 通用数据库”允许您对决定索引工作方式的主组件创建并应用您自己的逻辑。那些可以替换的组件包括：

- 索引维护。这使得可以将索引列内容映射至索引键。这样的映射是通过用户定义的映射函数完成的。扩充索引中只能有一个结构化类型列。与普通索引不同，扩充索引每行可以有多个索引项。每行有多个索引项允许文本文档作为特定对象存储，对于该对象，文档中的每个关键字都有单独的索引项。
- 索引利用。这使应用程序的设计者能够将过滤条件（范围谓词）与用户定义函数 (UDF) 相关联（否则，这些用户定义函数对优化器而言将是不透明的）。这使 DB2 能够避免对每一行进行单独的 UDF 调用，因而避免了客户机与服务器之间的上下文相关切换，从而大幅改进性能。

**注：**用户定义函数定义必须是确定的，并且一定不能允许外部操作，这样才能由优化器利用。

还可指定可选的数据过滤函数。在对用户定义函数求值之前，优化器对取到的元组使用过滤器。

只有结构化类型或单值类型列才能使用索引扩充来对这些对象创建用户定义扩充索引类型。用户定义扩充索引类型一定不能：

- 使用分群索引定义
- 带有 INCLUDE 列

## 有关索引维护的详细信息

通过 CREATE INDEX EXTENSION 语句定义构成索引操作的组件中的两个。

索引维护是这样的一个过程：将索引列内容（或源键）变换为目标索引键。变换过程由数据库中先前定义的表函数定义。

FROM SOURCE KEY 子句为此索引扩充所支持的源键列指定结构化数据类型或单值类型。给出了单个参数名和数据类型，它们与源键列相关联。

`GENERATE KEY USING` 子句指定用来生成索引键的用户定义表函数。必须在 `TARGET KEY` 子句规范中指定此函数的输出。也可将此函数的输出用作 `FILTER USING` 子句上指定的索引过滤函数的输入。

## 有关索引搜索的详细信息

索引搜索将搜索自变量映射至搜索范围。

`CREATE INDEX EXTENSION` 语句的 `WITH TARGET KEY` 子句指定目标键参数，它们是 `GENERATE KEY USING` 子句上指定的用户定义表函数的输出。给出了单个参数名和数据类型，它们与目标键列相关联。此参数与 `GENERATE KEY USING` 子句的用户定义表函数的 `RETURNS` 表的列相对应。

`SEARCH METHODS` 子句引入对该索引定义的一个或多个搜索方法。每个搜索方法都由一个方法名、搜索自变量、一个范围生成函数和一个可选的索引过滤函数组成。每个搜索方法都定义用户定义表函数生成下层用户定义索引的索引搜索范围的方法。此外，每个搜索方法都定义如何通过用户定义标量函数进一步限定特定搜索范围中的索引项，以返回单个值。

- `WHEN` 子句将一个标号与一个搜索方法相关联。此标号是一个 SQL 标识符，它与索引利用规则中指定的方法名（可在用户定义函数的 `PREDICATES` 子句中找到）相关。给出了一个或多个参数名和数据类型，用作范围函数和 / 或索引过滤函数中的自变量。`WHEN` 子句指定当 `CREATE FUNCTION` 语句的 `PREDICATES` 子句与入局查询相匹配时优化器可以执行的操作。
- `RANGE THROUGH` 子句指定一个用户定义外部表函数，该函数生成索引键范围。这使优化器能够在索引键落在键范围之外时避免调用相关联的 UDF。
- `FILTER USING` 子句是一种可选的指定用户定义外部表函数或情况表达式的方法，该函数或表达式用来过滤范围生成函数所返回的索引项。若索引过滤函数或情况表达式返回的值是 1，则从表中检索与索引项相对应的行。若返回的值不是 1，则废弃该索引项。当辅助过滤器的成本比求值原始方法的成本低，且辅助过滤器的选择性相对较低时，此功能非常有价值。

## 有关索引利用的详细信息

索引利用在对搜索方法进行求值时发生。

`CREATE FUNCTION`（外部标量）语句创建一个用户定义谓词，此谓词与对索引扩充定义的搜索方法配合使用。

`PREDICATES` 子句标识那些使用此函数并有可能利用索引扩充（且有可能使用该谓词的搜索条件的可选 `SELECTIVITY` 子句）的谓词。若指定 `PREDICATES` 子句，则必须将函数定义成 `DETERMINISTIC`，并指定 `NO EXTERNAL ACTION`。

- **WHEN** 子句引入谓词中使用比较运算符 (=、>、< 及其他) 和常量或表达式 (使用 **EXPRESSION AS** 子句) 定义的函数的特定使用。当谓词将此函数与同一比较运算符和给定的常量或表达式配合使用时, 可使用过滤和索引利用。允许使用常量主要是为了包括布尔表达式 (其结果类型是 1 或 0)。对于所有其他情况, 最好使用 **EXPRESSION AS** 子句。
- **FILTER USING** 子句标识可用来对结果表执行附加过滤的过滤函数。这是已定义函数 (在谓词中使用) 的备用且更快速的版本, 它减少了为了确定行是否有资格而必须对其执行用户定义谓词的行数。万一索引生成的结果接近用户定义谓词所预期的结果, 则此过滤函数的应用程序便可能是多余的。
- 可选择对索引扩充的每个搜索方法定义一组规则, 来利用索引。也可以在索引扩充中定义一个搜索方法来描述搜索目标、搜索自变量以及如何使用它们来执行索引搜索。
  - **SEARCH BY INDEX EXTENSION** 子句标识索引扩充。
  - 可选的 **EXACT** 子句指示索引查找正位于其谓词求值之中。此子句告知数据库在索引查找之后不要应用用户提供的原始谓词函数或过滤函数。若不使用索引查找, 则必须应用原始谓词和过滤函数。若不使用 **EXACT** 子句, 则在索引查找之后, 应用用户提供的原始谓词。当索引查找返回与 **EXACT** 谓词相同的结果时, 该谓词非常有用。这防止查询执行对从索引查找获取的结果应用用户定义谓词。若只期望索引提供谓词的近似值, 则不要指定 **EXACT** 子句。
  - **WHEN KEY** 子句定义搜索目标。只可对一个键指定一个搜索目标。 **WHEN KEY** 子句后面给出的值标识正在定义的函数的参数名。当命名的参数的值是那些根据指定的索引扩充被索引包括的列时, 此子句求值为真。
  - **USE** 子句定义搜索自变量。搜索自变量标识将要使用索引扩充中定义的哪一种方法。此处给定的方法名必须与索引扩充中定义的方法相匹配。一个或多个参数值标识正在定义的函数的参数名, 这些参数一定不能与搜索目标中指定的任何参数名相同。参数值的数目和每个参数的数据类型必须与对索引扩充中该方法定义的参数相匹配。内部和单值数据类型必须精确匹配, 且必须在相同的结构化类型中。

## 定义索引扩充的一个方案

下面是一个定义索引扩充的方案:

1. (为形状) 定义结构化类型。使用 **CREATE TYPE** 语句来定义一个类型层次结构, 形状是超类型, 空形状、点、线和多边形是子类型。这些结构化类型建立空间实体的模型。例如, 商店的位置是一个点, 河流的路径是一条线; 而商业区的边界是一个多边形。最小定界矩形 (**mbr**) 是一个属性。 **gtype** 属性标识相

关联的实体是一个点、一条线还是一个多边形。地理边界通过 `numpart`、`numpoint` 和 `geometry` 属性建立模型。所有其他属性都被忽略，因为此方案对它们不感兴趣。

2. 创建索引扩充。

- 使用 `CREATE FUNCTION` 语句来创建用于下列各项的函数：键变换 (`gridentry`)、范围生成 (`gridrange`) 和索引过滤 (`checkduplicate` 和 `mbroverlap`)。
- 使用 `CREATE INDEX EXTENSION` 语句来创建所需的其余索引组件。

3. 创建与索引的索引维护组件相对应的键变换。

```
CREATE INDEX EXTENSION iename (parm_name datatype, ...)
FROM SOURCE KEY (parm_name datatype)
GENERATE KEY USING table_function_invocation
...
```

`FROM SOURCE KEY` 子句标识键变换的参数和数据类型。`GENERATE KEY USING` 子句标识一个特定函数，该函数用来映射带有从该函数生成的值的源键。

4. 定义与索引的索引搜索组件相对应的范围生成与索引过滤函数。

```
CREATE INDEX EXTENSION iename (parm_name datatype, ...)
...
WITH TARGET KEY
  WHEN method_name (parm_name datatype, ...)
  RANGE THROUGH range_producing_function_invocation
  FILTER USING index_filtering_function_invocation
```

`WITH TARGET KEY` 子句标识搜索方法定义。`WHEN` 子句标识方法名。`RANGE THROUGH`子句标识用来限制要使用的索引作用域的函数。`FILTER USING` 子句标识一个特定函数，该函数用来从生成的索引值中消去不必需的项目。

注: `FILTER USING` 子句可以标识一个情况表达式，而不是索引过滤函数。

5. 定义谓词来利用索引扩充。

```
CREATE FUNCTION within (x shape, y shape)
RETURNS INTEGER
...
PREDICATES
  WHEN = 1
  FILTER USING mbrWithin (x..mbr..xmin, ...)
  SEARCH BY INDEX EXTENSION grid_extension
  WHEN KEY (parm_name) USE method_name(parm_name)
```

`PREDICATES` 子句引入一个或多个谓词，这些谓词以 `WHEN` 子句开始。`WHEN` 子句以一个比较运算符开始指定谓词，然后指定常量或 `EXPRESSION AS` 子句。`FILTER USING` 子句标识可用来对结果表执行附加过滤的过滤函数。

这是已定义函数（在谓词中使用）的廉价版本，它减少了为了确定有资格的行而必须对其执行用户定义谓词的行数。 **SEARCH BY INDEX EXTENSION** 子句指定发生索引利用的位置。索引利用使用可用来利用索引的索引扩充的搜索方法来定义一组规则。**WHEN KEY** 子句指定利用规则。利用规则描述搜索目标和搜索自变量以及如何通过搜索方法使用它们来执行索引搜索。

#### 6. 定义过滤器函数。

```
CREATE FUNCTION mbrWithin (...)
```

创建此处定义的函数是为了在索引扩充的谓词中使用。

为了使查询优化器能够成功地利用创建的索引来改进查询性能，在函数调用上提供了 **SELECTIVITY** 选项。在您对谓词可能返回的行的百分比有一定了解的情况下，可在函数调用上使用 **SELECTIVITY** 选项来帮助 DB2 优化器选择更有效的访问路径。

在以下示例中，**within** 用户定义函数计算中心和半径（分别根据第一个和第二个参数），并用适当的选择性构建语句字符串：

```
SELECT * FROM customer
WHERE within(loc, circle(100, 100, 10)) = 1 SELECTIVITY .05
```

在此示例中，指示的谓词 (**SELECTIVITY .05**) 过滤掉 **customer** 表中 95% 的行。





---

## 第4章 改变数据库

本章主要讨论在改变数据库之前必须考虑的事项；以及如何改变或删除数据库对象。

---

### 在改变数据库之前

在实现了一个数据库设计之后的某个时间，可能需要更改数据库设计。您应重新考虑以前设计中的主要设计问题。您应特别注意下列各项：

- 『更改逻辑设计和物理设计特征』
- 『更改许可证信息』
- 『更改实例』
- 第156页的『更改环境变量和简要表注册表变量』
- 第157页的『更改节点配置文件』
- 第157页的『更改数据库配置』

### 更改逻辑设计和物理设计特征

在进行影响整个数据库的更改之前，应该复查所有逻辑和物理设计决策。例如，当改变表空间时，应该复查关于 SMS 或 DMS 存储器类型使用的设计决策。（有关更多信息，参考《管理指南：计划》。）

### 更改许可证信息

在管理 DB2 产品的许可证期间，可能发现需要增加许可证数。可使用“控制中心”中的“许可证中心”来检查已安装产品的使用情况，并根据该使用情况来增加许可证数。

### 更改实例

实例都设计为尽可能与以后产品的安装和除去所产生的影响无关。

在大多数情况下，现存实例自动继承或失去要安装或除去的产品功能的访问权。但是，若安装或除去了特定的可执行文件或组件，则现存实例不会自动继承新的系统配置参数或获得所有附加功能的访问权。必须更新该实例。

若通过安装“程序临时修订” (PTF) 或补丁程序来更新 DB2，则应使用 **db2iupdt** 命令更新所有现存的 DB2 实例。还应使用 **dasiupdt** 命令更新“管理服务器” (DAS)。

在尝试更改或删除实例前，应确保了解那些实例和实例中已有的数据库分区服务器。

## 列出实例

要使用“控制中心”获得系统上可用的所有实例的列表：

1. 展开对象树，直到您看到**数据库**文件夹为止。
2. 右键单击要列出其实例的数据库，并从弹出菜单中选择**添加**。
3. 单击**刷新**，并单击**数据库名**字段末尾的箭头来查看实例列表。
4. 按**取消**。

要使用命令行获得系统上可用的所有实例的列表，输入：

```
db2ilist
```

要确定哪一个实例适用于当前会话（在 OS/2 或受支持的 Windows 平台上），使用：

```
set db2instance
```

## 更新实例配置

运行 **db2iupdt** 命令，并执行以下操作来更新指定的实例：

- 替换实例所有者主目录下 **sqllib** 子目录中的文件。
- 若更改了节点类型，则会创建一个新的数据库管理器配置文件。为此，可将现存的数据库管理器配置文件的相关值与新节点类型的缺省数据库管理器配置文件合并。若创建了一个新的数据库管理器配置文件，则将旧文件备份到实例所有者主目录下的 **sqllib** 子目录的 **backup** 子目录中。

可在正确的版本和发行版子目录（准确名称随操作系统的不同而改变）下的 **instance** 子目录中找到 **db2iupdt** 命令。

按如下所示使用该命令：

```
db2iupdt InstName
```

**InstName** 是实例所有者的登录名。

此命令还有其他可选参数：

- **-h** 或 **-?**  
显示此命令的帮助菜单。
- **-d**  
设置在问题确定期间要使用的调试方式。

- **-a AuthType**

指定实例的认证类型。有效的认证类型是 **SERVER**、**CLIENT**、**DCS** 或 **DCE**。若未指定此参数，且若已安装了 **DB2** 服务器，则缺省值为 **SERVER**。否则，将它设置为 **CLIENT**。该实例的认证类型适用于实例拥有的所有数据库。

在 **UNIX** 操作系统上，**DCE** 不是有效的认证类型。

- **-e**

允许更新存在的每个实例。可以使用 **db2ilist** 来显示存在的实例。

- **-u FencedID**

命名受防护的用户定义函数 (UDF) 和存储过程执行期间所归属的用户。若安装了 **DB2** 客户机或“**DB2** 软件开发工具箱”，则不需要这样做。对于其他 **DB2** 产品，它是必需的参数。

**注：**FencedID 不能是 “root” 或 “bin”。

- **-k**

此参数保留当前实例类型。若不指定此参数，当前实例将按以下次序升级到可用的最高实例类型：

- 具有本地和远程客户机的分区数据库服务器（“**DB2** 扩充企业版”缺省实例类型）
- 具有本地和远程客户机的数据库服务器（“**DB2** 通用数据库企业版”缺省实例类型）
- 客户机（**DB2** 客户机缺省实例类型）

示例：

- 若在创建实例后安装了“**DB2** 通用数据库工作组版”或“**DB2** 通用数据库企业版”，可输入以下命令来更新该实例：

```
db2iupdt -u db2fenc1 db2inst1
```

- 若在创建实例后安装了“**DB2** Connect 企业版”，也可将该实例用作 Fenced ID：

```
db2iupdt -u db2inst1 db2inst1
```

- 要更新客户机实例，可使用以下命令：

```
db2iupdt db2inst1
```

## 除去实例

使用“控制中心”来除去实例:

1. 展开对象树，直到您看到要除去的实例为止。
2. 右键单击该实例名，并从弹出菜单中选择**除去**。
3. 选择**确认框**，并单击**确定**。

要使用命令行除去实例，输入:

```
db2idrop <instance_name>
```

使用命令行除去实例的准备工作和详细信息包括:

1. 停止当前使用该实例的所有应用程序。
2. 在每个 DB2 命令窗口中，运行 **db2 terminate** 命令来停止命令行处理器。
3. 运行 **db2stop** 命令来停止该实例。
4. 备份由 DB2INSTPROF 注册表变量指示的实例目录。在 UNIX 操作系统上，请考虑备份 INSTHOME/sql1lib 目录中的文件（其中 INSTHOME 是实例所有者的主目录）。例如，可能想保存数据库管理器配置文件 db2system、db2nodes.cfg 文件、用户定义函数 (UDF) 或受防护的存储过程应用程序。
5. （仅在 UNIX 操作系统上）作为实例所有者注销。
6. （仅在 UNIX 操作系统上）作为具有超级权限的用户登录。
7. 发出 **db2idrop** 命令:

```
db2idrop InstName
```

其中 InstName 是要删除的实例的名称。

此命令从实例列表中除去该实例项并除去该实例目录。

8. （仅在 UNIX 操作系统上）可选择作为具有超级权限的用户，除去该实例所有者的用户标识和组（若只用于该实例）。若计划重建该实例，则不要除去它们。

此步骤是可选的，因为实例所有者和实例所有者组可用于其他用途。

**db2idrop** 命令从实例列表中除去实例项，并除去实例所有者主目录下的 sql1lib 子目录。

## 更改环境变量和简要表注册表变量

必须考虑在特定操作系统上需要更改哪些环境变量（若存在的话）。若更改了任何环境变量且不在 UNIX 平台上，则需要重新启动系统，以使新的环境变量生效。在改变数据库之前，复查您是否应该复位“全局简要表”注册表中的简要表注册表

变量。然后将简要表注册表变量复位为最适合新数据库环境的那些值。若只更改了简要表注册表变量，则不需要重新启动系统。

## 更改节点配置文件

若计划更改任何节点组（添加或删除节点，或移动现存的节点），应参考《管理指南：性能》中的“通过添加处理器调整配置”以了解应执行哪些操作的详细信息。

## 更改数据库配置

若正在计划更改该数据库，应复查配置参数的值。根据数据库的使用方式更改数据库时，可经常调整某些参数值。

要更改数据库配置，使用“控制中心”中的“性能配置向导”。此向导通过建议修改某些配置参数并给它们提供建议值，来帮助调整性能和平衡每个实例中单个数据库的内存需求。

**注：**若修改任何参数，则在发生下列各项之前，不更新值：

- 对于数据库参数，在断开所有应用程序之后，与数据库建立第一个新的连接时。
- 对于数据库管理器参数，下一次停止和启动该实例时。

在大多数情况中，“性能配置向导”所建议的值都会提供比缺省值更好的性能，因为它们是根据有关工作负荷和特定服务器的信息确定的。但是，要注意这些值是为改进数据库系统的性能而设计的，并不一定能优化该系统。应该将它们当作一个起点，然后进一步调整来获得优化的性能。

要使用“控制中心”来更改数据库配置：

1. 展开对象树，直到您看到**数据库**文件夹为止
2. 右键单击要更改的实例或数据库，并从弹出菜单中选择**使用向导配置性能**。
3. 单击每一页并根据需要更改信息。
4. 单击**结果**页，复查您的工作并应用建议的任何配置参数。
5. 完成应用更新之后，单击**完成**。

要使用命令行更改数据库管理器配置，输入：

```
UPDATE DBM CFG FOR <database_alias>  
USING <config_keyword>=<value>
```

可以在单个命令中更新一个或多个 <config\_keyword>=<value> 组合。对数据库管理器配置文件的大多数更改只有在将它们装入内存之后才会生效。对于服务器配

置参数，这在运行 `START DATABASE MANAGER` 命令期间发生。对于客户机配置参数，这在重新启动应用程序时发生。

要查看或打印当前数据库管理器配置参数，使用 `GET DATABASE MANAGER CONFIGURATION` 命令。

有关如何通过基准测试来细化系统和配置系统的详细信息，参考《管理指南：性能》中的“基准测试”和“配置 DB2”。

**对于多分区：** 当有一个分布在多个分区上的数据库时，数据库配置文件在所有数据库分区上应是相同的。一致性是必需的，因为 SQL 编译程序根据节点配置文件中的信息来编译分布式 SQL 语句，并创建一个访问计划以满足 SQL 语句的需要。维护数据库分区上的不同配置文件可能产生不同的访问计划，这取决于准备该语句所在的数据库分区。使用 `db2_all` 来跨所有数据库分区维护配置文件。

---

## 改变数据库

改变数据库的任务几乎与创建数据库的任务一样多。这些任务更新或删除先前创建的数据库的部件。这些任务包括：

- 第159页的『删除数据库』
- 第159页的『改变节点组』
- 第159页的『改变表空间』
- 第164页的『删除模式』
- 第165页的『修改表的结构和内容』
- 第179页的『改变用户定义结构化类型』
- 第179页的『删除和更新类型表的行』
- 第179页的『重命名现存表』
- 第180页的『删除表』
- 第182页的『删除触发器』
- 第182页的『删除用户定义函数 (UDF)、类型映射或方法』
- 第183页的『删除用户定义类型 (UDT) 或类型映射』
- 第183页的『改变或删除视图』
- 第185页的『删除摘要表』
- 第186页的『改变或删除服务器』
- 第187页的『改变或删除别名』
- 第188页的『删除索引、索引扩充或索引规范』
- 第189页的『更改对象时的语句从属性』

## 删除数据库

虽然数据库中的某些对象可以改变，但是数据库本身不能改变：必须删除它，然后重新创建。因为删除数据库会删除它的所有对象、容器和相关的文件，所以此操作可产生广泛的影响。删除的数据库被从数据库目录中除去（取消编目）。

要使用“控制中心”删除数据库：

1. 展开对象树，直到您看到**数据库**文件夹为止
2. 右键单击要删除的数据库，并从弹出菜单中选择**删除**。
3. 单击**确认**框，并单击**确定**。

要使用命令行来删除数据库，输入：

```
DROP DATABASE <name>
```

以下命令删除数据库 **SAMPLE**：

```
DROP DATABASE SAMPLE
```

**注：**若打算继续用 **SAMPLE** 数据库做实验，则不应将其删除。若已删除 **SAMPLE** 数据库而又发现再次需要它，可重新创建它。

## 改变节点组

有关改变节点组的详细信息可在《*管理指南：性能*》中的“通过添加处理器调整配置”一章中找到。

一旦添加或删除了节点，就必须将当前的数据重新分布至节点组中的新节点集合中。为此，使用 **REDISTRIBUTE NODEGROUP** 命令。有关信息，参考《*管理指南：性能*》中的“跨数据库分区重新分布数据”，并参考 *Command Reference*。

## 改变表空间

当创建一个数据库时，至少要创建三个表空间：一个目录表空间 (**SYSCATSPACE**)；一个用户表空间（缺省名称是 **USERSPACE1**）；以及一个系统临时表空间（缺省名称是 **TEMPSPACE1**）。必须至少使这三种表空间各有一个。您可以在您希望的情况下添加其他用户和临时表空间。

**注：**不能删除目录表空间 **SYSCATSPACE** 或另创建一个，且必须始终至少存在一个系统临时表空间。可以创建其它系统临时表空间。在创建表空间之后，您也不能更改它的页大小或数据块大小。

本节讨论如何更改表空间，如下所示：

- 第160页的『将容器添加至 **DMS** 表空间』

- 『修改 DMS 表空间中的容器』
- 第162页的『将容器添加至分区上的 SMS 表空间』
- 第162页的『重命名表空间』
- 第162页的『转换表空间的状态』
- 第163页的『删除用户表空间』
- 第163页的『删除系统临时表空间』。

有关表空间的设计信息，参考《管理指南：计划》。

### 将容器添加至 DMS 表空间

可以通过将一个或多个容器添加至该表空间来增加 DMS 表空间（即，使用 MANAGED BY DATABASE 子句创建的）的大小。

在所有容器中重新平衡该表空间的内容。在重新平衡期间，不限制对该表空间的访问。若需要添加多个容器，应该同时添加它们。

要使用“控制中心”来向 DMS 表空间添加容器：

1. 展开对象树，直到您看到**表空间**文件夹为止。
2. 右键单击要添加容器的表空间，并从弹出菜单中选择**改变**。
3. 单击**添加**，填写信息，并单击**确定**。
4. 若该表空间在分区数据库环境中，且您需要更改该表空间的性能参数，则单击**高级**。
5. 单击**确定**。

要使用命令行来向 DMS 表空间添加容器，输入：

```
ALTER TABLESPACE <name>
  ADD (DEVICE '<path>' <size>)
```

以下示例举例说明如何将两个新设备容器（各含 10 000 页）添加至 UNIX 系统上的一个表空间中：

```
ALTER TABLESPACE RESOURCE
  ADD (DEVICE '/dev/rhd9' 10000,
      DEVICE '/dev/rhd10' 10000)
```

注意，ALTER TABLESPACE 语句允许更改可以影响性能的表空间的其他特性。有关更多信息，参考《管理指南：性能》中的“表空间对查询优化的影响”。

### 修改 DMS 表空间中的容器

可以通过调整一个或多个与表空间相关联的容器的大小或扩充一个或多个这样的容器来增加 DMS 表空间（即，使用 MANAGED BY DATABASE 子句创建的）



中的容器的大小。若您知道容器大小的新上限，则应该考虑调整大小的方法。若您不知道（也不关心）容器的当前大小，则应该考虑扩充方法。

要使用命令行来调整 DMS 表空间中一个或多个容器的大小，输入：

```
ALTER TABLESPACE <name>
RESIZE (DEVICE '<path>' <size>)
```

以下示例举例说明如何在基于 UNIX 的系统上的表空间中增加两个设备容器（各含 1 000 页）：

```
ALTER TABLESPACE HISTORY
RESIZE (DEVICE '/dev/rhd7' 2000,
        DEVICE '/dev/rhd8' 2000)
```

在此操作之后，两个设备的大小都从 1 000 页增加至 2 000 页。与添加新容器类似，在所有容器中重新平衡该表空间的内容。在重新平衡期间，不限制对该表空间的访问。

要使用命令行来扩充 DMS 表空间中一个或多个容器，输入：

```
ALTER TABLESPACE <name>
EXTEND (DEVICE '<path>' <size>)
```

以下示例举例说明如何在基于 UNIX 的系统上的表空间中增加两个设备容器（各含 1 000 页）：

```
ALTER TABLESPACE HISTORY
EXTEND (DEVICE '/dev/rhd11' 1000,
        DEVICE '/dev/rhd12' 1000)
```

在此操作之后，两个设备的大小都从 1 000 页增加至 2 000 页。与添加新容器类似，在所有容器中重新平衡该表空间的内容。在重新平衡期间，不限制对该表空间的访问。

通过预取程序以并行方式执行 DMS 容器（文件容器和原始设备容器），这些容器是在表空间创建期间或之后添加的，或者是在表空间创建之后扩充的。要增加创建或调整容器大小这些操作的并行性，可以增加系统中运行的预取程序的数目。不以并行方式执行的唯一进程是记录这些操作以及在创建容器的情况下标记这些容器。

**注：**要使 CREATE TABLESPACE 或 ALTER TABLESPACE 语句的并行性最大（对于将新的容器添加至现有的空间），确保预取程序数大于或等于要添加的容器数。

**注：**不能缩小容器的大小。

注意，ALTER TABLESPACE 语句允许更改可以影响性能的表空间的其他特性。有关更多信息，参考《管理指南：性能》中的“表空间对查询优化的影响”。

### 将容器添加至分区上的 SMS 表空间

要使用命令行来向 SMS 表空间添加容器，输入以下内容：

```
ALTER TABLESPACE <name>
  ADD ('<path>')
  ON NODE (<partition_number>)
```

由数字指定的分区和分区范围内的每个分区（或节点）必须在定义表空间的节点组中存在。partition\_number 可能仅在语句的正好一个 ON NODE 子句中显式出现或在某个范围内。

以下示例显示如何在基于 UNIX 的操作系统上将新的容器添加至由表空间 "plans" 所使用的节点组的第 3 号分区：

```
ALTER TABLESPACE plans
  ADD ('/dev/rhdisk0')
  ON NODE (3)
```

### 重命名表空间

可以给予现存表空间新名称，而无需关心该表空间中的个别对象。重命名表空间时，将更改所有引用该表空间的目录记录。

不能重命名 SYSCATSPACE 表空间。

不能重命名处于“前滚暂挂”或“正在前滚”状态的表空间。

当复原在备份后已被重命名的表空间时，必须在 RESTORE DATABASE 命令中使用新的表空间名。若使用先前表空间名，则将找不到它。同样，若使用 ROLLFORWARD DATABASE 命令前滚该表空间，也需确保使用新名称。若使用先前表空间名，则将找不到它。

### 转换表空间的状态

若与表空间相关的容器已变得可访问，可以使用 ALTER TABLESPACE 语句的 SWITCH ONLINE 子句从表空间中除去 OFFLINE 状态。表空间已除去 OFFLINE 状态，而数据库的其余部分仍在运行并在使用中。

使用此子句的一个替代方法是将所有应用程序与数据库断开连接，然后再次将这些应用程序连接至数据库。这样就可以从表空间中除去 OFFLINE 状态。

要使用命令行从表空间中除去 OFFLINE 状态，输入：

```
ALTER TABLESPACE <name>
  SWITCH ONLINE
```

## 删除用户表空间

当删除用户表空间时，也会删除该表空间中的所有数据，释放容器，除去目录项，并导致该表空间中定义的所有对象都被删除或标记为无效。

可以通过删除表空间来重新使用该空的表空间中的容器，但是，在试图重新使用这些容器之前，必须 COMMIT 该 DROP TABLESPACE 命令。

可删除一个包含所有表数据的用户表空间，包括在该单个用户表空间中的索引和 LOB 数据。也可删除所包含的表跨几个表空间的一个用户表空间。即，在一个表空间中可能有数据，在另一个表空间中有索引，而在第三个表空间中有任何 LOB。必须在一条语句中同时删除所有三个表空间。包含跨越的表的所有表空间必须全部纳入此单条语句中，否则该删除请求将失败。有关如何删除包含跨越表数据的表空间详细信息，参考 *SQL Reference*。

要使用“控制中心”来删除用户表空间：

1. 展开对象树，直到您看到**表空间**文件夹为止。
2. 右键单击要删除的表空间，并从弹出菜单中选择**删除**。
3. 选择**确认**框，并单击**确定**。

要使用命令行来删除用户表空间，输入：

```
DROP TABLESPACE <name>
```

以下 SQL 语句将删除表空间 ACCOUNTING：

```
DROP TABLESPACE ACCOUNTING
```

## 删除系统临时表空间

因为数据库必须始终有至少一个系统临时表空间，所以若不首先创建另一系统临时表空间，则不能删除系统临时表空间。例如，若希望将一个容器添加到一个 SMS 临时表空间中，必须先添加一个新的系统临时表空间，然后删除旧的系统临时表空间。

要使用“控制中心”来删除系统表空间：

1. 展开对象树，直到您看到**表空间**文件夹为止。
2. 若只有一个系统临时表空间，则右键单击**表空间**文件夹，并从弹出菜单中选择**创建** → **使用向导创建表空间**。否则，跳过步骤 4。
3. 遵循向导中的步骤，以创建新的系统临时表空间（若需要的话）。
4. 再次单击**表空间**文件夹，以在窗口右边（“内容”窗格）显示表空间列表。
5. 右键单击要删除的系统临时表空间，并从弹出菜单中单击**删除**。
6. 选择**确认框**，并单击**确定**。

若只有一个系统临时表空间，则在删除它之前，必须创建另一个。这可以使用命令行完成，即输入：

```
CREATE SYSTEM TEMPORARY TABLESPACE <name>  
MANAGED BY SYSTEM USING ('<device>')
```

之后，要使用命令行来删除系统表空间，输入：

```
DROP TABLESPACE <name>
```

以下 SQL 语句创建一个称为 **TEMPSPACE2** 的新的系统临时表空间：

```
CREATE SYSTEM TEMPORARY TABLESPACE TEMPSPACE2  
MANAGED BY SYSTEM USING ('d')
```

一旦创建了 **TEMPSPACE2**，则可使用以下命令删除原来的系统临时表空间 **TEMPSPACE1**：

```
DROP TABLESPACE TEMPSPACE1
```

可以通过删除表空间来重新使用该空的表空间中的容器，但是，在试图重新使用这些容器之前，必须 **COMMIT** 该 **DROP TABLESPACE** 命令。

### 删除用户临时表空间

仅当用户临时表空间中未定义当前已说明临时表时，才能删除该表空间。当删除表空间时，不会尝试删除该表空间中的所有已说明临时表。

**注：**已说明临时表是在说明它的应用程序与数据库断开连接时隐式删除的。

## 删除模式

在删除模式之前，必须删除该模式中的所有对象或将它们移至另一个模式。当尝试 **DROP** 语句时，该模式名必须在目录中；否则会返回错误。

要使用“控制中心”删除模式:

1. 展开对象树, 直到您看到**模式**文件夹为止。
2. 右键单击要删除的模式, 并从弹出菜单中选择**删除**。
3. 选择**确认框**, 并单击**确定**。

要使用命令行来删除模式, 输入:

```
DROP SCHEMA <name>
```

在以下示例中, 删除了模式 "joeschma":

```
DROP SCHEMA joeschma RESTRICT
```

RESTRICT 关键字强制执行一个规则, 即不能在指定的模式中为要从数据库中删除的模式定义对象。

## 修改表的结构和内容

修改表的结构和内容所需的任务包括下列各项:

- 『将列添加至现存表』
- 第166页的『修改列定义』
- 第167页的『从表或视图中除去行』
- 第168页的『改变约束』
- 第172页的『在现存表上定义生成列』
- 第175页的『将表说明为易失的』
- 第176页的『更改分区键』
- 第176页的『更改表属性』
- 第179页的『刷新摘要表中的数据』

注意, 不能改变表的触发器; 必须删除任何不再适合的触发器(参见第182页的『删除触发器』), 然后添加它的替换项目(参见第120页的『创建触发器』)。

### 将列添加至现存表

列定义包括列名、数据类型和任何需要的约束。

当将列添加至表时, 必须在逻辑上将列放置在最右边的现有列定义的右边。当将新列添加至现存的表时, 只修改系统目录中的表描述, 所以表的访问时间不会立即受到影响。在使用 UPDATE 语句修改现存记录之前, 不会实际改变它们。当从

表中检索现存的行时，根据新列的定义，会为新列提供空值或缺省值。在创建表之后添加的列不能定义为 NOT NULL：必须将它们定义为 NOT NULL WITH DEFAULT 或可空的。

要使用“控制中心”对现存表添加列：

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要向其添加列的表，并从弹出菜单中选择**改变**。
3. 检查列页，填写列的信息，并单击**确定**。

要使用命令行对现存表添加列，输入：

```
ALTER TABLE <table_name>  
ADD <column_name> <data_type> <>null_attribute>
```

可以使用 SQL 语句来添加列。以下语句使用 ALTER TABLE 语句将三列添加至 EMPLOYEE 表：

```
ALTER TABLE EMPLOYEE  
ADD MIDINIT CHAR(1) NOT NULL WITH DEFAULT  
ADD HIREDATE DATE  
ADD WORKDEPT CHAR(3)
```

### 修改列定义

可增加现存的 VARCHAR 列的长度来改变列的特征。字符数可增加到与所用的页大小相关的一个值。

要使用“控制中心”来修改现存表的列：

1. 展开对象树，直到您看到表文件夹为止。
2. 在右窗格中的表列表中，右键单击要修改其一个列的表，并从弹出菜单中选择**改变**。
3. 检查列页，选择该列，并单击**更改**。
4. 在**长度**中输入该列的新字节计数，并单击**确定**。

要使用命令行来修改现存表的列，输入：

```
ALTER TABLE ALTER COLUMN  
<column_name> <modification_type>
```

例如，要将一列增加到 4000 个字符，使用类似于以下的语句：

```
ALTER TABLE ALTER COLUMN  
COLNAM1 SET DATA TYPE VARCHAR(4000)
```

不能改变类型表的列。然而，可将一个作用域添加到尚未定义作用域的现存的引用类型列中。例如：

```
ALTER TABLE ALTER COLUMN  
COLNAMT1 ADD SCOPE TYPTAB1
```

有关 ALTER TABLE 语句的更多信息，参考 *SQL Reference* 手册。

### 从表或视图中除去行

可以通过删除行来更改表或视图的内容。从视图中删除行也会从基于该视图的表中删除行。使用 DELETE 语句来：

- 删除已由搜索条件任选确定的一行或多行。这称为 *搜索 DELETE*。
- 只删除已由游标的当前位置确定的一行。这称为 *定位 DELETE*。

可以将 DELETE 语句嵌入应用程序中或作为动态 SQL 语句发出。

若正在修改的表通过参考约束涉及其它表，则对行执行删除有一些注意事项。若标识的表或标识的视图的基本表是父代，则为删除选择的行不能有任何从属与删除规则 RESTRICT 有关。此外，DELETE 不得级联至具有与删除规则 RESTRICT 有关的从属的派生行。

若 RESTRICT 删除规则未阻止删除操作，则删除所选行。有关具有所选行的从属的行会发生什么情况的附加信息，应参考 *SQL Reference*。

例如，要从表 (DEPARTMENT) 中删除部门 (DEPTNO) "D11"，使用：

```
DELETE FROM department WHERE deptno='D11'
```

若在运行多行 DELETE 期间出错，则不会对表进行更改。若发生错误而阻止删除与搜索条件匹配的所有行且阻止现有的参考约束所需要的所有操作，则不会对表进行更改。

除非适当的锁定已经存在，否则在运行成功的 DELETE 语句期间需要互斥锁定。在 COMMIT 或 ROLLBACK 语句之后释放锁定。锁定可以防止其它应用程序对表执行操作。

### 修改标识列定义

若正在重新创建表然后执行导入或装入操作，并且在表中具有 IDENTITY 列，则在重新创建表的内容之后将复位表以开始生成 IDENTITY 值，从 1 开始。当将新值插入此重新创建的表时，您不想 IDENTITY 列再次从 1 开始。您不希望在 IDENTITY 列中有重复值。要防止此情况发生，您应该：

1. 重新创建表。

2. 使用 `MODIFIED BY IDENTITYOVERRIDE` 子句将数据装入表中。将数据装入表中但不会对行生成标识值。

3. 运行查询以获取 `IDENTITY` 列的最后一个计数器值:

```
SELECT MAX(<IDENTITY column>)
```

此查询将返回表的 `IDENTITY` 列值的等价值。

4. 使用 `ALTER TABLE` 语句的 `RESTART` 子句:

```
ALTER TABLE <table name> ALTER COLUMN <IDENTITY column>  
RESTART WITH <last counter value>
```

5. 将新行插入表中。将基于在 `RESTART WITH` 子句中指定的值生成 `IDENTITY` 列值。

### 改变约束

只能通过删除旧约束，然后添加新约束取代它，来改变约束。有关更多信息，参见:

- 『添加约束』
- 第171页的『删除约束』

有关约束的更多信息，参见第108页的『定义约束』。

### 添加约束

使用 `ALTER TABLE` 语句添加约束。有关此语句的更多信息，包括它的语法，参考 *SQL Reference* 手册。

有关约束的更多信息，参见第108页的『定义约束』。

**添加唯一约束:** 可以将唯一约束添加至现存表。约束名不能与在 `ALTER TABLE` 语句内指定的任何其他约束相同，且必须在该表内是唯一的（这包括定义的任何参考完整性约束的名称）。在成功执行该语句之前，会对照新条件检查现存数据。

以下 `SQL` 语句将一个唯一约束添加至 `EMPLOYEE` 表，它表示唯一标识该表中的雇员的一个新方法:

```
ALTER TABLE EMPLOYEE  
ADD CONSTRAINT NEWID UNIQUE(EMPNO,HIREDATE)
```

**添加主键和外键:** 要将约束添加至大表，更有效的方法是，使该表处于检查暂挂状态，再添加约束，然后检查该表中违规行的合并列表。使用 `SET INTEGRITY` 语句来显式设置检查暂挂状态: 若该表是父表，则为所有从属表和子表隐式设置检查暂挂。



要使用“控制中心”来添加主键:

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要修改的表，并从弹出菜单中选择**改变**。
3. 在**主键**页上，选择一个或多个列作为主键，并单击箭头来移动它们。
4. 可选：输入主键的约束名。
5. 单击**确定**。

要使用命令行添加主键，输入:

```
ALTER TABLE <name>
  ADD CONSTRAINT <column_name>
  PRIMARY KEY <column_name>
```

当将一个外键添加至表中时，包含下列语句的软件包和高速缓存动态 SQL 可能被标记为无效:

- 插入或更新包含外键的表的语句
- 更新或删除父表的语句。

有关信息，参见第189页的『更改对象时的语句从属性』。

要使用“控制中心”来添加外键:

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要修改的表，并从弹出菜单中选择**改变**。
3. 在**外键**页上，单击**添加**。
4. 在**添加外键**窗口上，指定父表信息。
5. 选择一个或多个要作为外键的列，并单击箭头来移动它们。
6. 指定当删除或更新父表的行时要对从属表执行的操作。还可以为该外键添加约束名。
7. 单击**确定**。

要使用命令行添加外键，输入:

```
ALTER TABLE <name>
  ADD CONSTRAINT <column_name>
  FOREIGN KEY <column_name>
  ON DELETE <action_type>
  ON UPDATE <action_type>
```

以下示例显示 ALTER TABLE 语句如何将主键和外键添加至一个表:

```

ALTER TABLE PROJECT
  ADD CONSTRAINT PROJECT_KEY
    PRIMARY KEY (PROJNO)
ALTER TABLE EMP_ACT
  ADD CONSTRAINT ACTIVITY_KEY
    PRIMARY KEY (EMPNO, PROJNO, ACTNO)
  ADD CONSTRAINT ACT_EMP_REF
    FOREIGN KEY (EMPNO)
      REFERENCES EMPLOYEE
        ON DELETE RESTRICT
  ADD CONSTRAINT ACT_PROJ_REF
    FOREIGN KEY (PROJNO)
      REFERENCES PROJECT
        ON DELETE CASCADE

```

**添加表检查约束：** 可以使用 ALTER TABLE 语句将检查约束添加至现存的表。约束名不能与在 ALTER TABLE 语句内指定的任何其他约束相同，且必须在该表内是唯一的（这包括定义的任何参考完整性约束的名称）。在成功执行该语句之前，会对照新条件检查现存数据。

要将约束添加至大表，更有效的方法是，使该表处于检查暂挂状态，添加约束，然后检查该表中违规行的合并列表。使用 SET INTEGRITY 语句来显式设置检查暂挂状态：若该表是父表，则为所有从属表和子表隐式设置检查暂挂。

当添加表检查约束时，插入或更新该表的软件包和高速缓存的动态 SQL 可能被标记为无效。有关更多信息，参见第189页的『更改对象时的语句从属性』。

要使用“控制中心”来添加表检查约束：

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要修改的表，并从弹出菜单中选择**改变**。
3. 在**检查约束**页上，单击**添加**。
4. 在**添加检查约束**窗口上，填写信息，并单击**确定**。
5. 在**检查约束**页上，单击**确定**。

要使用命令行来添加表检查约束，输入：

```

ALTER TABLE <name>
  ADD CONSTRAINT <name> (<constraint>)

```

以下 SQL 语句将一个约束添加至 EMPLOYEE 表，即每个雇员的工资加佣金必须超过 \$25,000：

```

ALTER TABLE EMPLOYEE
  ADD CONSTRAINT REVENUE CHECK (SALARY + COMM > 25000)

```

## 删除约束

使用 ALTER TABLE 语句删除约束。有关此语句的更多信息，包括它的语法，参考 *SQL Reference* 手册。

有关约束的更多信息，参见第108页的『定义约束』。

**删除唯一约束：** 可以使用 ALTER TABLE 语句来显式删除唯一约束。一个表上的所有唯一约束的名称可以在 SYSCAT.INDEXES 系统目录视图找到。

以下 SQL 语句从 EMPLOYEE 表中删除唯一约束 NEWID:

```
ALTER TABLE EMPLOYEE
DROP UNIQUE NEWID
```

删除此唯一约束使使用该约束的任何软件包或高速缓存的动态 SQL 失效。

**删除主键和外键：** 要使用“控制中心”来删除主键:

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要修改的表，并从弹出菜单中选择**改变**。
3. 在**主键**页上，在右边选择要删除的主键，并单击箭头来将其移至左边的**可用列框**。
4. 单击**确定**。

要使用命令行来删除主键，输入:

```
ALTER TABLE <name>
DROP PRIMARY KEY
```

当删除外键约束时，包含下列语句的软件包或高速缓存的动态 SQL 语句可能被标记为无效:

- 插入或更新包含外键的表的语句
- 更新或删除父表的语句。

有关更多信息，参见第189页的『更改对象时的语句从属性』。

要使用“控制中心”来删除外键:

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要修改的表，并从弹出菜单中选择**改变**。
3. 在**外键**页上，单击**添加**。
4. 在右边选择要删除的外键，并单击箭头来将其移至左边的**可用列框**。
5. 在**外键**页上，单击**确定**。

要使用命令行来删除外键，输入：

```
ALTER TABLE <name>
  DROP FOREIGN KEY <foreign_key_name>
```

下列示例在 ALTER TABLE 语句中使用 DROP PRIMARY KEY 和 DROP FOREIGN KEY 子句来删除表上的主键和外键：

```
ALTER TABLE EMP_ACT
  DROP PRIMARY KEY      DROP FOREIGN KEY ACT_EMP_REF
  DROP FOREIGN KEY ACT_PROJ_REF
ALTER TABLE PROJECT
  DROP PRIMARY KEY
```

有关 ALTER TABLE 语句的信息，参考 *SQL Reference* 手册。

**删除表检查约束：** 可以使用 ALTER TABLE 语句显式删除或更改表检查约束，或作为 DROP TABLE 语句的结果隐式删除它。

当删除一个表检查约束时，与该表有 INSERT 或 UPDATE 关系的所有软件包和高速缓存的动态 SQL 语句失效。（有关更多信息，参见第189页的『更改对象时的语句从属性』。）一个表上的所有检查约束的名称可以在 SYSCAT.INDEXES 目录视图中找到。在尝试删除带有系统生成的名称的表检查约束之前，在 SYSCAT.CHECKS 目录视图中查找该名称。

要使用“控制中心”来删除表检查约束：

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要修改的表，并从弹出菜单中选择**改变**。
3. 在**检查约束**页上，选择要删除的检查约束，单击**除去**，并单击**确定**。

要使用命令行来删除表检查约束：

```
ALTER TABLE <table_name>
  DROP CHECK <check_constraint_name>
```

以下 SQL 语句从 EMPLOYEE 表中删除表检查约束 REVENUE：

```
ALTER TABLE EMPLOYEE
  DROP CHECK REVENUE
```

### 在现存表上定义生成列

生成列在基本表上定义，在这些列中，存储的值是使用表达式计算得出的，而不是通过插入或更新操作指定。可以在创建表时创建生成列，也可以作为对现存表的修改来创建它。

执行下列步骤来定义生成列：

1. 将表置于检查暂挂状态。

```
SET INTEGRITY FOR t1 OFF
```

2. 改变表，以添加一个或多个生成列。

```
ALTER TABLE t1 ADD COLUMN c3 DOUBLE GENERATED ALWAYS AS (c1 + c2),  
ADD COLUMN c4 GENERATED ALWAYS AS  
(CASE WHEN c1 > c3 THEN 1 ELSE NULL END)
```

3. 此时，根据要对该表执行的工作的不同，有几种方法来完成任务：

- 该表非常大，您不确定是否有足够的日志空间来完成任务。装入数据后，在重新打开完整性检查之前，您需要：

```
COMMIT
```

然后，需要使用 `db2gncol` 实用程序来为生成的列建立值。此实用程序位于 `sqlllib` 目录的 `bin` 子目录下面。使用此实用程序的方法如下所示：

```
db2gncol -d <dbname> -s <schema> -t <table_name>  
-c <commitcount>
```

`dbname` 指定该表所在的数据库的别名。`schema`指定该表的模式名，区分大小写。`table_name` 指定一个表，该表的由表达式生成的列的新值要进行计算。`schema` 和 `table_name` 都区分大小写。`commitcount` 是每两次内部提交（目的是清理日志）之间处理的行数。此参数影响生成列值所需的日志空间大小。

还有两个可选参数在上述示例中没有显示出来。它们是 `-u <username>` 和 `-p <password>`，它们标识用户和密码。用户必须具有 `SYSADM` 或 `DBADM` 权限。若未标识用户和密码，则将使用当前用户标识。

若您希望得到有关此实用程序的帮助信息，输入：

```
db2gncol -h
```

当使用帮助参数时，所有其他参数都被忽略。

该表在整个过程期间被锁定（即使它处于检查暂挂状态）。进行锁定的原因是：其他实用程序可以访问处于检查暂挂状态的表。此锁定可以防止与这些其他实用程序发生冲突。

- 您预期用于更新生成列的日志空间能够满足 `SET INTEGRITY`。这将是正常的情况。装入数据后，使用以下命令对生成列重新计算并重新赋值：

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED  
FORCE GENERATED
```

**注：**此时，可使用异常表。

- 该表非常大，您不确定是否有足够的日志空间来完成任务，您不选择上面介绍的第一种方法。装入数据后，在重新打开完整性检查之前，您需要：
  - a. 获取对表的互斥锁定。这将防止除未提交的读取事务之外的所有事务访问该表。

```
LOCK TABLE t1
```

- b. 将该表置于数据未经检查的联机状态。

```
SET INTEGRITY FOR t1 ALL IMMEDIATE UNCHECKED
```

- c. 使用间歇性的提交和谓词更新生成列，以避免填充日志。

```
UPDATE t1 SET (c3, c4) = (DEFAULT, DEFAULT) WHERE <predicate>
```

- d. 使该表联机，并检查其完整性。

```
SET INTEGRITY FOR t1 OFF
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
```

- e. 通过使用提交语句完成事务来解锁该表。

```
COMMIT
```

- 您知道该表是使用“最初不记录”选项创建的。这样，该表的记录是关闭的，当使用生成列值时，存放普通的蕴含项和风险。

- a. 激活“最初不记录”选项。

```
ALTER TABLE t1 ACTIVATE NOT LOGGED INITIALLY
```

- b. 生成值。

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED FORCE GENERATED
```

- c. 通过提交事务再次关闭“最初不记录”。

```
COMMIT
```

也可以简单地通过应用表达式（就象该表达式是相等检查约束一样）检查生成列的值：

```
SET INTEGRITY FOR t1 IMMEDIATE CHECKED
```

若已在生成列中放置了值，例如使用 `LOAD`，且您知道这些值与生成表达式相匹配，则可以使该表脱离检查暂挂状态，而不必检查或指定值：

```
SET INTEGRITY FOR t1 GENERATED COLUMN IMMEDIATE UNCHECKED
```

只能在已对其定义了相等比较的数据类型上定义生成列。被生成列排除的数据类型包括：结构化类型、LOB、CLOB、DBCLOB、LONG VARCHAR、LONG VARGRAPHIC 以及使用相同的被排除数据类型定义的用户定义类型。

不能在约束、唯一索引、参考约束、主键和全局临时表中使用生成列。使用 LIKE 和具体化视图创建的表不继承生成列属性。

不指定关键字 `DEFAULT` 时，不能插入或更新生成列。进行插入时，使用 `DEFAULT` 使得无需枚举列列表中的列。而是，可以在值列表中将生成列设置为 `DEFAULT`。进行更新时，`DEFAULT` 允许重新计算特定生成列，`SET INTEGRITY` 已使这些生成列联机，但未进行检查。

处理触发器的次序要求 `BEFORE` 触发器不能在它们的头部（在更新之前）或在它们的主体中引用生成列。在处理次序中，生成列是在 `BEFORE` 触发器之后处理的。

`db2look` 实用程序将看不到生成列所生成的检查约束。

当使用复制时，目标表一定不能在其映射中使用生成列。使用复制时，有两个选项：

- 目标表必须将生成列定义为正常列，即非生成列
- 目标表必须在映射中省略掉生成列

当使用生成列时，有几个限制：

- 生成列一定不能相互依赖。
- 用来创建生成列的表达式一定不能包含子查询。这包括带有读取 `SQL` 数据 (`READS SQL DATA`) 的函数的表达式。
- 生成列上不允许检查约束。
- 生成列上不允许唯一索引。这包括唯一约束和主键。

### 将表说明为易失的

易失的表是在运行时它的内容可能从空变为很大的一种表。此类型表的易失性或极端的可变性使 `RUNSTATS` 收集的统计信息变得不准确。统计信息是在一个时间点收集的，它也只反映该时间点的情况。若生成一个使用易失表的访问计划，可导致不正确或性能不好的计划。例如，若易失表为空时收集统计信息，优化器可能会使用表扫描而不是索引扫描来访问易失表。

为了防止这种情况，应考虑使用 `ALTER TABLE` 语句将该表说明为易失的。通过说明该表为易失的，优化器将考虑使用索引扫描而不是表扫描。使用说明的易失表的访问计划将不依赖于该表的现存统计信息。

要使用“控制中心”来将表说明为易失的：

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要修改的表，并从弹出菜单中选择**改变**。
3. 在表页中，选择**基数在运行时显著变化**复选框，并单击**确定**。

要使用命令行来将表说明为“易失的”，输入：

```
ALTER TABLE <table_name>
VOLATILE CARDINALITY
```

## 更改分区键

只能在单分区节点组中更改表上的分区键。首先删除现存的分区键，然后创建另一个。

以下 SQL 语句从 MIXREC 表中删除分区键 MIX\_INT:

```
ALTER TABLE MIXREC
DROP PARTITIONING KEY
```

有关更多信息，在 *SQL Reference* 手册中查看 ALTER TABLE 语句。

不能更改一个多分区节点组中的一个表的分区键。若尝试删除它，则会返回错误。

要更改多分区节点组的分区键:

- 将所有数据导出至单分区节点组，然后遵循以上指示信息。
- 导出所有数据，删除表，重新创建表并重新定义分区键，然后导入所有数据。

这两种方法都不适用于大数据库；因此，在实现大数据库的设计之前，要定义适当的分区键。

## 更改表属性

您可能有理由要更改表属性，如数据捕捉选项、每页上空闲空间的百分比 (PCTFREE)、锁定大小或追加方式。

在一个表的每一页上要留下的空闲空间量是通过 PCTFREE 指定的，且是有效地使用群集索引的重要注意事项。要指定的数量取决于现存数据和预计的将来数据的性质。LOAD 和 REORG 会考虑 PCTFREE，但插入、更新和导入活动将忽略它。

将 PCTFREE 设置为更大的值将在更长的时期内维持群集，但是，也将需要更多的磁盘空间。

可指定使用 LOCKSYZE 参数访问表时使用的锁定的大小（粒度）。在缺省情况下，在创建表时，定义行级锁定。使用表级锁定可提高查询的性能，方法是限制需要获取和释放的锁定的数目。

通过指定 APPEND ON，可提高表的整体性能。它允许更快的插入，而无需维护关于空闲空间的信息。

不能将带群集索引的表改变为打开追加方式。类似地，不能在具有追加方式的表上创建群集索引。



## 改变标识列

使用 ALTER TABLE 语句修改现有标识列的属性。有关此语句的更多信息，包括它的语法，参考 *SQL Reference*。

有多种方式来修改标识列，以便它具有序列的某些特征。

有某些任务专用于 ALTER TABLE 语句和标识列：

- **RESTART** 将与标识列关联的序列复位为隐式或显式指定的值，该值是在最初创建标识列时作为起始值指定的。
- **RESTART WITH <numeric-constant>** 将与标识列关联的序列复位为准确的数字常量值。数字常量可以是任何正的或负的值，而且任何小数点右边不带有非零数字，可以将该值赋给标识列。

## 改变序列

使用 ALTER SEQUENCE 语句修改现有序列的属性。有关此语句的更多信息，包括它的语法，参考 *SQL Reference*。

可以修改的序列属性包括：

- 更改将来值的递增
- 建立新的最小值或最大值
- 更改高速缓存序列号的数目
- 更改序列是否将循环
- 更改是否必须按请求顺序生成序列号
- 重新启动序列

有两种任务不是序列创建的一部分。它们是：

- **RESTART**。将序列复位为隐式或显式指定的值，该值是在创建序列时作为起始值指定的。
- **RESTART WITH numeric-constant**。将序列复位为精确的数字常量值。数字常量可以是任何正的或负的值，而且任何小数点右边不带有非零数字。

在重新启动序列或更改为 **CYCLE** 之后，可能会生成重复序列号。ALTER SEQUENCE 语句仅影响将来的序列号。

不能更改序列的数据类型。而是必须删除当前序列，然后创建新序列，指定新的数据类型。

当改变序列时，会丢失 DB2 未使用的所有高速缓存序列值。

## 删除序列

要删除序列，使用 `DROP` 语句。有关此语句的更多信息，包括它的语法，参考 *SQL Reference*。

可以通过使用下列命令来删除特定序列：

```
DROP SEQUENCE sequence_name
```

其中 `sequence_name` 是要删除的序列名，它包括隐式或显式模式名以正确标识现有的序列。

不能使用 `DROP SEQUENCE` 语句删除系统为 `IDENTITY` 列创建的序列。

一旦删除序列，则也会删除对该序列的所有特权。

## 改变摘要表特性

在遵守一些限制的前提下，可将摘要表更改为正规表或将正规表更改为摘要表。不能更改其他表类型；只可更改正规表和摘要表。例如，不能将复制摘要表更改为正规表，反之亦然。

一旦将正规表改变成摘要表，该表便处于检查暂挂状态。当使用此方法进行改变时，摘要表定义中的全查询必须与原始表定义相匹配，即：

- 列数必须相同。
- 列名和位置必须匹配。
- 数据类型必须完全相同。

若摘要表是在原始表上定义的，则不能将原始表本身改变为摘要表。若原始表有触发器、检查约束、参考约束或定义了唯一索引，则不能将它改变为摘要表。若是在改变表特性以定义摘要表，则不允许在同一 `ALTER TABLE` 语句中以任何其他方法改变该表。

当将正规表改变为摘要表时，摘要表定义的全查询不能直接引用原始表或通过视图、别名或摘要表间接引用原始表。

要将摘要表更改为正规表，使用以下命令：

```
ALTER TABLE sumtable  
SET SUMMARY AS DEFINITION ONLY
```

要将正规表更改为摘要表，使用以下命令：

```
ALTER TABLE regtable  
SET SUMMARY AS <fullselect>
```

将正规表改变为摘要表时，有关全查询的限制与使用 CREATE SUMMARY TABLE 语句创建摘要表时的限制非常相似。

有关 CREATE SUMMARY TABLE 语句的其他信息，参考 *SQL Reference*。

### 刷新摘要表中的数据

可通过使用 REFRESH TABLE 语句来刷新一个或多个摘要表中的数据。此语句可嵌入应用程序中，或可动态执行。要使用此语句，必须具有 SYSADM 或 DBADM 权限，或对要刷新的表具有 CONTROL 特权。

以下示例显示如何刷新摘要表中的数据：

```
REFRESH TABLE SUMTAB1
```

有关 REFRESH TABLE 语句的更多信息，参考 *SQL Reference*。

## 改变用户定义结构化类型

创建结构化类型之后，可能会发现需要添加或删除与该结构化类型相关的属性。这可使用 ALTER TYPE（结构化）语句来完成。有关结构化类型上所需的所有信息，参考 *Application Development Guide*。

## 删除和更新类型表的行

可使用搜索式或定位式 DELETE 语句从类型表中删除行。可使用搜索式或定位式 UPDATE 语句更新类型表中的行。有关您所需的类型表的所有信息，参考 *Application Development Guide*。

## 重命名现存表

可以在一个模式内给予现存的表新名称，并保留在原表中创建的授权和索引。

要重命名的现存表可以是标识一个表的别名。要重命名的现存表不能是目录表、摘要表、类型表的名称，或者是除表或别名以外的对象的名称。

不能在下列任何一个对象中引用现存的表：

- 视图
- 触发器
- 参考约束
- 摘要表
- 现存引用列的作用域

并且，该表内不能有检查约束，也不能有除标识列之外的任何生成列。依赖于原表的任何软件包或高速缓存的动态 SQL 语句失效。最后，引用该原表的任何别名不会被修改。

应该考虑检查适当的系统目录表，以确保重命名的表不受以上任何一个限制的影响。

若软件包引用刚好已重命名的表，则必须重新绑定软件包。在下列情况中，可以隐式地重新绑定软件包：

- 使用该表的原名来重命名另一个表，或
- 使用该表的原名来创建一个别名或视图。

在尝试任何隐式或显式的重新绑定之前，必须完成这两种选择之一。若未选择任何一种，则任何重新绑定将失败。

要使用“控制中心”来重命名现存表：

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要重命名的表，并从弹出菜单中选择**重命名**。
3. 输入新表名，并单击**确定**。

要使用命令行来重命名现存的表，输入：

```
RENAME TABLE <schema_name>.<table_name> TO <new_name>
```

以下的 SQL 语句在 COMPANY 模式内将 EMPLOYEE 表重命名为 EMPL：

```
RENAME TABLE COMPANY.EMPLOYEE TO EMPL
```

有关 RENAME TABLE 语句的更多信息，参考 *SQL Reference* 手册。

## 删除表

可以使用 DROP TABLE SQL 语句删除表。

当删除一个表时，也会删除 SYSCAT.TABLES 目录中包含有关该表的信息的那一行，并会影响从属于该表的任何其他对象。例如：

- 会删除所有的列名。
- 会删除基于该表的任何列创建的索引。
- 将基于该表的所有视图标记为不起作用。（有关更多信息，参见第184页的『恢复不起作用的视图』。）
- 对删除的表和从属视图的所有特权被隐式取消。

- 会删除在其中该表为父表或从属表的所有参考约束。
- 从属于删除的表的所有软件包和高速缓存的动态 SQL 语句被标记为无效，且该状态会保持至重新创建了从属对象为止。这包括这样一些软件包，它们从属于将被删除的层次结构中子表上的任何超表。（有关更多信息，参见第189页的『更改对象时的语句从属性』。）
- 其引用的作用域为删除的表的任何引用列变为“无作用域”。
- 该表上的别名定义不受影响，因为不能取消别名的定义
- 将从属于该删除的表的所有触发器标记为不起作用。
- 通过任何 DATALINK 列链接的所有文件都断开。断开操作是异步执行的，表明这些文件可能不能立即用于其他操作。

要使用“控制中心”来删除表：

1. 展开对象树，直到您看到表文件夹为止。
2. 右键单击要删除的表，并从弹出菜单中选择删除。
3. 选择**确认**框，并单击**确定**。

要使用命令行来删除表，输入：

```
DROP TABLE <table_name>
```

以下语句删除 DEPARTMENT 表：

```
DROP TABLE DEPARTMENT
```

若一个表有子表，则不能删除它。但是，可用单个 DROP TABLE HIERARCHY 语句删除一个表 e 中的所有表，如以下示例所示：

```
DROP TABLE HIERARCHY person
```

DROP TABLE HIERARCHY 语句必须命名要删除的层次结构的根表。

删除表层次结构与删除特定的表之间有一些差别：

- DROP TABLE HIERARCHY 不会激活个别的 DROP 表语句将激活的删除触发器。例如，删除个别子表将激活其超表上的删除触发器。
- DROP TABLE HIERARCHY 不为删除的表的个别行建立日志项。而是将该层次结构的删除作为单个事件记录。

有关 DROP 语句的更多信息，参考 *SQL Reference*。

## 删除用户定义临时表

删除用户定义临时表时要注意一些注意事项；即，要注意使用 `DECLARE GLOBAL TEMPORARY TABLE` 语句创建的用户定义临时表。

删除这样的表时，表名必须由模式名 `SESSION` 限定，且该表名必须存在于创建该表的应用程序中。

软件包不能依赖于这种类型的表，因此，删除这样的表时，不使它们无效。

删除用户定义临时表时，若其创建时间早于活动工作单元或保存点，则在功能上删除该表，应用程序无法访问该表。然而，该表仍在其表空间中保留了一些空间，这使得在提交工作单元或结束保存点之前，不能删除该用户临时表空间。

有关 `DROP` 语句的更多信息，参考 *SQL Reference*。

## 删除触发器

可以使用 `DROP` 语句删除触发器对象，但是此过程将导致从属软件包被标记为无效，如下所示：

- 若删除不带显式列列表的更新触发器，则对目标表起更新作用的软件包就会失效。
- 若删除带一个列列表的更新触发器，则仅当该软件包也可更新 `CREATE TRIGGER` 语句的列名列表中至少一列时，用于更新目标表的该软件包才失效。
- 若删除插入触发器，则用于插入目标表的软件包失效。
- 若删除删除触发器，则用于删除目标表的软件包失效。

软件包将保持无效，直到显式绑定或重新绑定该应用程序，或运行它且数据库管理器自动重新绑定它为止。

## 删除用户定义函数 (UDF)、类型映射或方法

可以使用 `DROP` 语句来删除用户定义函数 (UDF)、函数模板或函数映射。

可用映射选项 `DISABLE` 禁用函数映射。有关更多信息，参考 *SQL Reference*。

若视图、触发器、表检查约束或另一个 UDF 从属于某个 UDF，则不能删除该 UDF。不能删除由 `CREATE DISTINCT TYPE` 语句隐式生成的函数。删除在 `SYSIBM` 模式或 `SYSFUN` 模式中的函数是不可能的。

其他对象可以从属于一个函数或函数模板。必须除去所有这类从属，包括函数映射，才能删除该函数，而标记为不起作用的软件包除外。此类软件包未隐式重新绑定。必须使用 `BIND` 或 `REBIND` 命令来重新绑定它，或使用 `PREP` 命令准备

它。有关这些命令的更多信息，参考 *Command Reference* 手册。删除 UDF 使使用它的任何软件包或高速缓存的动态 SQL 语句失效。

删除一个函数映射会将一个软件包标记为无效。将自动进行重新绑定，而优化器将尝试使用本地函数。对于本地函数是模板的情况，隐式重新绑定将失败。

(有关更多信息，参见第189页的『更改对象时的语句从属性』。)

## 删除用户定义类型 (UDT) 或类型映射

可使用 **DROP** 语句来删除用户定义类型 (UDT) 或类型映射。若 UDT 用于以下情况，则不能将其删除：

- 在现存的表或视图（单值类型）的列定义中
- 作为现存的类型表或类型视图（结构化类型）的类型
- 作为另一个结构化类型的超类型

不能删除缺省类型映射；只能创建另一个类型映射来覆盖它。

数据库管理器尝试删除从属于此单值类型的所有函数。若不能删除该 UDF，则也不能删除该 UDT。若视图、触发器、表检查约束或另一个 UDF 从属于某个 UDF，则不能删除该 UDF。删除 UDT 使使用它的任何软件包或高速缓存的动态 SQL 语句失效。

若为一个 UDT 创建了一个变换，且计划删除该 UDT，应考虑它是否需要删除该变换。这可通过 **DROP TRANSFORM** 语句来完成。有关此语句的详细信息，参考 *SQL Reference*。注意，只能删除您或其他应用程序开发人员定义的变换；不能删除内置变换及其相关的组定义。

有关用户定义类型的更多信息，参考 *SQL Reference* 和 *Application Development Guide* 手册。

## 改变或删除视图

**ALTER VIEW** 语句通过改变引用类型列以添加作用域，来修改现存的视图。您希望对视图所做的任何其他更改都要求删除该视图，然后重新创建它。

当改变视图时，必须将作用域添加到尚未定义作用域的现存的引用类型列中。此外，不能从超视图继承该列。

在 **ALTER VIEW** 语句中列名的数据类型必须是 **REF**（类型表名或类型视图名的类型）。

虽然软件包和高速缓存的动态语句都被标记为无效，但是不会影响其他数据库对象，如表和索引。有关更多信息，参见第189页的『更改对象时的语句从属性』。

有关 ALTER VIEW 语句的附加信息，参考 *SQL Reference*。

要使用“控制中心”来改变视图：

1. 展开对象树，直到您看到**视图**文件夹为止。
2. 右键单击要修改的视图，并从弹出菜单中选择**改变**。
3. 在**改变视图**窗口中，输入或修改注释，并单击**确定**。

要使用命令行来改变视图，输入：

```
ALTER VIEW <view_name> ALTER <column name>  
ADD SCOPE <typed table or view name>
```

要使用“控制中心”来删除视图：

1. 展开对象树，直到您看到**视图**文件夹为止。
2. 右键单击要删除的视图，并从弹出菜单中选择**删除**。
3. 选择**确认框**，并单击**确定**。

要使用命令行来删除视图，输入：

```
DROP VIEW <view_name>
```

以下示例显示如何删除 EMP\_VIEW：

```
DROP VIEW EMP_VIEW
```

从属于要删除的视图的任何其他视图将变得不起作用。（有关更多信息，参见『恢复不起作用的视图』。）

对于表层次结构，可以在一条语句中删除整个视图层次结构，方法是命名该层次结构的根视图，如以下示例所示：

```
DROP VIEW HIERARCHY VPerson
```

有关删除和创建视图的更多信息，参考 *SQL Reference* 手册。

### 恢复不起作用的视图

视图可以变为不起作用的：

- 由于取消对基础表的特权
- 若删除表、别名或功能



- 若超视图变为不起作用
- 当删除它们所从属的视图时。

下列步骤可以帮助您恢复不起作用的视图：

1. 确定最初用于创建该视图的 SQL 语句。可以从 SYSCAT.VIEW 目录视图的 TEXT 列获取此信息。
2. 使用 CREATE VIEW 语句并使用相同的视图名和相同的定义来重新创建该视图。
3. 使用 GRANT 语句重新授予先前在该视图上授予的所有特权。（注意，在不起作用的视图上授予的所有特权都被取消。）

若不希望恢复不起作用的视图，可以使用 DROP VIEW 语句显式删除它，或者可以使用相同的名称和不同的定义来创建一个新视图。

不起作用的视图只在 SYSCAT.TABLES 和 SYSCAT.VIEWS 目录视图中具有项；SYSCAT.VIEWDEP、SYSCAT.TABAUTH、SYSCAT.COLUMNS 和 SYSCAT.COLAUTH 目录视图中的所有项已被除去。

## 删除摘要表

不能改变摘要表，但可删除它。

所有引用该表的索引、主键、外键和检查约束均被删除。所有引用该表的视图和触发器均变得不起作用。从属于任何删除的对象的软件包或被标记为不起作用的软件包均失效。有关软件包从属性的更多信息，参见第189页的『更改对象时的语句从属性』。

要使用“控制中心”来删除摘要表：

- |   |
|---|
| <ol style="list-style-type: none"><li>1. 展开对象树，直到您看到表文件夹为止。</li><li>2. 右键单击要删除的摘要表，并从弹出菜单中选择删除。</li><li>3. 选择确认框，并单击确定。</li></ol> |
|---|

要使用命令行来删除摘要表，输入：

```
DROP TABLE <table_name>
```

以下 SQL 语句删除摘要表 XT：

```
DROP TABLE XT
```

## 恢复不起作用的摘要表

取消基础表的 `SELECT` 特权将会导致摘要表变得不起作用。

下列步骤可帮助您恢复不起作用的摘要表:

- 确定最初用于创建该摘要表的 `SQL` 语句。可以从 `SYSCAT.VIEW` 目录视图的 `TEXT` 列获取此信息。
- 使用 `CREATE SUMMARY` 语句并使用相同的摘要表名和相同的定义，来重新创建该摘要表。
- 使用 `GRANT` 语句重新授予先前在该摘要表上授予的所有特权。（注意，在不起作用的摘要表上授予的所有特权都被取消。）

若不希望恢复不起作用的摘要表，可以使用 `DROP TABLE` 语句显式地删除它，或者可以使用相同的名称但是不同的定义来创建新的摘要表。

不起作用的摘要表只在 `SYSCAT.TABLES` 和 `SYSCAT.VIEWS` 目录视图中具有项；在 `SYSCAT.VIEWDEP`、`SYSCAT.TABAUTH`、`SYSCAT.COLUMNS` 和 `SYSCAT.COLAUTH` 目录视图中的所有项已被除去。

## 删除包装器

`DROP` 语句可从数据库中除去包装器。以下示例显示如何删除 `DRDA` 包装器:

```
DROP WRAPPER DRDA
```

删除从属于该包装器的所有服务器定义、用户定义函数映射和用户定义数据类型映射。还会删除从属于所删除的服务器定义的所有用户定义的映射、别名、用户定义的类型映射和用户映射。删除从属于所删除的别名的任何索引规范，并且将从属于这些别名的任何视图都标记为不起作用。使所有从属于已删除的对象和不起作用的视图的软件包无效。

必须具有 `SYSADM` 或 `DBADM` 权限，才可删除包装器。

有关删除包装器的更多信息，参考 *SQL Reference*。

## 改变或删除服务器

`ALTER SERVER` 语句修改联合数据库目录中现存的服务器定义。使用此语句:

- 修改特定数据源的定义。
- 修改具有特定类型或版本的多个数据源的定义。
- 修改特定数据源的配置。例如，若要将特定服务器标识的 `DBMS` 迁移到具有更快处理器的一个新工作站，应更新 `cpu_ratio` 服务器选项。

不能使用此语句来修改 `dbname` 或 `node` 服务器选项。

以下示例显示如何改变 ORA1 服务器:

```
ALTER SERVER ORA1 OPTIONS (SET CPU_RATIO '5.0')
```

可从联合数据库中删除服务器。以下示例显示如何删除 ORALOC01 服务器:

```
DROP SERVER ORALOC01
```

删除驻留在数据源上的表和视图的所有别名。删除从属于这些别名的任何索引规范。还会删除从属于所删除的服务器定义的任何用户定义函数映射、用户定义类型映射和用户映射。使从属于已删除的服务器定义、函数映射、别名和索引规范的所有软件包无效。

必须具有 SYSADM 或 DBADM 权限, 才可改变或删除服务器。

有关删除和改变服务器的更多信息, 参考 *SQL Reference*。

## 改变或删除别名

ALTER NICKNAME 语句用于更新本地存储的有关数据源表或视图的信息。例如, 可使用此语句来更改某列的本地名称, 或将一种列数据类型映射为另一种数据类型。也可使用此语句来添加列选项。有关 ALTER NICKNAME 语法的更多信息, 参阅 *SQL Reference*。

当删除一个别名时, 基于该别名创建的视图被标记为不起作用。当在视图中引用别名时, 不能改变该别名的列名或数据类型。

必须对联合数据库具有 SYSADM 或 DBADM 权限, 或必须对该别名具有 CONTROL 或 ALL 数据库特权、ALTERIN (对当前模式) 模式特权, 或必须作为别名定义者, 才可使用此语句。

### 改变别名列和删除别名

以下示例显示如何改变别名 TESTNN, 将一列的本地名称从 COL1 更改为 NEWCOL:

```
ALTER NICKNAME TESTNN ALTER COLUMN COL1 LOCAL NAME NEWCOL
```

以下示例显示如何删除别名 TESTNN:

```
DROP NICKNAME TESTNN
```

### 改变别名列选项

可为称为列选项的参数指定值, 来指定列信息。可用大写或小写来指定任何一个值。下面的表描述这些值并提供其他信息。

表 3. 列选项及其设置

选项	有效设置	缺省设置
numeric_string	<p>‘Y’ 是，此列只包含数字数据串。要点：若此列只包含数字串且带尾部空格，建议不要指定 ‘Y’。</p> <p>‘N’ 否，此列不限于数字数据串。</p> <p>将一系列的 numeric_string 设置为 ‘Y’，以告知优化器此列不包含可能干扰列数据排序的空格。当数据源的整理顺序与 DB2 的不同时，此选项可发挥作用。由于整理顺序不同，将从本地（数据源）求值中排除用此选项标记的那些列。</p>	‘N’
varchar_no_trailing_blanks	<p>指示特定的 VARCHAR 列是否缺少尾部空格：</p> <p>‘Y’ 是，此 VARCHAR 列缺少尾部空格。</p> <p>‘N’ 否，此 VARCHAR 列不缺少尾部空格。</p> <p>若数据源的 VARCHAR 列不包含填充的空格，那么优化器用于访问这些列的策略在一定程度上取决于这些列是否包含尾部空格。缺省情况下，优化器“假定”这些列确实包含尾部空格。基于这种假设，它制定了一个访问策略，包括修改查询，以使从这些列返回的值是用户期望的值。但是，若 VARCHAR 列没有尾部空格，而您又让优化器知道这点，它可能会制定一个更有效的访问策略。要告诉优化器特定的列没有尾部空格，在 ALTER NICKNAME 语句中指定该列（有关语法，参见 <i>SQL Reference</i>）。</p>	‘N’

## 删除索引、索引扩充或索引规范

不能更改索引定义、索引扩充或索引规范的任何子句；必须删除该索引或索引扩充，并再次创建它。（删除索引或索引规范不会导致删除任何其他对象，但可能导致一些软件包失效。）

要使用“控制中心”来删除索引、索引扩充或索引规范：

1. 展开对象树，直到您看到索引文件夹为止。
2. 右键单击要删除的索引，并从弹出菜单中选择删除。
3. 选择**确认**框，并单击**确定**。

要使用命令行来删除索引、索引扩充或索引规范，输入：

```
DROP INDEX <index_name>
```

以下 SQL 语句删除称为 PH 的索引：

```
DROP INDEX PH
```

以下 SQL 语句删除称为 IX\_MAP 的索引扩充:

```
DROP INDEX EXTENSION ix_map RESTRICT
```

索引扩充的名称必须标识目录中描述的索引扩充。RESTRICT 子句强制执行一个规则，即不能定义依赖于索引扩充定义的索引。若一个下层索引依赖于此索引扩充，则删除失败。

不能显式删除主键或唯一键索引（除非它是索引规范）。必须使用下列其中一种方法删除它:

- 若主索引或唯一约束是为主键或唯一键自动创建的，则删除主键或唯一键将会使该索引也被删除。使用 ALTER TABLE 语句来执行删除。
- 若主索引或唯一约束是用户定义的，则必须使用 ALTER TABLE 语句先将主键或唯一键删除。在删除主键或唯一键之后，该索引就不再被认为是主索引或唯一索引了，这时可以显式删除它。

任何从属于删除的索引的软件包和高速缓存的动态 SQL 语句都被标记为无效。有关更多信息，参见『更改对象时的语句从属性』。应用程序不受添加或删除索引所导致的更改的影响。

## 更改对象时的语句从属性

语句从属性包括软件包和高速缓存的动态 SQL 语句。软件包是一个数据库对象，它包含数据库管理器为了以最有效的方式访问特定应用程序的数据所需的信息。绑定是创建软件包的过程，当执行应用程序时数据库管理器需要该软件包来访问该数据库。Application Development Guide 详细讨论了如何创建软件包。

软件包和高速缓存的动态 SQL 语句可以从属于很多类型的对象。有关那些对象的完整列表，参考 SQL Reference。

可以显式引用这些对象，例如，在一个 SQL SELECT 语句中涉及的一个表或用户定义函数。也可以隐式引用这些对象，例如，当删除父表中的一行时，为确保不违反参考约束需要检查的从属表。软件包还与已授予软件包创建者的特权有关。

若一个软件包或高速缓存的动态 SQL 语句从属于一个对象而该对象被删除，则该软件包或高速缓存的动态 SQL 语句将被置于“无效”状态。若一个软件包从属于一个用户定义函数而该函数被删除，则该软件包被置于“不起作用”状态。

处于无效状态的高速缓存动态 SQL 语句在下次使用时将被自动重新优化。若该语句需要的一个对象已被删除，则执行该动态 SQL 语句可能失败，并伴有出错信息。

处于无效状态的软件包在下次使用时将被隐式重新绑定。也可以显式重新绑定这种软件包。若由于删除一个触发器而将一个软件包标记为无效，则重新绑定后的软件包不再调用该触发器。

必须显式重新绑定处于不起作用状态的软件包，然后才能使用它。有关绑定和重新绑定软件包的更多信息，参考 *Application Development Guide*。

联合数据库对象具有相似的从属性。例如，删除服务器将使引用与该服务器相关的别名的任何软件包或高速缓存动态 SQL 无效。

在某些情况下，重新绑定软件包是不可能的。例如，若一个表已删除但尚未重新创建，则不能重新绑定该软件包。在这种情况下，需要重新创建该对象或更改该应用程序，以使它不使用删除的对象。

在许多其他情况中，例如，若删除了一个约束，则重新绑定该软件包是可能的。

下列系统目录视图可帮助您确定软件包的状态和软件包的从属性：

- SYSCAT.PACKAGEAUTH
- SYSCAT.PACKAGEDEP
- SYSCAT.PACKAGES

有关对象从属性的更多信息，参考 *SQL Reference* 手册中的 DROP 语句。

---

## 第3部分 数据库安全性





---

## 第5章 控制数据库访问

数据库管理员和系统管理员的一个最重要的责任是保证数据库的安全性。使数据库保密涉及到以下几个任务:

- 防止由于设备或系统失效而意外损失数据或损害数据的完整性。
- 防止越权访问有价值的信息。必须确保敏感信息不被无需知道这些信息的人访问。
- 防止越权人员通过恶意删除或篡改数据来进行破坏。
- 监视用户对数据的访问, 这在第239页的『第6章 审计 DB2 活动』中做了讨论。

讨论下列主题:

- 『为安装选择用户标识和组』
- 第196页的『为服务器选择认证方法』
- 第200页的『远程客户机的认证注意事项』
- 第201页的『分区数据库注意事项』
- 第201页的『使用“DCE 安全性服务”来认证用户』
- 第206页的『联合数据库认证处理』
- 第211页的『特权、权限和授权』
- 第224页的『控制对数据库对象的访问』
- 第233页的『任务和必需的授权』
- 第234页的『使用系统目录』.

**制定安全性计划:** 首先定义数据库访问控制计划的目标, 并指定哪些人在什么条件下可访问哪些内容。您的计划也应当描述如何通过使用数据库函数、其他程序的函数和管理过程来实现这些目标。

---

### 为安装选择用户标识和组

从安装产品的那一刻起, DB2 管理员就应当开始重视安全性问题。各个平台特定的快速入门一书提供了计划、安装和配置 DB2 所需的全部信息。

完成 DB2 安装的步骤需要用户名、组名和密码。在安装期间, 管理员可以选择每个需求的缺省值。一旦在安装 DB2 期间使用了缺省值, 极力建议管理员在创建数

数据库将驻留的实例之前，创建新的用户名、组名和密码。使用新的用户名、组名和密码，将把不是管理员的用户获知缺省值并在实例和数据库中以不当方式使用缺省值的风险降至最低。

密码在认证用户时很重要。若在操作系统级别没有设置认证要求，而数据库正在使用操作系统来认证用户，则将会允许用户连接。例如，在 UNIX 操作系统上，将未定义的密码当作 NULL。而且任何没有已定义的密码的用户都将当作假设它们有一个 NULL 密码。从操作系统方面来看，这是一种匹配，而且验证了用户，且用户能够连接至数据库。若您想要操作系统对您的数据库进行用户认证。则您需要操作系统级别的密码。

在安装 DB2 之后，另一个安全性建议为，更改授予给用户的缺省特权。在安装过程期间，在缺省情况下会将“系统管理” (SYSADM) 特权授予每个操作系统上的下列用户：

**OS/2** 属于“用户简要表管理 (UPM) 管理员”或“本地管理员”组的有效 DB2 用户标识。

**Windows 95 或 Windows 98** 任何 Windows 95 或 Windows 98 用户。

**Windows NT 或 Windows 2000** 属于“管理员”组的有效 DB2 用户名。

**UNIX** 属于实例所有者用户标识的主组的有效 DB2 用户名。

SYSADM 特权是 DB2 中可用的能力最强的一组特权。（将在本章后面讨论特权。）因此，您可能不希望在缺省情况下所有这些用户都具有 SYSADM 特权。DB2 给管理员提供了授予和取消组和个别用户标识的特权的能力。

针对各平台的、创建和指定组和用户标识的信息可在各个快速入门手册中找到。通过更新数据库管理器配置参数 SYSADM\_GROUP，管理员可以控制将哪个组定义为具有“系统管理员”特权的“系统管理”组。必须遵循下面的准则，来满足安装 DB2 及创建实例和数据库的安全性需求。

（通过更新 SYSADM\_GROUP）定义为“系统管理”组的任何组必须存在。此组的名称应该很容易标识为对实例所有者创建的组。属于此组的用户标识和组对它们各自的实例都具有系统管理员权限。

应考虑创建一个容易被识别为与特定实例相关的实例所有者用户标识。此用户标识作为其组的一员，应具有以上创建的 SYSADM 组的名称。另一个建议是只使用此实例所有者用户标识作为实例所有者组的一个成员，而不要在任何其他组中使用它。这种做法应该控制可以修改实例环境的用户标识和组的增加。

创建的用户标识应始终与一个密码相关，以允许在访问该实例内的数据和数据库之前进行认证。创建密码时的建议是，遵循您所在机构的密码命名准则。

## Windows NT 平台注意事项

当在“扩充企业版 Windows NT 版”中工作时，在定义帐户的机器上，任何属于“本地管理员”组的有效 DB2 用户帐户都被授予“系统管理”(SYSADM) 权限。

例如，若用户向一个域帐户登录，并尝试访问 DB2 数据库，则 DB2 转至“域控制器”来列举组（包括“管理员”的组）。可以使用两种方法更改此行为：

1. 设置注册表变量 DB2\_GRP\_LOOKUP = local，并向本地“管理员”组添加域帐户（或全局组）。
2. 更新数据库管理器配置文件，指定一个新组。若要让该组列举在本地机器上，则还必须设置 DB2\_GRP\_LOOKUP 注册表变量。

在 Windows NT 域环境中，缺省情况下，只有属于“主域控制器”(PDC) 上“管理员”组的域用户才对实例具有 SYSADM 权限。因为 DB2 总是在定义帐户的机器上执行授权，所以向服务器上的本地“管理员”组添加域用户并不将域用户 SYSADM 权限授予该组。

为了避免向 PDC 上的“管理员”组添加域用户，应创建一个全局组并添加要授予其 SYSADM 权限的用户（域用户和本地用户）。为此，输入下列命令：

```
DB2STOP
DB2 UPDATE DBM CFG USING SYSADM_GROUP global_group
DB2START
```

## UNIX 平台注意事项

在基于 UNIX 的平台上，必须为受防护的“用户定义函数”(UDF) 和存储过程创建一个组，而且使用受防护的 UDF 或存储过程的任何用户标识必须是此组的一个成员。对于 SYSADM 组，受防护的 UDF 或存储过程组的名称应该能够很容易识别。属于受防护的 UDF 或存储过程组的用户标识将与该组相关的任何权限和特权作为他们的缺省权限。

鉴于安全性原因，我们建议不要使用实例名作为 Fenced ID。但若计划不使用受防护的 UDF 或存储过程，可以将 Fenced ID 设置为实例名，而不用创建另一个用户标识。

建议创建一个将被识别为与此组相关的用户标识。将受防护的 UDF 和存储过程的用户指定为实例创建脚本的参数 (db2icrt ... -u <FencedID>)。若安装了“DB2 客户机”或“DB2 软件开发者工具箱”，则不需要这样做。

## 一般规则

全部对象和用户命名都有规则。某些规则与所用的平台有关。例如，有的规则是关于名称中的大小写使用。

- 在 UNIX 平台上，名称必须小写。
- 在 OS/2 上，名称必须大写。
- 在 Windows 平台上，名称可以是大写、小写或大小写混合。

有关 DB2 命名规则，参见第277页的『附录A. 命名规则』。

db2icrt 命令在实例所有者的主目录下创建主 SQL 库 (sqlib) 目录。

---

## 为服务器选择认证方法

访问实例或数据库首先要求认证用户。每个实例的认证类型确定如何以及在何处验证用户。认证类型存储在服务器上的数据库管理器配置文件中。它是在创建实例时进行的初始设置。有关认证数据库管理器配置参数的更多信息，参考《管理指南：性能》中的“配置 DB2”。每个实例都有一个认证类型，该类型控制对数据库服务器和该服务器控制下的所有数据库的访问。

若打算从联合数据库访问数据源，必须考虑数据源认证处理和联合认证类型的定义。有关更多信息，参见第206页的『联合数据库认证处理』。

提供了下列认证类型：

### SERVER

指定使用本地操作系统安全性在服务器上进行认证。若在连接尝试期间指定了用户标识和密码，则在服务器上将它们与有效的用户标识和密码比较，以确定是否允许该用户访问这个实例。这是缺省的安全性机制。

**注：**服务器代码检测一个连接是本地的还是远程的。对于本地连接，当认证是 SERVER 时，要使认证成功，不需要用户标识和密码。

若远程实例具有 SERVER 认证，则可以有两种进行认证的方式：

- 由用户提供用户标识和密码。
- 由 DB2 检索用户标识和密码，然后将它们传送至服务器进行验证。（用户已登录至本地机器或登录至域。）

### SERVER\_ENCRYPT

指定服务器接受加密的 SERVER 认证方案。若未指定客户机认证，使用在服务器中选择的方法认证客户机。

若客户机认证是 DCS 或 SERVER，可将用户标识和密码传送给该服务器来认证客户机。若客户机认证是 DCS\_ENCRYPT 或 SERVER\_ENCRYPT，可传送用户标识和加密的密码来认证客户机。

若在客户机中指定 SERVER\_ENCRYPT，在服务器中指定 SERVER，由于认证级别不匹配会返回一个错误。

## CLIENT

指定使用操作系统安全性在调用应用程序所在的数据库分区上执行认证。在客户机节点上，将在一个连接尝试期间指定的用户标识和密码与有效的用户标识和密码的组合比较，以确定是否允许此用户标识访问这个实例。不在数据库服务器上执行其他认证。

若用户执行一个本地登录或客户机登录，则只有该本地客户机工作站认识该用户。

若远程实例有 CLIENT 认证，则另两个参数确定最终的认证类型：*trust\_allclnts* 和 *trust\_clntauth*。

### 只用于 TRUSTED 客户机的 CLIENT 级安全性:

可信的客户机是具有可靠的、本地安全性系统的客户机。具体地说，除 Windows 95 和 Windows 98 操作系统外，所有客户机都是可信的客户机。

当已选择认证类型 CLIENT 时，可选择一个附加选项来保护其操作环境没有内在安全性的客户机。

要保护未采取任何保护措施的客户机，管理员可以将 *trust\_allclnts* 参数设置为 NO，来选择“可信的客户机认证”。这意味着所有可信的平台都可以代表服务器认证用户。在服务器上认证不可信的客户机，必须提供一个用户标识和密码。使用 *trust\_allclnts* 配置参数来指示您是否信赖客户机。此参数的缺省值是 YES。

**注：**可以信赖所有客户机（*trust\_allclnts* 为 YES），而让一些客户机作为没有本机保密安全性系统来认证的那些客户机。

甚至对于可信的客户机，您也可能希望在服务器上完成认证。要指示在哪里验证可信的客户机，使用 *trust\_clntauth* 配置参数。此参数的缺省值是 CLIENT。有关此参数的更多信息，参考《管理指南：性能》中的“配置 DB2”。

**注：**仅对于可信的客户机，若在试图 CONNECT 或 ATTACH 时没有显式提供用户标识或密码，则对用户的验证在客户机上进行。*trust\_clntauth* 参数只用于确定在何处验证在 USER/USING 子句上提供的信息。

要避免除 DRDA 客户机外的所有其他客户机访问 DB2 MVS 版和 OS/390 版、DB2 VM 版和 VSE 版以及 DB2 OS/400 版，将 *trust\_allclnts* 参数设置为 DRDAONLY。只有这些客户机受信赖，因此才可执行客户机端的认证。所有其他客户机必须提供用户标识和密码，以供服务器认证。

*trust\_clntauth* 参数用于确定在何处认证以上客户机：若 *trust\_clntauth* 是 "client"，则在客户机处进行认证。若 *trust\_clntauth* 是 "server"，没有提供密码时在客户机处认证，而提供密码时则在服务器处认证。

表 4. 使用 TRUST\_ALLCLNTS 和 TRUST\_CLNTAUTH 参数组合的认证方式。

TRUST_ ALLCLNTS	TRUST_ CLNTAUTH	不提供密码的不可信的非 DRDA 客户机认证	提供密码的不可信的非 DRDA 客户机认证	不提供密码的可信的非 DRDA 客户机认证	提供密码的可信的非 DRDA 客户机认证	不提供密码的 DRDA 客户机认证	提供密码的 DRDA 客户机认证
YES	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT	CLIENT
YES	SERVER	CLIENT	SERVER	CLIENT	SERVER	CLIENT	SERVER
NO	CLIENT	SERVER	SERVER	CLIENT	CLIENT	CLIENT	CLIENT
NO	SERVER	SERVER	SERVER	CLIENT	SERVER	CLIENT	SERVER
DRDAONLY	CLIENT	SERVER	SERVER	SERVER	SERVER	CLIENT	CLIENT
DRDAONLY	SERVER	SERVER	SERVER	SERVER	SERVER	CLIENT	SERVER

**DCS** 主要用于编目使用 DB2 Connect 访问的数据库。（有关此主题的更多详细信息，参考《DB2 Connect 用户指南》中有关“安全性”一节。）当在数据库管理器配置文件中使用时来指定实例的认证类型时，除非正在使用“高级程序到程序通信”（APPC）协议并通过“分布式关系数据库体系结构”（DRDA）的“应用程序服务器”（AS）体系结构来访问服务器，否则，它意味着与 **SERVER** 认证的情况相同。在这种情况下，使用 **DCS** 指示认证将在服务器上进行，但只在 APPC 层进行。进一步的认证将不在 DB2 代码中进行。仅当将连接的 APPC 的 SECURITY 参数指定为 SAME 或 PROGRAM 时，才支持此值。

### DCS\_ENCRYPT

指定 DB2 Connect 接受加密的 SERVER 认证方案。若未指定客户机认证，使用在服务器中选择的方法认证客户机。

若客户机认证是 DCS 或 SERVER，可将用户标识和密码传送给 DB2 Connect 来认证客户机。若客户机认证是 DCS\_ENCRYPT 或 SERVER\_ENCRYPT，可传送用户标识和加密的密码来认证客户机。

若在客户机中指定 DCS\_ENCRYPT，在服务器中指定 DCS，由于认证级别不匹配会返回一个错误。

**DCE** 指定使用“DCE 安全性服务”认证此用户。有关“DCE 安全性”的更多信息，参见第201页的『使用“DCE 安全性服务”来认证用户』。

### **DCE\_SERVER\_ENCRYPT**

指定服务器接受 DCE 认证或加密的 SERVER 认证方案。若客户机认证是 DCE 或未指定，可使用“DCE 安全性服务”认证客户机。有关“DCE 安全性”的更多信息，参见第201页的『使用“DCE 安全性服务”来认证用户』。

若客户机认证是 SERVER 或 DCS，可将用户标识和密码传送给服务器来认证客户机。若客户机认证是 SERVER\_ENCRYPT 或 DCS\_ENCRYPT，可传送用户标识和加密的密码来认证客户机。不能将客户机的认证类型指定为 DCE\_SERVER\_ENCRYPT。若将实例的认证类型指定为 DCE\_SERVER\_ENCRYPT，所有本地应用程序将使用 DCE 作为认证方案。这也适用于不需要数据库连接或实例连接的任何实用程序命令。

除允许 DCE 和 SERVER\_ENCRYPT 的组合认证类型外，DCE\_SERVER\_ENCRYPT 认证类型还消除在 DCE 内使用组时存在的一个限制。当将认证类型设置为 DCE\_SERVER\_ENCRYPT 时，假定在非认证时间请求的组列表来自基本操作系统而非 DCE。作为管理员，可在服务器上设置一个用户来匹配短的 DCE 名称，以便在非认证时间提供组列表支持。

### **KERBEROS**

当 DB2 客户机和服务器都在支持 Kerberos 安全性协议的操作系统上时使用。通过使用传统加密技术来创建共享密钥，Kerberos 安全性协议作为第三方认证服务执行认证。此密钥成为用户的凭证，在所有请求本地或网络服务的场合中，都使用它来验证用户的身份。此密钥消除了将用户名和密码作为明码通信报文通过网络传送这一需要。通过使用 Kerberos 安全性协议，您能够使用单一注册来向远程 DB2 服务器注册。

### **KRB\_SERVER\_ENCRYPT**

指定服务器接受 KERBEROS 认证或加密的 SERVER 认证方案。若客户机认证是 KERBEROS，则使用 Kerberos 安全性系统认证客户机。若客户机认证不是 KERBEROS，则系统认证类型等价于 SERVER\_ENCRYPT。

**注：**Kerberos 认证类型只有在运行 Windows 2000 的客户机和服务器上才受支持。

注:

1. 仅当您有远程数据库客户机在访问数据库时，或当您使用联合数据库功能时，选择的认证类型才显得重要。通过本地客户机来访问数据库的大多数用户始终在此数据库所在的相同机器上进行认证。使用“DCE 安全性服务”时例外。有关支持和使用远程客户机的信息，参考《快速入门》手册。
2. 因为对配置文件本身的访问受到配置文件中信息的保护，所以在更改认证信息时，不要无意中将自己锁在实例之外。下列数据库管理器配置文件参数控制对实例的访问：
  - AUTHENTICATION \*
  - SYSADM\_GROUP \*
  - TRUST\_ALLCLNTS
  - TRUST\_CLNTAUTH
  - SYSCTRL\_GROUP
  - SYSMANT\_GROUP

\* 指示两个最重要的参数，以及最可能引起问题的那些参数。

可以采取一些措施来确保这种情况不会发生：若意外将自己锁在 DB2 系统外，所有平台上都提供了一个故障保险选项，它将允许您使用具有很高特权的本地操作系统安全性用户，以越过通常的 DB2 安全性检查，而更新数据库管理器配置文件。此用户**始终**具有更新数据库管理器配置文件并校正该问题的特权。但是，这种绕过安全性检查的做法只限于对数据库管理器配置文件进行本地更新。不能在远程或对任何其他 DB2 命令使用故障保险用户。此特殊用户被标识为：

- UNIX 平台：实例所有者
  - NT 平台：本地“管理员”组的成员
  - OS/2 平台：UPM 管理员
  - 其他平台：由于在其他平台上没有本地安全性，因此所有用户无论如何都要通过本地安全性检查
3. 有关 Windows NT 安全性的其他信息，参见第323页的『附录E. DB2 Windows NT 版如何使用 Windows NT 的安全性』。

---

## 远程客户机的认证注意事项

当编目一个数据库以便远程访问时，可在数据库目录项中指定认证类型。

对于使用 DB2 Connect 来访问的数据库：若未指定值，假定进行 SERVER 认证。



对远程访问但不是使用 DB2 Connect 来访问的数据库：不需要认证类型。但是，若未指定它，则客户机在开始认证流程前，必须首先与服务器联系以获得此值。若已指定，则只要指定的值与服务器上的值匹配，认证就可以立即开始。若检测到不匹配，则 DB2 会尝试恢复，这可能导致更多的流量来协调该差别，或导致错误（若 DB2 不能恢复的话）。在不匹配的情况下，认为服务器上的值是正确的。

---

## 分区数据库注意事项

在一个分区数据库中，必须为数据库的每个分区定义同一组用户和组。若这些定义不相同，则用户也许是被授权在不同的分区上做不同的事情。建议所有分区保持一致。

---

## 使用“DCE 安全性服务”来认证用户

当考虑分布式数据库环境的安全性时，“分布式计算环境 (DCE) 安全性服务”是一个比较好的选择，因为 DCE 提供：

- 用户和密码的集中管理。
- 不传透明码文本密码和用户标识。
- 每个用户单独注册。

DB2 支持 DCE 缺省登录上下文、连接登录上下文和代理上下文。缺省登录上下文是当用户在客户机上执行 `dce_login` 时建立的。后续的 DB2 命令可以访问此上下文，并且可以执行用户认证而不需要进一步的用户干预（即，不需要用户标识或密码）。连接登录上下文是使用用户标识和密码为 DB2 会话建立的，而用户标识和密码是在使用 `USER/USING` 子句的 `CONNECT` 或 `ATTACH` 上提供的。最后，代理登录上下文是在将一个 DB2 客户机用作一个 DCE 服务器应用程序的一部分时发生的。DCE 服务器应用程序（也是 DB2 客户机）从一个 DCE 客户机应用程序接收到请求，用户的原始身份从该处产生。只要正确配置 DCE 客户机和 DCE 服务器以允许 DCE 服务器成为 DCE 客户机的一个代理，则 DB2 将获得代理标记，并将它转发给 DB2 服务器。这允许 DB2 服务器使用 DCE 客户机的原始身份，而不是使用 DCE 服务器的身份，来处理请求。有关如何建立一个代理登录上下文的信息，可以参考适合您的平台的 DCE 文档。

**注：**有若干种支持 DCE 的供应商产品。为确保 DB2 UDB Windows NT 版可以在安全性服务范围内使用 IBM 的 DCE 产品，提供了两个新的 DLL：`db2dces.ibm` 和 `db2dcec.ibm`。（这些 DLL 文件仅适用于 Windows NT。）若购买并使用 IBM 的 DCE 产品来提供安全性服务，则必须将这两个文件分别复制为 `db2dces.dll` 和 `db2dcec.dll`。若考虑另一个供应商的 DCE 产品，则应该与该供应商服务机构和 DB2 UDB 服务机构联系，以讨论该供应商为安全性服务所提供的 DCE 实现是否将使用 DB2 UDB。

## 如何设置 DB2 用户来使用 DCE

在使用 DB2 前，用户必须在“分布式计算环境”(DCE)的注册表中注册，且具有正确的属性。有关如何创建一个 DCE 主体的信息，参见针对该平台的合适的 DCE 文档。

对于希望使用 DCE 认证服务器的每个 DB2 用户，必须在启用了客户机标志的“DCE 注册表”中定义一个 DCE 主体和帐户。此主体也必须在它的“扩充注册表属性”(ERA)部分中有一项，该项目显示当此主体与特定的 DCE 认证服务器连接时将用于此主体的授权名。

您也许希望让用户主体成为组的成员，以便在数据库中使用组特权。在组 ERA 中的相似信息会将组名映射为一个 DB2 授权名。此授权名是辅助授权名，但存在相同的限制。有关如何创建组和添加成员的其他信息，请参考您的 DCE 文档。

ERA 中的信息将一个用户的 DCE 主体或组名映射为一个特定服务器的 DCE 主体名称的 DB2 授权名。要使用 ERA，必须定义一个 ERA 模式，以指示此属性的格式。这需要对每个 DCE 单元执行一次，并且通过执行如下步骤来完成：

1. 作为一个有效的 DCE 管理员登录到 DCE
2. 调用 `dcecp` 并在提示符处输入下列内容：

```
> xattrschema create ./:/sec/xattrschema/db2map \  
> -aclmgr {{principal r m r m } {group r m r m }} \  
> -annotation {Schema entry for DB2 database access} \  
> -encoding stringarray \  
> -multivalued no \  
> -uuid 1cbe84ca-9df3-11cf-84cd-02608c2cd17b
```

这就创建了“扩充注册表属性”`db2map`。

要查看此映射，在 `dcecp` 提示符处发出如下命令：

```
> xattrschema show ./:/sec/xattrschema/db2map
```

将看到下列内容：

```
{axlmgr  
  {{principal {{query r} {update m} {test r} {delete m}}}  
   {group {{query r} {update m} {test r} {delete m}}}}}  
{annotation {Schema entry for DB2 database access}}  
{applydefs no}  
{intercell rejects}  
{multivalued no}  
{reserved no}  
{scope {}}  
{trigbind {}}  
{trigtype none}  
{unique no}  
{uuid 1cbe84ca-9df3-11cf-84cd-02608c2cd17b}
```

**注:** DCE 不强制对 ERA 中记录的授权名的内容施加限制。若一个 DCE 主体或组被授予一个无效的授权名, 则当 DB2 尝试认证该用户时会出错。(在 CONNECT、ATTACH、DB2START 或需要认证的任何其他操作中可能会重新调用该认证。) 而且极力建议您保证将授权名逐个赋予 DCE 主体, 并且该名称应是唯一的。DCE 不检查这些条件。

若 DB2 客户机要访问 DB2 UDB 服务器, 则一旦它们作为 DCE 主体注册, 就必须添加该 ERA 信息以提供主体名称至授权名的映射。这必须对每个用户或组执行一次, 并通过执行如下步骤来完成:

- 作为一个有效的 DCE 管理员登录到 DCE
- 调用 dcecp 并在提示符处输入如下内容:

```
> principal modify principal_name \  
> -add {db2map map_1 map_2...map_n}
```

其中 map\_n 使用如下格式:

```
DCE_server_principal,DB2_authid
```

其中 DCE\_server\_principal 是一个 DB2 UDB 服务器的有效的 DCE 主体名 (或者是通配符 \*, 它指示此映射对还未在另一个 map\_n 项中指定的任何 DB2 服务器有效), 而 DB2\_authid 是有效的 DB2 授权名。

若 DCE 组将用于 DCE 主体, 则它还必须具有一个至 DB2 authid 的映射, 该 authid 具有正确的权限, 如 SYSADM 或 SYSCTRL 权限。

请注意, 在 DCE 模式中指定的授权标识符 (authid) 必须用大写字体指定, 该模式用于将 DCE 主体名映射为 DB2 authid。使用小写字体或大小写混合的 authid 将导致错误。

## 如何设置 DB2 服务器来使用 DCE

服务器必须在“分布式计算环境”(DCE) 的注册表中注册为主体, 并具有正确的属性, 才可使用 DB2。有关如何创建 DCE 服务器主体的信息, 参见针对该平台相关的合适的 DCE 文档。

必须安装“DCE 安全性”客户机运行时代码, 并且服务器实例可以访问它。

希望使用 DCE 作为认证机制的每个 DB2 服务器, 必须在发出 DB2START 时向 DCE 登录。为避免必须以手工方式执行此操作, DCE 提供了一个方法, 通过此方法, 服务器在一个称为键表文件的特殊文件中维护它自己的用户标识和密码(密钥)信息。在执行 DB2START 时, DB2 读取数据库管理器配置文件, 并获取实例的认证类型。若它发现认证类型是 DCE, 则 DB2 服务器执行 DCE 调用以从键

表文件获取信息。此信息用于向 DCE 注册该服务器。此注册允许服务器从 DCE 客户机接受 DCE 标记，并使用它们来认证这些用户。

实例管理员必须使用 DCE 命令为实例创建键表文件。有关如何创建键表文件的详细信息包括在适合您平台的 DCE 文档中。在该文档中，参考与键表文件及命令 *dcecp keytab* 或 *rgy\_edit* 相关的详细信息。DB2 键表文件必须命名为 *keytab.db2*，并且必须驻留在此实例的 *sqllib* 目录下 *security* 子目录中。（对于基于 Intel 的操作系统，此文件必须驻留在 *sqllib* 目录的 *INSTANCENAME* 子目录的 *security* 子目录中。*INSTANCENAME* 是正使用的实例的实例名。）对于指定实例的服务器主体，它应只包含一项；若有任何其他内容，在执行 *DB2START* 时将会出错。在 UNIX 操作系统平台上，必须借助文件许可权来保护此文件，以便只允许实例所有者进行读/写。

以下是创建键表文件的一个示例：

- 以一个有效的 DCE 用户身份登录到 DCE
- 调用 *rgy\_edit*，并在提示符处输入如下内容：

```
> ktadd -p principal_name -pw principal_password \  
> -f keytab.db2
```

要在 DCE 配置完成时使用 DCE 认证来启动 DB2，必须用“DCE”认证类型更新数据库管理器配置文件，以告诉 DB2 它要使用 DCE 认证。这可通过发出如下 CLP 命令来完成：

```
db2 update database manager configuration using authentication DCE  
sysadm_group DCE_group_name
```

然后执行 *dce\_login*，以成为一个具有 *SYSADM* 权限的有效的 DB2 DCE 用户，并发出 *DB2START*。

**注：**在使用 DCE 认证来启动 DB2 之前，确保已定义了一个 DCE 用户主体以用作该实例的 *SYSADM*，从而使您有一个有效的 DCE 用户标识，可使用它来启动、停止和管理这个实例。有关如何执行此操作的指示，请参见第202页的『如何设置 DB2 用户来使用 DCE』。

除这些指示之外，确保创建的主体是实例的 *SYSADM\_GROUP* 的一个成员。缺省情况下，当没有明确指定任何组时（即，当 *SYSADM\_GROUP* 为空值时），此组名是 DCE 认证的 *DB2ADMIN*，但是在将该实例的认证类型更改为您所选的组名（授权名）之前可以更新它。必须为所选的 DCE 组定义一个 ERA，此 ERA 把它映射为指定的 *SYSADM\_GROUP* 授权名。

“DB2 管理服务器”的功能之一是启动 DB2 实例。当 *AUTHENTICATION* = *DCE* 时，在该实例的 DB2 键表文件中使用的 DCE 主体必须具有一个从

DCE 主体至 DB2 authid 的有效映射。“DB2 管理服务器”需要此映射，才能启动 DB2 实例。有效的映射允许此标识充当客户机和服务器。

## 如何设置 DB2 客户机实例来使用 DCE

可建立一个纯客户机实例，来对本地操作使用 DCE 认证，方法是更新数据库管理器配置文件，并将认证类型设置为 DCE。不要求纯客户机实例有一个键表文件，因为没有服务器需要向 DCE 注册。通常，不建议（或不需要）纯客户机的 DB2 实例使用 DCE 认证，但是支持它。

希望使用 DCE 安全性来访问远程数据库的客户机需要访问适用的“DCE 安全性”产品。或者客户机可选择在数据库目录中为目标数据库编目认证类型。若客户机选择指定 DCE 认证，必须也指定全限定 DCE 服务器主体名。若在该目录中未指定 DCE 认证，则在 CONNECT 时从服务器获取认证和主体信息。

## 使用 DCE 安全性的 DB2 限制

使用 DCE 认证会对 DB2 提供的、与组支持相关的特定 SQL 函数施加一些限制。当使用 DCE 认证时存在如下限制：

- 当使用 GRANT 或 REVOKE 语句时，必须指定关键字 USER 和 GROUP 以限定指定的授权名，否则会发出错误。
- 当使用 CREATE SCHEMA 语句的 AUTHORIZATION 子句时，在评估执行此子句后的语句所需的授权时，将不考虑指定的授权名的组成员资格。这可能导致在执行 CREATE SCHEMA 语句期间授权失败。
- 当软件包的原始绑定者以外的用户重新绑定该软件包时，要重新评估此原始绑定者特权。在这种情况下，在重新评估特权时不考虑原始绑定者的组成员资格。这可能导致在重新绑定期间授权失败。

由 DB2 执行的 DCE 认证会传送“DCE 凭单”，这些凭单是使用 OSF DCE 的“类属安全性服务应用程序编程接口”（GSSAPI）来获得的。这样，所有对“DCE 安全性”的认证都在数据库协议层发生。某些通信机制会提供附加的通信层安全性，该安全性不需要与 DCE 集成到一起。在通信层认证可以保持与数据库协议层认证完全无关的情况下，将不施加任何限制。但是，必须同时满足数据库协议层与通信层认证的标准，才能成功地建立连接。在数据库协议层与通信协议层认证机制交互作用的情况下，若某些组合导致安全性漏洞，那么可以限制它们的使用。

DCE 认证可以与 TCPIP SOCKS 支持一起使用；然而，这两个安全性机制的作用是相互独立的。这可能意味着用户不仅必须提供有效的 DCE 登录上下文，还必须登录为符合“SOCKS 服务器”标准的本地操作系统用户标识。

DCE 认证可以与“NT 命名管道”一起使用；然而，这两个安全性机制的作用是相互独立的。用户不仅必须提供有效的 DCE 登录上下文，还必须作为一个符合“NT 命名管道”支持标准的用户标识登录到“NT 域”。

为了避免同时使用 DCE 主体和本地操作系统用户标识来认证可能造成的混淆，如上述两个示例中的情况，可以使用集成的 DCE 登录。在这种情况下，当向系统登录时，用户同时还自动地登录为适当的 DCE 主体。有关如何使用此功能部件（若受支持的话）的详细信息，参见适合您平台的 DCE 文档。注意，在使用此方法时，DCE 主体与本地操作系统标识使用相同的名称。这可能意味着 DCE 加密凭单中所包含的相同值也在通信层的线路上以未加密的形式流动。

当 SECURITY 参数设置为 NONE 时，DCE 认证只可以用于 APPC 通信。这是为了避免这样一种可能性，即在通信层发送未加密的主体和 / 或密码，而在数据库协议层中对同一主体使用加密的 DCE 标记。这时 DB2 不支持 APPC 层的“DCE 安全性”。

---

## 联合数据库认证处理

若安装了分布式连接安装功能部件，并将数据库管理器配置变量 *federated* 设置为 'YES'，则 DB2 系统作为联合系统运行。在联合系统中的数据库认证设置与标准的 DB2 定义略有不同。更重要的是，在联合系统中必须考虑数据源的认证需求。通常将数据源（DB2、Oracle、DB2 OS/390 版等）设置为需要认证。这表示必须确保标识和密码（需要时）可传送至数据源。DB2 提供了几种方法来支持在数据源的认证，所有方法在本节内均有说明。

## 认证设置

### SERVER

指定与 DB2 连接的客户机要提供用户标识和密码来访问 DB2。在这种情况下，用户标识和密码可传送到数据源。通过服务器选项和用户映射来控制实际传送到数据源的数据，但是认证信息是可传送到数据源的。

### CLIENT

指定在使用操作系统安全性调用应用程序所在的数据库分区上进行认证。密码不能直接传送到数据源。在这种情况下，若数据源需要认证，必须创建一个或多个用户映射。还必须确保服务器选项设置正确，以便将正确的用户标识和密码信息传送到数据源。

当使用 CLIENT 认证时应格外小心。仅对安全网络考虑这种形式的认证。当满足下列条件时，用户对联合数据库拥有 SYSADM 权限：

- 将认证设置为 CLIENT。
- 用户在客户机上具有根状态。

- 用户知道 SYSADM 的授权名。
- 用户在客户机上定义一个与 DB2 上的 SYSADM 相同的授权名。

**DCS** 指定认证在数据源而不是在 DB2 中进行。在这种情况下，将绕过标准的 DB2 认证处理。取决于服务器选项设置，用户标识和密码可直接传送到数据源。认证仅在 Oracle 或 DB2 系列数据源进行。

当将认证设置为 DCS 时要小心。既不在客户机也不在 DB2 执行认证。知道 SYSADM 认证名称的任何用户都可以认为是具有联合服务器的 SYSADM 权限。

**DCE** 若将认证设置为 DCE，只有用户标识可传送到数据源。而密码则不能。若数据源需要认证处理（用户标识和密码），必须定义将密码（可能还有用户标识）发送到数据源的用户映射。若数据源信赖 DB2 连接，则不需要用户映射，因为可将外部安全性系统接收到的标识传送到数据源。

也可进行其他 DB2 认证设置，而一个或多个这种设置可将在 DB2 的密码传送到数据源。若 DB2 和客户机认证设置导致将密码传送到 DB2，该密码可用于在数据源的附加认证处理。有关更多信息，参见第198页的表4。

## 将用户标识和密码传送到数据源

有四种方法控制将认证信息传送到数据源：DB2 认证设置、用户映射、服务器选项和 APPC 安全性设置：

### 认证设置

本节的目的是说明认证设置如何影响联合系统中的全局认证处理（有关认证设置的定义，参见第206页的『认证设置』）。例如，若将 DB2 认证设置为 SERVER 或 DCS，则需要用户标识和密码来建立连接。因此，用户标识和密码可传送到数据源。若将认证设置为 DCE 或 CLIENT，且认证不在包含联合数据库的 DB2 系统执行，则只有用户标识可以发送。若数据源认证处理需要密码（也许是不同的用户标识和密码），则必须创建一个用户映射。若将认证设置为 CLIENT，且 *trust\_clntauth* 参数设置是 SERVER，则可以将密码发送到 DB2，也可以发送到数据源。

### 用户映射

DB2 可发送用于连接 DB2 的授权名或在 DB2 定义的授权名。用户映射存储在 DB2 定义的授权名。它们是用 CREATE USER MAPPING 语句创建的。

用户映射很灵活：可将一个标识映射为新的标识和密码或只映射为密码。可使用它们来提供丢失的信息，或将标识和密码更改为数据源接受的值。

要创建或改变用户映射，必须具有 SYSADM 或 DBADM 权限，或认证标识必须与为该语句指定的授权名匹配。

用户映射语句的一个示例是：

```
CREATE USER MAPPING FOR "SHAWN" SERVER DB21 OPTIONS (REMOTE_AUTHID "SHAWNBCA",  
REMOTE_PASSWORD "MAPLELEAF")
```

其中 DB2 认证标识 (SHAWN) 映射为名称为 DB21 的服务器的远程标识 SHAWNBCA 和远程密码 MAPLELEAF。

若 DB2 的授权名（或密码）与数据源的授权名（或密码）之间的唯一区别是传送字符串的字体，可考虑使用服务器选项将该字体转换为期望的设置，而不是创建新的标识和密码。有关更多信息，参见『服务器选项』。

当认证设置是 DCE 并且数据源需要认证处理（需要密码）时，必须创建用户映射。DB2 只把 DCE 用户标识传送到数据源。必须将密码映射为该用户标识，然后发送到数据源。

### 服务器选项

服务器选项可用于提供整体认证支持。使用这些选项指示是否将密码传送到数据源（典型情况为是），以及是否需要将用户标识和密码转换为大写或小写。使用 CREATE SERVER、ALTER SERVER 和 SET SERVER OPTION 语句设置服务器选项。

本节的其余部分讨论针对认证处理的服务器选项。在第134页的『使用服务器选项来帮助定义数据源并简化认证处理』中提供了服务器选项的更完整的列表。

**密码服务器选项：** 密码的缺省设置是 'Y'（将密码发送到数据源）。对于数据源将执行认证并且不期望加密密码的所有情况，保持或设置此选项为 'Y'。

DB2 可传送加密的密码。若应将密码以加密形式发送到 DB2 系列数据源，可将服务器密码选项设置为 'ENCRYPTION'。若在 DB2 的认证设置是 DCS\_ENCRYPT 或 SERVER\_ENCRYPT，建议将密码设置为 'ENCRYPTION'。

总是将用户标识发送到数据源。

**ID 和密码转换选项：** 某些情况下，授权名和密码可能需要更改。不同的数据源对标识和密码有不同的授权名和密码要求（关于大写或小写的使用）。

DB2 提供两个服务器选项，有助于您处理命名的不同。这两个选项名是 **fold\_id** 和 **fold\_pw**，它们的设置为：

'U' 在将授权名或密码发送到数据源之前，DB2 将它们转换为大写字体。



**'N'** DB2 不转换授权名或密码。

**'L'** 在将授权名或密码发送到数据源之前，DB2 将它们转换为小写字体。

**空值** DB2 首先以大写字体发送授权名或密码；若失败，DB2 将它转换为小写然后再发送。

空值设置似乎更有用，因为它涵盖了许多可能性。但是，从性能角度出发，最好对这些选项进行设置，以便只进行一次尝试就可建立连接。若 `fold_id` 和 `fold_pw` 选项均设置为空值，则 DB2 有可能将进行四次尝试来发送授权名和密码：

1. 授权名和密码都是大写。
2. 授权名是大写而密码是小写。
3. 授权名是小写而密码是大写。
4. 授权名和密码都是小写。

### **APPC 安全性设置**

若要通过 APPC 与需要用户标识和密码的 DRDA 数据源连接，或者若认证设置是 DCS 且您要在 DRDA 数据源进行认证，确保 DB2 和数据源之间的连接的 APPC 安全性设置是 PROGRAM。

## **联合数据库认证示例**

本节为联合系统认证和授权步骤提供了一个概述。有关联合数据库认证和授权处理的概述，参见第210页的图3。

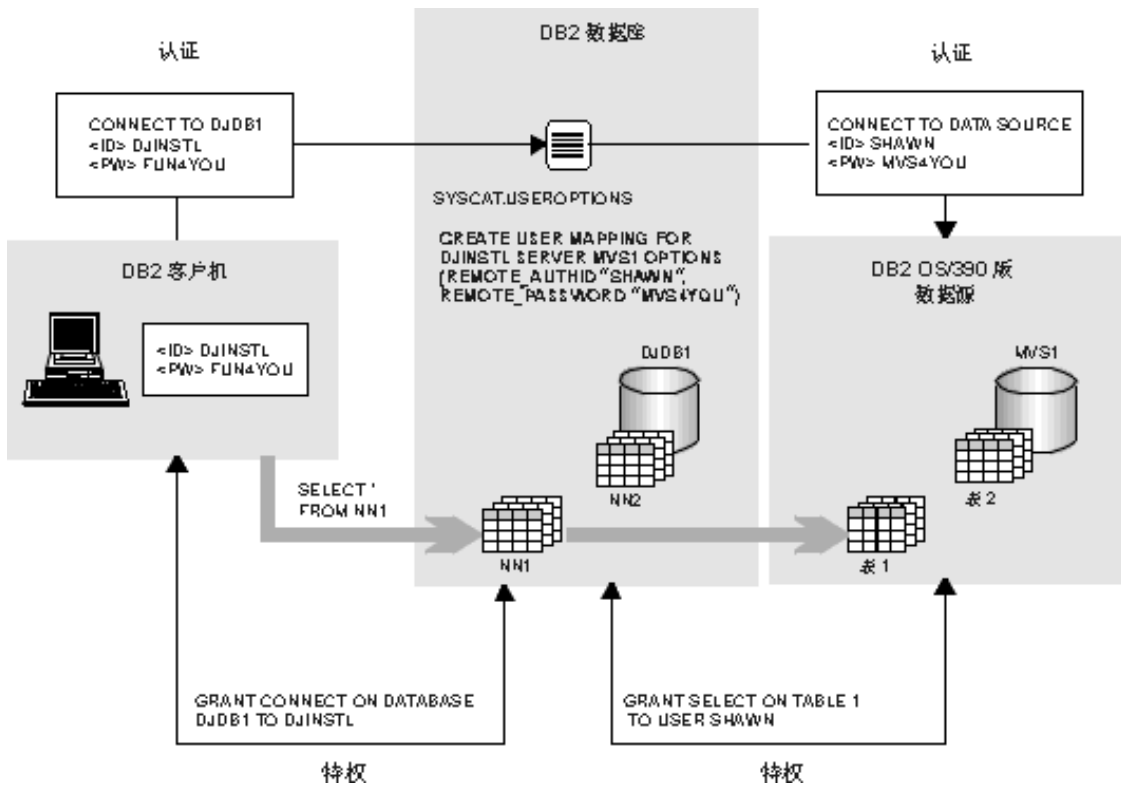


图 3. 联合数据库认证和授权处理

此方案中的任务是允许用户 DJINSTL 根据两个别名 (NN1 和 NN2) 执行 UNION 操作。这两个别名代表两个表。一个数据源是 DB2 OS/390 版系统，在该系统中 DJINSTL 具有名称为 MVS1 的另一个用户标识和密码（参见图3）。需要创建一个用户映射，以访问 MVS1 中的信息。另一个数据源是 DB2 系统，在该系统中 DJINSTL 的标识和密码是相同的。数据源 DB21 只要求以大写字体发送用户标识和密码。

将 DB2 认证设置为 SERVER。DJINSTL 将通过 TCP/IP 连接从 Windows NT 客户机访问 DB2。从 DB2 到 DB2 OS/390 版的连接也是 TCP/IP。联合数据库名是 DJDB1。

首先确保 DB2 期望密码并且已发送密码。还要确保客户机和服务器认证类型匹配。从 DB2 服务器发出以下命令：

```
GET DATABASE MANAGER CONFIGURATION
```

检查 DB2 服务器认证类型。从客户机发出以下命令：

## LIST DATABASE DIRECTORY

检查客户机认证类型。在这两种情况下，均要确保将认证设置为 `SERVER`。若客户机的设置是 `DCS` 或 `CLIENT`，可使用 `UNCATALOG DATABASE` 和 `CATALOG DATABASE` 命令更改它。

下一步，确保将密码发送到数据源。与联合数据库 `DJDB1` 连接后，发出下列命令：

```
ALTER SERVER MVS1 OPTIONS (SET password 'Y')
ALTER SERVER DB21 OPTIONS (SET password 'Y')
```

然后，确保将密码以正确的字体发送到 `DB21` 数据源：

```
ALTER SERVER DB21 OPTIONS (ADD fold_id 'U')
ALTER SERVER DB21 OPTIONS (ADD fold_pw 'U')
```

再下一步是授予特权，允许用户 `DJINSTL` 与联合数据库 `DJDB1` 连接并选择别名：

```
GRANT CONNECT ON DATABASE DJDB1 TO DJINSTL;
```

现在，将 `DJINSTL` 的 `DB2` 标识和密码映射为 `MVS1` 服务器的正确的用户标识和密码：

```
CREATE USER MAPPING FOR "DJINSTL" SERVER MVS1 OPTIONS (REMOTE_AUTHID "SHAWN",
    REMOTE_PASSWORD "MVS4YOU")
```

此时，`DB2` 用户标识 `DJINSTL` 可将请求发送到数据源。可能需要附加的步骤来访问由别名引用的数据源对象（通常需要特权，才可访问别名引用的表和视图）。

---

## 特权、权限和授权

特权使用户能够创建或访问数据库资源。权限级提供一个方法，以将特权及高级数据库管理器维护和实用程序操作分组。这两者一起用于控制对数据库管理器和它的数据库对象的访问。用户只能访问那些他们具有适当授权（即必需的特权或权限）的对象。

存在下列权限：

- 第213页的『系统管理权限 (SYSADM)』
- 第214页的『系统控制权限 (SYSCTRL)』
- 第215页的『系统维护权限 (SYSMAINT)』
- 第215页的『数据库管理权限 (DBADM)』
- 第216页的『LOAD 权限』

存在下列类型的特权：

- 第216页的『数据库特权』
- 第218页的『模式特权』
- 第220页的『表空间特权』
- 第220页的『表和视图特权』
- 第222页的『别名特权』
- 第222页的『服务器特权』
- 第223页的『软件包特权』
- 第223页的『索引特权』。

图4举例说明权限及其控制范围（数据库、数据库管理器）之间的关系。

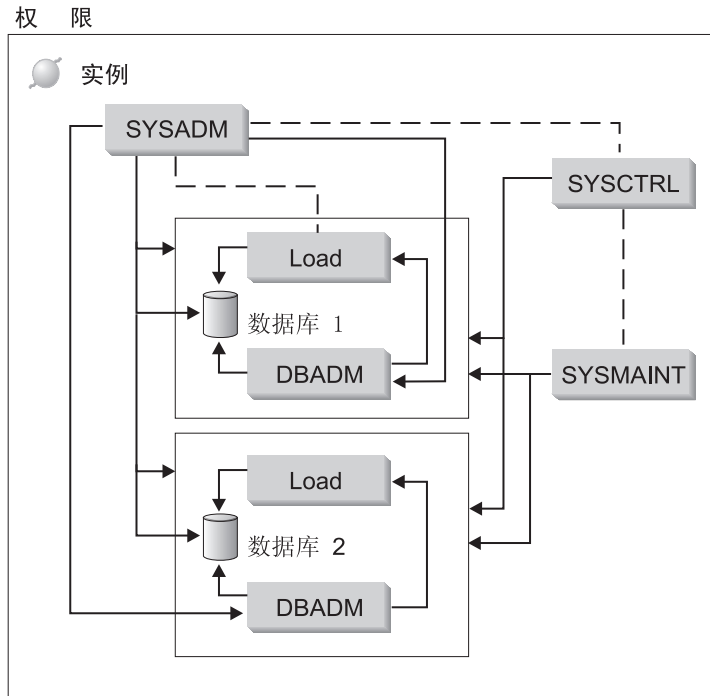


图 4. 权限的层次结构

用户或组可以具有一个或多个以下级别的授权：

- 管理权限（SYSADM 或 DBADM）提供对一组对象的全部特权。
- 系统权限（SYSCTRL 或 SYSMAINT）提供用于管理系统的全部特权，但是不允许访问数据。

- “装入”权限 (LOAD) 授予“装入”实用程序或“自动装入程序”实用程序将数据装入表的特权。
- 所有权特权 (在某些情况下又称为 CONTROL 特权) 提供对特定对象的全部特权。
- 可以授予个别特权以允许用户对特定对象执行特定的功能。
- 可以将隐式特权授予具有执行软件包的权限的用户。当用户可运行应用程序时，他们不一定需要对软件包内使用的数据对象具有显式特权。有关更多信息，参见第227页的『允许通过软件包授予间接特权』。

具有管理权限 (SYSADM 或 DBADM) 或所有权特权 (CONTROL) 的用户可以使用 GRANT 和 REVOKE 语句来向其他用户授予特权，或从其他用户取消特权。(参见第224页的『控制对数据库对象的访问』。) 还可以将一个表、视图或模式特权授予另一个用户 (若该特权保留有 WITH GRANT OPTION)。但是，WITH GRANT OPTION 不允许授予该特权的人在授权后取消该特权。必须具有 SYSADM 权限、DBADM 权限或 CONTROL 特权才能取消该特权。

可以将个别特权或权限的任何组合授予一个用户或组。当一个特权与一个资源相关时，该资源必须存在。例如，除非一个表先前已创建，否则不能授予用户对该表的 SELECT 特权。

**注：**当授予一个授权名权限和特权且没有使用该授权名创建用户时，必须小心。稍后，可以使用该授权名创建一个用户，并且该用户自动接收与该授权名相关的所有权限和特权。

有关特定的命令、API 或 SQL 语句需要何种授权的信息，参考 *Command Reference*、*Administrative API Reference* 或 *SQL Reference*。

## 系统管理权限 (SYSADM)

SYSADM 权限是最高级别的管理权限。具有 SYSADM 权限的用户可以运行实用程序，发出数据库及数据库管理器命令，访问在此数据库管理器实例内任何数据库中的任何表中的数据。它提供控制此实例中所有数据库对象的能力，这些对象包括数据库、表、视图、索引、软件包、模式、服务器、别名、数据类型、函数、过程、触发器、表空间、节点组、缓冲池和事件监视器。

将 SYSADM 权限赋予由 *sysadm\_group* 配置参数指定的组 (参考《管理指南：性能》中的“配置 DB2”)。通过您的平台上使用的安全性设施，在数据库管理器外控制该组的成员资格。有关如何使用系统安全性设施来创建、更改或删除 SYSADM 权限的信息，参考《快速入门》。

只有具有 SYSADM 权限的用户才可以执行下列功能：

- 迁移数据库
- 更改数据库管理器配置文件（包括指定具有 SYSCTRL 或 SYSMANT 权限的组）
- 授予 DBADM 权限。

另外，具有 SYSADM 权限的用户可以执行具有下列权限的用户的功能：

- 『系统控制权限 (SYSCTRL)』
- 第215页的『系统维护权限 (SYSMAINT)』
- 第215页的『数据库管理权限 (DBADM)』

**注：**当具有 SYSADM 权限的用户创建数据库时，将自动授予他们对此数据库的显式 DBADM 权限。若从 SYSADM 组中除去了此数据库创建者，而且您也想阻止他们作为 DBADM 访问该数据库，则必须显式取消此 DBADM 权限。

## 系统控制权限 (SYSCTRL)

SYSCTRL 权限是最高级别的系统控制权限。此权限提供对数据库管理器实例和其数据库执行维护和实用程序操作的能力。这些操作可以影响系统资源，但是它们并不允许直接访问此数据库中的数据。系统控制权限是为管理包含敏感数据的数据库管理器实例的用户而设计的。

将 SYSCTRL 权限赋予由 *sysctrl\_group* 配置参数指定的组（参考《管理指南：性能》中的“配置 DB2”）。若指定了一个组，则通过您的平台上使用的安全性设施，在数据库管理器外控制该组的成员资格。

只有具有 SYSCTRL 权限或更高权限的用户才可以执行下列操作：

- 更新数据库、节点或分布式连接服务 (DCS) 目录
- 迫使用户从系统注销
- 创建或删除一个数据库
- 删除、创建或改变一个表空间
- 复原为新数据库。

另外，具有 SYSCTRL 权限的用户可以执行具有第215页的『系统维护权限 (SYSMAINT)』权限的用户的功能。

具有 SYSCTRL 权限的用户也有与一个数据库连接的隐式特权。

**注：**当具有 SYSCTRL 权限的用户创建数据库时，将自动授予他们对此数据库的显式 DBADM 权限。若从 SYSCTRL 组中除去了此数据库创建者，而且您也想阻止他们作为 DBADM 访问该数据库，则必须显式取消此 DBADM 权限。

## 系统维护权限 (SYSMAINT)

SYSMAINT 权限是第二级别的系统控制权限。此权限提供对数据库管理器实例和其数据库执行维护和实用程序操作的能力。这些操作可以影响系统资源，但是它们并不允许直接访问此数据库中的数据。系统维护权限是为某些用户设计的，这些用户维护包含敏感数据的数据库管理器实例中的数据库。

将 SYSMAINT 权限赋予由 *sysmaint\_group* 配置参数指定的组（参考《管理指南：性能》中的“配置 DB2”）。若指定了一个组，则通过您的平台上使用的安全性设施，在数据库管理器外控制该组的成员资格。

只有具有 SYSMAINT 或更高系统权限的用户才可以执行下列操作：

- 更新数据库配置文件
- 备份数据库或表空间
- 复原为现存的数据库
- 执行前滚恢复
- 启动或停止实例
- 复原表空间
- 运行跟踪
- 捕捉数据库管理器实例或其数据库的数据库系统监控器快照。

具有 SYSMAINT、DBADM 或更高权限的用户可以执行下列操作：

- 查询表空间的状态
- 更新日志历史文件
- 停止表空间
- 重组表
- 使用 RUNSTATS 实用程序收集目录统计信息。

具有 SYSMAINT 权限的用户也有与一个数据库连接的隐式特权。

## 数据库管理权限 (DBADM)

DBADM 权限是第二个最高级别的管理权限。它只适用于特定的数据库，并允许用户运行特定的实用程序，发出数据库命令，访问该数据库中的任何表中的数据。当授予 DBADM 权限时，也同时授予了 BINDADD、CONNECT、CREATETAB、CREATE\_NOT\_FENCED 和 IMPLICIT\_SCHEMA 特权。只有具有 SYSADM 权限的用户才可以授予或取消 DBADM 权限。具有 DBADM 权限的用户可以授予其他用户对该数据库的特权，并可以从任何用户取消任何特权，而不管是谁授予的。

只有具有 DBADM 或更高权限的用户才可以执行下列操作:

- 读取日志文件
- 创建、激活和删除事件监视器。

具有 DBADM、SYSMAINT 或更高权限的用户可以执行下列操作:

- 查询表空间的状态
- 更新日志历史文件
- 停止表空间。
- 重组表
- 使用 RUNSTATS 实用程序收集目录统计信息。

注: DBADM 只能对您持有 DBADM 权限的数据库执行上述功能。

## LOAD 权限

在数据库级具有 LOAD 权限以及对表具有 INSERT 特权的用户可以使用 LOAD 命令或“自动装入程序”实用程序来将数据装入表中。

若先前装入操作是用来插入数据的装入操作,则在数据库级具有 LOAD 权限以及对表具有 INSERT 特权的用户可以执行 LOAD RESTART 或 LOAD TERMINATE。

若先前装入操作是“装入替换”,则在该用户可以执行 LOAD RESTART 或 LOAD TERMINATE 操作之前,必须将 DELETE 特权授予该用户。

若使用异常表作为 LOAD 的一部分,则用户必须对该异常表具有 INSERT 特权。

具有此权限的用户可以执行 QUIESCE TABLESPACES FOR TABLE、RUNSTATS 和 LIST TABLESPACES 命令。

## 数据库特权

第217页的图5显示数据库特权。



## 数据库特权

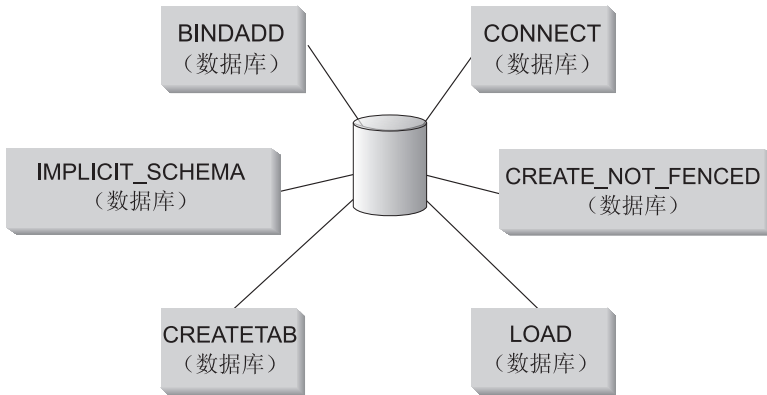


图 5. 数据库特权

数据库特权涉及到对作为一个整体的数据库所执行的操作:

- **CONNECT** 允许用户访问数据库
- **BINDADD** 允许用户在数据库中创建新的软件包
- **CREATETAB** 允许用户在数据库中创建新的表
- **CREATE\_NOT\_FENCED** 允许用户创建“未保护的”用户定义函数 (UDF) 或过程。“未保护”的 UDF 或过程必须经过严格的测试，因为数据库管理器不会阻止这些 UDF 或过程保护它的存储器或控制块。(因此，允许按“未保护”形式运行的、一个编写和测试都不严密的 UDF 或过程，可能在系统中引起严重问题。)(有关更多信息，参考 *Application Development Guide* 或 *SQL Reference*。)
- 通过在 **CREATE** 语句中使用尚未存在的模式名来创建对象，**IMPLICIT\_SCHEMA** 允许任何用户隐式创建一个模式。**SYSIBM** 成为隐式创建的模式的所有者，并且授予 **PUBLIC** 在此模式中创建对象的特权。
- **LOAD** 允许用户将数据装入表中。

只有具有 **SYSADM** 或 **DBADM** 权限的用户才可以向其他用户授予特权，并可从其他用户取消这些特权。

**注:** 当创建一个数据库时，会自动将下列特权授予 **PUBLIC**:

- **CREATETAB**
- **BINDADD**
- **CONNECT**
- **IMPLICIT\_SCHEMA**
- 对 **USERSPACE1** 表空间的 **USE** 特权

- 对系统目录视图的 SELECT 特权。

要除去任何特权，DBADM 或 SYSADM 必须从 PUBLIC 显式除去该特权。

### 隐式模式权限 (IMPLICIT\_SCHEMA) 注意事项

当创建一个新数据库时，或从先前发行版迁移一个数据库时，PUBLIC 会被授予 IMPLICIT\_SCHEMA 数据库权限。使用此权限，任何用户都可以创建一个对象并指定尚未存在的模式名，来创建一个模式。SYSIBM 成为隐式创建的模式的所有者，并且授予 PUBLIC 在此模式中创建对象的特权。

若数据库需要控制可隐式创建模式对象的人，则应当从 PUBLIC 取消 IMPLICIT\_SCHEMA 数据库权限。一旦取消了该权限，就只有三种创建模式对象的方法：

- 任何用户都可以在一个 CREATE SCHEMA 语句上使用他们自己的授权名来创建一个模式。
- 任何具有 DBADM 权限的用户都可以显式地创建任何尚不存在的模式，并且可以选择是否指定另一个用户作为此模式的所有者。
- 任何具有 DBADM 权限的用户都具有 IMPLICIT\_SCHEMA 数据库权限（与 PUBLIC 无关），这样，他们在创建其他数据库对象时可以使用任何名称隐式地创建一个模式。SYSIBM 成为隐式创建的模式的所有者，而且 PUBLIC 具有在此模式中创建对象的特权。

用户始终具有使用他们自己的授权名显式创建他们自己的模式的能力。

## 模式特权

模式特权属于对象特权类别。对象特权显示在第219页的图6中。

## 对象特权

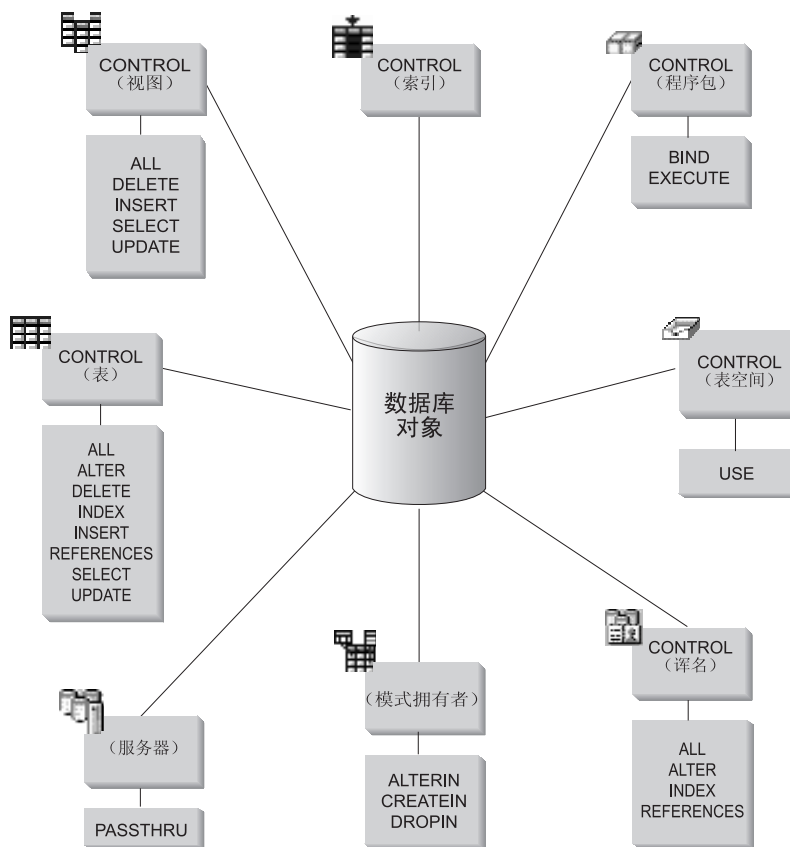


图 6. 对象特权

模式特权涉及到对一个数据库中的模式所执行的操作。可以授予用户下列任何一个特权：

- **CREATEIN** 允许用户在模式中创建对象。
- **ALTERIN** 允许用户在模式中改变对象。
- **DROPIN** 允许用户在模式中删除对象。

模式所有者具有所有这些特权，并且有将这些特权授予其他用户的能力。在模式对象中操纵的对象包括：表、视图、索引、软件包、数据类型、函数、触发器、过程和别名。

## 表空间特权

表空间特权涉及对数据库中表空间的操作。可以将对表空间的 `USE` 特权授予用户，这允许它们在该表空间中创建表。

表空间的所有者（通常是具有 `SYSADM` 或 `SYSCTRL` 权限的创建者）具有 `USE` 特权，有能力将此特权授予别人。在缺省情况下，当创建数据库时，将表空间 `USERSPACE1` 的 `USE` 特权授予 `PUBLIC`（虽然可以取消此特权）。

`USE` 特权不能配合 `SYSCATSPACE` 或任何系统临时表空间使用。

## 表和视图特权

表和视图特权涉及到对一个数据库中的表或视图所执行的操作。用户必须对数据库具有 `CONNECT` 特权，才可使用下列任何一个特权：

- `CONTROL` 给用户提供了对表或视图的所有特权，包括删除它、授予和取消个别表特权的能力。必须具有 `SYSADM` 或 `DBADM` 权限，才可授予 `CONTROL`。一个表的创建者自动接收表的 `CONTROL` 特权。仅当一个视图的创建者对视图定义中引用的所有表和视图具有 `CONTROL` 特权时，或他们具有 `SYSADM` 或 `DBADM` 权限时，才自动接收 `CONTROL` 特权。
- `ALTER` 允许用户向一个表添加列、添加或更改有关一个表及其列的注解、添加主键或唯一约束以及创建或删除表检查约束。用户也可以在表上创建触发器，虽然需要对此触发器中引用的所有对象的附加权限（若触发器引用该表的任何列，则包括对该表的 `SELECT`）。对所有子表具有 `ALTER` 特权的用户可以删除主键；对该表具有 `ALTER` 特权并对父表具有 `REFERENCES` 特权的用户，或对适当的列具有 `REFERENCES` 特权的用户可以创建或删除外键。具有 `ALTER` 特权的用户也可以 `COMMENT ON` 一个表。
- `DELETE` 允许用户从表或视图中删除行。
- `INDEX` 允许用户对表创建一个索引。索引创建者自动具有索引的 `CONTROL` 特权。有关更多信息，参见第223页的『索引特权』。
- `INSERT` 允许用户将行插入表或视图，并运行 `IMPORT` 实用程序。
- `REFERENCES` 允许用户创建和删除一个外键，并指定该表为关系中的父表。用户可能只对特定的列拥有此特权。
- `SELECT` 允许用户检索表或视图中的行，对表创建视图，并运行“导出”实用程序。
- `UPDATE` 允许用户更改表或视图中的项，或表或视图中的一个或多个特定列的项。用户只能对特定的列拥有此特权。

将这些特权授予其他用户的特权也可在 `GRANT` 语句上使用 `WITH GRANT OPTION` 来授予。

**注:** 当授予一个用户或组对某个表的 `CONTROL` 特权时, 将使用 `WITH GRANT OPTION` 自动授予对该表的所有其他特权。若接着从某个用户取消了对该表的 `CONTROL` 特权, 该用户将仍然保留自动授予的其他特权。要取消使用 `CONTROL` 特权授予的所有特权, 必须显式取消各个的特权, 或者在 `REVOKE` 语句上指定 `ALL` 关键字, 例如:

```
REVOKE ALL
ON EMPLOYEE FROM USER HERON
```

当使用类型表时, 表和视图的特权有特别的意义。

**注:** 可在一个表层次结构的每一层单独授予特权。因此, 对于类型表的层次结构中的一个超表, 被授予对该超表的特权的用户也可间接影响任何子表。然而, 若对一个子表持有需要的特权, 则用户只能直接对该子表操作。

在一个表层次结构中, 表之间的超表/子表关系表示象 `SELECT`、`UPDATE` 和 `DELETE` 这样的操作将影响该操作的目标表和其所有子表(若有)中的行。这种行为称为“可替换能力”。例如, 假设创建了一个类型为 `Employee_t` 的 `Employee` 表, 它具有类型为 `Manager_t` 的子表 `Manager`。经理是一种特殊的雇员, 这由结构化类型 `Employee_t` 与 `Manager_t` 之间的类型/子类型关系, 和对应的在表 `Employee` 与 `Manager` 之间的表/子表关系来指示。由于这种关系, `SQL` 查询:

```
SELECT * FROM Employee
```

将返回雇员和经理的对象标识符和 `Employee_t` 属性。类似地, 更新操作:

```
UPDATE Employee SET Salary = Salary + 1000
```

将给经理和正式雇员加薪一千元。

对 `Employee` 具有 `SELECT` 特权的用户可以执行这个 `SELECT` 操作, 即使他们对 `Manager` 没有显式 `SELECT` 特权。但是, 将不允许这类用户直接对 `Manager` 子表执行 `SELECT` 操作, 因此, 这类用户将不能访问 `Manager` 表的任何非继承列。

类似地, 对 `Employee` 具有 `UPDATE` 特权的用户将能够对 `Manager` 执行 `UPDATE` 操作, 从而影响正规的雇员和经理, 即使该用户对 `Manager` 表不具有显式的 `UPDATE` 特权。但是, 将不允许这类用户直接对 `Manager` 子表执行 `UPDATE` 操作, 因而, 这类用户将不能更新 `Manager` 表的任何非继承列。

下列手册提供了有关执行特定的命令、API 或 `SQL` 语句所需的授权的信息:

- *SQL Reference*
- *Command Reference*
- *Administrative API Reference.*

有关更新目录统计信息所需的授权的信息，参考《管理指南：性能》。

有关如何确定视图特权的信息，参考 *SQL Reference* 手册中的 CREATE VIEW 语句。

## 别名特权

别名特权涉及到对数据库中的别名所执行的操作。这些特权不影响对别名引用的数据源对象拥有的特权。用户必须对数据库具有 CONNECT 特权，才可使用下列任何一个特权：

- CONTROL 给用户提供别名的所有特权，包括删除它、授予和取消个别别名特权的能力。必须具有 SYSADM 或 DBADM 权限，才可授予 CONTROL。一个别名的创建者自动接收该别名的 CONTROL 特权。
- ALTER 允许用户更改别名中的列名、添加或更改该列的数据类型要映射成的 DB2 类型，并为别名列设置列选项。
- INDEX 允许用户基于别名创建索引规范。索引规范的创建者自动具有此索引的 CONTROL 特权。
- REFERENCES 允许用户创建和删除外键，并指定该别名为关系中的父级。用户只能对特定的列拥有此特权。

将这些特权授予其他用户的特权也可在 GRANT 语句上使用 WITH GRANT OPTION 来授予。

**注：**当授予一个用户或组对某个别名的 CONTROL 特权时，对该别名的所有其他特权将使用 WITH GRANT OPTION 自动授予。若接着从某个用户取消了对该别名的 CONTROL 特权，该用户将仍然保留自动授予的其他特权。

要访问数据源数据，还必须对数据源中别名引用的对象具有适当的授权。

当用户访问引用一个或多个别名的视图时，必须授权用户来访问该视图和在数据源中别名引用的对象。

## 服务器特权

有一个服务器特权：PASSTHRU。此特权控制哪些授权标识可将 DDL 和 DML 语句直接发送（通过操作）到数据源。

DB2 提供两条 SQL 语句来控制通过操作：

- GRANT PASSTHRU 语句，授予对数据源发出 SET PASSTHRU 语句并将 DML 和 DDL 语句发送到该数据源的权限。
- REVOKE PASSTHRU 语句，取消对数据源发出 SET PASSTHRU 语句并将 DML 和 DDL 语句发送到该数据源的权限。

对服务器 ORACLE1 的用户 SHAWN 授予通过授权的一个样本语句如下：

```
GRANT PASSTHRU ON SERVER ORACLE1 TO USER SHAWN
```

有关 PASSTHRU 语句语法的完整信息，参阅 *SQL Reference*。

## 软件包特权

软件包是一个数据库对象，它包含数据库管理器为了以最有效的方式访问特定应用程序的数据所需的信息。软件包特权使用户能够创建和操纵软件包。用户必须对数据库具有 CONNECT 特权，才可使用下列任何一个特权：

- CONTROL 给用户重新绑定、删除或执行软件包的能力，以及将那些特权授予其他用户的能力。软件包的创建者自动接收此特权。具有 CONTROL 特权的用户被授予 BIND 和 EXECUTE 特权，又可以将 BIND 和 EXECUTE 特权授予其他用户。要授予 CONTROL 特权，用户必须具有 SYSADM 或 DBADM 权限。
- BIND 允许用户重新绑定现存的软件包。
- EXECUTE 允许用户执行软件包。

除这些软件包特权外，BINDADD 数据库特权还允许用户创建新软件包或重新绑定数据库中的现存软件包。

有权执行包含别名的软件包的用户不需要软件包内的别名具有附加的特权权限级；但是，这些用户需要通过在包含别名引用的对象的数据源处的认证检查。另外，软件包用户必须对数据源中的数据源对象拥有恰当的特权或权限级。

包含别名的软件包可能需要附加的授权步骤，因为与 DB2 系列数据源通信时，DB2 使用动态 SQL。在数据源运行软件包的授权标识必须有恰当的权限，才可在该数据源动态执行此软件包。有关 DB2 如何处理静态和动态 SQL 的更多信息，参阅 *SQL Reference*。

## 索引特权

索引或索引规范的创建者自动接收该索引的 CONTROL 特权。索引的 CONTROL 特权实际是删除此索引的能力。要授予对一个索引的 CONTROL 特权，用户必须具有 SYSADM 或 DBADM 权限。

表级 INDEX 特权允许用户在该表上创建索引（参见第220页的『表和视图特权』）。

## 序列特权

序列的创建者自动接收 `USAGE` 特权。`USAGE` 特权需要使用序列的 `NEXTVAL` 和 `PREVVAL` 表达式。要允许其它用户使用 `NEXTVAL` 和 `PREVVAL` 表达式，必须将序列特权授予 `PUBLIC`。这就允许所有用户使用具有指定序列的表达式。

---

## 控制对数据库对象的访问

控制数据访问需要了解直接和间接特权、管理权限和软件包。本节解释这些主题并提供一些示例。

直接授予的特权存储在系统目录中。在第234页的『使用系统目录』中讨论了审核数据库访问控制计划实现的方法。

控制授权有三种方法：

- 通过使用 `GRANT` 和 `REVOKE` 语句控制的特权来控制显式授权
- 通过创建和删除对象来控制隐式授权
- 间接特权与软件包相关。

讨论下列主题：

- 『授予特权』
- 第225页的『取消特权』
- 第226页的『通过创建和删除对象来管理隐式授权』
- 第227页的『允许通过软件包授予间接特权』
- 第229页的『使用视图控制对数据的访问』
- 第231页的『使用审计设施监视对数据的访问』。

## 授予特权

`GRANT` 语句允许授权用户授予特权。可以在一条语句中将一个特权授予一个或多个授权名；或授予 `PUBLIC`，这使该特权可供所有用户使用。注意授权名可以是个别用户，也可以是组。

在存在具有相同名称的用户和组的操作系统上，应当指定是将该特权授予用户还是授予组。`GRANT` 和 `REVOKE` 语句都支持关键字 `USER`和 `GROUP`。若未使用这些可选的关键字，则数据库管理器检查操作系统安全性设施，以确定该授权名是标识用户还是组。若该授权名可能既是用户又是组，则返回一个错误。

以下示例将 `EMPLOYEE` 表的 `SELECT` 特权授予用户 `HERON`：

```
GRANT SELECT
ON EMPLOYEE TO USER HERON
```



以下示例将 EMPLOYEE 表的 SELECT 特权授予组 HERON:

```
GRANT SELECT
ON EMPLOYEE TO GROUP HERON
```

要授予对大多数数据库对象的特权，用户必须对该对象具有 SYSADM 权限、DBADM 权限或 CONTROL 特权；或者，用户必须持有使用 WITH GRANT OPTION 授予的特权。只能授予对现存对象的特权。要将 CONTROL 特权授予其他用户，此用户必须具有 SYSADM 或 DBADM 权限。要授予 DBADM 特权，用户必须具有 SYSADM 权限。

有关 GRANT 语句的更多信息，参考 *SQL Reference*。

## 取消特权

REVOKE 语句允许授权用户取消先前已授予其他用户的特权。要取消对数据库对象的特权，必须对该对象具有 DBADM 权限、SYSADM 权限或 CONTROL 特权。注意持有使用 WITH GRANT OPTION 授予的特权并不足以取消该特权。要从另一个用户取消 CONTROL 特权，必须具有 SYSADM 或 DBADM 权限。要取消 DBADM 权限，必须具有 SYSADM 权限。只能取消对现存对象的特权。

**注：**对某个表或视图没有 DBADM 权限或 CONTROL 特权的用户，不能取消他们使用 WITH GRANT OPTION 授予的特权。另外，由被取消特权的人授予特权的那些人不会被取消特权。有关取消特权所需权限的更多信息，参考 *SQL Reference* 手册。

若已经把一特权授予有相同名称的一个用户和一个组，当取消此特权时必须指定 GROUP 或 USER 关键字。以下示例从用户 HERON 取消对 EMPLOYEE 表的 SELECT 特权:

```
REVOKE SELECT
ON EMPLOYEE FROM USER HERON
```

以下示例从组 HERON 取消对 EMPLOYEE 表的 SELECT 特权:

```
REVOKE SELECT
ON EMPLOYEE FROM GROUP HERON
```

注意从一个组中取消特权并不能从该组的所有成员中取消该特权。若个别名称已被直接授予一特权，则此名称将保留它，直到被直接取消该特权为止。

若从一个用户取消了表特权，则也取消对该用户创建的任何视图的特权，这取决于已取消的表特权。但是，只取消系统隐式授予的那些特权。若该视图的一个特权是另一个用户直接授予的，则此特权仍然会被保留。

您可能会遇到这样的情况，您想要将一种特权授予一个组，然后仅从组中的其中一个成员取消该特权。仅有两种方式执行该操作而不会接收到出错信息 SQL0556N:

- 您可以从组中除去该成员；或者创建具有较少成员的新组，并将特权授予新组。
- 可以从组中取消特权，然后将特权授予个别用户（授权标识）。

若从具有 DBADM 权限的用户取消一个显式授予的表（或视图）特权，**将不会**从在该表上定义的其他视图取消特权。这是因为视图特权可通过 DBADM 权限得到，并不依赖于基础表上的显式特权。

若在一个或多个基础表或视图的基础上定义了一个视图，并且又失去了对那些表或视图中一个或多个的 SELECT 特权，则不能使用该视图。

**注：**当从一个用户取消对一个表或视图的 CONTROL 特权时，此用户仍继续有将特权授予其他用户的能力。当授予 CONTROL 特权时，用户也可接收使用 WITH GRANT OPTION 提供的所有其他特权。一旦取消了 CONTROL，则使用 WITH GRANT OPTION 提供的所有其他特权会保留，一直到显式取消它们为止。

取决于被取消的特权的所有软件包都标记为无效，但是若一个具有合适权限的用户重新绑定它们，则可以变得有效。若随后又将特权授予应用程序的绑定者，也可以重建软件包；运行此应用程序将触发一个成功的隐式重新绑定。若从 PUBLIC 取消了特权，则所有由只能根据 PUBLIC 特权绑定的用户绑定的软件包都失效。若从用户取消了 DBADM 权限，则该用户绑定的所有软件包全都失效，包括与数据库应用程序相关的那些软件包。试图使用标记为无效的软件包将引起系统试图重新绑定此软件包。若此重新绑定尝试失败，则发生错误 (SQLCODE -727)。在这种情况下，必须由具有如下权限的用户显式地重新绑定软件包：

- 重新绑定软件包的权限
- 对软件包中使用的对象的适当权限

应当在取消特权时重新绑定这些软件包。有关 REVOKE 和 REBIND PACKAGE 语句的更多信息，参考 *SQL Reference*。

若根据一个或多个特权定义了一个触发器，并且又失去了那些特权中的一个或多个，则不能使用该触发器。

## 通过创建和删除对象来管理隐式授权

数据库管理器隐式地将特定特权授予以下用户：发出 CREATE SCHEMA、CREATE TABLESPACE、CREATE TABLE、CREATE VIEW 或 CREATE INDEX 语句的

用户，或使用 `PREP` 或 `BIND` 命令创建新软件包的用户。当具有 `SYSADM` 或 `DBADM` 权限的用户创建对象时，就授予了特权。类似地，当删除一个对象时，就除去了特权。

当创建的对象是表空间、表、索引或软件包时，用户会接收到该对象的 `CONTROL` 特权。当对象是一个视图时，只有在用户对该视图定义中引用的所有表和视图具有 `CONTROL` 特权时，才被隐式地授予此视图的 `CONTROL` 特权。

当显式创建的对象是一个模式时，将使用 `WITH GRANT OPTION` 授予此模式所有者 `ALTERIN`、`CREATEIN` 和 `DROPIN` 特权。隐式创建的模式具有授予 `PUBLIC` 的 `CREATEIN`。

有关如何确定视图特权的信息，参考 *SQL Reference* 手册中的 `CREATE VIEW` 语句。

## 建立计划或软件包的所有权

`BIND` 和 `PRECOMPILE` 命令创建或更改应用软件包。在任何一个命令上，使用 `OWNER` 选项以命名结果软件包的所有者。命名软件包的所有者有简单的规则：

- 任何用户可命名自己为所有者。若未指定 `OWNER` 选项，则这是缺省值。
- 具有 `SYSADM` 或 `DBADM` 权限的标识可使用 `OWNER` 选项命名任何授权标识作为所有者。

并非所有可使用 `DB2` 数据库产品绑定软件包的操作系统都支持 `OWNER` 选项。

有关 `BIND` 和 `PRECOMPILE` 命令的更多信息，参考 *Command Reference*。

## 允许通过软件包授予间接特权

对一个数据库中数据的访问可以由应用程序请求，也可由参与交互式工作站会话的人请求。软件包包含允许用户对许多数据库对象执行不同操作的语句。其中每个操作需要一个或多个特权。

授予绑定软件包的个人和 `PUBLIC` 的特权用于在绑定静态 `SQL` 时的授权检查。通过组授予的特权不用于在绑定静态 `SQL` 时的授权检查。除非在绑定软件包时指定 `VALIDATE RUN`，否则绑定软件包的具有有效 `authID` 的用户必须被显式授予了执行该软件包中的静态 `SQL` 语句所需的全部特权，或者通过 `PUBLIC` 被隐式授予了需要的特权。若执行 `BIND` 时指定了 `VALIDATE RUN`，则此软件包中任何静态 `SQL` 语句的所有授权失败都不会导致 `BIND` 失败，那些 `SQL` 语句在运行时重新验证。当检查以确保用户具有合适的授权（`BIND` 或 `BINDADD` 特权）来绑定软件包时，`PUBLIC`、组和用户特权**全部**都会用到。

软件包可包括静态和动态的 SQL。要处理包含静态 SQL 的软件包，用户只需要对该软件包具有 EXECUTE 特权。然后对于该软件包中的任何静态 SQL，此用户可以间接获得该软件包绑定者的特权，但是只在此软件包所施加的限制内。

要处理包含任何动态 SQL 语句的一个软件包，用户必须对该软件包具有 EXECUTE 特权。用户需要对该软件包具有 EXECUTE 特权，以及执行该软件包中的动态 SQL 语句所需的任何特权。绑定者的权限和特权用于处理软件包中的任何静态 SQL。

## 允许通过包含别名的软件包授予间接特权

当软件包包含对别名的引用时，对软件包创建者和软件包用户的授权处理比较复杂。当软件包创建者成功绑定包含别名的软件包时，软件包创建者不必通过在数据源处对别名引用的表和视图所做的认证检查或特权检查。但是，此软件包的执行者必须通过数据源的认证和权限检查。

例如，假设软件包创建者的 .SQC 文件包含几条 SQL 语句。一条静态语句引用了本地表。另一条动态语句引用了别名。当绑定此软件包时，使用软件包创建者的授权标识来验证对本地表的特权，但不对别名标识的数据源对象执行检查。当另一个用户执行此软件包时，假设他对该软件包具有 EXECUTE 特权，则该用户不必通过对引用此表的语句所做的任何附加的特权检查。但是，对于引用别名的语句，执行此软件包的用户必须通过数据源的认证检查和特权检查。

当 .SQC 文件仅包含动态 SQL 语句和表与别名的混合引用时，对本地对象和别名的 DB2 授权检查是类似的。软件包用户必须通过在语句内设置的任何本地对象（表、视图）的特权检查，还要通过别名对象的特权检查（软件包用户必须通过在包含别名标识的对象的数据源处的认证检查和特权检查）。在这两种情况下，软件包的用户都必须具有 EXECUTE 特权。

软件包执行者的标识和密码用于所有数据源认证和特权处理。可通过创建用户映射更改此信息。

**注：**不能在静态 SQL 中指定别名。不要对包含别名的软件包使用 DYNAMICRULES 选项（设置为 BIND）。

包含别名的软件包可能需要附加的授权步骤，因为与 DB2 系列数据源通信时，DB2 使用动态 SQL。在数据源运行软件包的授权标识必须有恰当的权限，才可在该数据源动态执行此软件包。有关 DB2 如何处理静态和动态 SQL 的更多信息，参阅 *SQL Reference*。

## 使用视图控制对数据的访问

视图通过允许下列操作，提供了一种方法来控制对表的访问或扩展对表的特权：

- 只访问表的指定列。

对于要求只访问一个表的特定列的用户和应用程序，授权用户可以创建一个视图来限制这些列只被需要它们的那些人访问。

- 只访问表的所有行的一个子集。

通过在一个视图定义的子查询中指定 **WHERE** 子句，授权用户可以限制通过一个视图访问的行。

- 只访问数据源表或视图中的行或列的一个子集。若通过别名访问数据源，可创建引用别名的本地 **DB2** 视图。这些视图可从一个或多个数据源引用别名。

**注：**因为可以创建一个视图来包含对多个数据源的别名引用，因此用户可从一个视图访问多个数据源中的数据。这些视图称为**多位置视图**。当将一个分布式环境中各敏感表的列信息合并时，或个别用户缺少在数据源需要的特定对象的特权时，这类视图可以发挥作用。

要创建一个视图，用户必须对此视图定义中引用的每个表或视图具有 **SYSADM** 权限、**DBADM** 权限、**CONTROL** 或 **SELECT** 特权。用户还必须能够在为此视图指定的模式中创建对象。即，对现存模式具有 **CREATEIN** 特权，或者，若此模式还不存在，则对数据库具有 **IMPLICIT\_SCHEMA** 权限。有关更多信息，参见第127页的『创建视图』。

若要创建引用别名的视图，不需要对视图中别名引用的数据源对象（表和视图）有附加的权限；但是，当用户访问该视图时，必须对基础数据源对象具有 **SELECT** 权限或等效的授权级。

若用户在数据源处对基础对象（表和视图）没有合适的权限，可执行下列操作：

1. 以用户可访问的数据源表中的那些列为基础，创建一个数据源视图
2. 授予用户对此视图的 **SELECT** 特权
3. 创建一个引用此视图的别名

然后用户可发出引用新别名的 **SELECT** 语句来访问那些列。

以下方案提供了如何使用视图来限制访问信息的一个更详细的示例。

因种种原因，许多人可能需要访问 **STAFF** 表中的信息。例如：

- 人事部门需要能更新和查看整个表。

通过授予组 **PERSONNL** 对 **STAFF** 表的 **SELECT** 和 **UPDATE** 特权，可以很容易地满足此要求：

```
GRANT SELECT,UPDATE ON TABLE STAFF TO GROUP PERSONNL
```

- 个别部门的经理需要查看他们雇员的工资信息。

可以通过为每个部门经理创建一个视图来满足此要求。例如，可以为部门号为 51 的经理创建如下视图：

```
CREATE VIEW EMP051 AS
  SELECT NAME,SALARY,JOB FROM STAFF
  WHERE DEPT=51
GRANT SELECT ON TABLE EMP051 TO JANE
```

具有授权名 JANE 的经理将象查询 STAFF 表一样查询 EMP051 视图。当访问 STAFF 表的 EMP051 视图时，此经理会看到如下信息：

NAME	SALARY	JOB
Fraye	45150.0	Mgr
Williams	37156.5	Sales
Smith	35654.5	Sales
Lundquist	26369.8	Clerk
Wheeler	22460.0	Clerk

- 所有用户都需要能够找到其他雇员。可以根据 STAFF 表的 NAME 列和 ORG 表的 LOCATION 列创建一个视图，并通过两个表各自的 DEPT 和 DEPTNUMB 列将这两个表连接，来满足此要求：

```
CREATE VIEW EMPLOCS AS
  SELECT NAME, LOCATION FROM STAFF, ORG
  WHERE STAFF.DEPT=ORG.DEPTNUMB
GRANT SELECT ON TABLE EMPLOCS TO PUBLIC
```

访问雇员位置视图的用户将看到如下信息：

NAME	LOCATION
Molinare	New York
Lu	New York
Daniels	New York
Jones	New York
Hanes	Boston
Rothman	Boston
Ngan	Boston
Kermisch	Boston
Sanders	Washington

NAME	LOCATION
Pernal	Washington
James	Washington
Sneider	Washington
Marenghi	Atlanta
O'Brien	Atlanta
Quigley	Atlanta
Naughton	Atlanta
Abrahams	Atlanta
Koonitz	Chicago
Plotz	Chicago
Yamaguchi	Chicago
Scoutten	Chicago
Fraye	Dallas
Williams	Dallas
Smith	Dallas
Lundquist	Dallas
Wheeler	Dallas
Lea	San Francisco
Wilson	San Francisco
Graham	San Francisco
Gonzales	San Francisco
Burke	San Francisco
Quill	Denver
Davis	Denver
Edwards	Denver
Gafney	Denver

## 使用审计设施监视对数据的访问

DB2 审计设施为一系列预定义数据库事件生成一个审计记录，并允许您维护它。当不是阻止访问数据的设施时，审计设施可以监视对访问或修改数据对象的尝试，并可对这些尝试做一个记录。

要使用审计设施管理员工具 `db2audit`，必需 `SYSADM` 权限。

有关 DB2 审计设施的详细描述，参见第239页的『第6章 审计 DB2 活动』。

## 数据加密

安全性计划的一个部分可能包括加密您的数据。为此，可以使用加密和解密内置函数：ENCRYPT、DECRYPT\_BIN、DECRYPT\_CHAR 和 GETHINT。有关这些函数的更多信息，参考 *SQL Reference*。

ENCRYPT 函数使用基于密码的加密方法对数据进行加密。这些函数还允许您封装密码提示。密码提示嵌入在加密数据中。一旦加密，对数据进行解密的唯一方式是通过使用正确的密码来解密。选择使用这些函数的开发者应该对忘记密码和不能用的数据如何管理进行计划。

ENCRYPT 函数的结果数据类型与第一个变量的相同。

仅可以加密 VARCHAR。

结果的声明长度是下列其中一项：

- 当指定可选的提示参数时，数据变量的长度加上 42。
- 当未指定可选的提示参数时，数据变量的长度加上 10。

DECRYPT\_BIN 和 DECRYPT\_CHAR 函数使用基于密码的解密对数据进行解密。

DECRYPT\_BIN 和 DECRYPT\_CHAR 函数的结果数据类型与第一个变量的相同。

结果的声明长度是原始数据的长度。

GETHINT 函数返回封装的密码提示。密码提示是将帮助数据所有者回忆起密码的短语。例如，单词“大海”可以用作回忆密码“太平洋”的提示。

以下列两种方式之一确定用于对数据进行加密的密码：

- 密码变量。密码是当调用 ENCRYPT 函数时显式传送的字符串。使用给出的密码对数据进行加密和解密。
- 专用寄存器密码。SET ENCRYPTION PASSWORD 语句对密码值进行加密，并将加密后的密码发送至数据库管理器以存储在专用寄存器中。未使用密码参数调用的 ENCRYPT、DECRYPT\_BIN 和 DECRYPT\_CHAR 函数使用 ENCRYPTION PASSWOED 专用寄存器中的值。

专用寄存器的初始或缺省值是一个空字符串。

密码的有效长度在 6 和 127 之间，包括 6 和 127。提示的有效长度在 0 和 32 之间，包括 0 和 32。



当从客户机设置 ENCRYPTION PASSWORD 专用寄存器时，在客户机上对密码进行加密，并将密码发送至数据库服务器，然后解密。要确保密码不具有可读性，还要在数据库服务器中对密码进行重新加密。DECRYPT\_BIN 和 DECRYPT\_CHAR 函数在使用前必须解密专用寄存器。在 ENCRYPTION PASSWORD 中找到的值也不能具有可读性。不支持网关安全性。

## 任务和必需的授权

并非所有机构都以相同方式来划分工作职责。表5列出了其他一些常见的职务、与这些职务通常对应的任务，和完成这些任务需要的权限或特权。

表 5. 常见职务、任务和必需的授权

职务	任务	必需的授权
部门管理员	监督部门系统；创建数据库	SYSCTRL 权限。若部门有其自己的实例，则为 SYSADM 权限。
安全性管理员	授予其他用户一些或所有授权和特权	SYSADM 或 DBADM 权限。
数据库管理员	设计、开发、操作、保护和维护一个或多个数据库	对一个或多个数据库的 DBADM 和 SYSMOINT 权限。某些情况下，为 SYSCTRL 权限。
系统操作员	监视数据库并执行备份功能	SYSMOINT 权限。
应用程序员	开发和测试数据库管理器的应用程序；也可创建测试数据表	对现存软件包的 BINDADD、BIND，对一个或多个数据库的 CONNECT 和 CREATETAB，某些特定模式特权，以及对某些表的特权的列表。
用户分析员	通过检查系统目录视图来定义一个应用程序的数据需求	对目录视图的 SELECT；对一个或多个数据库的 CONNECT。
程序最终用户	执行应用程序	对软件包的 EXECUTE；对一个或多个数据库的 CONNECT。参见此表后的注释。
信息中心顾问	定义查询用户的数据需求；通过创建表和视图，并授予对数据库对象的访问权来提供数据	对一个或多个数据库的 DBADM 权限。
查询用户	发出 SQL 语句来检索、添加、删除或更改数据；可以将结果作为表来保存	对一个或多个数据库的 CONNECT；对要创建的表和视图的模式 CREATEIN；以及对某些表和视图的 SELECT、INSERT、UPDATE、DELETE。

**注意：**若一个应用程序包含动态 SQL 语句，则“程序最终用户”除需要 EXECUTE 和 CONNECT 外可能需要其他特权（如 SELECT、INSERT、DELETE 和 UPDATE）。

---

## 使用系统目录

有关每个数据库的信息自动在一个称为系统目录的视图集合中维护，系统目录是在生成数据库时创建的。此系统目录描述表、列、索引、程序、特权和其他对象。

这些视图中的六个视图列出了用户持有的特权以及授予每个特权的用户的身份：

<b>SYSCAT.DBAUTH</b>	列出数据库特权
<b>SYSCAT.TABAUTH</b>	列出表和视图的特权
<b>SYSCAT.COLAUTH</b>	列出列的特权
<b>SYSCAT.PACKAGEAUTH</b>	列出软件包特权
<b>SYSCAT.INDEXAUTH</b>	列出索引特权
<b>SYSCAT.SCHEMAAUTH</b>	列出模式特权
<b>SYSCAT.PASSTHROUGH</b>	列出服务器特权

系统授予用户的特权将让 SYSIBM 作为授予者。SYSADM、SYSMAINT 和 SYSCTRL 未在系统目录中列出。

CREATE 和 GRANT 语句在系统目录中设置特权。具有 SYSADM 和 DBADM 权限的用户可以授予和取消对系统目录视图的 SELECT 特权。下列示例显示如何使用这些 SQL 查询来抽取有关特权的信息：

- 『检索具有授予的特权的授权名』
- 第235页的『检索具有 DBADM 权限的全部名称』
- 第235页的『检索被授权访问表的名称』
- 第236页的『检索授予用户的全部特权』
- 第236页的『保密系统目录视图』。

### 检索具有授予的特权的授权名

没有单个系统目录视图包含全部特权的信息。如下语句检索具有特权的所有授权名：

```
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'DATABASE' FROM SYSCAT.DBAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'TABLE ' FROM SYSCAT.TABAUTH
UNION
```

```

SELECT DISTINCT GRANTEE, GRANTEETYPE, 'PACKAGE ' FROM SYSCAT.PACKAGEAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'INDEX ' FROM SYSCAT.INDEXAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'COLUMN ' FROM SYSCAT.COLAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SCHEMA ' FROM SYSCAT.SCHEMAAUTH
UNION
SELECT DISTINCT GRANTEE, GRANTEETYPE, 'SERVER ' FROM SYSCAT.PASSTHROUGHAUTH
ORDER BY GRANTEE, GRANTEETYPE, 3

```

应定期将此语句检索到的列表与系统安全性设施中定义的用户名和组名的列表比较。然后，可以标识不再有效的那些授权名。

**注：**若您支持远程数据库客户机，有可能只在远程客户机定义了此授权名，而没有在数据库服务器上定义。

## 检索具有 **DBADM** 权限的全部名称

如下语句检索被直接授予 **DBADM** 权限的所有授权名：

```

SELECT DISTINCT GRANTEE FROM SYSCAT.DBAUTH
WHERE DBADMAUTH = 'Y'

```

## 检索被授权访问表的名称

如下语句检索被直接授权访问具有限定符 **JAMES** 的表 **EMPLOYEE** 的所有授权名：

```

SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
WHERE TABNAME = 'EMPLOYEE'
AND TABSCHEMA = 'JAMES'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
WHERE TABNAME = 'EMPLOYEE'
AND TABSCHEMA = 'JAMES'

```

要了解谁可以更新具有限定符 **JAME** 的表 **EMPLOYEE**，发出如下语句：

```

SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.TABAUTH
WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
(CONTROLAUTH = 'Y' OR
UPDATEAUTH = 'Y' OR UPDATEAUTH = 'G')
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.DBAUTH
WHERE DBADMAUTH = 'Y'
UNION
SELECT DISTINCT GRANTEETYPE, GRANTEE FROM SYSCAT.COLAUTH
WHERE TABNAME = 'EMPLOYEE' AND TABSCHEMA = 'JAMES' AND
PRIVTYPE = 'U'

```

以上语句检索具有 DBADM 权限的任何授权名，以及被直接授予 CONTROL 或 UPDATE 特权的那些名称。然而，它不会返回只持有 SYSADM 权限的用户的授权名。

记住某些授权名可以是组，而不只是个别用户。

## 检索授予用户的全部特权

通过在系统目录视图上进行查询，用户可以检索他们持有的特权的列表和他们授予其他用户的特权的列表。例如，如下语句检索直接授予个别授权名的数据库特权的列表：

```
SELECT * FROM SYSCAT.DBAUTH
WHERE GRANTEE = USER AND GRANTEETYPE = 'U'
```

如下语句检索由一个特定用户直接授予的表特权的列表：

```
SELECT * FROM SYSCAT.TABAUTH
WHERE GRANTOR = USER
```

如下语句检索由一个特定用户直接授予的个别列特权的列表：

```
SELECT * FROM SYSCAT.COLAUTH
WHERE GRANTOR = USER
```

这些语句中的关键字 USER 始终等于一个用户的授权名的值。USER 是一个只读专用寄存器。有关专用寄存器的更多信息，参考 *SQL Reference*。

## 保密系统目录视图

在创建数据库期间，会将系统目录视图的 SELECT 特权授予 PUBLIC。（有关自动授予 PUBLIC 的其他特权，参见第216页的『数据库特权』。）大多数情况下，这样做不会引起任何安全性问题。但是，对于特别敏感的数据，这可能不恰当，因为这些表描述数据库中的每个对象。若是这种情况，要考虑从 PUBLIC 取消 SELECT 特权；然后按需要将 SELECT 特权授予特定用户。授予和取消对系统目录视图的 SELECT 与对任何其他视图授予和取消权限的方式相同，但是必须具有 SYSADM 或 DBADM 权限，才可执行此操作。

至少，应当考虑限制访问下列目录视图：

- SYSCAT.DBAUTH
- SYSCAT.TABAUTH
- SYSCAT.PACKAGEAUTH
- SYSCAT.INDEXAUTH
- SYSCAT.COLAUTH
- SYSCAT.PASSTHROUGH

- SYSCAT.SCHEMAAUTH

这将防止有关用户特权的信息对可访问该数据库的任何人可用。若具有此信息，不道德的用户可以不经授权而访问该数据库。

还应检查作为统计信息的那些列（参考《管理指南：性能》中的“目录统计信息”）。记录在系统目录中的某些统计信息包含可能是您环境中的敏感信息的数据值。若这些统计信息包含敏感数据，可能希望从 PUBLIC 取消对 SYSCAT.COLUMNS 和 SYSCAT.COLDIST 目录视图的 SELECT 特权。

若希望限制对系统目录视图的访问，您可以定义视图，来让每个授权名检索有关它自己特权的信息。

例如，如下视图 MYSELECTS 包括每个特定的表的所有者和名称，已将该表的 SELECT 特权直接授予了一个用户的授权名：

```
CREATE VIEW MYSELECTS AS
  SELECT TABSCHEMA, TABNAME FROM SYSCAT.TABAUTH
  WHERE GRANTEETYPE = 'U'
  AND GRANTEE = USER
  AND SELECTAUTH = 'Y'
```

在此语句中的关键字 USER 始终等于该授权名的值。

如下语句使此视图可供每个授权名使用：

```
GRANT SELECT ON TABLE MYSELECTS TO PUBLIC
```

最后，应记住要取消对基表的 SELECT 特权：

```
REVOKE SELECT ON TABLE SYSCAT.TABAUTH FROM PUBLIC
```



---

## 第6章 审计 DB2 活动

认证、权限和特权可以用于控制对数据的已知访问或预期访问，但是这些方法可能不足以防止对数据的未知访问或未预期访问。为了辅助检测后一种类型的数据访问，DB2 提供了一个审计设施。成功监视不需要的数据访问和后续分析，可改善对数据访问的控制，并最终防止对数据的恶意访问或粗心的未经授权的访问。监视应用程序和单独的用户访问（包括系统管理操作）可提供有关数据库系统活动的历史记录。

DB2 审计设施为一系列预定义数据库事件生成一个审计记录，并允许您维护它。将此设施生成的记录条目保存在审计日志文件中。对这些记录条目的分析可揭示标识系统误用的使用模式。一旦标识出该模式，则可执行操作来减少或消除这类系统误用。

审计设施在实例级运行，记录所有实例级活动和数据库级活动。

当在分区数据库环境中工作时，许多可审计的事件将在与用户连接的分区（协调程序节点）或目录节点（若它们不是相同的分区）中发生。这意味着审计记录条目可由多个分区生成。每个审计记录条目的一部分包含有关协调程序节点和始发节点标识符的信息。

审计日志 (db2audit.log) 和审计配置文件 (db2audit.cfg) 位于该实例的 security 子目录中。在您创建一个实例时，操作系统会尽可能设置对这些文件的读 / 写许可权。缺省情况下，许可权仅是该实例所有者的读 / 写权限。建议您不要更改这些许可权。

审计设施管理工具 db2audit 的用户必须具有 SYSADM 权限 / 特权。

必须明确停止和启动审计设施。当启动时，审计设施使用现存的审计配置信息。因为审计设施与 DB2 服务器无关，所以，即使停止该实例，审计设施将仍然是活动的。事实上，当停止该实例时，可在审计日志中生成审计记录条目。

审计设施的授权用户可控制审计设施内的下列操作：

- 开始记录 DB2 实例内可审计的事件。
- 停止记录 DB2 实例内可审计的事件。
- 配置审计设施的行为，包括选择要记录的可审计事件的类别。
- 请求当前审计配置的描述。
- 从该实例中清除任何暂挂的审计记录条目，并将它们写入审计日志中。

- 格式化审计日志中的审计记录条目，并将它们复制到一个平面文件或 ASCII 定义的文件，来抽取这些审计记录条目。因为下列两个原因之一来执行抽取：准备分析日志记录，或者准备修剪日志记录。
- 修剪当前审计日志中的审计记录条目。

可生成不同类别的审计记录条目。在可用于审计的事件类别的描述中（下面），您应注意每个类别的名称后面是一个单词的关键字，它用来标识该类别的类型。可用于审计的事件类别是：

- 审计 (AUDIT)。当更改审计设置或访问审计日志时，生成记录条目。
- 授权检查 (CHECKING)。在对访问或操纵 DB2 对象或功能的尝试进行授权检查期间，生成记录条目。
- 对象维护 (OBJMAINT)。当创建或删除数据对象时生成记录条目。
- 安全性维护 (SECMAINT)。当授予或取消对象或数据库特权或 DBADM 权限时生成记录条目。当修改数据库管理器安全性配置参数 SYSADM\_GROUP、SYSCTRL\_GROUP 或 SYSMAINT\_GROUP 时也生成记录条目。
- 系统管理 (SYSADMIN)。当执行需要 SYSADM、SYSMAINT 或 SYSCTRL 权限的操作时生成记录条目。
- 用户验证 (VALIDATE)。当认证用户或检索系统安全性信息时生成记录条目。
- 操作上下文 (CONTEXT)。当执行数据库操作时，生成记录条目以显示该操作上下文。此类别允许对审计日志文件进行更好的解释。当与该日志的事件相关因子字段一起使用时，可将一组事件重新与单个数据库操作关联。例如，动态 SQL 的 SQL 语句、静态 SQL 的软件包标识符或可执行的操作类型的指示符（如 CONNECT）均可提供分析审计结果时所需的场景。

**注：**提供该操作上下文的 SQL 语句可能很长，并可在 CONTEXT 记录条目内完全显示。这可能使 CONTEXT 记录条目变得很大。

- 您可以审计失败的操作和 / 或成功的操作。

对该数据库执行的任何操作可能生成几个记录条目。生成的和移至审计日志的实际记录数目取决于审计设施配置所指定的要记录的事件类别数。它还取决于是审计成功的操作，还是失败的操作，或二者兼有。由于此原因，对要审计的事件的选择十分重要。



---

## 审计设施的行为

审计设施记录可审计的事件，包括那些影响的数据库实例。由于此原因，审计设施是 DB2 中可运行的一个独立部件，即使停止了 DB2 实例，它也可运行。若审计设施是活动的，当启动一个已停止的实例时，对该实例中的数据库事件的审计将继续。

将审计记录条目写入审计日志的时间安排可对该实例中的数据库的性能有显著影响。写入审计记录条目可与导致生成那些记录条目的事件同步或异步发生。`audit_buf_sz` 数据库管理器配置参数的值确定何时写入审计记录条目。

若此参数值为零 (0)，则同步写入。生成审计记录条目的事件将等到将该记录条目写入磁盘为止。与每个记录条目相关的等待导致 DB2 性能降低。

若 `audit_buf_sz` 的值大于零，则异步写入该记录条目。`audit_buf_sz` 的值大于零时，该值是 4 KB 页的一个倍数，用来创建内部缓冲区。该内部缓冲区用来在将一组审计记录条目写入磁盘之前保存大量审计记录条目。作为审计事件的结果生成审计记录条目的语句将不会等到将该记录条目写入磁盘，它可继续其操作。

在异步情况下，审计记录条目可能可以在未填写的缓冲区中保留一段时间。要防止这种情况的持续时间过长，数据库管理器将强制定期写入审计记录条目。审计设施的授权用户也可用显式请求清除审计缓冲区。

记录是同步写入还是异步写入，使发生错误时情况有些不同。在异步方式中，可能会有某些记录丢失，因为审计记录条目是在写入磁盘之前缓冲存储的。在同步方式中，可能有一个记录丢失，因为错误只能阻止最多一个审计记录条目写入。

`ERRORTYPE` 审计设施参数的设置控制在 DB2 和审计设施之间如何管理错误。当审计设施是活动的，并且 `ERRORTYPE` 审计设施参数的设置是 `AUDIT` 时，将审计设施作为 DB2 的任何其他部件一样对待。对于与被视为成功的语句相关的一个审计事件，必须写入审计记录条目（在同步方式下，写至磁盘；在异步方式下，写至审计缓冲区）。当在此方式下运行时，不管何时遇到错误，都将一个负的 `SQLCODE` 返回至生成审计记录条目的语句的应用程序。若将错误类型设置为 `NORMAL`，则忽略来自 `db2audit` 的任何错误，并返回该操作的 `SQLCODE`。有关 `ERRORTYPE` 审计设施参数（和其他相关参数）附加详细信息，参见第242页的『审计设施使用方案』。

根据该 API 或 SQL 语句以及 DB2 实例的审计设置，可为特定的事件生成一个或几个审计记录条目，或不生成审计记录条目。例如，具有一个 `SELECT` 子查询的 `SQL UPDATE` 语句可生成两个审计记录条目，一个记录条目包含对一个表的 `UPDATE` 特权的授权检查结果，另一个记录条目包含对一个表的 `SELECT` 特权的授权检查结果。

对于动态数据操纵语言 (DML) 语句, 在准备该语句时会对所有授权检查生成审计记录条目。不会再次审计同一用户对那些语句的重新使用, 因为那时不进行授权检查。然而, 若已更改包含特权信息的目录表之一, 则在下一个工作单元中, 再次检查高速缓存的动态 SQL 语句的语句特权, 并创建一个或多个新的审计记录。

对于仅包含静态 DML 语句的软件包, 可生成审计记录条目的唯一可审计的事件是授权检查, 它查看用户是否具有执行该软件包的特权。在预编译或绑定该软件包时, 执行该软件包中静态 SQL 语句所需的权限检查和可能的审计记录条目创建。软件包内的静态 SQL 语句的执行是不可审计的。当用户显式地再次绑定软件包时, 或系统隐式再次绑定软件包时, 则为该静态 SQL 语句所需的授权检查生成审计记录条目。

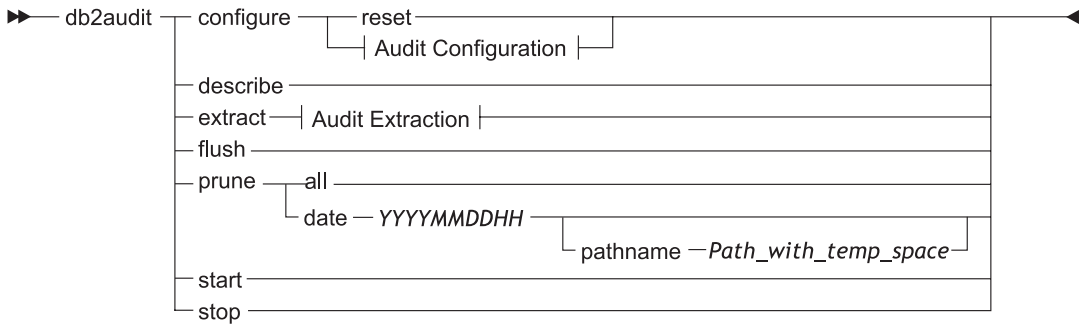
对于在执行语句时执行授权检查的语句 (例如, 数据定义语言 (DDL)、GRANT 和 REVOKE 语句), 不管何时使用这些语句, 都将生成审计记录条目。

**注:** 当执行 DDL 时, 无论该语句的实际段号可能是什么, 在审计记录条目中为所有事件 (除场景事件外) 记录的段号都将为零 (0)。

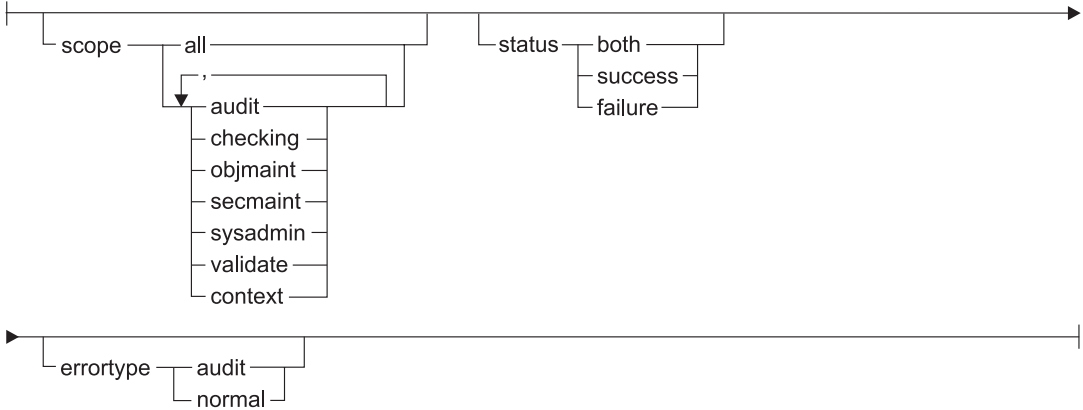
---

## 审计设施使用方案

复查下列语法图中的每一部分将会有助于您理解可以如何使用审计设施。



审查配置:



审查抽取:

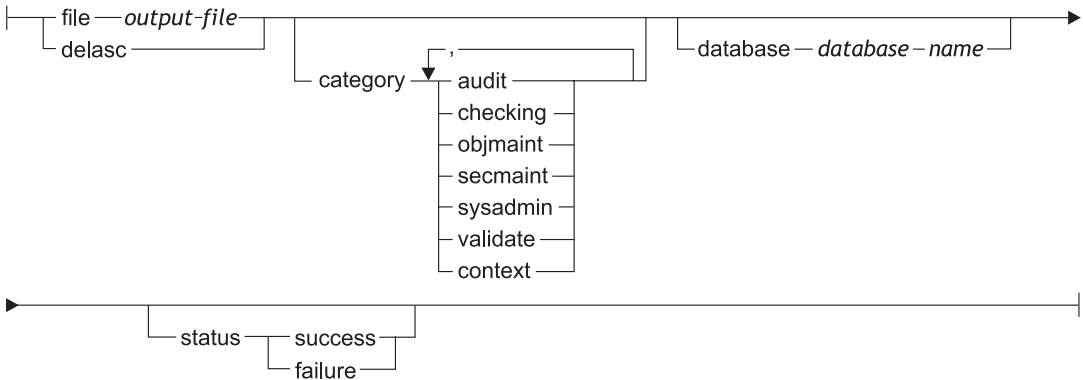


图 7. DB2AUDIT 语法

以下是每个参数的描述和隐含用法:

### configure

此参数允许修改该实例的 security 子目录中的 db2audit.cfg 配置文件。

即使在关闭该实例时也可更新此文件。当实例是活动时发生的更新会动态影响所有分区上 DB2 正在执行的审计。若已启动审计设施，并且正在审计可审计事件的审计类别，则配置文件上发生的配置操作会导致创建一条审计记录条目。

以下是可能对配置文件执行的操作：

- **RESET**。此操作导致配置文件还原为初始配置（其中，SCOPE 是除 CONTEXT 外的所有类别，STATUS 为 FAILURE，ERRORTYPE 为 NORMAL，而审计设施为 OFF）。若原始文件已丢失或损坏，此操作将创建一个新的审计配置文件。
- **SCOPE**。此操作指定要审计哪一类或哪几类的事件。此操作也允许集中审计并减少日志的增长。建议尽可能地限制所记录的事件的数量和类型，否则审计日志将迅速增长。

**注：** 请注意，缺省 SCOPE 是除 CONTEXT 外的所有类别，并且它可能导致迅速生成记录条目。与同步或异步方式一起使用，类别的选择可能导致性能明显降低以及磁盘需求的明显增加。

- **STATUS**。此操作指定是记录成功的事件，还是记录失败的事件，还是既记录成功事件又记录失败事件。

**注：** 上下文事件发生在操作的状态已知之前。因此，无论与此参数相关的值如何，都记录这类事件。

- **ERRORTYPE**。此操作指定是将审计错误返回给用户还是忽略它。此参数的值可以是：
  - **AUDIT**。所有错误，包括在审计设施中发生的错误，均由 DB2 管理，并将所有负的 SQLCODE 报告回给调用程序。
  - **NORMAL**。忽略 db2audit 生成的任何错误，并只将与执行的操作相关的错误的 SQLCODE 返回至该应用程序。

### **describe**

此参数将当前审计配置信息和状态显示至标准输出。

### **extract**

此参数允许将审计记录条目从审计日志移动至指示的目的地。若未指定可选的子句，则抽取所有审计记录条目并将它们放在平面报告文件中。若未指定 "extract" 参数，则将审计记录条目放在 security 目录下称为 db2audit.out 的文件中。若 output\_file 已存在，则返回一条出错信息。

以下是在抽取时可使用的可能选项：

- **FILE**。将抽取的审计记录条目放在文件 (output\_file) 中。

- DELASC。使用定界的 ASCII 格式来放置抽取的审计记录条目，以便适合装入 DB2 关系表中。将输出放在单独的文件中：每种类别一个文件。这些文件名是：
  - audit.del
  - checking.del
  - objmaint.del
  - secmaint.del
  - sysadmin.del
  - validate.del
  - context.del

当从审计日志中抽取时，DELASC 选项也允许您覆盖缺省审计字符串定界符 (“0xff”)。将使用 DELASC DELIMITER，后接您希望使用的新定界符，以便准备装入将容纳审计记录条目的表中。新装入定界符可以是单个字符（如 !）或代表十六进制数的四字节字符串（如 0xff）。有关更多信息，参考第261页的『审计设施提示和技术』。

- CATEGORY。将抽取指定类别的审计事件的审计记录条目。若未指定，则所有类别都适合抽取。
- DATABASE。将抽取指定数据库的审计记录条目。若未指定，则所有数据库都适合抽取。
- STATUS。将抽取指定状态的审计记录条目。若未指定，则所有记录条目都适合抽取。

**flush** 此参数强制将任何暂挂的审计记录条目写入审计日志。若审计设施处于错误状态，则也将引擎中的审计状态从“不能记录”复位为“准备记录”。

**prune** 此参数允许从审计日志中删除审计记录条目。若审计设施是活动的并已由审计指定了事件的“审计”类别，则在修剪审计日志之后将记录一条审计记录条目。

以下是在修剪时可使用的可能选项：

- ALL。将删除审计日志中的所有审计记录条目。
- DATE `yyyymmddhh`。用户可指定从审计日志中删除在指定的日期/时间当时或之前发生的所有审计记录条目。作为选项，用户可提供一个 `pathname`

审计设施在修剪审计日志时将该路径名作为一个临时空间。当日志文件驻留的磁盘已满，并且没有足够的磁盘空间允许修剪操作时，此临时空间允许修剪审计日志。

**start** 此参数导致审计设施根据 `db2audit.cfg` 文件的内容开始审计事件。在分区

的 DB2 实例中，当指定此子句时，将在所有分区上开始审计。若已为审计指定了事件的“审计”类别，则当启动审计设施时，将记录一条审计记录条目。

**stop** 此参数导致审计设施停止审计事件。在分区的 DB2 实例中，当指定此子句时，将在所有分区上停止审计。若已为审计指定了事件的“审计”类别，则当停止审计设施时，将记录一条审计记录条目。

---

## 审计设施消息

---

### SQL1322N 当写入审计日志文件时出错。

**解释：** 当调用 DB2 审计设施以将审计事件记录到审计日志文件中时，该设施遇到错误。审计日志驻留的文件系统上没有空间。

**用户回答：** 系统管理员应在此文件系统中释放空间，或修剪审计日志以减小其大小。

当有更多空间可用时，使用 `db2audit` 来删除内存中的任何数据，并将该审计程序复位为就绪状态。确保在修剪该日志前，进行适当的抽取，或建立该日志的副本，因为删除的记录条目是不可恢复的。

| **sqlcode:** -1322

| **sqlstate:** 50830

---

### SQL1323N 访问审计配置文件时出错。

**解释：** 审计配置文件 (`db2audit.cfg`) 不能打开或者无效。此错误的可能是原因是 `db2audit.cfg` 文件不存在或已损坏。

**用户回答：** 执行下列其中一项操作：

- 从该文件的保存版本复原。
- 通过发出以下命令复位审计设施配置文件：

```
db2audit reset
```

| **sqlcode:** -1323

| **sqlstate:** 57019

---

## 审计设施记录条目的布局

当使用 `DELTASC` 抽取选项从审计日志中抽取审计记录条目时，每个记录条目将具有下列表中显示的格式之一。每个表将从显示样本记录条目的内容开始。该记录条目中每一项的描述显示在相关的表中，一次显示一行。若该项很重要，将突出显示（**黑体**）该项的名称。这些项包含您最感兴趣的信息。

**注：**

1. 并非样本记录条目中的所有字段都有值。
2. 某些字段（如“尝试的访问”）以定界的 ASCII 格式存储为位图。然而，在此平面报告文件中，这些字段将显示为一组字符串，表示位图值。

表 6. *AUDIT* 事件的审计记录条目布局

timestamp=1998-06-24-11.54.05.151232;category=AUDIT;audit event=START; event correlator=0;event status=0; userid=boss;authid=BOSS;		
名称	格式	描述
Timestamp	CHAR(26)	审计事件的日期和时间。
<b>Category</b>	CHAR(8)	审计事件的类别。可能的值是: AUDIT
<b>Audit Event</b>	VARCHAR(32)	特定的审计事件。  可能的值包括: CONFIGURE、DB2AUD、EXTRACT、FLUSH、 PRUNE、START、STOP 和 UPDATE_ADMIN_CFG
Event Correlator	INTEGER	正审计的操作的相关标识符。用来标识哪些审计记录条目与单个事件相关。
Event Status	INTEGER	审计事件的状态, 由 SQLCODE 表示, 如下所示 成功的事件 > = 0 失败的事件 < 0
User ID	VARCHAR(1024)	审计事件发生时的用户标识。
Authorization ID	VARCHAR(128)	审计事件发生时的授权标识。

表 7. *CHECKING* 事件的审计记录条目布局

timestamp=1998-06-24-08.42.11.622984;category=CHECKING;audit event=CHECKING_OBJECT; event correlator=2;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; object name=F00;object type=DATABASE; access approval reason=DATABASE;access attempted=CONNECT;		
名称	格式	描述
Timestamp	CHAR(26)	审计事件的日期和时间。
<b>Category</b>	CHAR(8)	审计事件的类别。可能的值是: CHECKING
<b>Audit Event</b>	VARCHAR(32)	特定的审计事件。  可能的值包括: CHECKING_OBJECT 和 CHECKING_FUNCTION
Event Correlator	INTEGER	正审计的操作的相关标识符。用来标识哪些审计记录条目与单个事件相关。

表 7. CHECKING 事件的审计记录条目布局 (续)

<pre>timestamp=1998-06-24-08.42.11.622984;category=CHECKING;audit event=CHECKING_OBJECT; event correlator=2;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; object name=F00;object type=DATABASE; access approval reason=DATABASE;access attempted=CONNECT;</pre>		
名称	格式	描述
<b>Event Status</b>	INTEGER	审计事件的状态，由 SQLCODE 表示，如下所示 成功的事件 > = 0 失败的事件 < 0
Database Name	CHAR(8)	为之生成该事件的数据库的名称。若它是实例级审计事件，则为空白。
User ID	VARCHAR(1024)	审计事件发生时的用户标识。
Authorization ID	VARCHAR(128)	审计事件发生时的授权标识。
Origin Node Number	SMALLINT	审计事件发生时所在的节点号。
Coordinator Node Number	SMALLINT	协调程序的节点号。
Application ID	VARCHAR(255)	审计事件发生时正在使用的应用程序标识。
Application Name	VARCHAR(1024)	审计事件发生时正在使用的应用程序名。
Package Schema	VARCHAR(128)	审计事件发生时正在使用的软件包的模式。
Package Name	VARCHAR(128)	审计事件发生时正在使用的软件包的名称。
Package Section Number	SMALLINT	审计事件发生时正在使用的软件包中的段号。
Object Schema	VARCHAR(128)	为之生成审计事件的对象的模式。
Object Name	VARCHAR(128)	为之生成审计事件的对象的名称。
Object Type	VARCHAR(32)	为之生成审计事件的对象的类型。可能的值包括： TABLE、VIEW、ALIAS、FUNCTION、INDEX、PACKAGE、 DATA_TYPE、NODEGROUP、SCHEMA、 STORED_PROCEDURE、BUFFERPOOL、TABLESPACE、 EVENT_MONITOR、TRIGGER、DATABASE、INSTANCE、 FOREIGN_KEY、PRIMARY_KEY、UNIQUE_CONSTRAINT、 CHECK_CONSTRAINT、WRAPPER、SERVER、NICKNAME、 USER MAPPING、SERVER OPTION、TRANSFORM、 TYPE MAPPING、FUNCTION MAPPING、SUMMARY TABLES、和 NONE。
<b>Access Approval Reason</b>	CHAR(18)	指示为此审计事件批准访问的原因。可能的值包括：显示在此表后第一个列表中的那些值。



表 7. CHECKING 事件的审计记录条目布局 (续)

<pre>timestamp=1998-06-24-08.42.11.622984;category=CHECKING;audit event=CHECKING_OBJECT; event correlator=2;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; object name=F00;object type=DATABASE; access approval reason=DATABASE;access attempted=CONNECT;</pre>		
名称	格式	描述
<b>Access Attempted</b>	CHAR(18)	指定尝试的访问类型。可能的值包括： 显示在此表后第二个列表中的那些值。

以下是可能的 CHECKING 访问批准原因的列表：

**0x0000000000000001 ACCESS DENIED**

未批准访问；确切地说，拒绝了访问。

**0x0000000000000002 SYSADM**

批准访问；该应用程序 / 用户具有 SYSADM 权限。

**0x0000000000000004 SYSCTRL**

批准访问；该应用程序 / 用户具有 SYSCTRL 权限。

**0x0000000000000008 SYSMaint**

批准访问；该应用程序 / 用户具有 SYSMaint 权限。

**0x0000000000000010 DBADM**

批准访问；该应用程序 / 用户具有 DBADM 权限。

**0x0000000000000020 DATABASE PRIVILEGE**

批准访问；该应用程序 / 用户具有使用该数据库的显式特权。

**0x0000000000000040 OBJECT PRIVILEGE**

批准访问；该应用程序 / 用户具有使用该对象或功能的显式特权。

**0x0000000000000080 DEFINER**

批准访问；该应用程序 / 用户是该对象或功能的定义者。

**0x0000000000000100 OWNER**

批准访问；该应用程序 / 用户是该对象或功能的所有者。

**0x0000000000000200 CONTROL**

批准访问；该应用程序 / 用户对该对象或功能具有 CONTROL 特权。

**0x0000000000000400 BIND**

批准访问；该应用程序 / 用户对该软件包具有绑定特权。

以下是可能的 CHECKING 访问尝试类型的列表：

- 0x0000000000000002 ALTER**  
试图改变一个对象。
- 0x0000000000000004 DELETE**  
试图删除一个对象。
- 0x0000000000000008 INDEX**  
试图使用一个索引。
- 0x0000000000000010 INSERT**  
试图插入到对象中。
- 0x0000000000000020 SELECT**  
试图查询一个表或视图。
- 0x0000000000000040 UPDATE**  
试图更新对象中的数据。
- 0x0000000000000080 REFERENCE**  
试图在对象间建立关系约束。
- 0x0000000000000100 CREATE**  
试图创建一个对象。
- 0x0000000000000200 DROP**  
试图删除一个对象。
- 0x0000000000000400 CREATEIN**  
试图在另一个模式内创建一个对象。
- 0x0000000000000800 DROPIN**  
试图删除另一个模式内找到的对象。
- 0x0000000000001000 ALTERIN**  
试图改变或修改在另一个模式内找到的对象。
- 0x0000000000002000 EXECUTE**  
试图执行或运行一个应用程序。
- 0x0000000000004000 BIND**  
试图绑定或准备一个应用程序。
- 0x0000000000008000 SET EVENT MONITOR**  
试图设置事件监视器开关。
- 0x0000000000010000 SET CONSTRAINTS**  
试图设置对一个对象的约束。
- 0x0000000000020000 COMMENT ON**  
试图创建有关一个对象的注解。

**0x0000000000040000 GRANT**

试图将使用一个对象的特权授予另一个用户标识。

**0x0000000000080000 REVOKE**

试图从一个用户标识取消使用一个对象的特权。

**0x0000000000100000 LOCK**

试图锁定一个对象。

**0x0000000000200000 RENAME**

试图重命名一个对象。

**0x0000000000400000 CONNECT**

试图与一个对象连接。

**0x0000000000800000 SYS 组的成员**

试图访问或使用 SYS 组的成员。

**0x0000000001000000 Access All**

试图执行语句，对对象的所有必需特权被挂起（仅用于 DBADM/SYSADM）。

**0x0000000002000000 Drop All**

尝试删除多个对象。

**0x0000000004000000 LOAD**

尝试在表空间中装入表。

**0x0000000008000000 USE**

试图在表空间中创建表。

表 8. OBJMAINT 事件的审计记录条目布局

timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;audit event=CREATE_OBJECT; event correlator=3;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=AUDIT;object type=TABLE;		
名称	格式	描述
Timestamp	CHAR(26)	审计事件的日期和时间。
Category	CHAR(8)	审计事件的类别。可能的值是: OBJMAINT
Audit Event	VARCHAR(32)	特定的审计事件。  可能的值包括: CREATE_OBJECT、RENAME_OBJECT 和 DROP_OBJECT

表 8. OBJMAINT 事件的审计记录条目布局 (续)

timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;audit event=CREATE_OBJECT; event correlator=3;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=AUDIT;object type=TABLE;		
名称	格式	描述
Event Correlator	INTEGER	正审计的操作的相关标识符。可用来标识哪些审计记录条目与单个事件相关。
Event Status	INTEGER	审计事件的状态, 由 SQLCODE 表示, 如下所示 成功的事件 > = 0 失败的事件 < 0
Database Name	CHAR(8)	为之生成该事件的数据库的名称。若它是实例级审计事件, 则为空白。
User ID	VARCHAR(1024)	审计事件发生时的用户标识。
Authorization ID	VARCHAR(128)	审计事件发生时的授权标识。
Origin Node Number	SMALLINT	审计事件发生时所在的节点号。
Coordinator Node Number	SMALLINT	协调程序节点的节点号。
Application ID	VARCHAR(255)	审计事件发生时正在使用的应用程序标识。
Application Name	VARCHAR(1024)	审计事件发生时正在使用的应用程序名。
Package Schema	VARCHAR(128)	审计事件发生时正在使用的软件包的模式。
Package Name	VARCHAR(128)	审计事件发生时正在使用的软件包的名称。
Package Section Number	SMALLINT	审计事件发生时正在使用的软件包中的段号。
Object Schema	VARCHAR(128)	为之生成审计事件的对象的模式。
Object Name	VARCHAR(128)	为之生成审计事件的对象的名称。
Object Type	VARCHAR(32)	为之生成审计事件的对象的类型。可能的值包括: TABLE、VIEW、ALIAS、FUNCTION、INDEX、PACKAGE、 DATA_TYPE、NODEGROUP、SCHEMA、 STORED_PROCEDURE、BUFFERPOOL、TABLESPACE、 EVENT_MONITOR、TRIGGER、DATABASE、INSTANCE、 FOREIGN_KEY、PRIMARY_KEY、UNIQUE_CONSTRAINT、 CHECK_CONSTRAINT、WRAPPER、SERVER、NICKNAME、 USER MAPPING、SERVER OPTION、TRANSFORM、 TYPE MAPPING、FUNCTION MAPPING、SUMMARY TABLES、和 NONE。

表 9. SECMAINT 事件的审计记录条目布局

timestamp=1998-06-24-11.57.45.188101;category=SECMAINT;audit event=GRANT; event correlator=4;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155728;application name=db2bp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=T1;object type=TABLE; grantor=BOSS;grantee=WORKER;grantee type=USER;privilege=SELECT;		
名称	格式	描述
Timestamp	CHAR(26)	审计事件的日期和时间。
<b>Category</b>	CHAR(8)	审计事件的类别。可能的值是： SECMAINT
<b>Audit Event</b>	VARCHAR(32)	特定的审计事件。  可能的值包括：GRANT、REVOKE、IMPLICIT_GRANT、 IMPLICIT_REVOKE 和 UPDATE_DBM_CFG。
Event Correlator	INTEGER	正审计的操作的相关标识符。可用来标识哪些审计记录条目与 单个事件相关。
Event Status	INTEGER	审计事件的状态，由 SQLCODE 表示，如下所示 成功的事件 > = 0 失败的事件 < 0
Database Name	CHAR(8)	为之生成该事件的数据库的名称。若它是实例级审计事件，则 为空白。
User ID	VARCHAR(1024)	审计事件发生时的用户标识。
Authorization ID	VARCHAR(128)	审计事件发生时的授权标识。
Origin Node Number	SMALLINT	审计事件发生时所在的节点号。
Coordinator Node Number	SMALLINT	协调程序节点的节点号。
Application ID	VARCHAR(255)	审计事件发生时正在使用的应用程序标识。
Application Name	VARCHAR(1024)	审计事件发生时正在使用的应用程序名。
Package Schema	VARCHAR(128)	审计事件发生时正在使用的软件包的模式。
Package Name	VARCHAR(128)	审计事件发生时正在使用的软件包的名称。
Package Section Number	SMALLINT	审计事件发生时正在使用的软件包中的段号。
<b>Object Schema</b>	VARCHAR(128)	为之生成审计事件的对象的模式。
<b>Object Name</b>	VARCHAR(128)	为之生成审计事件的对象的名称。

表 9. SECMAINT 事件的审计记录条目布局 (续)

<pre>timestamp=1998-06-24-11.57.45.188101;category=SECMAINT;audit event=GRANT; event correlator=4;event status=0; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155728;application name=db2bp; package schema=NULLID;package name=SQLC28A1; package section=0;object schema=BOSS;object name=T1;object type=TABLE; grantor=BOSS;grantee=WORKER;grantee type=USER;privilege=SELECT;</pre>		
名称	格式	描述
<b>Object Type</b>	VARCHAR(32)	为之生成审计事件的对象的类型。可能的值包括: TABLE、VIEW、ALIAS、FUNCTION、INDEX、PACKAGE、DATA_TYPE、NODEGROUP、SCHEMA、STORED_PROCEDURE、BUFFERPOOL、TABLESPACE、EVENT_MONITOR、TRIGGER、DATABASE、INSTANCE、FOREIGN_KEY、PRIMARY_KEY、UNIQUE_CONSTRAINT、CHECK_CONSTRAINT、WRAPPER、SERVER、NICKNAME、USER MAPPING、SERVER OPTION、TRANSFORM、TYPE MAPPING、FUNCTION MAPPING、SUMMARY TABLES、和 NONE。
<b>Grantor</b>	VARCHAR(128)	授予者标识。
<b>Grantee</b>	VARCHAR(128)	被授予或取消特权或权限的接受者标识。
Grantee Type	VARCHAR(32)	被授予或取消权限的接受者的类型。可能的值包括: USER、GROUP 或 BOTH。
<b>Privilege or Authority</b>	CHAR(18)	指示授予或取消的特权或权限的类型。可能的值包括: 显示在此表后的列表中的那些值。

以下是可能的 SECMAINT 特权或权限的列表:

**0x0000000000000001 Control Table**

被授予的或取消的对一个表的控制特权。

**0x0000000000000002 ALTER TABLE**

被授予的或取消的改变一个表的特权。

**0x0000000000000004 ALTER TABLE with GRANT**

对于允许授予特权的一个表, 被授予的或取消的改变该表的特权。

**0x0000000000000008 DELETE TABLE**

被授予的或取消的删除一个表的特权。

**0x0000000000000010 DELETE TABLE with GRANT**

对于允许授予特权的一个表, 被授予的或取消的删除该索引的特权。

**0x0000000000000020 Table Index**

被授予的或取消的对索引的特权。

**0x0000000000000040 Table Index with GRANT**

对于允许授予特权的一个索引，被授予的或取消的对该索引的特权。

**0x0000000000000080 Table INSERT**

被授予的或取消的对表进行插入的特权。

**0x0000000000000100 Table INSERT with GRANT**

对于允许授予特权的一个表，被授予的或取消的对表进行插入的特权。

**0x0000000000000200 Table SELECT**

被授予的或取消的对表进行选择的特权。

**0x0000000000000400 Table SELECT with GRANT**

对于允许授予特权的一个表，被授予的或取消的对表进行选择的特权。

**0x0000000000000800 Table UPDATE**

被授予的或取消的对表进行更新的特权。

**0x0000000000001000 Table UPDATE with GRANT**

对于允许授予特权的一个表，被授予的或取消的对表进行更新的特权。

**0x0000000000002000 Table REFERENCE**

被授予的或取消的对表进行引用的特权。

**0x0000000000004000 Table REFERENCE with GRANT**

对于允许授予特权的一个表，被授予的或取消的对表进行引用的特权。

**0x0000000000008000 Package BIND**

被授予的或取消的对一个软件包的 BIND 特权。

**0x0000000000010000 Package EXECUTE**

被授予的或取消的对一个软件包的 EXECUTE 特权。

**0x0000000000020000 CREATEIN Schema**

被授予的或取消的对一个模式的 CREATEIN 特权。

**0x0000000000040000 CREATEIN Schema with GRANT**

对于允许授予特权的一个模式，被授予的或取消的对该模式的 CREATEIN 特权。

**0x0000000000080000 DROPIN Schema**

被授予的或取消的对一个模式的 DROPIN 特权。

**0x0000000000100000 DROPIN Schema with GRANT**

对于允许授予特权的一个模式，被授予的或取消的对该模式的 DROPIN 特权。

**0x000000000200000 ALTERIN Schema**

被授予的或取消的对一个模式的 ALTERIN 特权。

**0x000000000400000 ALTERIN Schema with GRANT**

对于允许授予特权的一个模式，被授予的或取消的对该模式的 ALTERIN 特权。

**0x000000000800000 DBADM Authority**

被授予的或取消的 DBADM 权限。

**0x0000000001000000 CREATETAB Authority**

被授予的或取消的 Createtab 权限。

**0x000000000200000 BINDADD Authority**

被授予的或取消的 Bindadd 权限。

**0x0000000004000000 CONNECT Authority**

被授予的或取消的 CONNECT 权限。

**0x0000000008000000 Create not fenced Authority**

被授予的或取消的 Create not fenced 权限。

**0x0000000010000000 Implicit Schema Authority**

被授予的或取消的 Implicit Schema 权限。

**0x0000000020000000 Server PASSTHRU**

被授予的或取消的对此服务器（联合数据库数据源）使用通过设施的特权。

**0x0000000100000000 Table Space USE**

被授予的或取消的在表空间中创建表的特权。

**0x0000000200000000 Table Space USE with GRANT**

被授予的或取消的在表空间中创建表并允许进行特权授权的特权。

**0x0000000400000000 Column UPDATE**

被授予的或取消的对一个表的一个或多个特定列进行更新的特权。

**0x0000000800000000 Column UPDATE with GRANT**

对于允许授予特权的一个表，被授予的或取消的对该表的一个或多个特定列进行更新的特权。

**0x0000001000000000 Column REFERENCE**

被授予的或取消的对一个表的一个或多个特定列进行引用的特权。

**0x0000002000000000 Column REFERENCE with GRANT**

对于允许授予特权的一个表，被授予的或取消的对该表的一个或多个特定列进行引用的特权。



## 0x0000004000000000 LOAD Authority

被授予的或取消的 LOAD 权限。

表 10. SYSADMIN 事件的审计记录条目布局

timestamp=1998-06-24-11.54.04.129923;category=SYSADMIN;audit event=DB2AUDIT; event correlator=1;event status=0; userid=boss;authid=BOSS; application id=*LOCAL.boss.980624155404;application name=db2audit;		
名称	格式	描述
Timestamp	CHAR(26)	审计事件的日期和时间。
Category	CHAR(8)	审计事件的类别。可能的值是: SYSADMIN
Audit Event	VARCHAR(32)	特定的审计事件。  可能的值包括: 显示在此表后的列表中的那些值。
Event Correlator	INTEGER	正审计的操作的相关标识符。可用来标识哪些审计记录条目与单个事件相关。
Event Status	INTEGER	审计事件的状态, 由 SQLCODE 表示, 如下所示 成功的事件 > = 0 失败的事件 < 0
Database Name	CHAR(8)	为之生成该事件的数据库的名称。若它是实例级审计事件, 则为空白。
User ID	VARCHAR(1024)	审计事件发生时的用户标识。
Authorization ID	VARCHAR(128)	审计事件发生时的授权标识。
Origin Node Number	SMALLINT	审计事件发生时所在的节点号。
Coordinator Node Number	SMALLINT	协调程序节点的节点号。
Application ID	VARCHAR(255)	审计事件发生时正在使用的应用程序标识。
Application Name	VARCHAR(1024)	审计事件发生时正在使用的应用程序名。
Package Schema	VARCHAR(128)	审计事件发生时正在使用的软件包的模式。
Package Name	VARCHAR(128)	审计事件发生时正在使用的软件包的名称。
Package Section Number	SMALLINT	审计事件发生时正在使用的软件包中的段号。

以下是可能的 SYSADMIN 审计事件的列表:

表 11. SYSADMIN 审计事件

START_DB2	SET_RUNTIME_DEGREE
STOP_DB2	SET_TABLESPACE_CONTAINERS
CREATE_DATABASE	UNCATALOG_DB
DROP_DATABASE	UNCATALOG_DCS_DB
UPDATE_DBM_CFG	UNCATALOG_NODE
UPDATE_DB_CFG	UPDATE_ADMIN_CFG
CREATE_TABLESPACE	UPDATE_MON_SWITCHES
DROP_TABLESPACE	LOAD_TABLE
ALTER_TABLESPACE	DB2AUDIT
RENAME_TABLESPACE	SET_APPL_PRIORITY
CREATE_NODEGROUP	CREATE_DB_AT_NODE
DROP_NODEGROUP	KILLDBM
ALTER_NODEGROUP	MIGRATE_SYSTEM_DIRECTORY
CREATE_BUFFERPOOL	DB2REMO
DROP_BUFFERPOOL	DB2AUD
ALTER_BUFFERPOOL	MERGE_DBM_CONFIG_FILE
CREATE_EVENT_MONITOR	UPDATE_CLI_CONFIGURATION
DROP_EVENT_MONITOR	OPEN_TABLESPACE_QUERY
ENABLE_MULTIPAGE	SINGLE_TABLESPACE_QUERY
MIGRATE_DB_DIR	CLOSE_TABLESPACE_QUERY
DB2TRC	FETCH_TABLESPACE
DB2SET	OPEN_CONTAINER_QUERY
ACTIVATE_DB	FETCH_CONTAINER_QUERY
ADD_NODE	CLOSE_CONTAINER_QUERY
BACKUP_DB	GET_TABLESPACE_STATISTICS
CATALOG_NODE	DESCRIBE_DATABASE
CATALOG_DB	ESTIMATE_SNAPSHOT_SIZE
CATALOG_DCS_DB	READ_ASYNC_LOG_RECORD
CHANGE_DB_COMMENT	PRUNE_RECOVERY_HISTORY
DEACTIVATE_DB	UPDATE_RECOVERY_HISTORY
DROP_NODE_VERIFY	QUIESCE_TABLESPACE
FORCE_APPLICATION	UNLOAD_TABLE
GET_SNAPSHOT	UPDATE_DATABASE_VERSION
LIST_DRDA_INDOUBT_TRANSACTIONS	CREATE_INSTANCE
MIGRATE_DB	DELETE_INSTANCE
RESET_ADMIN_CFG	SET_EVENT_MONITOR
RESET_DB_CFG	GRANT_DBADM
RESET_DBM_CFG	REVOKE_DBADM
RESET_MONITOR	GRANT_DB_AUTHORITIES
RESTORE_DB	REVOKE_DB_AUTHORITIES
ROLLFORWARD_DB	REDIST_NODEGROUP

表 12. VALIDATE 事件的审计记录条目布局

timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;audit event=CHECK_GROUP_MEMBERSHIP; event correlator=2;event status=-1092; database=F00;userid=boss;authid=BOSS;execution id=newton; application id=*LOCAL.newton.980624124210;application name=testapp; auth type=SERVER;		
名称	格式	描述
Timestamp	CHAR(26)	审计事件的日期和时间。
<b>Category</b>	CHAR(8)	审计事件的类别。可能的值是： VALIDATE
<b>Audit Event</b>	VARCHAR(32)	特定的审计事件。  可能的值包括： GET_GROUPS, GET_USERID, AUTHENTICATE_PASSWORD 和 VALIDATE_USER。
Event Correlator	INTEGER	正审计的操作的相关标识符。可用于标识哪些审计记录条目与单个事件相关。
Event Status	INTEGER	审计事件的状态，由 SQLCODE 表示，如下所示 成功的事件 > = 0 失败的事件 < 0
Database Name	CHAR(8)	为之生成该事件的数据库的名称。若它是实例级审计事件，则为空白。
User ID	VARCHAR(1024)	审计事件发生时的用户标识。
Authorization ID	VARCHAR(128)	审计事件发生时的授权标识。
<b>Execution ID</b>	VARCHAR(1024)	审计事件发生时正在使用的执行标识。
Origin Node Number	SMALLINT	审计事件发生时所在的节点号。
Coordinator Node Number	SMALLINT	协调程序节点的节点号。
Application ID	VARCHAR(255)	审计事件发生时正在使用的应用程序标识。
Application Name	VARCHAR(1024)	审计事件发生时正在使用的应用程序名。
<b>Authentication Type</b>	VARCHAR(32)	审计事件发生时的认证类型。
Package Schema	VARCHAR(128)	审计事件发生时正在使用的软件包的模式。
Package Name	VARCHAR(128)	审计事件发生时正在使用的软件包的名称。
Package Section Number	SMALLINT	审计事件发生时正在使用的软件包中的段号。

表 13. CONTEXT 事件的审计记录条目布局

<pre>timestamp=1998-06-24-08.42.41.476840;category=CONTEXT;audit event=EXECUTE_IMMEDIATE; event correlator=3; database=F00;userid=boss;authid=BOSS; application id=*LOCAL.newton.980624124210;application name=testapp; package schema=NULLID;package name=SQLC28A1; package section=203;text=create table audit(c1 char(10), c2 integer);</pre>		
名称	格式	描述
Timestamp	CHAR(26)	审计事件的日期和时间。
Category	CHAR(8)	审计事件的类别。可能的值是： CONTEXT
Audit Event	VARCHAR(32)	特定的审计事件。  可能的值包括：显示在此表后的列表中的那些值。
Event Correlator	INTEGER	正审计的操作的相关标识符。用来标识哪些审计记录条目与单个事件相关。
Database Name	CHAR(8)	为之生成该事件的数据库的名称。若它是实例级审计事件，则为空白。
User ID	VARCHAR(1024)	审计事件发生时的用户标识。
Authorization ID	VARCHAR(128)	审计事件发生时的授权标识。
Origin Node Number	SMALLINT	审计事件发生时所在的节点号。
Coordinator Node Number	SMALLINT	协调程序节点的节点号。
Application ID	VARCHAR(255)	审计事件发生时正在使用的应用程序标识。
Application Name	VARCHAR(1024)	审计事件发生时正在使用的应用程序名。
Package Schema	VARCHAR(128)	审计事件发生时正在使用的软件包的模式。
Package Name	VARCHAR(128)	审计事件发生时正在使用的软件包的名称。
Package Section Number	SMALLINT	审计事件发生时正在使用的软件包中的段号。
语句文本 (语句)	CLOB(32K)	SQL 语句的文本 (若适用)。若 SQL 语句文本不可用，则为空值。

以下是可能的 CONTEXT 审计事件的列表:

表 14. CONTEXT 审计事件

CONNECT	SET_APPL_PRIORITY
CONNECT_RESET	RESET_DB_CFG
ATTACH	GET_DB_CFG
DETACH	GET_DFLT_CFG
DARI_START	UPDATE_DBM_CFG
DARI_STOP	SET_MONITOR
BACKUP_DB	GET_SNAPSHOT
RESTORE_DB	ESTIMATE_SNAPSHOT_SIZE
ROLLFORWARD_DB	RESET_MONITOR
OPEN_TABLESPACE_QUERY	OPEN_HISTORY_FILE
FETCH_TABLESPACE	CLOSE_HISTORY_FILE
CLOSE_TABLESPACE_QUERY	FETCH_HISTORY_FILE
OPEN_CONTAINER_QUERY	SET_RUNTIME_DEGREE
CLOSE_CONTAINER_QUERY	UPDATE_AUDIT
FETCH_CONTAINER_QUERY	DBM_CFG_OPERATION
SET_TABLESPACE_CONTAINERS	DISCOVER
GET_TABLESPACE_STATISTIC	OPEN_CURSOR
READ_ASYNC_LOG_RECORD	CLOSE_CURSOR
QUIESCE_TABLESPACE	FETCH_CURSOR
LOAD_TABLE	EXECUTE
UNLOAD_TABLE	EXECUTE_IMMEDIATE
UPDATE_RECOVERY_HISTORY	PREPARE
PRUNE_RECOVERY_HISTORY	DESCRIBE
SINGLE_TABLESPACE_QUERY	BIND
LOAD_MSG_FILE	REBIND
UNQUIESCE_TABLESPACE	RUNSTATS
ENABLE_MULTIPAGE	REORG
DESCRIBE_DATABASE	REDISTRIBUTE
DROP_DATABASE	COMMIT
CREATE_DATABASE	ROLLBACK
ADD_NODE	REQUEST_ROLLBACK
FORCE_APPLICATION	IMPLICIT_REBIND

## 审计设施提示和技术

在大多数情况下，当使用 CHECKING 事件时，审计记录条目中的对象类型字段是要检查的对象，以了解试图访问该对象的用户标识是否拥有必需的特权或权限。例如，若一个用户试图通过添加列来改变一个表，则 CHECKING 事件审计记录条目将指示尝试的访问是 "ALTER"，且要检查的对象类型是 "TABLE"（注意：不是列，因为它是必须检查的表特权）。

然而，当该检查要验证是否存在一个数据库权限来允许用户标识创建或绑定对象或删除对象时，虽然存在对该数据库的检查，对象类型字段仍将指定要创建、绑定或删除的对象（而不是数据库本身）。

当在表上创建一个索引时，必需创建索引的特权，因此 CHECKING 事件审计记录条目将具有访问尝试类型“索引”而不是“创建”。

当绑定一个已存在的软件包时，会为该软件包的 DROP 创建 OBJMAINT 事件审计记录条目，然后为该软件包新副本的 CREATE 创建另一个 OBJMAINT 事件审计记录条目。

SQL“数据定义语言”(DDL)可生成记录为成功的 OBJMAINT 或 SECMAINT 事件。然而，在记录该事件后，一个后续的错误可能会导致 ROLLBACK 发生。这样就创建不了该对象；或者 GRANT 或 REVOKE 操作不能完成。在这种情况下，使用 CONTEXT 事件变得很重要。这类 CONTEXT 事件审计记录条目，特别是结束该事件的语句，将指示尝试的操作的完成性质。

当抽取使用定界的 ASCII 格式（以适合于装入 DB2 关系表）的审计记录条目时，您应对在该语句文本字段内使用的定界符的有关情况十分清楚。这可在抽取该定界的 ASCII 文件时执行，并使用下列语句来执行：

```
db2audit extract delasc delimiter <load delimiter>
```

装入定界符可以是单个字符（如 "），或是表示十六进制值的四字节字符串（如 "0xff"）。有效命令的示例是：

```
db2audit extract delasc
db2audit extract delasc delimiter !
db2audit extract delasc delimiter 0xff
```

若您在抽取时使用的定界符不是缺省装入定界符 ("），则应在 LOAD 命令上使用 MODIFIED BY 选项。下面是将 "0xff" 用作定界符的 LOAD 命令的示例一部分：

```
db2 load from context.del of del modified by chardel0xff replace into ...
```

它将覆盖缺省装入字符串定界符 "0xff"。

---

## 控制 DB2 审计设施活动

当讨论控制审计设施活动时，我们将使用一个简单的方案：用户 *newton* 运行一个名为 *testapp* 的应用程序，以连接和创建一个表。在下面讨论的每个示例中都使用这个相同的应用程序。

我们从一个极端情况的示例开始：您已确定审计所有成功和不成功的审计事件，因此将以下列方式配置审计设施：

db2audit configure scope all status both

**注：**它为每个可能的可审计事件创建审计记录条目。因此，会将许多记录条目写入审计日志，这降低了数据库管理器的性能。这里显示这个极端情况，仅用于演示；不建议您使用上面显示的命令配置审计设施。

在对此配置开始使用审计设施（使用“db2audit start”），并运行 *testapp* 应用程序之后，生成了下列记录条目，将它们放在审计日志中。通过从该日志中抽取审计记录条目，您将看到为该应用程序执行的两个操作生成的下列记录条目：

**操作 创建的记录条目的类型**

## **CONNECT**

```
timestamp=1998-06-24-08.42.10.555345;category=CONTEXT;  
audit event=CONNECT;event correlator=2;database=F00;  
application id=*LOCAL.newton.980624124210;  
application name=testapp;
```

```
timestamp=1998-06-24-08.42.10.944374;category=VALIDATE;  
audit event=AUTHENTICATION;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;execution id=newton;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.561187;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.594620;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.622984;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
object name=F00;object type=DATABASE;access approval reason=DATABASE;  
access attempted=CONNECT;
```

```
timestamp=1998-06-24-08.42.11.801554;category=CONTEXT;  
audit event=COMMIT;event correlator=2;database=F00;userid=boss;  
authid=BOSS;application id=*LOCAL.newton.980624124210;
```

```
application name=testapp;

timestamp=1998-06-24-08.42.41.450975;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=2;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;object schema=NULLID;
object name=SQLC28A1;object type=PACKAGE;
access approval reason=OBJECT;access attempted=EXECUTE;
```

## CREATE TABLE

```
timestamp=1998-06-24-08.42.41.476840;category=CONTEXT;
audit event=EXECUTE_IMMEDIATE;event correlator=3;database=F00;
userid=boss;authid=BOSS;application id=*LOCAL.newton.980624124210;
application name=testapp;package schema=NULLID;package name=SQLC28A1;
package section=203;text=create table audit(c1 char(10), c2 integer);
```

```
timestamp=1998-06-24-08.42.41.539692;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=3;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;package section=0;
object schema=BOSS;object name=AUDIT;object type=TABLE;
access approval reason=DATABASE;access attempted=CREATE;
```

```
timestamp=1998-06-24-08.42.41.570876;category=CHECKING;
audit event=CHECKING_OBJECT;event correlator=3;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;package section=0;
object name=BOSS;object type=SCHEMA;access approval reason=DATABASE;
access attempted=CREATE;
```

```
timestamp=1998-06-24-08.42.41.957524;category=OBJMAINT;
audit event=CREATE_OBJECT;event correlator=3;event status=0;
database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;package section=0;
object schema=BOSS;object name=AUDIT;object type=TABLE;
```

```
timestamp=1998-06-24-08.42.42.018900;category=CONTEXT;audit event=COMMIT;
event correlator=3;database=F00;userid=boss;authid=BOSS;
application id=*LOCAL.newton.980624124210;application name=testapp;
package schema=NULLID;package name=SQLC28A1;
```

正如您所见到的那样，由于审计配置请求所有可能的审计事件和类型的审计，因此从该审计配置生成了相当大量的审计记录条目。

在大多数情况下，您将配置审计设施，以获得您希望审计的事件的更具体或更集中的视图。例如，您可能只想审计那些失效的事件。在这种情况下，可按如下所示配置审计设施：



```
db2audit configure scope audit,checking,objmaint,secmaint,sysadmin,  
validate status failure
```

注：此配置是初始审计配置，或是在复位该审计配置时出现的配置。

在对此配置开始使用审计设施并运行 *testapp* 应用程序之后，生成了下列记录条目，将它们放在审计日志中。（而且我们假定 *testapp* 以前未运行过。）通过从该日志中抽取审计记录条目，您将看到为该应用程序执行的两个操作生成的下列记录条目：

**操作**    创建的记录条目的类型

### CONNECT

```
timestamp=1998-06-24-08.42.11.527490;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.561187;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

```
timestamp=1998-06-24-08.42.11.594620;category=VALIDATE;  
audit event=CHECK_GROUP_MEMBERSHIP;event correlator=2;  
event status=-1092;database=F00;userid=boss;authid=BOSS;  
execution id=newton;application id=*LOCAL.newton.980624124210;  
application name=testapp;auth type=SERVER;
```

### CREATE TABLE

（无）

由于此审计配置仅在该事件尝试失败时才请求所有可能的审计事件（除 CONTEXT 外）的审计，因此，此审计配置生成的审计记录条目少得多。通过更改该审计配置，您可控制生成的审计记录条目的类型和性质。

当您想审计的那些事件被成功授予对一个对象的特权时，审计设施可允许您创建审计记录条目。在这种情况下，您可按如下所示配置审计设施：

```
db2audit configure scope checking status success
```

在对此配置开始使用审计设施并运行 *testapp* 应用程序之后，生成了下列记录条目，将它们放在审计日志中。（而且我们假定 *testapp* 以前未运行过。）通过从该日志中抽取审计记录条目，您将看到为该应用程序执行的两个操作生成的下列记录条目：

**操作**    创建的记录条目的类型

### CONNECT

```
timestamp=1998-06-24-08.42.11.622984;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;
```

```
timestamp=1998-06-24-08.42.41.450975;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=2;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;object schema=NULLID;  
object name=SQLC28A1;object type=PACKAGE;  
access approval reason=OBJECT;access attempted=EXECUTE;
```

```
timestamp=1998-06-24-08.42.41.539692;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=3;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;package section=0;  
object schema=BOSS;object name=AUDIT;object type=TABLE;  
access approval reason=DATABASE;access attempted=CREATE;
```

```
timestamp=1998-06-24-08.42.41.570876;category=CHECKING;  
audit event=CHECKING_OBJECT;event correlator=3;event status=0;  
database=F00;userid=boss;authid=BOSS;  
application id=*LOCAL.newton.980624124210;application name=testapp;  
package schema=NULLID;package name=SQLC28A1;package section=0;  
object name=BOSS;object type=SCHEMA;access approval reason=DATABASE;  
access attempted=CREATE;
```

## CREATE TABLE

(无)

---

## 第4部分 移动数据



---

## 第7章 数据移动实用程序

“装入”实用程序将数据移入表中、扩展现存的索引并生成统计信息。当涉及到大量数据时，“装入”移动数据的速度较“导入”实用程序更快。使用“导出”实用程序卸载的数据可以使用“装入”实用程序装入。

“自动装入程序”实用程序将大量数据进行分割，然后将分割的数据装入一个分区数据库的不同分区。

“导入”和“导出”实用程序将数据在一个表或视图与另一个数据库或电子表格程序之间、DB2 数据库之间、以及 DB2 数据库与使用 DB2 Connect 的主机数据库之间移动。

“数据复制”（以前为 DataPropagator Relational (DPROPR)）是“DB2 通用数据库”的一个组件，它允许将表更新自动复制到其他 DB2 关系数据库中的其他表中。

**注：**有关这些主题的全部信息以及 *Command Reference* 和 *Administrative API Reference* 中对应的主题，已全部合并到 *Data Movement Utilities Guide and Reference* 中。

*Data Movement Utilities Guide and Reference* 是有关这些主题的主要的、单一的信息源。

要查找关于复制的更多信息，参见 *Replication Guide and Reference*。



---

## 第5部分 恢复





---

## 第8章 恢复数据库

当情况变坏时，您将需要能够恢复数据库。问题可能包括掉电故障、应用程序故障以及媒体和存储器故障。要确保当发生此类问题时可以恢复，您需要备份或复制整个数据库或构成数据库的表空间。然后，可以在发生数据库问题之后使用这些备份来复原数据库。

在发生问题之后重建数据库称为恢复。崩溃恢复在发生故障后自动尝试恢复数据库。崩溃恢复防止数据库处于不一致或不可用状态。当数据库发生故障时，针对该数据库的事务可能未完成。崩溃恢复回滚未完成的事务或提交已完成的事务。这些操作使数据库一致并可用。

当数据库损坏且您不能确定数据库的内容时，还有其它两种恢复类型。恢复损坏的数据库的两种方法是：版本恢复和前滚恢复。若您正在使用只读数据库或不关心数据库中正在记录的事务，则可能只需要版本恢复。若已经备份数据库，则可以应用或复原该数据库的副本。这称为版本恢复。若正在使用应用了事务的数据库，并且您需要知道数据库已应用了所有更改，则需要完成前滚恢复。前滚恢复涉及恢复数据库的备份。然后，您必须对数据库应用记录事务的日志的记录条目。日志的应用会重复针对数据库的所有活动，以便使数据库处于刚好在故障点之前的状态。使用此恢复方法，不会丢失对数据库的任何更改。记录时此恢复方法的关键所在。

**注：**有关这些主题的全部信息以及 *Command Reference* 和 *Administrative API Reference* 中对应的主题，已全部合并到 *Data Recovery and High Availability Guide and Reference* 中。

*Data Recovery and High Availability Guide and Reference* 是有关这些主题的主要的、单一的信息源。



---

## 第6部分 附录



---

## 附录A. 命名规则

转至描述您需要有关其信息的命名规则的章节:

- 『一般命名规则』
- 『对象命名规则』
- 第280页的『如何在联合系统中保留区分大小写的值』

---

### 一般命名规则

除非另有指定, 所有名称可以包括下列字符:

- A 至 Z。在大多数名称中使用, 会将字符 A 至 Z 从小写转换为大写。
- 0 至 9
- @、#、\$ 和 \_ (下划线字符)

名称不能以数字或下划线字符开始。

不要使用 SQL 保留字来命名表、视图、列、索引或授权标识。有关 SQL 保留字的列表, 参见 *SQL Reference*。

取决于操作系统以及使用 DB2 的环境, 还有其它特殊字符可能起作用。然而, 虽然它们可能起作用, 但不能保证它们将一定起作用。建议在命名数据库中的对象时不要使用其他特殊字符。

---

### 对象命名规则

所有对象遵循“一般命名规则”。另外, 某些对象还具有下面所显示的附加限制。

表 15. 数据库、数据库别名和实例命名规则

对象	准则
<ul style="list-style-type: none"> <li>• 数据库</li> <li>• 数据库别名</li> <li>• 实例</li> </ul>	<ul style="list-style-type: none"> <li>• 数据库名在它们被编目的位置中必须是唯一的。在基于 UNIX 的 DB2 实现上，此位置是一个目录路径，而在 Windows 实现上，它是盘符。</li> <li>• 数据库别名在系统数据库目录中必须是唯一的。当创建新数据库时，该别名缺省为数据库名。因此，不能使用已作为数据库别名的名称来创建数据库，即使没有具有该名称的数据库。</li> <li>• 数据库、数据库别名和实例名最多只能包含 8 个字节。</li> <li>• 在 Windows NT 和 Windows 2000 系统上，任何实例都不能与服务名同名。</li> </ul> <p><b>注：</b> 为避免潜在的问题，若您打算在通信环境中使用数据库，则不要在数据库名称中使用特殊字符 @、# 和 \$。而且，因为不是所有键盘都有这些字符，因此若您计划以另一语言使用该数据库，则不要使用这些特殊字符。</p>

表 16. 数据库对象命名规则

对象	准则
<ul style="list-style-type: none"> <li>• 别名</li> <li>• 缓冲池</li> <li>• 列</li> <li>• 事件监视器</li> <li>• 索引</li> <li>• 方法</li> <li>• 节点组</li> <li>• 模式</li> <li>• 存储过程</li> <li>• 表</li> <li>• 表空间</li> <li>• 触发器</li> <li>• UDF</li> <li>• UDT</li> <li>• 视图</li> </ul>	<p>最多可以包含 18 个字节，但下列名称除外：</p> <ul style="list-style-type: none"> <li>• 表名（包括视图名、摘要表名、别名和相关名），它们最多可以包含 128 个字节</li> <li>• 列名，它们最多可以包含 30 个字节</li> <li>• 模式名，它们最多可以包含 30 个字节</li> <li>• 对象名也可以包含： <ul style="list-style-type: none"> <li>- 有效的重音字符（如 ö）</li> <li>- 除多字节空格外的多字节字符（用于多字节环境）</li> </ul> </li> </ul>

### 关于模式名的附加信息

- 不能复制模式名超过 18 个字节的表。
- 用户定义类型 (UDT) 不能具有超过 8 个字节的模式名。

- 下列模式名是保留字，因此不能使用：  
SYSCAT、SYSFUN、SYSIBM、SYSSTAT。
- 要避免将来可能发生迁移问题，不要使用以 SYS 开头的模式名。数据库管理器将不允许您使用以 SYS 开头的模式名来创建触发器、用户定义类型或用户定义函数。
- 建议您不要将 SESSION 用作模式名。已说明临时表必须由 SESSION 限定。因此，应用程序有可能说明与持久表同名的临时表，在这种情况下，应用程序逻辑会变得过于复杂。除非是在处理已说明临时表，否则应避免使用模式 SESSION。

表 17. 用户、用户标识和组命名规则

对象	准则
<ul style="list-style-type: none"> <li>• 组名</li> <li>• 用户名</li> <li>• 用户标识</li> </ul>	<ul style="list-style-type: none"> <li>• 组名最多可以包含 8 个字节。</li> <li>• 基于 UNIX 的系统上的用户标识最多可以包含 8 个字符。</li> <li>• Windows 上的用户名最多可以包含 30 个字符。Windows NT 和 Windows 2000 当前的实际限制为 20 个字符。</li> <li>• 当使用 DCE 认证时，用户名的限制为 8 个字符。</li> <li>• 若不使用 DCE 或“客户机”认证，当显式指定了用户名和密码时，支持使用超过 8 个字符的用户名将非 Windows 32 位客户机连接至 Windows NT 和 Windows 2000。</li> <li>• 名称和标识不能：               <ul style="list-style-type: none"> <li>- 为 USERS、ADMINS、GUESTS、PUBLIC、LOCAL 或 <i>SQL Reference</i> 中列示的任何 SQL 保留字。</li> <li>- 以 IBM、SQL 或 SYS 开始。</li> <li>- 包含重音字符。</li> </ul> </li> </ul> <p>在基于 UNIX 的系统上，组和用户可以同名。对于 GRANT 语句，您必须指定您是引用组还是用户。对于 REVOKE 语句，指定用户或组取决于对于具有不同 GRANTEETYPE 值的 GRANTEE，在授权目录表中是否存在多行。</p> <p>在 Windows NT 上，本地组、全局组和用户不能有相同的名称。</p> <p>在 OS/2 上，组和用户不能有相同的名称。</p> <p>注：</p> <ol style="list-style-type: none"> <li>1. 有些操作系统允许区分大小写的用户标识和密码。您应检查操作系统文档，看看是否是这种情况。</li> <li>2. 从成功的 CONNECT 或 ATTACH 返回的授权标识截断为 8 个字符。省略号 (...) 附加到授权标识上，并且 SQLWARN 字段包含警告以指示发生了截断。有关更多信息，参见 <i>SQL Reference</i> 中的 CONNECT 语句。</li> </ol>

## 关于密码的附加信息

您可能需要执行密码维护任务。因为这类任务必须在服务器上进行，并且许多用户并不能或不能轻松地在服务器环境下工作，故执行这些任务可能是一个重大的挑战。DB2 UDB 提供了一个无需在服务器上更新和验证密码的方法。例如，DB2 OS/390 版的版本 5 支持这种更改用户密码的方法。若收到出错信息 SQL1404N “密码到期”，则使用 CONNECT 语句来更改密码，如下所示：

```
CONNECT TO <database> USER <userid> USING <password>  
NEW <new_password> CONFIRM <new_password>
```

也可使用 “DB2 客户机配置辅助程序” (CCA) 的 “密码更改” 对话来更改密码。有关这些密码更改方法的更多信息，参考 *SQL Reference* 和 CCA 联机帮助。

表 18. 联合数据库对象命名规则

对象	准则
<ul style="list-style-type: none"><li>函数映射</li><li>索引规范</li><li>别名</li><li>服务器</li><li>类型映射</li><li>用户映射</li><li>包装器</li></ul>	<ul style="list-style-type: none"><li>别名、映射、索引规范、服务器和包装器名不能超过 128 个字节。</li><li>服务器和别名选项以及选项设置限制为 255 个字节。</li><li>联合数据库对象的名称也可以包含：<ul style="list-style-type: none"><li>有效的重音字母（如 ö）</li><li>除多字节空格外的多字节字符（用于多字节环境）</li></ul></li></ul>

---

## 在对象名中使用定界标识符

可以使用关键字。若关键字用在一个它也可能被解释为 SQL 关键字的上下文中，则必须将它指定为一个定界标识符。

然而，使用定界标识符，有可能创建一个违反这些命名规则的对象，以后使用该对象可能会导致错误。例如，若创建在名称中带有 + 或 - 号的列，并且后来在索引中使用了该列，则当您尝试重组该表时将遇到问题。有关定界标识符的信息，参见 *SQL Reference* 的 “SQL Identifiers” 一节。

---

## 如何在联合系统中保留区分大小写的值

对于分布式请求，有时需要指定在数据源区分大小写的标识符和密码。为了确保将它们传送到数据源时大小写正确，请遵循下列准则：

- 用要求的大小写指定标识符和密码并用双引号引起来。



- 若指定的是用户标识，应将数据源的 *fold\_id* 服务器选项设置为 "n"（否，不更改大小写）。若指定的是密码，应将数据源的 *fold\_pw* 服务器选项设置为 "n"。

用户标识和密码有替代项。若数据源要求用户标识为小写，可以任何大小写指定它，并可将 *fold\_id* 服务器选项设置为 "l"（以小写形式将此标识发送至数据源）。若数据源要求此标识为大写，可以任何大小写指定它，并将 *fold\_id* 服务器选项设置为 "u"（以大写形式将此标识发送至数据源）。同样，若数据源要求密码为小写或大写，可将 *fold\_pw* 服务器选项设置为 "l" 或 "u" 来满足此要求。

有关服务器选项的更多信息，参见 第134页的『使用服务器选项来帮助定义数据源并简化认证处理』中的“使用服务器选项来帮助定义数据源并简化认证处理”。

- 若在操作系统命令提示符处用双引号括起区分大小写的标识符或密码，则必须确保系统可对双引号进行正确的语法分析。为此：
  - 在基于 UNIX 的操作系统上，用单引号括起该语句。
  - 在 Windows NT 操作系统上，在每个引号前加一个反斜杠。

例如，DB2 数据源中的许多定界标识符是区分大小写的。假设您希望为驻留在数据源 NORBASE 中的“DB2 CS 版”视图 "my\_schema"."wkly\_sal" 创建别名 NICK1。

在基于 UNIX 的系统的命令提示符处，应输入：

```
db2 'create nickname nick1 for norbase."my_schema"."wkly_sal"
```

在 Windows NT 命令提示符处，应输入：

```
db2 create nickname nick1 for norbase.\"my_schema\".\"wkly_sal\"
```

若从 DB2 命令提示符处输入语句（交互方式），或者若在应用程序中指定它，则不需要单引号或反斜杠。例如，在基于 UNIX 的系统或 Windows NT 上的 DB2 命令提示符处，应输入：

```
create nickname nick1 for norbase."my_schema"."wkly_sal"
```



---

## 附录B. 使用分布式计算环境 (DCE) 的目录服务

DCE 提供“单元目录服务”(CDS)和“全局目录服务”(GDS)。有关 DCE 概念和这些服务的更多信息,参考 *Introduction to OSF DCE* 手册。“DCE 目录服务”的 DB2 功能仅支持 CDS。通过使用此支持,用户就不必在每个单独的客户机上都创建一个数据库、节点和 DCS 数据库。所有这些信息都集中在 DCE CDS 中。

下列几节描述如何使用“DCE 目录服务”来设置和访问数据库:

- 创建目录对象
- 每个对象类的属性
- 目录服务安全性
- 配置参数和注册表变量
- CATALOG 和 ATTACH 命令以及 CONNECT 语句
- 客户机如何与数据库连接
- 如何搜索目录
- 临时替换 DCE 目录信息
- 目录服务任务
- 目录服务的限制

并非所有 DB2 客户机都能支持 DCE 目录服务。若某一个 DB2 客户机支持 DCE 目录服务,则您的《快速入门》手册会提供附加的信息。

---

### 创建目录对象

数据库管理员需要创建的目录对象有三种类型:

- 第284页的『数据库对象』
- 第285页的『数据库定位器对象』
- 第286页的『路由信息对象』

每个对象都包含属性。有关这些属性的完整描述,参考第288页的『每个对象类的属性』。

DCE 管理员需要将数据库信息添加至 CDS 表并授予数据库管理员创建特权,该数据库管理员才可以创建对象。有关详细信息,参考第303页的『DCE 管理员任务』。

## 数据库对象

数据库对象对于每个目标数据库是必需的。该对象的名称包含并置在一起的单元名、目录名和数据库名，例如：

```
./.../cell_name/dir_name1/dir_name2/OBJ_NAME
```

**注：**以下是对数据库名称的建议。该名称应不超过 8 个字符，且所有字符都应大写。若该名称为大小写混合或超过 8 个字符，则您需要使用 CATALOG GLOBAL DATABASE 命令来指定一个别名。有关该命令的详细信息，参见第 295 页的『CATALOG GLOBAL DATABASE 命令』。

以下是一个数据库对象的示例。存储在 DCE 目录中的对象包含其他信息，如时间戳记。每个属性左边的字母指示该属性是必需的 - R、可选的 - O 还是注解 - C。

```
Object name:                /.../CELL_TORONTO/subsys/database/AIXDB1
R DB_Object_Type:           D
C DB_Product_Name:          DB2_for_AIX
C DB_Product_Release:       V7R1M000
R DB_Native_Database_Name:  AIXDBASE
R DB_Database_Protocol:     DB2RA
R DB_Authentication:        CLIENT
O DB_Communication_Protocol:
O DB_Database_Locator_Name: /.../CELL_TORONTO/subsys/database/AIX_INST
C DB_Comment:                Test_database_on_AIX
```

若该数据库是与数据库管理器实例相关的多个数据库中的一个，则该数据库对象应该包含数据库定位器对象的名称且通信协议应是空白的。数据库定位器对象的名称是数据库管理器或 DB2 Connect 实例的全限定名。

以下是用于创建该对象的 DCE 命令的示例。在能够创建任何对象之前，DCE 管理员需要执行第 303 页的『DCE 管理员任务』中描述的步骤

首先，您必须在称为 *cdscp.inp* 的文件中输入下列内容：

```
create object ././subsys/database/AIXDB1

add object ././subsys/database/AIXDB1 DB_Object_Type           = D
add object ././subsys/database/AIXDB1 DB_Product_Name         = DB2_for_AIX
add object ././subsys/database/AIXDB1 DB_Product_Release      = V7R1M000
add object ././subsys/database/AIXDB1 DB_Native_Database_Name = AIXDBASE
add object ././subsys/database/AIXDB1 DB_Database_Protocol    = DB2RA
add object ././subsys/database/AIXDB1 DB_Authentication        = CLIENT
add object ././subsys/database/AIXDB1 DB_Database_Locator_Name = /...
/CELL_TORONTO/subsys/database/AIX_INST
add object ././subsys/database/AIXDB1 DB_Comment               = Test_database_on_AIX
```

然后您必须运行

- dcelogin principal password (在 OS/2 上)；或
- dce\_login principal password (在 UNIX、Windows 操作系统上)。

然后应运行

- `cdscp < cdscp.inp`

使用下列命令以显示该对象:

```
cdscp show object ././subsys/database/AIXDB1
```

若该数据库是与数据库管理器实例相关的唯一数据库, 则该数据库对象应包含“通信协议”属性的值, 而数据库定位器对象的名称应是空白的。例如:

```
Object name:                /.../CELL_TORONTO/subsys/database/MVSDB
R DB_Object type:           D
C DB_Product_Name:          DB2_for_MVS
C DB_Product_Release:       V7R1M00
R DB_Native_Database_Name:  MVSDBASE
R DB_Database_Protocol:     DRDA
R DB_Authentication:        SERVER
O DB_Communication_Protocol: APPC;NET1;TARGETLU1;DB2DRDA;MODE1;PROGRAM
O DB_Database_Locator_Name:
C DB_Comment:                Test_database_on_MVS
```

## 数据库定位器对象

这些对象包含有关数据库管理系统实例或 DB2 Connect 实例所使用的所有通信协议的详细信息。对于下列每一项, 必需一个数据库定位器对象:

- 同时具有数据库管理系统和 DB2 Connect 的每个实例
- 与多个数据库相关但没有相关的 DB2 Connect 的每个数据库管理系统实例
- 没有相关的数据库管理系统的每个 DB2 Connect 实例。

该对象的名称包含并置在一起的单元名、目录名和数据库实例的部分名称, 例如:

```
/.../cell_name/dir_name1/dir_name2/AIX_INST
```

**注:** 若该实例被用作 ATTACH 的目标, 则部分名称必须不超过 8 个字符且全部大写。

以下是数据库定位器对象的一个示例。存储在 DCE 目录中的对象包含其他信息, 如时间戳记。每个属性左边的字母指示该属性是必需的 - R、可选的 - O 还是注解 - C。

```
Object name:                /.../CELL_TORONTO/subsys/database/AIX_INST
R DB_Object_Type:           L
C DB_Product_Name:          DB2_for_AIX
C DB_Product_Release:       V7R1M00
R DB_Communication_Protocol: TCPIP;HOSTNAME1;1234
R DB_Communication_Protocol: APPC;NET1;TARGETLU1;TPN1;MODE;PROGRAM
C DB_Comment:                Test_instance_on_AIX
```

当在数据库对象和数据库定位器对象中都定义了某个属性时，使用数据库对象中的值。

以下是用于创建该对象的 DCE 命令的示例。在能够创建任何对象之前，DCE 管理员需要执行第303页的『DCE 管理员任务』中描述的步骤

首先，您必须在称为 *cdscp.inp* 的文件中输入下列内容：

```
create object ../subsys/database/AIX_INST

add object ../subsys/database/AIX_INST DB_Object_Type           = L
add object ../subsys/database/AIX_INST DB_Product_Name         = DB2_for_AIX
add object ../subsys/database/AIX_INST DB_Product_Release     = V7RIM00
add object ../subsys/database/AIX_INST DB_Communication_Protocol = TCP/IP;
HOSTNAME1;1234
add object ../subsys/database/AIX_INST DB_Communication_Protocol = APPC;NET1;
TARGETLU;TPN1;MODE;PROGRAM
add object ../subsys/database/AIX_INST DB_Comment              = Test_instance_on_AIX
```

然后您必须运行

- `dcelogin principal password` (在 OS/2 上)；或
- `dce_login principal password` (在 UNIX、Windows 操作系统上)。

然后应运行

- `cdscp < cdscp.inp`

使用下列命令以显示该对象：

```
cdscp show object ../subsys/database/AIX_INST
```

## 路由信息对象

对于主机访问，路由信息对象是必需的。当客户机所用的数据库协议与目标数据库所用的数据库协议不匹配时，路由对象告知客户机要使用哪个 DB2 Connect 实例。每个目标数据库都存在属性，这些属性包括可用的数据库协议和 DB2 Connect 实例的数据库定位器对象的名称。该对象的名称包含并置在一起的单元名、目录名和唯一的部分名称，例如：

```
../cell_name/dir_name1/dir_name2/ROUTE1
```

以下是路由信息对象的一个示例。存储在 DCE 目录中的对象包含其他信息，如时间戳记。每个属性左边的字母指示该属性和属性内的每个标记是必需的 - R、可选的 - O 还是注解 - C。

第 1 组客户机包括第297页的图8中的 Client\_1、Client\_2 和 Client\_3

```
对象名:          /.../CELL_TORONTO/subsys/database/ROUTE1
R DB_Object_Type: R
C DB_Comment:    Routing_for_client_group_1
```

```

R DB_Target_Database_Info
R Database name = ../../CELL_TORONTO/subsys/database/MVSDB
R Outbound protocol from router = DRDA
R Inbound protocol to router = DB2RA
R Authenticate at gateway = 1
O Parameter string = NOMAP,D,INTERRUPT_ENABLED
R DB_Database_Locator_Name = ../../CELL_TORONTO/subsys/database/GW_INST

R DB_Target_Database_Info
R Database name = *OTHERDBS
R Outbound protocol from router = DRDA
R Inbound protocol to router = DB2RA
R Authenticate at gateway = 0
O Parameter string =
R DB_Database_Locator_Name = ../../CELL_TORONTO/subsys/database/OTH_INST

```

数据库名 \*OTHERDBS 是一个特殊值，它标识用于访问未在路由信息对象中明确定义的任何目标数据库的公共路由器。

以下是用于创建该对象的 DCE 命令的示例。反斜杠 (\) 字符是一个延续字符。

在能够创建任何对象之前，DCE 管理员需要执行第303页的『DCE 管理员任务』中描述的步骤

首先，您必须在称为 *cdscp.inp* 的文件中输入下列内容：

```

create object ././subsys/database/ROUTE1

add object ././subsys/database/ROUTE1 DB_Object_Type = R
add object ././subsys/database/ROUTE1 DB_Comment = Routing_for_client_group_1
add object ././subsys/database/ROUTE1 DB_Target_Database_Info = \
    ../../CELL_TORONTO/subsys/database/MVSDB;\
drda;db2ra;1;NOMAP,D,INTERRUPT_ENABLED;\
    ../../CELL_TORONTO/subsys/database/GW_INST
add object ././subsys/database/ROUTE1 DB_Target_Database_Info = \
    *OTHERDBS;drda;db2ra;0;;\
    ../../CELL_TORONTO/subsys/database/OTH_INST

```

然后您必须运行

- dcelogin principal password (在 OS/2 上)；或
- dce\_login principal password (在 UNIX、Windows 操作系统上)。

然后应运行

- `cdscp < cdscp.inp`

使用下列命令以显示该对象：

```
cdscp show object ././subsys/database/ROUTE1
```

有关 DCE 命令的更多信息，参考下列 OSF DCE 出版物：

- *OSF DCE Administration Guide*
- *OSF DCE Administration Reference*

## 每个对象类的属性

在 DCE 环境中，每个对象和对象属性由一个对象标识 (OID) 标识。每个 OID 是从分配权限的层次结构中获得的，其中最高权限为“国际标准化组织” (ISO)。

表19显示每个对象类的属性，而表20显示它们的属性。

表 19. 对象属性类

对象类	对象标识 (OID)	必需的属性	可选的属性
(DB) Database_Object	1.3.18.0.2.6.12	DAU, DOT, DDP, DNN	DCO, DPN, DRL, DLN, DCP, DPR
(DL) Database_Locator_Object	1.3.18.0.2.6.13	DOT, DCP	DCO, DPN, DRL
(RI) Routing_Information_Object	1.3.18.0.2.6.14	DOT, DTI	DCO, DPN, DRL

表 20. 对象类属性

属性名	OID	最小长度	最大长度	语法
(DAU) DB_Authentication	1.3.18.0.2.4.39	1	1024	Char
(DCO) DB_Comment	1.3.18.0.2.4.30	1	1024	Char
(DCP) DB_Communication_Protocol	1.3.18.0.2.4.31	1	1024	Char
(DDP) DB_Database_Protocol	1.3.18.0.2.4.32	1	1024	Char
(DLN) DB_Database_Locator_Name	1.3.18.0.2.4.33	1	1024	Char
(DNN) DB_Native_Database_Name	1.3.18.0.2.4.34	1	1024	Char
(DOT) DB_Object_Type	1.3.18.0.2.4.35	1	1	Char
(DPN) DB_Product_Name	1.3.18.0.2.4.36	1	1024	Char
(DRL) DB_Product_Release	1.3.18.0.2.4.37	1	1024	Char
(DTI) DB_Target_Database_Info	1.3.18.0.2.4.38	1	1024	Char
(DPR) DB_Principal	1.3.18.0.2.4.63	1	1024	Char

注：DCP、DDP 和 DTI 允许多个值。其他属性只允许一个值。



## 有关每个属性的详细信息

下面一节描述每个属性。

**注：**“DCE 目录服务”不检查这些项目是否对于 DB2 有效。确保输入必需的属性，并确保输入正确的值。

### **DB\_Authentication (DAU)**

对象必需的认证方法。对于 DB2 服务器的数据库对象，此属性是必需的。该值必须为 CLIENT、SERVER 或 DCE。

### **DB\_Principal (DPR)**

若认证方法为 “DCE”，则在此属性中输入 DCE 所有者。

### **DB\_Comment (DCO)**

仅用于编写文档。

### **DB\_Communication\_Protocol (DCP)**

一个多值属性，其中的每个值由标记组成，这些标记描述受支持的网络协议。网络协议的示例有 TCP/IP、APPC、IPX/SPX 和 NetBIOS。标记之间由分号隔开。不要在标记之间加入空格。

• TCP/IP 的标记是：

1. tcpip
2. 目标节点的主机名
3. 该对象用于监听入局 TCP/IP 连接请求的端口号
4. （可选的）安全性可以是 NONE 或 SOCKS。

例如： tcpip;HOSTNAME;1234

• APPC 的标记是：

1. appc
2. 对象所属的目标的网络标识。
3. 可找到该目标的 LU 名。
4. 表示 LU 中的对象的“事务程序名” (TPN)（对于 DB2 MVS/ESA 版，使用 DB2DRDA 作为 TPN）。
5. 方式名
6. 目标使用的安全性类型。值是：
  - 无
  - PROGRAM
  - SAME

例如： appc;NETID;TARGETLU;TPNAME;MODE;PROGRAM

**注:** 对于 APPC, 客户机必须使用它的本地控制点 (CP) 作为其 LU 名。

- (限于 OS/2 和受支持的 Windows 操作系统) IPX/SPX 的记号是:

1. ipxspx
2. 文件服务器的名称
3. 对象的名称

例如: ipxspx;SVR\_NAME;OBJ\_NAME

- (限于 OS/2 和受支持的 Windows 操作系统) NetBIOS 的记号是:

1. netbios
2. 服务器的节点名

例如: netbios;SVR\_NNME, 其中客户机适配器号可在注册表值 *db2clientadpt* 或数据库管理器配置参数 *dft\_client\_adpt* 中找到

- (仅适用于受支持的 Windows 操作系统) NPIPE 的记号是:

1. NPIPE
2. 服务器的计算机名
3. 服务器的实例名

例如: npipe;computername;instance

### **DB\_Database\_Protocol (DDP)**

目标数据库支持的一个或多个数据库协议。这些值的示例有 DB2RA 和 DRDA。以下是添加两个协议的 *cdscp* 命令

```
add object ././subsys/database/AIXDB1 DB_Database_Protocol db2ra
add object ././subsys/database/AIXDB1 DB_Database_Protocol drda
```

### **DB\_Database\_Locator\_Name (DLN)**

数据库定位器对象的 DCE 名称。在数据库对象中, 该名称是用于 DBMS 实例的。在路由信息对象中, 该名称是用于 DB2 Connection 实例的。

例如, /.../CELL\_TORONTO/subsys/database/AIX\_INST

### **DB\_Native\_Database\_Name (DNN)**

在包含该数据库的实例内识别该数据库所用的数据库名或别名。它是在该实例中的本地应用程序用于与该数据库连接的名称。

对于 DB2 通用数据库, 该名称最多为 8 个字符。对于其他数据库, 该名称的长度可能会不同。例如, 对于 DB2 MVS/ESA 版上的数据库, 该名称可以多达 18 个字符。

## DB\_Object\_Type (DOT)

对象的类型。对于所有对象，此属性是必需的，且它可以是下列其中一个：

- D 数据库对象
- L 数据库定位器对象
- R 路由信息对象

## DB\_Product\_Name (DPN)

产品的标识。仅用于编写文档。

## DB\_Product\_Release (DRL)

产品发行版级别。仅用于编写文档。

## DB\_Target\_Database\_Info (DTI)

一个多值属性，其中的每个值由固定数目的标记组成，这些标记之间由分号隔开。不要在标记之间加入空格。这些标记必须按下列次序排列：

1. 数据库名。需要提供路由服务的目标数据库的 DCE 名。值 \*OTHERDBS 为未在路由信息对象中明确定义的任何目标数据库指定一个缺省网关。
2. 自路由器的出站协议。目标数据库所用的数据库协议，或路由 DB2 Connect 实例用于与该目标数据库通信的数据库协议。例如，DRDA。
3. 至路由器的入站协议。路由 DB2 Connect 实例对象接受的数据库协议。例如，DB2RA。
4. 在网关处的认证。有效值是 0 或 1。有关更多详细信息，参见第292页的表21。
5. 包含有关 DB2 Connect 网关的特定信息的参数字符串。该字符串包含必须按以下描述的次序排列的标记。这些标记由逗号隔开。对于未指定的标记，使用缺省值。
  - 映像文件名。SQLCODE 映射文件的全限定名，该文件覆盖缺省 SQLCODE 映射。要关闭 SQLCODE 映射，指定 NOMAP。
  - D. 当返回特定的 SQLCODE 时，该应用程序与 DRDA 服务器数据库断开。有关 SQLCODE 的详细信息，参考《DB2 Connect 用户指南》。
  - INTERRUPT\_ENABLED。若一个客户机在与 DRDA 服务器连接时发出中断，DB2 Connect 将断开该连接并回滚该工作单元。

下面是一些示例：

```
NOMAP
/u/username/sqllib/map/dcs1new.map,D
/u/username/sqllib/map/dcs1new.map,D,INTERRUPT_ENABLED
```

在此例中使用了缺省值，使用逗号来保留这些标记的次序，例如：

,D

和 / 或

,,INTERRUPT\_ENABLED

有关参数字符串的详细信息，参考《DB2 Connect 用户指南》。

#### 6. 提供路由服务的 DB2 Connect 实例的 DCE 名。

以下是 DB\_Target\_Database\_Info 的一个示例：

```
./.../CELL_TORONTO/subsys/database/MVSDB;\
drda;db2ra;0;;\
./.../CELL_TORONTO/subsys/database/GW_INST
```

注：在以上的示例中，反斜杠 (\) 是行延续字符。

---

## 目录服务安全性

当在不具有 DB2 Connect 网关的环境中使用 DCE 目录服务时，认证的使用与在访问数据库服务器的其他客户机中的使用是相同的。有关更多信息，参见第196页的『为服务器选择认证方法』。

当在具有 DB2 Connect 网关的环境中使用 DCE 目录服务时，DB2 Connect 管理员确定在何处验证用户名和密码。使用 DCE 目录，指定下列各项：

- 表示 DB2 Connect 工作站的数据库定位器对象中通信协议的安全性类型。（若远程客户机通过 APPC 连接与“DB2 Connect 扩充版”网关连接，则在该网关的 DCE 定位器对象中指定安全性类型 NONE。）
- 数据库对象中的认证类型。
- 数据库对象（或其相关的定位器对象）中的通信协议的安全性类型。
- 路由信息对象中网关记号处的认证。

表21显示这些值的可能组合，并对使用 APPC 连接的每个组合执行验证此表中显示的组合受具有“DCE 目录服务”的 DB2 Connect 支持。

表 21. 使用 APPC 连接的 DCE 的有效安全性方案

实例	服务器的数据库对象		路由对象	验证
	认证	安全性	网关处的认证	
1	CLIENT	SAME	0	远程客户机（或 DB2 Connect 工作站）
2	CLIENT	SAME	1	DB2 Connect 工作站

表 21. 使用 APPC 连接的 DCE 的有效安全性方案 (续)

实例	服务器的数据库对象		路由对象	验证
	认证	安全性	网关处的认证	
3	SERVER	PROGRAM	0	DRDA 服务器
4	SERVER	PROGRAM	1	DB2 Connect 工作站和 DRDA 服务器
5	DCE	NONE	不适用	DCE

表22显示这些值的可能组合，并对使用 TCP/IP 连接的每个组合执行验证此表中显示的组合受具有“DCE 目录服务”的 DB2 Connect 支持。

表 22. 使用 TCP/IP 连接的 DCE 的有效安全性方案

实例	认证	网关处的认证	验证
1	CLIENT	0	客户机
2	CLIENT	1	DB2 Connect 工作站
3	SERVER	0	DRDA 服务器
4	不适用	不适用	无
5	DCE	不适用	DCE

每个组合都适用于 APPC 和 TCP/IP，且下面将更详细地描述这些组合：

- 只在远程客户机上验证用户名和密码。（对于本地客户机，只在 DB2 Connect 工作站上验证用户名和密码。）  
期望在用户第一次注册至的位置认证他或她。用户标识是跨网络发送的，而密码不是这样。仅当所有客户机工作站都有足够的安全性设施时，才使用此类安全性。
- 只在 DB2 Connect 工作站上验证用户名和密码。通过网络，将密码从远程客户机发送至 DB2 Connect 工作站，而不发送至 DRDA 服务器。
- 只在 DRDA 服务器上验证用户名和密码。通过网络，将密码从远程客户机发送至 DB2 Connect 工作站，再从 DB2 Connection 工作站发送至 DRDA 服务器。
- 在 DB2 Connect 工作站和 DRDA 服务器上验证用户名和密码。通过网络，将密码从远程客户机发送至 DB2 Connect 工作站，再从 DB2 Connection 工作站发送至 DRDA 服务器。  
因为在这两处都执行验证，所以在 DB2 Connect 工作站和 DRDA 服务器上必须维护相同的一组用户名和密码。
- 从“DCE 安全性服务器”获取 DCE 标记。

注:

1. 对于基于 AIX 的系统, 使用安全性类型 SAME 的所有用户必须属于 AIX 系统组。
2. 对于具有远程客户机的基于 AIX 的系统, 在 DB2 Connect 工作站上运行的 DB2 Connect 产品的实例必须属于 AIX 系统组。
3. 对 DRDA 服务器的访问受它自己的安全性机制或子系统控制; 例如, “虚拟远程通信访问方法” (VTAM) 和 “资源访问控制设施” (RACF)。对受保护的数据库对象的访问受 SQL **GRANT** 和 **REVOKE** 语句控制。

---

## 配置参数和注册表变量

下列配置参数在 DCE 目录中使用。显示了这些值的示例。有关详细信息, 参考《管理指南: 性能》中 “配置 DB2” 一章中的 “分布式服务”。

- *dir\_obj\_name* 是与 *dir\_path\_name* 并置在一起的数据库实例名。若该实例名被用作 ATTACH 命令的目标, 则该名称必须不超过 8 个字符且全部大写, 例如:

AIX\_INST

- *dir\_type* 标识是否要使用 DCE 目录服务 要启用 DCE 目录服务, 必须将此参数设置为:

DCE

注意, 要将 *dir\_type* 设置为 NONE, 且不能在不支持 DCE 目录服务的使用的数据库客户机上更新它。

- *dir\_path\_name* 是 DCE 管理员提供的目录路径名, 例如:

././subsys/database/

- *route\_obj\_name* 是提供路由信息对象的 DCE 目录服务名称的可选参数。该名称可以是全限定的, 例如:

././subsys/database/ROUTE1

或者是将与 *dir\_path\_name* 并置在一起的部分名称, 例如:

ROUTE1

- *dft\_client\_comm* 是指定客户机使用的通信协议的可选 DCE 参数, 例如:

TCPIP

此参数也可以指定多个协议, 例如:

TCPIP, APPC (在基于 UNIX 的平台上)

TCPIP, APPC, IPXSPX, NETBIOS (在 OS/2 平台上)

TCPIP, APPC, IPXSPX, NETBIOS, NPIPE (在受支持的 Windows 操作系统上)

- *dft\_client\_adpt* 是可选的 DCE 参数，它指定 OS/2 和受支持的 Windows 操作系统上的 NetBIOS 协议的缺省客户机适配器号。该编号的有效范围是 0 至 15。若此参数包含非数值值，则该值会缺省为 0。若此参数包含允许范围之外的值，则该值会缺省为 0。

对于下列参数，注册表变量可以覆盖参数值。

配置参数	注册表变量
<i>dir_path_name</i>	DB2DIRPATHNAME
<i>route_obj_name</i>	DB2ROUTE
<i>dft_client_comm</i>	DB2CLIENTCOMM
<i>dft_client_adpt</i>	DB2CLIENTADPT

设置这些注册表变量的规则与它们对应的配置参数的规则相同。例如，同 *dft\_client\_comm* 参数一样，DB2CLIENTCOMM 是一个可以有多个值的字符串，值之间由逗号隔开，如：

```
db2set DB2CLIENTCOMM=TCPIP,APPC
```

## CATALOG 和 ATTACH 命令以及 CONNECT 语句

DCE 信息需要在下列命令中指定：

- CATALOG GLOBAL DATABASE 命令
- CONNECT 语句
- ATTACH 命令

### CATALOG GLOBAL DATABASE 命令

当客户机和服务器有不同的路径名时，或当数据库名包含 8 个以上的字符或大小写混合的字符时，使用 CATALOG GLOBAL DATABASE 命令。数据库管理委员会输入该数据库的 DCE 名和目录类型 DCE。

例如：

- 例如，当路径名不同时，若 *dir\_path\_name* = *...*  

```

/.../CELL_TORONTO/subsys/database/:
CATALOG GLOBAL DATABASE
/.../CELL_VANCOUVER/subsys/database/VMDB AS VANVMD
USING DIRECTORY DCE WITH "comment-string"

```
- 当数据库名包含 8 个以上的字符时，如名称 DB\_LONGNAME:

```
CATALOG GLOBAL DATABASE
/.../CELL_VANCOUVER/subsys/database/DB_LONGNAME AS VANVMD
USING DIRECTORY DCE WITH "comment-string"
```

## CONNECT 语句

要检索适当的 DCE 目录服务对象，客户机必须知道该数据库或 DBMS 实例的全限定的 DCE 名。下面是在 CONNECT 语句中指定名称的一些方法。

- 输入别名，例如：

```
CONNECT TO VANVMD
```

- 输入部分名称，例如：

```
CONNECT TO VMD
```

在这种情况下，在客户机上指定的路径名必须与在服务器上指定的路径名相同。（该路径名由 *dir\_path\_name* 配置参数或对应的注册表值指定。）

## ATTACH 命令

客户机的有效路径名必须与目标 DBMS 实例的路径名相同。

若 *dir\_path\_name* 对于客户机和服务器是相同的（例如，*/.../CELL\_TORONTO/subsys/database/*）且数据库服务器上的 *dir\_obj\_name* 是 AIX\_INST，则连接该实例的命令为：

```
ATTACH TO AIX_INST
```

---

## 客户机如何与数据库连接

第297页的图8显示具有两个 DCE 单元的数据库网络的样本配置 */.../CELL\_TORONTO* 和 */.../CELL\_VANCOUVER* 是单元名。（这些单元中的每一个都包含一个称为 *./subsys/database/* 的目录，且当该图表中未列举时，则在其他示例中使用。）



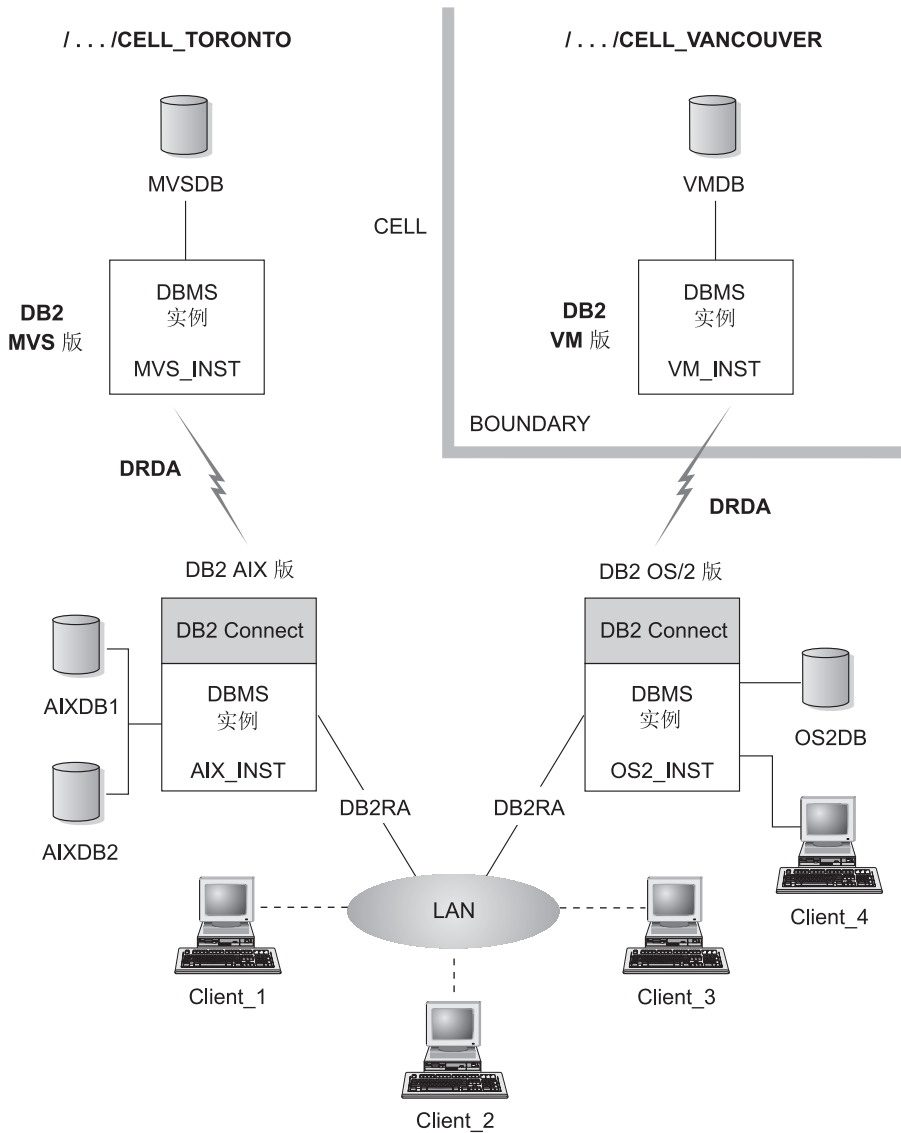


图 8. 一个网络数据库的配置

要允许 TORONTO 单元中的客户机访问两个单元中的所有数据库，必须在数据库管理器配置参数中指定值，且必须创建下列对象：

- 每个数据库的数据库对象。
- “DB2 AIX 版”和“DB2 OS/2 版”的两个数据库服务器上的目标数据库的数据库定位器对象。

- 所有客户机都认识的单个路由信息对象。这些属性指定哪个 DB2 Connect 节点要用于 MVSDB 和 VMDB 数据库。

下面提供一个客户机如何与一个数据库连接的示例:

- 与相同单元中的数据库连接
- 与不同单元中的数据库连接.

这些示例包括必须指定的数据库管理器配置参数。

## 与相同单元中的数据库连接

本节描述客户机如何与相同单元中的数据库连接的几个示例。

1. Client\_1 与 AIXDB2 连接。该数据库与客户机共享相同的目录路径名。

数据库管理员需要:

- 在配置参数 *dir\_path\_name* (或 DB2DIRPATHNAME 注册表值) 中指定目录路径名的值
- 在配置参数 *dir\_type* 中将目录服务类型值指定为 DCE。
- 在配置参数 *dft\_client\_comm* (或 DB2CLIENTCOMM 注册表值) 中指定通信协议。

本地系统数据库目录不包含 AIXDB2, 所以要使用全限定名来搜索 DCE 目录。该名称是通过将配置参数 *dir\_path\_name* 的值 (或 DB2DIRPATHNAME 注册表值) 与 AIXDB2 并置在一起创建的

事件的顺序是:

- a. Client\_1 使用该数据库的 DCE 名 `../../CELL_TORONTO/subsys/database/AIXDB2` 来获取 AIXDB2 的数据库对象。
- b. 从此对象中, Client\_1 获知 AIXDB2 使用 DB 协议 DB2RA, 与 Client\_1 使用的协议相同。
- c. DB 协议匹配, 所以 Client\_1 读取 AIX\_INST 的数据库管理系统定位器对象, 检索与它使用的协议匹配的通信协议属性值, 并使用该信息来启动与该数据库管理系统实例的对话。

2. Client\_3 与 MVSDB 连接。该数据库与客户机共享相同的目录路径名, 并使用与该客户机不同的数据库协议。

数据库管理员需要:

- 在配置参数 *dir\_path\_name* (或 DB2DIRPATHNAME 注册表值) 中指定目录路径名的值
- 在配置参数 *dir\_type* 中将目录服务类型值指定为 DCE。

- 在配置参数 *dft\_client\_comm*（或 DB2CLIENTCOMM 注册表值）中指定通信协议。
- 在配置参数 *route\_obj\_name*（或 DB2ROUTE 注册表值）中指定缺省路由信息对象的 DCE 名。

事件的顺序是:

- a. Client\_3 使用数据库的 DCE 名 `../../CELL_TORONTO/subsys/database/MVSDB` 来获取 MVSDB 的数据库对象。
- b. 从此对象中, Client\_3 发现 MVSDB 只使用 DB 协议 DRDA, 它不是 Client\_3 使用的那种协议。
- c. 然后, Client\_3 使用在 *route\_obj\_name* 配置参数或 DB2ROUTE 注册表值中定义的名称来获取路由信息对象。这样, 该客户机可查找 MVSDB 的目标数据库信息。
- d. Client\_3 读取与 MVSDB 目标数据库信息相关的数据库定位器对象, 检索通信协议, 并向路由器发送 SQL CONNECT 请求。
- e. 然后路由器建立与 MVSDB 的 APPC 连接。

## 与不同单元中的数据库连接

本节描述当数据库协议不同时, 客户机如何与不同单元中的数据库连接的示例。

1. Client\_3 先前已配置为使用下列功能:

- DCE 目录服务, 通过为 *dir\_type* 参数指定 DCE。
- 非 CELL\_VANCOUVER 的单元, 通过配置参数 *dir\_path\_name*, 例如:

```
../../CELL_TORONTO/subsys/database/
```

2. 为使 Client\_3 与 VMDB 连接, 数据库管理员需要:

- 在本地系统数据库目录中明确编目 VMDB。将 VMDB 的 DCE 名与本地唯一的数据库别名关联, 并发出带该别名值的 CONNECT 语句。例如:

```
CATALOG GLOBAL DATABASE
../../CELL_VANCOUVER/subsys/database/VMDB AS VANVMDB
USING DIRECTORY DCE WITH "comment-string"
```

后跟:

```
CONNECT TO VANVMDB
```

- 在配置参数 *dft\_client\_comm*（或 DB2CLIENTCOMM 注册表值）中指定通信协议。
- 在配置参数 *route\_obj\_name*（或 DB2ROUTE 注册表值）中指定缺省路由信息对象的 DCE 名。

事件的顺序是:

- a. Client\_3 在其系统数据库目录中查找 VANVMDB 的全限定 DCE 名。
- b. Client\_3 使用数据库的 DCE 名 `../CELL_VANCOUVER/subsys/database/VMDB` 来获取 VMDB 的数据库对象。
- c. 从此对象中, Client\_3 发现 VMDB 只使用 DB 协议 DRDA, 它不是 Client\_3 使用的那种协议。
- d. 然后, Client\_3 使用在 `route_obj_name` 配置参数或 DB2ROUTE 注册表值中定义的名称来获取路由信息对象。这样, 该客户机可查找 VMDB 的目标数据库信息。
- e. Client\_3 读取与 VMDB 目标数据库信息相关的数据库定位器对象, 检索通信协议, 并向路由器发送 SQL CONNECT 请求。
- f. 然后路由器建立与 VMDB 的 APPC 连接。

---

## 如何搜索目录

若在所有目标数据库共享相同的目录路径名的环境中使用 DCE 目录, 则客户机上不需要任何本地目录。

本节描述为下列目的搜索目录的次序:

- ATTACH 命令
- CONNECT 语句

### ATTACH 命令

第301页的图9显示当一个客户机与名为 ABC\_INST 的数据库管理系统实例连接时如何搜索目录

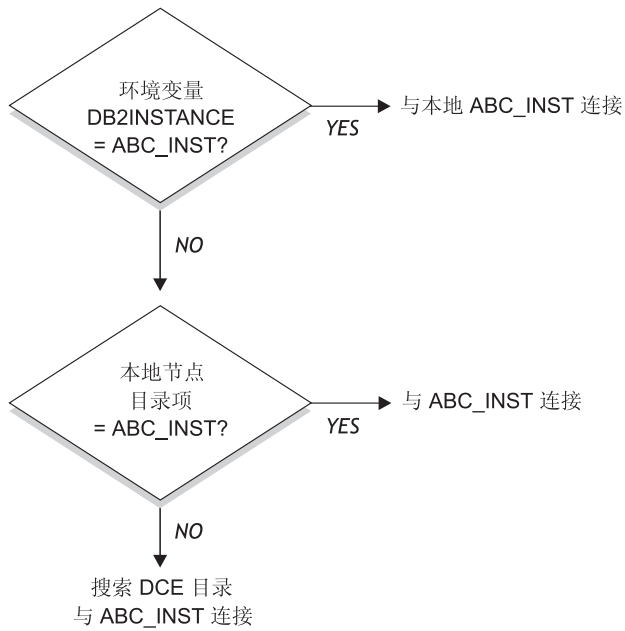


图 9. 如何使用目录来连接数据库

## CONNECT 语句

第302页的图10显示当一个客户机与称为 DBTEST 的数据库连接时如何搜索目录

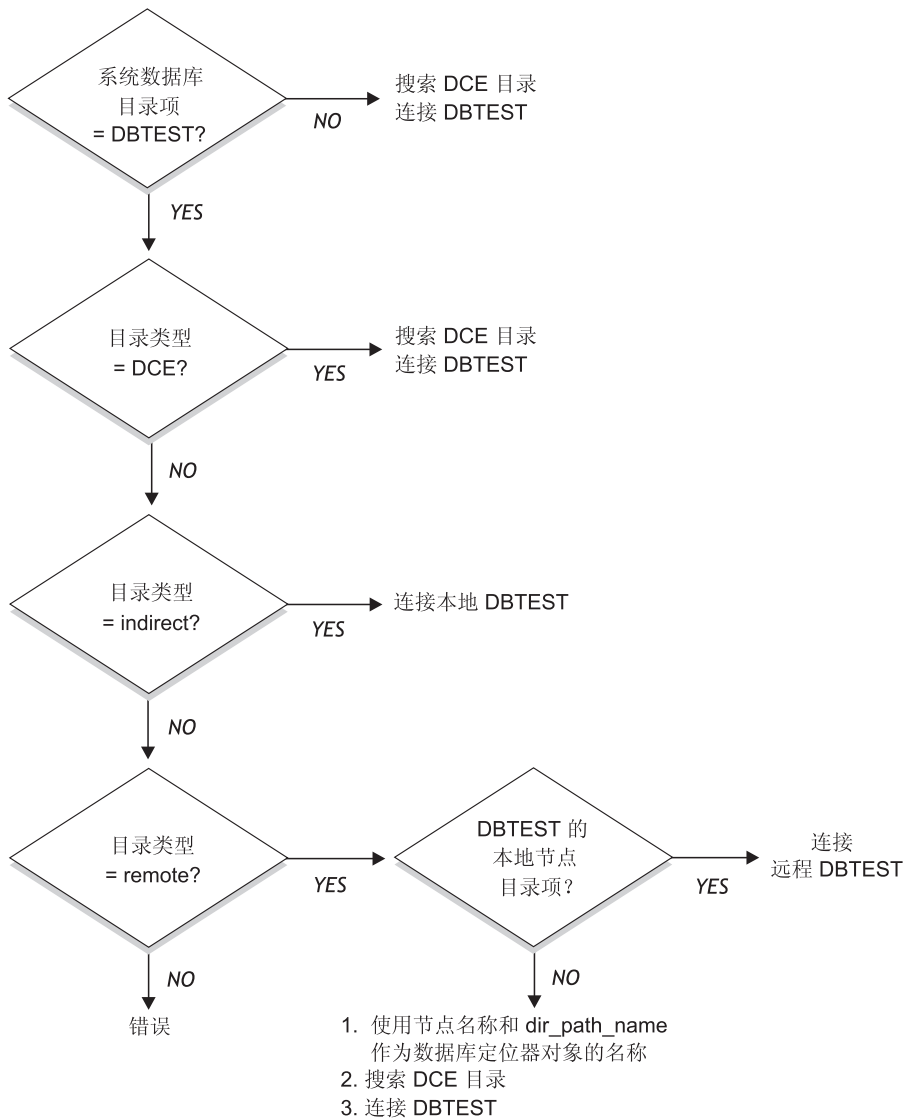


图 10. 如何使用目录来连接数据库

---

## 临时替换 DCE 目录信息

您可以使用本地数据库目录来覆盖 DCE 目录信息。例如，若您 CONNECT TO DBTEST，其中 `./:/subsys/database/DBTEST` 当驻留在称为 JAGUAR 的主机上时是在 DCE 目录中定义的，您可以临时将 DBTEST 更改为驻留在称为 STORM 的主机上的另一个数据库。将 DBTEST 在本地编目为一个远程数据库，并带有指向 STORM 的节点目录项。

您可以为其 DCE 名之前未加客户机的目录路径名的数据库创建别名。有关该命令的更多信息，参见第295页的『CATALOG GLOBAL DATABASE 命令』。

---

## 目录服务任务

下面列示了设置和使用“DCE 目录服务”必须执行的任务。下面几节描述每个任务的详细信息。

- DCE 管理员任务  
DCE 管理员必须更新 DCE 目录，以便可以添加新的数据库资源信息。
- 数据库管理员任务  
数据库管理员必须更新 DCE 目录，并为 DB2 的安装和配置提供信息。
- 数据库用户任务  
数据库用户必须登录至 DCE，并且要知道目标数据库名。

此外，网络管理员要为每个用户节点设置网络访问权。有关详细信息，参考网络文档。

### DCE 管理员任务

在可以创建或读取目录对象之前，DCE 管理员必须执行下列任务：

- 为 DB2 指定目录子树，例如 `./:/subsys/database`
- 授予数据库管理员创建目录对象的特权。
- 授予数据库用户读取目录对象的特权。
- 将新 DCE 目录对象属性的信息添加至 DCE 属性表。

编辑 CDS 属性文件（在 UNIX 平台上为 `/etc/dce/cds_attributes`；在 OS/2 上为 `X:\opt\dcelocal\etc\cds_attr`，其中“X”是适当的驱动器），并追加下列内容：

```
1.3.18.0.2.4.30 DB_Comment char
1.3.18.0.2.4.31 DB_Communication_Protocol char
1.3.18.0.2.4.32 DB_Database_Protocol char
1.3.18.0.2.4.33 DB_Database_Locator_Name char
1.3.18.0.2.4.34 DB_Native_Database_Name char
1.3.18.0.2.4.35 DB_Object_Type char
1.3.18.0.2.4.36 DB_Product_Name char
```

1.3.18.0.2.4.37	DB_Product_Release	char
1.3.18.0.2.4.38	DB_Target_Database_Info	char
1.3.18.0.2.4.39	DB_Authentication	char
1.3.18.0.2.4.63	DB_Principal	char

- 确保当用户需要使用“DCE 目录服务”来访问该数据库时，DCE 正在运行。

有关更多信息，参考您使用的平台的 DCE 文档。

## 数据库管理员任务

数据库管理员必须执行下列任务：

- 从 DCE 管理员处获取数据库资源的目录子树。例如，`././subsys/database`
- 在 DB2 数据库管理器的安装期间，请 DCE 管理员添加 DB2 所需的新 DCE 目录对象属性。
- 在 DCE 目录子树中为每个数据库管理系统实例指定一个唯一名称。例如，`././subsys/database/AIX_INST`
- 对于每个数据库管理系统实例，指定 DCE 的数据库管理器配置参数。

- *dir\_type*
- *dir\_obj\_name*
- *dir\_path\_name*
- *route\_obj\_name*
- *dft\_client\_comm*
- *dft\_client\_adpt*

客户机设置的注册表变量可以临时覆盖某些配置参数。有关更多信息，参考第 294 页的『配置参数和注册表变量』。

- 在 DCE 目录子树中为每个数据库指定唯一的名称。在数据库配置文件的 *dir\_obj\_name* 参数中指定该名称
- 使用 DCE *cdscp* 命令来创建和显示对象，以创建“DCE 目录服务”的对象。分别从数据库管理器的安装过程和数据库管理器的实例启动过程创建对象。

存在三种类型的对象。

- 数据库对象对于每个目标数据库是必需的。
- 数据库定位器对象对于与多个数据库相关的每个 DB2 Connect 实例和每个数据库管理系统实例（没有 DB2 Connect）是必需的。
- 路由信息对象是访问主机数据库所必需的。
- 根据每个环境，数据库管理员必须确定：
  - 在考虑客户机访问什么数据库和使用什么通信协议的前提下，如何将客户机分为逻辑组。



- 需要多少个路由信息对象。
- 哪些目标数据库应该记录在每个路由信息对象中。
- 哪组客户机应认识哪些路由信息对象。

有关这些对象的详细信息，参考第283页的『创建目录对象』。

## 数据库用户任务

数据库用户必须执行下列任务：

- 从数据库管理员处获取数据库的名称。此名称可以是简单的部分名称，或全限定 DCE 名。
- 若需要，在注册表变量中指定“DCE 目录服务”所需的值。客户机设置的注册表变量可以临时覆盖配置参数。
  - 若必需主机数据库访问，则从数据库管理员处获取路由信息对象的全限定 DCE 名。若未在 *route\_obj\_name* 中指定此名称，或它是另一个名称，则在尝试与主机数据库连接之前在 *DB2ROUTE* 注册表变量中指定此名称。
  - 若未在 *dft\_client\_comm* 中指定首选通信协议，或它是另一个协议，则在 *DB2CLIENTCOMM* 注册表变量中指定客户机的通信协议。以下是一些 UNIX 示例：

```
db2set DB2CLIENTCOMM=tcpip
db2set DB2CLIENTCOMM=appc
db2set DB2CLIENTCOMM=tcpip,appc
db2set DB2CLIENTCOMM=appc,tcpip
```

以下是一些 OS/2 示例：

```
db2set DB2CLIENTCOMM=ipxspx
db2set DB2CLIENTCOMM=netbios
db2set DB2CLIENTCOMM=tcpip,ipxspx,netbios
db2set DB2CLIENTCOMM=netbios,tcpip,ipxspx,appc
```

以下是一些 Windows 操作系统示例：

```
db2set DB2CLIENTCOMM=npipe
db2set DB2CLIENTCOMM=netbios
db2set DB2CLIENTCOMM=tcpip,ipxspx,netbios
db2set DB2CLIENTCOMM=netbios,tcpip,ipxspx,appc,npipe
```

若存在多个通信协议，则使用指定的第一个协议。

- 若任何一个数据库具有一个 DCE 名称，而该名称未在 *dir\_path\_name* 配置参数或 *DB2DIRPATHNAME* 注册表变量中定义的目录路径中，则使用 *CATALOG GLOBAL DATABASE* 命令显式编目该数据库。有关更多信息，参考第295页的『*CATALOG GLOBAL DATABASE* 命令』。

- 在与目标数据库连接或与数据库实例连接之前，登录至 DCE。有关登录命令的更多信息，参考 *OSF DCE Administration Guide*。

---

## 目录服务的限制

本节描述不支持的项目。

- 并非所有数据库客户机都可受支持。参见您的《快速入门》手册，以确定 DCE 目录服务是否受 DB2 客户机支持。当前，只是为所有 UNIX、OS/2 和受支持的 Windows 操作系统的“DB2 客户机”提供了支持。
- 客户机不能使用“DCE 目录服务”来与 DB2 OS/2 版的版本 1 服务器连接。
- 只有受支持的 Windows 操作系统客户机才可使用 TCP/IP、APPC、NetBIOS、IPX/SPX 或 NPIPE 中的任何一个或全部协议。只有 OS/2 客户机才可使用 TCP/IP、APPC、NetBIOS 和 IPX/SPX 中的任何一个或全部协议。所有受支持的 UNIX 客户机只能使用 TCP/IP 和 APPC 协议。
- LIST DATABASE (或 NODE) DIRECTORY COMMANDS 只提供本地目录的项目，而不提供 DCE 目录的项目。您可以在 DCE 中使用 *cdscp show object* 命令以显示对象。
- 当下列所有条件都存在时，数据库管理器实例的所有者必须在启动数据库管理器（使用 *db2start* 命令）之前登录至 DCE。
  - 通过 *dir\_type* 配置参数配置数据库管理器实例以支持 DCE 目录服务。
  - 单元目录服务对象只能通过明确地登录至 DCE 来读取
  - 必须访问 DCE 目录，以支持下列任何一项：
    - 位于另一个实例上的事务管理器数据库（由 *tm\_database* 配置参数指定）
    - 不能支持 DCE 目录服务或未配置成使用 DCE 目录服务的客户机。

**注：**当执行 DCE 登录时，您应该使用具有较长访问权限有效期的所有者。

- 当使用 DDCS 版本 2.2 (或更早的版本) 网关将使用 DCE 目录服务的客户机与 DRDA 服务器连接时，您必须在网关的本地目录中编目该数据库别名。此数据库别名必须于客户机上的别名相同，且它必须表示同一个数据库。
- 当使用受支持的 Windows 操作系统客户机时，将使用 DB2DCE.DLL。在 *sqliib* 子目录的 *bin* 子目录中可找到此文件。若 DCE 供应商是 Gradient\*\*, 缺省情况下，文件 DB2DCE.GRD 等效于 DB2DCE.DLL。若 DCE 供应商是 IBM，则必须将文件 DB2DCE.IBM 复制为 DB2DCE.DLL。

---

## 附录C. 数据库恢复的用户出口

此处提供的概述信息与“数据移动”一章中所提供的类似。

**注：**有关这些主题的全部信息以及*Command Reference*和 *Administrative API Reference* 中对应的主题，已全部合并到 *Data Recovery and High Availability Guide and Reference* 中。

*Data Recovery and High Availability Guide and Reference* 是有关这些主题的主要的、单一的信息源。



---

## 附录D. 向多个数据库分区发出命令

在分区数据库系统中，您可能想要发出将在实例中的机器或在多个数据库分区服务器（节点）上运行的命令。为此，您可以使用 **rah** 命令或 **db2\_all** 命令。**rah** 命令允许您发出将在实例中的机器上运行的命令。若想要命令在实例中的多个数据库分区服务器上运行，则运行 **db2\_all** 命令。本章节概述了这些命令。以下的信息仅适用于分区数据库系统。

注:

1. 在基于 UNIX 的平台上，您的登录界面可以是 Korn shell 或任何其他 shell；但是，不同 shell 处理包含特殊字符的命令所用的方式不同。
2. 在 Windows NT 上，要运行 **rah** 命令或 **db2\_all** 命令，您必须使用“管理员”组成员的用户帐户来登录。

要确定命令的范围，参考 *Command Reference*。本书指示命令是在单个数据库分区服务器上还是在所有数据库分区服务器上运行。若命令在一个数据库分区服务器上运行，而您想该命令在所有数据库分区服务器上运行，则使用 **db2\_all**。例外情况是 **db2trc** 命令，它在一台机器的所有逻辑节点（数据库分区服务器）上运行。若想要在所有机器的所有逻辑节点上运行 **db2trc**，则使用 **rah**。

---

### 命令

可以依次在各个数据库分区服务器上按顺序运行命令，或可以用并行的方式运行命令。在基于 UNIX 的平台上，若以并行方式运行这些命令，您可以选择将输出发送至缓冲区并收集该输出以便显示（缺省行为），或者可在发出该命令的机器上显示该输出。在 Windows NT 上，若以并行方式运行这些命令，则在发出该命令的机器上显示输出。

要使用 **rah** 命令，输入:

```
rah command
```

要使用 **db2\_all** 命令，输入:

```
db2_all command
```

有关 **rah** 语法的帮助，输入

```
rah "?"
```

该命令几乎可以是在交互式提示符下输入的任何内容，例如，要按顺序运行的多个命令。在基于 UNIX 的平台上，使用分号 (;) 将多个命令分开。在 Windows NT 上，使用 & 符号将多个命令分开。不要在最后一个命令之后使用分隔符。

以下示例显示如何使用 **db2\_all** 命令来更改在节点配置文件中指定的所有数据库分区上的数据库配置。因为 ; 字符在双引号之内，所以将同时运行请求：

```
db2_all ";UPDATE DB CFG FOR sample USING LOGFILSIZ=100"
```

## 命令描述

可以使用下列命令：

命令	描述
----	----

<b>rah</b>	在所有机器上运行该命令。
------------	--------------

<b>db2_all</b>	在指定的所有数据库分区服务器上运行该命令。
----------------	-----------------------

<b>db2_kill</b>	突然停止正在多个数据库服务器上运行的所有进程，并清除所有数据库分区服务器上的所有资源。此命令使数据库变得不一致。除非有 IBM 服务指导，否则不要发出此命令。
-----------------	---

<b>db2_call_stack</b>	
-----------------------	--

在基于 UNIX 的平台上，使在所有数据库分区服务器上运行的所有进程将调用追溯写入 syslog。

在 Windows NT 上，使在所有数据库分区服务器上运行的所有进程将调用追溯写入实例目录中的 *Pxxxx.nnn* 文件，其中 *Pxxxx* 是进程标识，而 *nnn* 是节点号。

在基于 UNIX 的平台上，这些命令执行带特定隐式设置的 **rah**，如：

- 以并行方式在所有机器上运行
- 将命令输出分别缓存到 `/tmp/$USER/db2_kill` 和 `/tmp/$USER/db2_call_stack` 中。

在 Windows NT 上，这些命令执行 **rah**，以便在所有机器上以并行方式运行。

## 指定要运行的命令

可以：

- 从命令行指定命令作为参数
- 为响应提示而指定命令（若不指定任何参数）。

若命令包含下列特殊字符则应该使用提示方法：

```
| & ; < > ( ) { } [ ] unsubstituted $
```

若在命令行上指定命令作为参数，而且若它包含刚才列出的任何特殊字符，必须以双引号将命令括起来。

**注：**在基于 UNIX 的平台上，将把该命令添加至您的命令历史中，就象您在提示符处输入它一样。

可以正常输入命令中的所有特殊字符（除 \ 外，不必以双引号括起来）。若需要在命令中包括 \，则必须输入两个反斜杠 (\)。

**注：**在基于 UNIX 的平台上，若您未使用 Korn shell，则可以正常地输入该命令中的所有特殊字符（除 "、\、不可替换的 \$ 和单引号 (') 外，其他字符都不需要用引号引起来）。若需要在命令中包括这些字符之一，则必须在字符前加三个反斜杠 (\)。例如，若需要在命令中包括一个 \，则必须输入四个反斜杠 (\\)。

若需要在命令中包括双引号 (")，则必须在双引号前加三个反斜杠，例如 \\"。

**注：**

1. 在基于 UNIX 的平台上，您不能在命令中包括单引号 (')，除非命令 shell 提供了在被括上单引号的字符串中输入单引号的某种方法。
2. 在 Windows NT 上，您不能在命令中包括单引号 (')，除非命令窗口提供了在被括上单引号的字符串中输入单引号的某种方法。

当运行任何 korn-shell shell-script，而该脚本包含从后台中的标准输入进行读取的逻辑时，您应该明确地将标准输入重定向到一个源，在此源上进程可以读取而不必在终端上停止 (SIGTTIN 消息)。要重定向标准输入，可以用下列格式运行脚本：

```
shell_script </dev/null &
```

若没有提供输入。

同样，当在后台运行 db2\_all 时，应该始终指定 </dev/null。例如：

```
db2_all ";run_this_command" </dev/null &
```

通过执行此操作，可以重定向标准输入，并避免在终端上停止。

当您不关心来自远程命令的输出时，此方法的一种替代方法是在 db2\_all 前缀中使用 "daemonize" 选项：

```
db2_all ";daemonize_this_command" &
```

## 在基于 UNIX 的平台上以并行方式运行命令

**注：**本节中的信息只适用于基于 UNIX 的平台。

缺省情况下，命令在每台机器上顺序运行，但通过在命令前加上某些前缀序列，可以指定使用后台 `rshell` 以并行方式运行命令。若 `rshell` 在后台运行，则每个命令将输出放在在其远程机器的缓冲文件中。此进程分两个部分检索输出：

1. 在远程命令完成后。
2. 在 `rshell` 终止后，若某些进程仍在运行，则 `rshell` 可能会过一段时间才终止。

缺省情况下，缓冲区文件的名称为 `/tmp/$USER/rahout`，但可以通过环境变量 `$RAHBUFDIR/$RAHBUFNAME` 指定该名称。

当指定想要命令同时运行时，缺省情况下，此脚本将附加命令作为前缀加到发送至所有主机的命令上，以检查 `$RAHBUFDIR` 和 `$RAHBUFNAME` 是否可用于缓冲文件。这会创建 `$RAHBUFDIR`。要避免这样，导出环境变量 `RAHCHECKBUF=no`。若知道目录存在并可用，则可以执行此操作以节省时间。

在使用 `rah` 同时在多台机器上运行命令之前：

- 确保对于您的用户标识，目录 `/tmp/$USER` 在每台机器上存在。要在目录尚不存在的情况下创建目录，运行：

```
rah ")mkdir /tmp/$USER"
```

- 将下一行添加至 `.kshrc`（对于 Korn shell 语法）或 `.profile`，并将它输入到当前的会话中：

```
export RAHCHECKBUF=no
```

- 确保您运行远程命令的每台机器的标识在其 `.rhosts` 文件中有一个项目对应于运行 `rah` 的标识；并且运行 `rah` 的标识在其 `.rhosts` 文件中有一个项目对应于运行远程命令的每台机器的标识。

## 在基于 UNIX 的平台上监视 `rah` 进程

**注：**本节中的信息只适用于基于 UNIX 的平台。

当任何远程命令仍在运行或仍在累加缓冲输出时，由 `rah` 启动的进程对活动进行监视以：

- 将消息写入终端，指示哪些命令尚未运行
- 检索缓冲输出。

在环境变量 `RAHWAITTIME` 控制的时间间隔写出信息消息。参考帮助信息以获取如何指定时间间隔的详细信息。通过导出 `RAHWAITTIME=0`，可以完全不显示所有信息消息。



主监视过程是一个命令，其命令名（由 `ps` 命令显示）为 **rahwaitfor**。第一条信息消息告诉您此进程的 `pid`（进程标识）。所有其他监视进程将表现为运行 **rah** 脚本（或符号链路的名称）的 **ksh** 命令。若愿意，可通过以下命令停止所有监视进程：

```
kill <pid>
```

其中 `<pid>` 是主监视进程的进程标识。不要指定信号号。让它为缺省值 15。这根本不会影响远程命令，但会阻止自动显示缓冲输出。注意，在执行单个 **rah** 时，可能有两组或更多组不同监视进程在不同时间执行。然而，若任何时候停止当前组监视进程，则不会再启动其他监视进程。

若正规登录界面不是 Korn shell（例如 `/bin/ksh`），则可以使用 **rah**，但如何输入包含下列特殊字符的命令的规则稍微有些差异：

```
" unsubstituted $ "
```

有关更多信息，输入 `rah "?"`。而且，在基于 UNIX 的环境中，若执行远程命令的标识的登录界面不是 Korn shell，则以该标识执行 **rah** 的登录界面也不得是 Korn shell。（**rah**根据本地标识来决定远程标识的 shell 是否为 Korn shell）。该 shell 不得对单引号内的字符串执行任何替换或特殊处理。必须照原样保持它。

## 附加 **rah**（运行所有主机）信息（仅适用于 Solaris 和 AIX）

为了增强性能，在大型系统上扩充了 **rah** 以使用 `tree_logic`。也就是说，**rah** 将检查该列表包含多少个节点，若该数目超过阈值，它会构造列表的一个子集，并将它自己的递归调用发送到那些节点。在那些节点上，递归调用的 **rah** 遵循相同的逻辑，直到该列表小得能够符合将该命令发送到列表中所有节点的标准逻辑（现在称为“树叶”逻辑）为止。该阈值可由环境变量 `RAHTREETHRESH` 指定，缺省值为 15。

对于每个物理节点存在多个逻辑节点的系统，`db2_all` 比较愿意将递归调用发送到各个不同的物理节点，然后 `rsh` 到同一个物理节点上的其他逻辑节点，这样可减少物理节点间的通信量。（这种方法只适合 `db2_all`，不适合 **rah**，因为 **rah** 始终只发送到不同的物理节点。）

---

## 前缀序列

前缀序列是一个或多个特殊字符。在命令字符前输入一个或多个前缀序列而不插入任何空格。若想指定多个序列，可以任何顺序输入它们，但任何多字符序列中的输入字符必须按顺序输入。若您输入任何前缀序列，则您必须将整个命令（包括该前缀序列）置于双引号内，如下例所示：

- 在基于 UNIX 的平台上：

```
rah "};ps -F pid,ppid,etime,args -u $USER"
```

- 在 Windows NT 上:

```
rah "||db2 get db cfg for sample"
```

前缀序列有:

顺序	目的
	在后台按顺序运行命令。
&	在后台按顺序运行这些命令，并在所有远程命令完成之后，终止该命令，即使有一些进程仍在运行。例如，若子进程（在基于 UNIX 的平台上）或后台进程（在 Windows NT 上）仍在运行，可能造成延迟。在此情况下，该命令启动独立的后台进程来检索命令终止之后产生的任何远程输出，并将该输出写回至源机器。  <b>注：</b> 在基于 UNIX 的平台上，指定 & 会降低性能，因为需要运行更多的 <b>rsh</b> 命令。
	在后台以并行方式运行命令。
&	在后台以并行方式运行命令并在所有远程命令完成之后终止命令，如以上  & 的情况所述。  <b>注：</b> 在基于 UNIX 的平台上，指定 & 会降低性能，因为需要运行更多的 <b>rsh</b> 命令。
;	与以上的   & 相同。这是一个较短的替代格式。  <b>注：</b> 在基于 UNIX 的平台上，指定 ; 会降低性能（相对于    而言），因为需要运行更多的 <b>rsh</b> 命令。
]	在执行命令之前预先暂挂用户简要表的点执行。  <b>注：</b> 仅在基于 UNIX 的平台上可用。
}	在执行命令之前预先暂挂在 \$RAHENV 中命名的文件的点执行（可能是 .kshrc）。  <b>注：</b> 仅在基于 UNIX 的平台上可用。
]]	在执行命令之前，预先暂挂用户简要表的点执行，然后执行在 \$RAHENV 中命名的文件（可能是 .kshrc）。  <b>注：</b> 仅在基于 UNIX 的平台上可用。
)	停止执行用户简要表和 \$RAHENV 中命名的文件。  <b>注：</b> 仅在基于 UNIX 的平台上可用。

- ' 将命令调用回送至机器。
- < 发送至除此机器外的所有机器。
- <<-nnn< 发送至除数据库分区服务器 *nnn* 外的所有数据库分区服务器 (db2nodes.cfg 中除节点号 *nnn* 外的所有数据库分区服务器, 参见本表中最后一个前缀序列后的第一段)。
- <<+nnn< 仅发送至数据库分区服务器 *nnn* (db2nodes.cfg 中节点号为 *nnn* 的数据库分区服务器, 参阅本表中最后一个前缀序列后的第一段)。
- (空格字符) 在后台运行远程命令, stdin、stdout 和 stderr 全部关闭。此选项仅当在后台运行命令时才有效, 即仅在还包括 \ 或 ; 的前缀序列中有效。它允许命令尽快完成 (远程命令一启动就完成)。若在 **rah** 命令行上指定此前缀序列, 则将该命令用单引号括起来, 或用双引号括起该命令并在前缀字符之前加 \ 。例如,
  - rah ' ; mydaemon'
  - 或
  - rah " ; \ mydaemon"

当做为后台进程运行时, **rah** 命令从不会等待返回任何输出。
- > 用机器名替换找到的 <>。
- " 用机器索引替换找到的 (), 用节点号替换找到的 ##。

**注:**

1. 机器索引是与数据库系统中的机器关联的号码。若不在运行多个逻辑节点, 则机器的机器索引对应于节点配置文件中该机器的节点号。要在多逻辑节点环境中获取机器的机器索引, 不要计算那些运行多个逻辑节点的机器的重复项目。例如, 若 MACH1 正在运行两个逻辑节点, MACH2 也正在运行两个逻辑节点, 则节点配置文件中 MACH3 的节点号为 5。但是, MACH3 的机器索引应是 3。  
在 Windows NT 上, 不要编辑节点配置文件。要获取机器索引, 使用 **db2nlist** 命令。有关详细信息, 参考 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings* 手册。
2. 当指定 " 时, 未从机器列表中删除重复。若想删除重复, 参见第316页的『从机器列表中删除重复项』。

当使用 <<-nnn< 和 <<+nnn< 前缀序列时, *nnn* 是任何 1、2 或 3 位分区号, 该分区号必须与 db2nodes.cfg 文件中的 *nodenum* 值匹配。

注：前缀序列被认为是命令的一部分。若指定前缀序列作为命令的一部分，必须将整个命令，包括前缀序列，括在双引号内。

---

## 指定机器列表

缺省情况下，机器列表取自节点配置文件 `db2nodes.cfg`。可以通过以下方法覆盖此设置：

- 通过导出（在基于 UNIX 的平台上）或设置（在 Windows NT 上）环境变量 `RAHOSTFILE`，指定包含机器列表的文件的名称。
- 通过导出（在基于 UNIX 的平台上）或设置（在 Windows NT 上）环境变量 `RAHOSTLIST`，明确指定该列表为由空格分隔的一连串名称。

注：若这两个环境变量都被指定，则 `RAHOSTLIST` 具有优先权。

注：在 Windows NT 上，要避免将不一致性引入该节点配置文件，不要以手工方式编辑它。要获取实例中机器的列表，使用 `db2nlist` 命令。有关详细信息，参考 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings* 手册。

## 从机器列表中删除重复项

若正在一台机器上运行具有多个逻辑节点（数据库分区服务器）的 DB2 扩充企业版，则 `db2nodes.cfg` 文件将包含那台机器的多个项目。在此情况下，`rah` 命令需要知道您是希望该命令在每台机器上只执行一次，还是为 `db2nodes.cfg` 文件中列出的每个逻辑节点均执行一次。使用 `rah` 命令来指定机器。使用 `db2_all` 命令来指定逻辑节点。

注：在基于 UNIX 的平台上，若您指定机器，则 `rah` 将会正常地从机器列表中删除重复的项，例外情况：若您指定逻辑节点，则 `db2_all` 会将下列赋值追加至您的命令之前：

```
export DB2NODE=nnn （对于 Korn shell 语法）
```

其中 `nnn` 取自 `db2nodes.cfg` 文件中的相应行的节点号，以便将命令按路径发送至所希望的数据库分区服务器。

当指定逻辑节点时，可以使用 `<<-nnn<` 和 `<<+nnn<` 前缀来限制列表包括除某个逻辑节点之外的所有逻辑节点，或指定一个数据库分区服务器。若想要首先在目录节点运行命令，并在该命令完成时，在所有其他数据库分区服务器上运行同一命令（可能以并行形式），则可能要执行此操作。当运行 `db2 restart database` 命令时，它常常是必需的。必须知道目录节点的节点号才能执行此操作。参见第 313 页的『前缀序列』以获取关于前缀序列的信息。

若您使用 **rah** 命令来执行 **db2 restart database**，重复项目就会从机器列表中被删除。然而，若指定 ” 前缀，则不删除重复项，因为认为使用 ” 前缀会隐含发送至每个数据库分区服务器，而不是发送至每台机器。

## 控制 rah 命令

可以使用下列环境变量来控制 **rah** 命令。

表 23.

名称	含义	缺省值
\$RAHBUFDIR 注：仅在基于 UNIX 的平台上可用。	缓冲区目录	/tmp/\$USER
\$RAHBUFNAME 注：仅在基于 UNIX 的平台上可用。	缓冲区文件名	rahout
\$RAHOSTFILE（在基于 UNIX 的平台上）； RAHOSTFILE（在 Windows NT 上）	包含主机列表的文件	db2nodes.cfg
\$RAHOSTFILE（在基于 UNIX 的平台上）； RAHOSTLIST（在 Windows NT 上）	字符串形式的主机列表	取自 \$RAHOSTFILE
\$RAHCHECKBUF 注：仅在基于 UNIX 的平台上可用。	若设置为 "no"，则绕过检查	未设置
\$RAHSLEEPTIME （在基于 UNIX 的平台上）； RAHSLEEPTIME （在 Windows NT 上）	以秒计的时间，此脚本将在此时间内等待来自以并行形式运行的命令的初始输出	对于 <b>db2_kill</b> 为 86400 秒，对于其他所有命令则为 200 秒

表 23. (续)

名称	含义	缺省值
\$RAHWAITTIME (在基于 UNIX 的 平台上);	在 Windows NT 上, 连续检查远程作业是否仍在运行的时间间隔 (以秒计)。	45 秒
RAHWAITTIME (在 Windows NT 上)	在基于 UNIX 的平台上, 连续检查远程作业是否仍在运行以及 rah: 正在等待 <pid> ... 消息的时间间隔 (以秒计)。  对于所有平台, 指定任何正整数。具有前导零的前缀值可抑制消息, 例如, 导出 RAHWAITTIME=045。  不必指定较小的值, 因为 rah 不依靠这些检查来检测作业是否完成。	
\$RAHENV 注: 仅在基于 UNIX 的平台上可用。	若 \$RAHDOTFILES=E 或 K 或 PE 或 B, 则指定要执行的文件名	\$ENV
\$RAHUSER (在基于 UNIX 的平台上);	在基于 UNIX 的平台上, 运行远程命令所使用的用户标识。	\$USER
RAHUSER (在 Windows NT 上)	在 Windows NT 上, 与 DB2 “远程命令服务” 相关的登录帐户	

注: 在基于 UNIX 的平台上, 将使用运行 rah 的 \$RAHENV 的值, 而不是远程 shell 设置的值 (若有的话)。

## 在基于 UNIX 平台上的 \$RAHDOTFILES

注: 本节中的信息只适用于基于 UNIX 的平台。

以下是未指定前缀序列时运行的 . 文件:

- P** .profile
- E** 在 \$RAHENV 中命名的文件 (可能是 .kshrc)
- K** 与 E 相同
- PE** 后跟 \$RAHENV 中命名的文件的 .profile (可能是 .kshrc)
- B** 与 PE 相同
- N** 无

**注:** 若登录界面不是 Korn shell, 则将在 Korn shell 进程中执行您指定要执行的任何点文件, 因此, 必须遵守 Korn shell 语法。例如, 若登录界面是 C shell, 要对 **rah** 执行的命令设置 .cshrc 环境, 则应该创建等效于 .cshrc 的 Korn shell INSHOME/.profile, 并在 INSHOME/.cshrc 中指定:

```
setenv RAHDOTFILES P
```

或应该创建等效于 .cshrc 的 Korn shell INSHOME/.kshrc, 并在 INSHOME/.cshrc 中指定:

```
setenv RAHDOTFILES E
setenv RAHENV INSHOME/.kshrc
```

并且, 若没有 tty (由 **rsh** 调用时), 则 .cshrc 必须不写入标准输出。通过将写至标准输出的任何行用引号括起来, 可确保这一点, 例如,

```
if { tty -s } then echo "executed .cshrc";
endif
```

## 在 Windows NT 上设置缺省环境简要表

**注:** 本节中的信息只适用于 Windows NT。

要设置 **rah** 命令的缺省环境简要表, 使用文件 db2rah.env, 应该在实例目录中创建该文件。该文件应该有以下格式:

```
; This is a comment line
DB2INSTANCE=instancename
DB2DBDFT=database
; End of file
```

可以指定为 **rah** 进行环境初始化所需要的所有环境变量。

---

## 在基于 UNIX 的平台上确定 rah 中的问题

**注:** 本节中的信息只适用于基于 UNIX 的平台。

以下是一些建议, 告诉您如何处理在运行 **rah** 时可能遇到的某些问题:

### 1. **rah** 挂起 (或运行很长的时间)

此问题可能是由下列原因引起的:

- **rah** 已确定它需要缓冲输出, 而您未导出 RAHCHECKBUF=no。因此, 在运行您的命令之前, **rah** 发送一个命令至所有机器以检查缓冲目录是否存在, 若不存在, 则会创建一个。
- 您在其中发送命令的一个或多台机器不响应。**rsh** 命令最终将超时, 但超时时间间隔相当长, 通常为 60 秒左右。

### 2. 已接受到下列各类消息:

- 登录不正确
- 拒绝许可权

其中一台机器没有在其 `.hosts` 文件中正确定义运行 **rah** 的标识，或运行 **rah** 的标识没有在其 `.rhosts` 文件中正确定义其中一台机器。

3. 当使用后台 `rshell` 以并行方式运行命令时，虽然这些命令在期望的时间内在机器上运行并完成，但是 **rah** 要耗费较长的时间来检测它并设置 `shell` 提示符。运行 **rah** 的标识没有在其 `.rhosts` 文件中正确定义其中某台机器。
4. 然 **rah** 从 `shell` 命令行运行时运行情况良好，但是若使用 `rsh` 远程运行 **rah**，例如，

```
rsh somewhere -l $USER db2_kill
```

，则 **rah** 将永不会完成。

这是正常的。**rah** 会启动后台监视进程，这些进程在 **rah** 退出后继续运行。那些进程通常将在与您运行的命令关联的所有进程将它们终止之后才结束。在 **db2\_kill** 的情况下，这意味着终止所有数据库管理器。可通过寻找其命令是 **rahwaitfor** 和 `kill <process_id>` 的进程来终止监视进程。不要指定信号号。而是要使用缺省值 (15)。

5. 当在同一 `$RAHUSER` 下发出多个 **rah** 命令时，**rah** 的输出未正确显示，或 **rah** 错误地报告 `$RAHBUFNAME` 不存在。

这是因为，并行执行多个 **rah** 正在尝试使用同一缓冲文件（例如 `$RAHBUFDIR/$RAHBUFNAME`）对输出进行缓冲。要避免此问题，将不同的 `$RAHBUFNAME` 用于每个并行 **rah** 命令，例如，在下列 `ksh` 中：

```
export RAHBUFNAME=rahout
rah ";$command_1" &
export RAHBUFNAME=rah2out
rah ";$command_2" &
```

或使用使 `shell` 自动选择唯一名称的方法，如：

```
RAHBUFNAME=rahout.$$ db2_a11 "....."
```

若磁盘空间有限，则无论使用哪种方法，都必须确保在某个时间清除缓冲文件。**rah** 不会在结束时删除缓冲文件，不过，下次您指定与现存文件相同的缓冲文件时，它将删除并重新使用该现存文件。

6. 输入

```
rah '"print from ()'
```

并接收到消息：

```
ksh: syntax error at line 1 : (' unexpected
```



替代 () 和 ## 的先决条件是:

- 使用 **db2\_all**，而不是使用 **rah**。
- 确保通过导出 **RAHOSTFILE** 或缺省为您的 `/sql1lib/db2nodes.cfg` 文件来使用 **RAHOSTFILE** 没有这些先决条件，**rah** 将照原样保持 () 和 ##。您接收到错误，因为命令 **print from ()** 无效。

要获取以并行方式运行命令时的性能提示，除非确实需要由 & 提供的功能，否则，使用 | 代替 |&，并使用 || 代替 ||& 或 ;。指定 & 需要更多 **rsh** 命令，因此会降低性能。



## 附录E. DB2 Windows NT 版如何使用 Windows NT 的安全性

当安装 Windows NT 时，它允许您创建两个管理员用户名：

- 一个称为 "Administrator"
- 另一个是由您选择的名称。它必须具有管理员权限且必须符合 DB2 的命名规则。有关 DB2 命名规则的更多信息，参见第277页的『附录A. 命名规则』。

用户可能要登录到本地机器，或者当在“Windows NT 域”中安装机器时，用户可能要登录到该域。DB2 Windows NT 版支持这两个选项。要认证用户，DB2 首先检查本地机器，然后检查当前域的“域控制器”，最后检查“域控制器”认识的任何一个“可信的域”。

为举例说明这是如何进行的，假设 DB2 实例需要服务器认证。该配置如下所示：

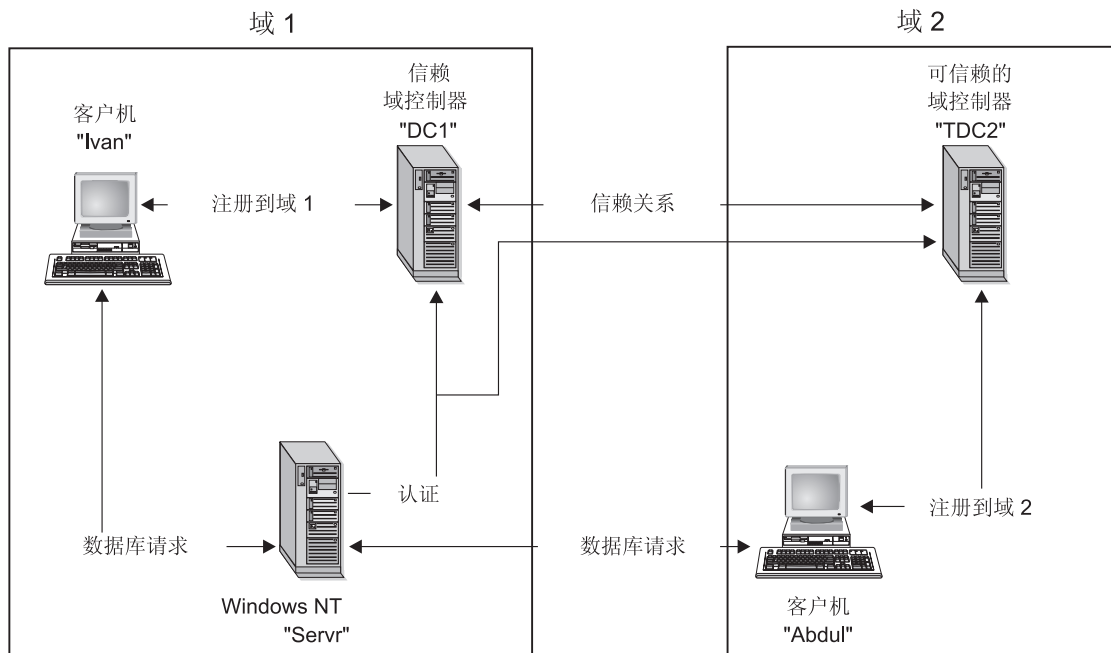


图 11. 使用 Windows NT 域的认证

除非客户机正在运行 Windows 9x, 否则, 每台机器应有一个安全性数据库, 即“安全性访问管理”(SAM)。Windows 9x 机器没有 SAM 数据库。DC1 是域控制器, 其中登记了客户机 Ivan 和“DB2 Windows NT 版服务器” Servr。TDC2 是 DC1 的一个可信的域, 客户机 Abdul 是 TDC2 的域的一个成员。

---

### 使用服务器认证的样本方案:

1. Abdul 注册到 TDC2 域 (亦即, 它在 TDC2 SAM 数据库中是可识别的)。
2. Abdul 然后与一个 DB2 数据库连接, 该数据库将被编目以驻留在 SRV3 上:  

```
db2 connect to remotedb user Abdul using fredpw
```
3. SRV3 确定可识别 Abdul 的位置。用于查找此信息的 API 首先搜索本地机器 (SRV3), 然后搜索域控制器 (DC1), 最后尝试搜索任何可信的域。在 TDC2 上找到用户名 Abdul。此搜索次序需要用户和组的单个名称空间。
4. 然后 SRV3:
  - a. 在 TDC2 上验证用户名和密码。
  - b. 通过询问 TDC2 来了解 Abdul 是否是管理员。
  - c. 通过询问 TDC2 来列举所有 Abdul 的组。

---

### 使用客户机认证和 Windows NT 客户机的样本方案:

1. 管理员 Dale 登录到 SRV3, 并将数据库实例的认证更改为“客户机”:  

```
db2 update dbm cfg using authentication client
db2stop myinst
db2start myinst
```
2. Ivan 在 Windows 客户机上登录到 DC1 域 (即, 他在 DC1 SAM 数据库中是可识别的)。
3. Ivan 然后与一个 DB2 数据库连接, 该数据库将被编目以驻留在 SRV3 上:  

```
DB2 CONNECT to remotedb user Ivan using johnpw
```
4. Ivan 的机器验证用户名和密码。用于查找此信息的 API 首先搜索本地机器 (Ivan), 然后搜索域控制器 (DC1), 最后尝试搜索任何可信的域。在 DC1 上找到用户名 Ivan。
5. 然后, Ivan 的机器用 DC1 来验证用户名和密码。
6. 然后 SRV3:
  - a. 确定可识别 Ivan 的位置。
  - b. 通过询问 DC1 来了解 Ivan 是否是管理员。
  - c. 通过询问 DC1 来列举所有 Ivan 的组。

**注:** 在尝试连接 DB2 数据库之前, 确保已启动 “DB2 安全性服务”。 “安全性服务” 由 DB2 安装并设置为作为 Windows NT 服务运行; 但它不是自动启动的。要启动 “DB2 安全性服务”, 输入 NET START DB2NTSECSERVER 命令。

---

## 使用客户机认证和 Windows 95 客户机的样本方案:

1. 管理员 Dale 登录到 SRV3, 并将数据库实例的认证更改为 “客户机”:

```
db2 update dbm cfg using authentication client
db2stop myinst
db2start myinst
```

2. 在 Windows 95 客户机中的 Ivan 登录到 DC1 域 (即, 他在 DC1 SAM 数据库中是可识别的)。
3. Ivan 然后与一个 DB2 数据库连接, 该数据库将被编目以驻留在 SRV3 上:

```
db2 connect to remotedb user Ivan using johnpw
```
4. Ivan 的 Windows 95 机器不能验证用户名和密码。因此, 假设该用户名和密码有效。
5. 然后 SRV3:
  - a. 确定可识别 Ivan 的位置。
  - b. 通过询问 DC1 来了解 Ivan 是否是管理员。
  - c. 通过询问 DC1 来列举所有 Ivan 的组。

**注:** 因为 Windows 95 客户机不能验证给出的用户名和密码, 所以在 Windows 95 下的客户机认证本质上是不安全的。但是, 若 Windows 95 机器可访问 Windows NT 安全性提供器, 则可将 Windows 95 系统配置为要进行验证的联通登录, 以便建立一定程度的安全性。有关如何用这种方式配置 Windows 95 系统的详细信息, 参考 Windows 95 的 Microsoft 文档。

DB2 还支持全局组。为使用全局组, 您必须将全局组包括在位于安全性服务器上的本地组中。当 DB2 列举某个人所属的所有组时, 它也列出用户间接所属的本地组 (由于在一个全局组中, 而该全局组本身是一个或多个本地组的成员)。

---

## 对 DB2 使用备份域控制器

若用于 DB2 的服务器也充当一个备份域控制器, 则当您配置 DB2 来使用备份域控制器时, 可改进 DB2 性能并减少网络通信。

通过设置 *DB2DMNBCKCTL*R 注册表变量, 来指定 DB2 的备份域控制器。

若您知道 DB2 服务器是其备份域控制器的域的名称, 则使用:

```
db2dmnbckctlr=DOMAIN_NAME
```

其中 *DOMAIN\_NAME* 必须大写。

要让 DB2 确定本地机器是其备份域控制器的域，则使用：

```
DB2DMNBCKCTRL=?
```

**注：**缺省情况下，DB2 不使用现存的备份域控制器，因为备份域控制器可以脱离与主域控制器的同步，从而导致安全性漏洞。当更新了主域控制器的安全性数据库，但没有将该更改传送至备份域控制器时，则域控制器可能脱离同步。若存在网络等待时间，或者若计算机浏览器服务未运行，也可能发生这种情况。

---

## 使用 DB2 Windows NT 版的用户认证

对于 Windows NT 用户而言，用户认证可能会产生问题，原因在于操作系统认证的方法。本节描述 DB2 Windows NT 版下的用户认证的一些注意事项：

- 『用户名和组名限制』
- 『DB2 Windows NT 版安全性服务』
- 第327页的『在备份域控制器上安装 DB2』
- 第328页的『使用组和域安全性的认证』

### 用户名和组名限制

下面是此环境中的限制：

- 在 DB2 中，用户名限长 30 个字符。组名限长 8 个字符。
- Windows NT 下的用户名不区分大小写；然而，密码区分大小写。
- 用户名和组名可以是大写字符与小写字符的组合。然而，当在 DB2 中使用时，它们通常被转换为大写字符。例如，若连接数据库并创建表 `schema1.table1`，则此表作为 `SCHEMA1.TABLE1` 存储在数据库中。（若您希望使用小写对象名，则从命令行处理器发出命令并将对象名括在引号中，或使用第三方 ODBC 前端工具。）

### DB2 Windows NT 版安全性服务

在“DB2 通用数据库”中，已将用户名和密码的认证集成到了“DB2 系统控制器”中。仅当将客户机与配置了 CLIENT 认证的服务器相连时，“安全性服务”才是必需的。

## 在备份域控制器上安装 DB2

在 Windows NT 环境中，可以在主或备份控制器上认证用户。在每个站点都有一个中央主域控制器和一个或多个备份域控制器 (BDC) 的大型分布式 LAN 中，此特点非常重要。可以在用户的站点上的备份域控制器上认证用户，而无需呼叫主域控制器 (PDC) 来进行认证。

在这种情况下，具有备份域控制器的优点是：认证用户的速度更快，LAN 不会象在没有 BDC 的情况下那么拥挤。

在下列条件下，会在 BDC 上发生认证：

- DB2 Windows NT 版服务器已安装在备份域控制器上。
- 已适当地设置了 DB2DMNBCKCTRL 简要表注册表变量。

若 DB2DMNBCKCTRL 简要表注册表变量未设置或设置为空白，则 DB2 Windows NT 版在主域控制器上执行认证。

DB2DMNBCKCTRL 的有效声明设置只有 “?” 或域名。

若 DB2DMNBCKCTRL 简要表注册表变量设置为问号 (DB2DMNBCKCTRL=?)，则在下列条件下，DB2 Windows NT 版将在备份域控制器上执行其认证：

- `cachedPrimaryDomain` 是设置为此机器所属的域的名称的注册表值。(可以在 **HKEY\_LOCAL\_MACHINE—> Software—> Microsoft—> Windows NT—> Current Version—> WinLogon** 下面找到此值。)
- “服务器管理器”显示备份域控制器活动且可用。(即，此机器的图标不是灰色的。)
- DB2 Windows NT 服务器的注册表指示该系统是指定的域上的备份域控制器。

DB2DMNBCKCTRL=? 设置在正常情况下可以起作用；但并非所有环境下都能起作用。域上提供的关于服务器的信息是动态的，必须运行“计算机浏览器”才能保持此信息准确和最新。大型 LAN 可能未运行“计算机浏览器”，因而“服务器管理器”的信息可能不是最新的。在这种情况下，还有一种方法来告知 DB2 Windows NT 版在备份域控制器上执行认证：设置 `DB2DMNBCKCTRL=xxx`，其中，`xxx` 是 DB2 服务器的 Windows NT 域名。借助此设置，在下列条件下，认证将发生在备份域控制器上：

- 注册表值 `cachedPrimaryDomain` 设置为此机器所属的域的名称。(可以在 **HKEY\_LOCAL\_MACHINE—> Software—> Microsoft—> Windows NT—> Current Version—> WinLogon** 下面找到此值。)
- 该机器配置成指定域的备份域控制器。(若该机器设置为另一个域的备份域控制器，则此设置将导致错误。)

## 使用组和域安全性的认证

DB2 Windows NT 版支持下列类型的组:

- 本地组
- 全局组
- 作为本地组成员的全局组。

DB2 Windows NT 版使用用户所在的安全性数据库来列举本地组和全局组, 该用户是这些组的成员。“DB2 通用数据库”提供了一种覆盖, 即强制组列举发生在安装 DB2 的本地 Windows NT 服务器上, 而不考虑找到用户帐户的位置。此覆盖可以使用下列命令实现:

- 对于全局设置:

```
db2set -g DB2_GRP_LOOKUP=local
```

- 对于实例设置:

```
db2set -i DB2_GRP_LOOKUP=local
```

要查看设置的所有 DB2 简要注册表变量, 输入

```
db2set -all
```

要让 DB2 Windows NT 版使用域安全性, 必须将权限和特权授予本地组。本地和全局组中的用户名必须在该本地或全局组所在的域中定义, 才能进行正确的认证。

若 DB2\_GRP\_LOOKUP 简要表注册表变量设置为 local, 则 DB2 只尝试在本地机器上查找用户。若在本地机器上找不到该用户, 或未将该用户定义为本地或全局组的成员, 则认证失败。DB2 并不尝试在该域中的另一机器上或在域控制器上查找该用户。

若 DB2\_GRP\_LOOKUP 简要表注册表变量未设置, 则:

1. DB2 首先尝试在同一机器上查找用户。
2. 若用户名是以本地方式定义的, 则以本地方式认证该用户。
3. 若以本地方式找不到该用户, 则 DB2 尝试在它的域上查找该用户, 然后在可信的域上查找。

下列示例说明了 DB2 Windows NT 版对域安全性的支持。在第一个示例中, 因为用户名和本地组在同一域上, 所以可以进行连接。在第二个示例中, 因为用户名与本地或全局组在不同的域上, 所以不能进行连接。

**成功连接的示例:** 在以下方案中, 因为用户名与本地或全局组在同一域上, 所以可以进行连接。



注意，用户名与本地或全局组无需在运行数据库服务器的域上定义，但它们必须在同一个域上。

表 24. 使用域控制器的成功连接

Domain1	Domain2
存在与 Domain2 的信赖关系。	<ul style="list-style-type: none"> <li>存在与 Domain1 的信赖关系。</li> <li>定义了本地或全局组 grp2。</li> <li>定义了用户名 id2。</li> <li>用户名 id2 是 grp2 的一部分。</li> </ul>
DB2 服务器在此域中运行。从它发出下列 DB2 命令：  <pre>REVOKE CONNECT ON db FROM public GRANT CONNECT ON db TO GROUP grp2 CONNECT TO db USER id2</pre>	
扫描本地或全局域，但找不到 id2。扫描域安全性。	
	在此域中找到用户名 id2。DB2 获得关于此用户名的其他信息（即，它是组 grp2 的一部分）。
因为用户名与本地或全局组在同一域上，所以可以进行连接。	

**不成功连接的示例：**在以下方案中，因为用户名与本地或全局组在不同域上定义，所以不能进行连接。

表 25. 使用域控制器的不成功连接

Domain1	Domain2
存在与 Domain2 的信赖关系。	<ul style="list-style-type: none"> <li>存在与 Domain1 的信赖关系。</li> <li>定义了本地或全局组 grp2。</li> </ul>
<ul style="list-style-type: none"> <li>定义了全局组 grp1。</li> <li>定义了用户名 id1。</li> <li>用户名 id1 是 grp1 的一部分。</li> </ul>	
	Domain1\grp1 是 grp2 的一部分。
DB2 服务器在此域中运行。从它发出下列 DB2 命令：  <pre>REVOKE CONNECT ON db FROM public GRANT CONNECT ON db TO GROUP grp2 CONNECT TO db USER id2</pre>	

表 25. 使用域控制器的不成功连接 (续)

Domain1	Domain2
扫描本地或全局，找到 id1。DB2 获得关于此用户名的信息（即，用户名 id1 是 grp1 的一部分，组 grp1 是 Domain2\grp2 的一部分）。	
	组 grp2 存在于此域上。
因为本地或全局组在 Domain2 上，而实际用户名在 Domain1 上定义，所以不能进行连接。	
相反，若发出以下命令，则可以进行连接：GRANT CONNECT ON db TO GROUP grp1	

---

## 附录F. 使用 Windows NT 性能监视器

DB2 Windows NT 版的用户有两个性能监视器可用:

- **DB2 性能监视器**

“DB2 性能监视器”只提供与 DB2 和 DB2 Connect 相关的快照和事件数据。  
(有关更多信息,单击控制中心中的帮助按钮,查看“入门”联机帮助。)

- **Windows NT 性能监视器**

“Windows NT 性能监视器”可用来监视数据库和系统性能,从向系统注册的任何性能数据提供者中检索信息。Windows NT 也提供有关机器运行各个方面的性能信息数据,包括:

- CPU 的使用
- 内存使用率
- 磁盘活动
- 网络活动

---

### 向 Windows NT 性能监视器注册 DB2

设置程序自动向“Windows NT 性能监视器”注册 DB2。

要使“Windows NT 性能监视器”可访问 DB2 和 DB2 Connect 性能信息,必须注册“DB2 Windows NT 性能计数器”的 DLL。这也就允许任何其他使用 Win32 性能 API 的 Windows NT 应用程序获取性能数据。

要安装并向“Windows NT 性能监视器”注册“DB2 Windows NT 版性能计数器”DLL (DB2Perf.DLL),输入:

```
db2perfi -i
```

注册该 DLL 的同时会在注册表的 services 选项中创建一个新键。其中一项给出 DLL 的名称,它提供计数器支持。另外三项给出该 DLL 中提供的函数的名称。这些函数包括:

- **Open**

在一个进程中系统首次装入该 DLL 时调用。

- **Collect**

从 DLL 请求性能信息时调用。

- **Close**

卸载 DLL 时调用。

---

## 允许远程访问 DB2 性能信息

若 DB2 Windows NT 版工作站与其他 Windows NT 机器联网，可使用本节中描述的功能部件。

为了从另一个 DB2 Windows NT 版的机器查看 Windows NT 性能对象，必须向 DB2 注册管理员用户名和密码。（“Windows NT 性能监视器”的缺省用户名 **SYSTEM** 是 DB2 保留字，因此不能使用。）要注册该用户名，输入：

```
db2perfr -r username password
```

**注：**使用的 username 必须符合 DB2 命名规则。

用户名和密码数据保存在注册表内的一个键中，并设置了安全性，它只允许管理员和 **SYSTEM** 帐户访问。编码该数据，以避免在注册表中存储管理员密码出现安全性问题。

**注：**

1. 一旦将用户名与密码一起注册到 DB2 中，甚至“性能监视器”的本地实例也将使用该用户名和密码显式登录。这就表示，若向 DB2 注册的用户名信息不匹配，“性能监视器”的本地会话将不显示 DB2 性能信息。
2. 必须维护该用户名与密码的组合，以便与 Windows NT 安全性数据库中存储的用户名和密码的值匹配。若在 Windows NT 安全性数据库中更改了用户名或密码，必须复位用于监视远程性能的用户名与密码的组合。
3. 要注销，输入：

```
db2perfr -u <username> <password>
```

---

## 显示 DB2 和 DB2 Connect 性能值

要使用“性能监视器”显示 DB2 和 DB2 Connect 性能值，只需从**添加至**框中选择您想要显示其值的性能计数器。此框显示性能对象的列表及其性能数据。选择一个对象，查看该对象提供的计数器列表。

一个性能对象也可以有多个实例。例如，LogicalDisk 对象提供象“% 磁盘读时间”和“磁盘字节 / 秒”这样的计数器；它还为机器上的每个逻辑驱动器提供一个实例，包括“C:”和“D:”。

Windows NT 提供下列性能对象：

- **DB2 数据库管理器**

此对象提供单个 Windows NT 实例的一般信息。受监视的 DB2 实例表现为对象实例。

由于实际原因和性能原因，一次只能从一个 DB2 实例获取性能信息。“性能监视器”显示的 DB2 实例受“性能监视器”进程中的 db2instance 注册表变量控制。若您有多个 DB2 实例在同时运行，并且想查看多个实例的性能信息，对于每个要监视的 DB2 实例，将 db2instance 设置为相关值，然后启动一个单独的“性能监视器”会话。

若是在运行分区数据库系统，则一次只能从一个数据库分区服务器（节点）获取性能信息。在缺省情况下，显示缺省节点（例如，带有逻辑端口 0 的节点）的性能信息。要查看另一节点的性能信息，必须启动“性能监视器”的一个单独会话，并将 DB2NODE 环境变量设置为要监视的节点的节点号。

- **DB2 数据库**

此对象提供特定数据库的信息。每个当前活动的数据库都有可用的信息。

- **DB2 应用程序**

此对象提供特定的 DB2 应用程序的信息。每个当前活动的 DB2 应用程序都有可用的信息。

- **DB2 DCS 数据库**

此对象提供特定的 DCS 数据库的信息。每个当前活动的数据库都有可用的信息。

- **DB2 DCS 应用程序**

此对象提供特定的 DB2 DCS 应用程序的信息。每个当前活动的 DB2 DCS 应用程序都有可用的信息。

“Windows NT 性能监视器”将列出哪些对象，取决于 Windows NT 机器上安装了什么内容以及哪些应用程序是活动的。例如，若安装了 DB2 UDB 且已启动数据库管理器，将列出“DB2 数据库管理器”对象。若同一台机器上还有一些 DB2 数据库和应用程序当前是活动的，也将列出那些 DB2 数据库和 DB2 应用程序对象。若将 Windows NT 系统用作 DB2 Connect 网关，且有一些 DCS 数据库和应用程序当前是活动的，将列出那些 DB2 DCS 数据库和 DB2 DCS 应用程序的对象。

---

## 访问远程 DB2 性能信息

前面已讨论过允许远程访问“DB2 性能信息”。在**添加至**框中选择要监视的另一台计算机。这将显示一个列表，列出该计算机上所有可用的性能对象。

为了能够监视远程计算机上的 DB2 性能对象，安装在那台计算机上的 DB2 UDB 或 DB2 Connect 代码的级别必须是版本 6 或更高版本。

---

## 复位 DB2 性能值

当一个应用程序调用 DB2 监视器 API 时，返回的信息通常是累积值，因为先前已启动了 DB2 服务器。但它经常可用于：

- 复位性能值
- 运行测试
- 再次复位值
- 重新运行测试。

要复位数据库性能值，使用 **db2perf** 程序。输入：

```
db2perf
```

缺省情况下，此命令将复位所有活动的 DB2 数据库的性能值。但也可指定要复位的数据库的列表。还可使用 **-d** 选项指定应复位 DCS 数据库的性能值。例如：

```
db2perf
db2perf dbalias1 dbalias2 ... dbaliasn

db2perf -d
db2perf -d dbalias1 dbalias2 ... dbaliasn
```

第一个示例复位所有活动的 DB2 数据库的性能值。第二个示例复位特定的 DB2 数据库的性能值。第三个示例复位所有活动的 DB2 DCS 数据库的性能值。最后一个示例复位特定的 DB2 DCS 数据库的性能值。

**db2perf** 程序复位当前正在访问相关的 DB2 服务器实例（即，运行 **db2perf** 所在的会话中 `db2instance` 中保存的服务器实例）的数据库性能信息的所有程序的值。

当执行 **db2perf** 命令时，调用 **db2perf** 也能复位远程访问 DB2 性能信息的任何人看到的值。

**注：**有一个 DB2 API `sqlmrset`，它允许应用程序复位它在本地而非全局看到的特定数据库的值。有关更多信息，参见 *Administrative API Reference*。

---

## 附录G. 使用 Windows NT 或 Windows 2000 数据库分区服务器

当在 Windows NT 或 Windows 2000 环境中更改配置的特征时，涉及的任务是使用特定实用程序执行的。其他操作系统环境使用《管理指南：性能》的“通过添加处理器调整配置”一章中显示的方法。

此处讨论的实用程序包括：

- 『列示实例中的数据库分区服务器』
- 『向实例添加数据库分区服务器』
- 第337页的『更改数据库分区』
- 第338页的『从实例中删除数据库分区』

---

### 列示实例中的数据库分区服务器

在 Windows NT 或 Windows 2000 上，使用 **db2nlist** 命令来获取参与一个实例的数据库分区服务器的列表。

按如下所示使用该命令：

```
db2nlist
```

当按以上所示使用此命令时，缺省实例是当前实例（由 **DB2INSTANCE** 环境变量设置）。要指定特定的实例，可使用以下命令指定该实例：

```
db2nlist /i:instName
```

其中 **instName** 是想要的特定实例名。

作为选项，也可使用以下命令请求每个分区服务器的状态：

```
db2nlist /s
```

每个数据库分区服务器的状态可能是：正在启动、正在运行、正在停止或已停止。

---

### 向实例添加数据库分区服务器

在 Windows NT 或 Windows 2000 上，使用 **db2ncrt** 命令来向实例添加数据库分区服务器（节点）。

**注：**若此实例已包含数据库，则不要使用 **db2ncrt** 命令。而使用 **db2start addnode** 命令。这确保可正确地将该数据库添加至新的数据库分区服务器。  
不要编辑 `db2nodes.cfg` 文件，因为更改文件可能导致分区数据库系统中的不一致性

该命令具有下列必需的参数：

```
db2ncrt /n:node_number
        /u:username,password
        /p:logical_port
```

- /n:  
用于标识数据库分区服务器的唯一节点号。该号码可以是按递升顺序排列的 1 到 999 中的任何一个值。
- /u:  
DB2 服务的登录帐户名和密码。
- /p:logical\_port  
用于数据库分区服务器的逻辑端口号（若逻辑端口不是零 (0)）。若不指定，则将逻辑端口号指定为 0。

仅当在机器上创建第一个节点时，逻辑端口参数才是可选的。若创建逻辑节点，则必须指定此参数并选择一个不在使用中的端口号。有几项限制：

- 在每台机器上，都必须要有有一个逻辑端口为 0 的数据库分区服务器。
- 端口号不能超过 `x:\winnt\system32\drivers\etc\` 目录中的服务文件中为 FCM 通信保留的端口范围。例如，若为当前实例保留四个端口，则最大端口号将是 3（端口 1、2 和 3；端口 0 用于缺省逻辑节点）。端口范围是在使用带有 `/r:base_port, end_port` 参数的 **db2icrt** 时定义的。

还有几个可选参数：

- /g:network\_name  
指定数据库分区服务器的网络名。若不指定此参数，则 DB2 使用它在系统上检测到的第一个 IP 地址。  
若机器上有多个 IP 地址，且您想要对数据库分区服务器指定特定 IP 地址，则使用此参数。可输入使用网络名或 IP 地址的 `network_name` 参数。
- /h:host\_name  
FCM 用于内部通信的 TCP/IP 主机名（若该主机名不是本地主机名）。若在远程机器上添加数据库分区服务器，则此参数是必需的。
- /i:instance\_name  
实例名；缺省值为当前实例。



- /m:machine\_name  
该节点所驻留的 Windows NT 工作站的计算机名；缺省名称是本地机器的计算机名。
- /o:instance\_owning\_machine  
该拥有实例的机器的计算机名称；缺省值是本地机器的计算机名称。当在非该拥有实例的任何机器上调用 **db2ncrt** 命令时，此参数是必需的。

例如，若要向拥有实例的机器 MYMACHIN 上的实例 TESTMPP 添加新的数据库分区服务器（以运行多个逻辑节点），且您想要让这个新节点成为使用逻辑端口 1 的节点 2，则输入：

```
db2ncrt /n:2 /p:1 /u:my_id,my_pword /i:TESTMPP
/M:TEST /o:MYMACHIN
```

---

## 更改数据库分区

在 Windows NT 或 Windows 2000 上，使用 **db2nchg** 命令来执行下列各项：

- 将数据库分区从一台机器移至另一台机器。
- 更改机器的 TCP/IP 主机名。  
若计划使用多个网络适配器，则必须使用此命令来为 *db2nodes.cfg* 文件中的 "netname" 字段指定 TCP/IP 地址。
- 使用另一逻辑端口号。
- 对数据库分区服务器（节点）使用另一名称。

该命令具有下列必需的参数：

```
db2nchg /n:node_number
```

参数 /n: 是您想要更改的数据库分区服务器的配置的节点号。此参数是必需的。

可选参数包括：

- /i:instance\_name  
指定此数据库分区服务器所参与的实例。若不指定此参数，则缺省值是当前实例。
- /u:username,password  
更改 DB2 服务的登录帐户名和密码。若不指定此参数，则登录帐户和密码保持不变。
- /p:logical\_port  
更改数据库分区服务器的逻辑端口。若将数据库分区服务器移至另一机器，则必须指定此参数。若不指定此参数，则逻辑端口号保持不变。

- /h:host\_name

更改 FCM 用于内部通信的 TCP/IP 主机名。若不指定此参数，则主机名不更改。

- /m:machine\_name

将数据库分区服务器移至另一机器。仅当实例中没有现存数据库时，才可移动数据库分区服务器。

- /g:network\_name

更改数据库分区服务器的网络名。

若机器上有多个 IP 地址，且您想要对数据库分区服务器使用特定 IP 地址，则使用此参数。可使用网络名或 IP 地址输入 network\_name。

例如，要将分配给节点 2（它参与实例 TESTMPP）的逻辑端口更改为使用逻辑端口 3，则输入以下命令：

```
db2nchg /n:2 /i:TESTMPP /p:3
```

---

## 从实例中删除数据库分区

在 Windows NT 或 Windows 2000 上，使用 **db2ndrop** 命令来从不带数据库的实例删除数据库分区服务器（节点）。若删除了一个数据库分区服务器，则它的节点号可以再次用于新的数据库分区服务器。

当从实例中删除数据库分区服务器时务必小心。若从该实例删除拥有实例的数据库分区服务器节点 0，该实例将变成不可用的。若要删除该实例，使用 **db2idrop** 命令。

**注：**若此实例包含数据库，则不要使用 **db2ndrop** 命令。而使用 **db2stop drop nodenum** 命令。这确保可正确地从数据库分区中除去该数据库。不要编辑 db2nodes.cfg 文件，因为更改文件可能导致分区数据库系统中的不一致性

若要从运行多个逻辑节点的机器中删除被分配了逻辑端口 0 的节点，则在删除被分配了逻辑端口 0 的节点之前，必须删除分配给其他逻辑端口的所有其他节点。每个数据库分区服务器都必须带有一个分配给逻辑端口 0 的节点。

该命令具有下列参数：

```
db2ndrop /n:node_number /i:instance_name
```

- /n:

用于标识数据库分区服务器的唯一节点号。这是一个必需的参数。该号码可以是按递升顺序排列的 0 到 999 中的任何一个值。记住节点 0 表示该拥有实例的机器。

- /i:instance\_name

实例名。这是一个可选参数。若不给出，则缺省值是当前实例（由 DB2INSTANCE 注册表变量设置）。



---

## 附录H. 配置多个逻辑节点

通常，将“DB2 扩充企业版”配置为对每台机器指定一个数据库分区服务器。然而，在几种情况下，在同一机器上运行数个数据库分区服务器很有好处。这意味着配置包含的节点数可以多于机器数。在这些情况下，我们称该机器正在运行多个逻辑节点（若这些逻辑节点参与同一实例的话）。若它们参与不同的实例，则此机器不主管多个逻辑节点。

借助多逻辑节点支持，您可以从三种类型的配置中进行选择：

- 标准配置，即每台机器只有一个数据库分区服务器。
- 多逻辑节点配置，即机器有多个数据库分区服务器。
- 在数台机器的每一台上运行数个逻辑节点的配置。

当系统在具有对称多处理机 (SMP) 体系结构的机器上运行查询时，使用多逻辑节点的配置非常有用。万一有机器失效，在机器上配置多个逻辑节点的能力也是很有用的。若一台机器失效（导致其上的数据库分区服务器失效），则可以在另一机器上使用 `DB2START NODENUM` 命令重新启动数据库分区服务器。这确保用户数据保持可用。

另一个好处是多个逻辑节点可利用 SMP 硬件配置。另外，因为数据库分区较小，所以当执行诸如备份和复原数据库和表空间以及创建索引之类的任务时，可以获得较佳的性能。

可用两种方式之一配置多个逻辑节点：

- 在 `db2nodes.cfg` 文件中配置逻辑节点（数据库分区）。然后，可使用 `DB2START` 命令或它的相关 API 启动所有逻辑节点和远程节点。

**注：**对于 Windows NT，若系统中没有数据库，则必须使用 `db2ncrt` 来添加节点，否则，若有一个或多个数据库，则应使用 `DB2START ADDNODE` 命令。在 Windows NT 中，绝对不应手工编辑 `db2nodes.cfg` 文件。

- 在另一个处理器上重新启动一个逻辑节点，其他逻辑数据库分区（节点）已在该处理器上运行。这允许您覆盖在 `db2nodes.cfg` 中为逻辑数据库分区指定的主机名和端口号。

要在 `db2nodes.cfg` 中配置一个逻辑数据库分区（节点），您必须在该文件中建立一项以便为该节点分配一个逻辑端口号。以下是应使用的语法：

```
nodenumber hostname logical-port netname
```

**注:** 对于 Windows NT, 若系统中没有数据库, 则必须使用 *db2ncrt* 来添加节点, 否则, 若有一个或多个数据库, 则应使用 `DB2START ADDNODE` 命令。在 Windows NT 中, 绝对不应手工编辑 *db2nodes.cfg* 文件。

Windows NT 上 *db2nodes.cfg* 文件的格式与 Unix 上同一文件的格式并不相同。在 Windows NT 上, 列格式是:

```
nodenumber hostname computername logical_port netname
```

必须确保在 `etc` 目录的 `services` 文件中为 FCM 通信定义了足够的端口。

---

## 附录I. 高速节点间通信

使用“DB2 通用数据库扩充企业版”时，您可能是在一个通信密集型环境中工作，在此环境中，系统整体吞吐量对于您的商业而言非常重要。

有两种类型的网络可用于分区环境。一种使用基于公共 LAN 的 TCP/IP。另一种类型使用基于专用互连的 TCP/IP 或“虚拟接口”(VI) 体系结构。

公共互连使用现存的 TCP/IP。TCP/IP 几乎在所有地方都可用作通信协议。这是“局域网”(LAN) 环境。此环境的优点是您可以选择立即连接群集，而不需要其他私人硬件和软件。此环境的缺点是附加的群集通信会影响基于整个 LAN 的服务的质量。例如，群集内的数据库活动可能会产生通信“爆炸”作用，这会影响通过 LAN 的通信。并且，LAN 环境其余部分的通信使得难以维护群集中的数据库处理的一致性能。

专用互连作为单独的网络工作。该网络可能是唯一可在群集中使用的网络，它可以配合 LAN 环境使用。该网络专门用来在群集的成员之间提供通信。这称为“系统区网络”(SAN)。数据库的性能不受外部通信(如在 LAN 中)影响，反之亦然。此环境的缺点是两个网络可能需要单独的管理，并且 LAN 和 SAN 都需要附加的单独硬件、软件和协议成本。专用互连的一个示例是 100 Mb / 秒以太网。

您可能希望维护事先存在的公共 LAN 环境，但您也希望有能力通过(群集中的) SAN 大量传送数据。若您希望在群集之上进行通信访问，则这样的安排非常方便。在 Windows NT 操作环境中，您可能需要为对“NT 域控制器”的通信访问保留公共 LAN。(有关“域控制器”的信息，参见第323页的『附录E. DB2 Windows NT 版如何使用 Windows NT 的安全性』。)

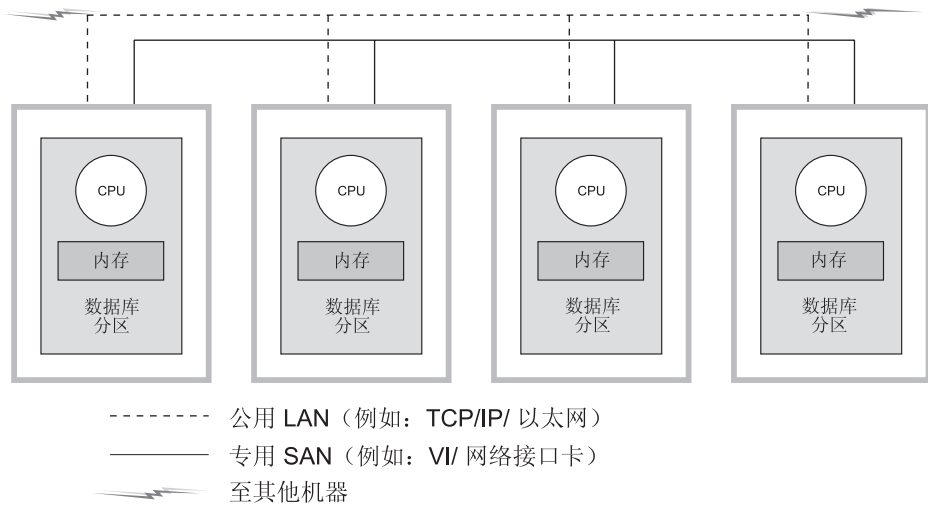


图 12. 将专用 SAN 与公共 LAN 组合

其余各节讨论下列各项:

- 『使用 TCP/IP 的高速互连』
- 第345页的『使用 VI 的高速互连』

## 使用 TCP/IP 的高速互连

使用 TCP/IP 的网络硬件设置的先决条件的示例包括:

- 标准以太网。  
没有不寻常的硬件、软件或协议需求。
- IBM Netfinity SP Switch。  
下一节将概述这些需求。

## 使用 IBM Netfinity SP Switch 的先决条件

要了解 Netfinity, 请访问以下 URL: <http://www.ibm.com/pc/us/netfinity>

有关其他文档和软件升级, 请访问 IBM Support Web 站点, 地址是:  
<http://www.ibm.com/pc/support>

1. 单击**服务器**
2. 在“系列”下面, 单击**分群**
3. 在“技术信息”下面, 单击**可下载的文件**以获取软件升级, 或单击**联机出版物**以获取文档



找出 IBM Netfinity SP Switch 主题并下载需要的文件。

### IBM Netfinity SP Switch 的设置过程

可在 *IBM Netfinity SP Switch Installation and User's Guide* 中找到有关安装 IBM Netfinity SP Switch 的指导。

应使用伴随各种硬件和软件组件（如服务器机架外壳、主机适配器和 SP 开关软件）的硬件和软件指南来安装、配置和测试这些组件。

DB2 在安装后，如不进行任何其他修改，它将使用 IBM Netfinity SP Switch。

---

## 使用 VI 的高速互连

“虚拟接口” (VI) 体系结构是在有大量并行处理 (MPP) 的 Windows NT 环境中代替 TCP/IP 的节点间通信协议。VI 是一种新的通信体系结构，它是由 Intel、Microsoft 和 Compaq 为改进“系统区域网” (SAN) 的性能而联合开发的。有关此体系结构的更多信息，访问 <http://www.viarch.org>。

有一些产品可能是与“DB2 通用数据库”分开订购的，它们具有启用 VIA 的网络接口卡 (NIC)、开关和软件驱动程序实现。几家“独立硬件供应商” (IHV) 已经发行或计划发行这类产品。

与 TCP/IP 相比，“VI 体系结构”具有等待时间短、高带宽和低 CPU 消耗量的特点。在通信密集型环境中，使用 VI 体系结构可提高整个系统的吞吐量。群集中的节点数越多，而且传输的数据量越大，使用 VI 体系结构获得的好处就越多。

“DB2 通用数据库”支持“VI 体系结构”实现，这些实现符合 *Virtual Interface Architecture Specification, Version 1.0* 和 *Intel Virtual Interface (VI) Architecture Developers' Guide, Version 1.0* 并通过“虚拟接口体系结构一致性规则”。可在 Web 上的 [http://www.intel.com/design/servers/vi/the\\_spec/specification.htm](http://www.intel.com/design/servers/vi/the_spec/specification.htm) 中找到该规范。可在 Web 上的 [http://www.intel.com/design/servers/vi/developer/ia\\_imp\\_guide.htm](http://www.intel.com/design/servers/vi/developer/ia_imp_guide.htm) 中找到 Developer's Guide。有关一致性规则的信息也可在这个 URL 中找到。

IBM 已宣布“DB2 通用数据库 EEE” V5.2 支持“虚拟接口” (VI) 体系结构。

要了解符合 VI 体系结构并受“DB2 通用数据库 EEE”支持的其他产品，请与“DB2 通用数据库”支持机构联系，网址为 <http://www.software.ibm.com/data>，或致电 1-800-237-5511（仅在美国和加拿大）。

已使用“DB2 通用数据库”测试的产品包括：

- GigaNet Interconnect，详细信息请参见第346页的『GigaNet Interconnect 的设置过程』。

- Compaq ServerNet Interconnect, 详细信息请参见第349页的『ServerNet Interconnect 的设置过程』。
- Fujitsu Synfinity Interconnect, 详细信息请参见第351页的『Synfinity Interconnect 的设置过程』。

可能还有使用“DB2 通用数据库”的其他产品。与该产品的供应商以及“IBM 服务与支持机构”联系, 确保其他产品是受支持的。

## 虚拟接口 (VI) 硬件设置

使用 VI 的网络硬件设置的先决条件的示例包括:

- GigaNet Interconnect。  
『GigaNet Interconnect 的设置过程』概述了此选项所需的硬件、软件和协议信息。  
要了解 GigaNet 产品或与“GigaNet 服务与支持机构”联系, 请访问以下 URL: <http://www.giganet.com/>
- Compaq ServerNet Interconnect。  
第349页的『ServerNet Interconnect 的设置过程』概述了此选项所需的硬件、软件和协议信息。  
要了解 ServerNet 产品或与“ServerNet 服务与支持机构”联系, 请访问以下 URL: <http://www.servernet.com/>
- Fujitsu Synfinity Interconnect。  
第351页的『Synfinity Interconnect 的设置过程』概述了此选项所需的硬件、软件和协议信息。  
要了解 Synfinity 产品或与 Fujitsu System Technologies 的“Synfinity 服务与支持机构”联系, 请访问以下 URL: <http://www.fjst.com/>

必须将 DB2 配置为使用 VI。第353页的『启用 DB2 以使用 VI 运行』带有您使用 VI 时所必需的信息。

### GigaNet Interconnect 的设置过程

设置此环境所需的硬件和软件的列表包括下列产品:

- GigaNet GNN1000 网络接口卡
- GigaNet GNX5000 开关
- GigaNet GNCxx11 铜线连接电缆
- GigaNet cLAN 软件版本 2.0。

确保 GigaNet Interconnect 可使用“DB2 通用数据库”所必需的步骤如下所示。每个步骤只是简单介绍了该步骤所需的操作：此处不提供与每个步骤相关的所有详细信息。您还应参考每个步骤中引用的文档，该文档提供了详细的指示信息和指导。

每个 GigaNet GNN1000 与一个 GigaNet cLAN 软件 CD-ROM 封装在一起。该 CD-ROM 包含设置 GigaNet Interconnect 需要的所有软件。另外，该 CD-ROM 还包含“VI 体系结构”软件开发工具箱 (SDK) 和 Adobe Acrobat 阅读器。“VI 体系结构”SDK 将用来开发和测试支持 VI 的应用程序。Adobe Acrobat 阅读器将用来查看 CD-ROM 上解释如何开发支持 VI 的应用程序的文档。

步骤摘要：

1. 安装适配卡
2. 安装开关和电缆
3. 安装适配器驱动程序
4. 安装 cLAN 管理控制台
5. 测试 Interconnect

以下是各步骤：

1. 安装 GigaNet GNN1000 网络接口卡。有关安装说明，请参考 *GigaNet GNN1000 User Guide*。
2. 安装 GigaNet GNX5000 开关和电缆。有关安装说明，请参考 *GigaNet GNX5000 User Guide*。
3. 在与 GNX5000 开关连接的每个节点上安装“GigaNet GNN1000 适配器驱动程序”软件。有关安装说明，请参考 *GigaNet GNN1000 User Guide*。若要安装 GigaNet 提供的驱动程序，以下是附加详细信息：
  - a. 除去已安装的任何先前版本的 GNN1000 驱动程序。除去操作要求重新启动节点。
  - b. 使用 开始->设置->控制面板->网络->适配器->添加来安装驱动程序
  - c. 单击**从磁盘安装...**并指定 CD-ROM 上的 Driver 目录。例如，若 F: 是 CD-ROM 驱动器，则应使用 F:\Driver
  - d. 选择“GNN1000 NDIS 适配器”，然后单击**确定**。
  - e. 配置“网络协议”以完成安装。

“GigaNet 适配器驱动程序”软件也可在 GigaNet 的 Web 站点获取：<http://www.giganet.com>。请参考可在 GigaNet 的 Web 站点的支持页面上找到的下载和安装指南。

“GNN1000 适配器驱动程序”的安装导致节点重新启动。

4. 可使用“GigaNet cLAN 管理控制台”(GMC)测试 GigaNet Interconnect 的完整性。“GigaNet cLAN 管理控制台”由两部分组成:“控制台”和“代理程序”。“代理程序”必须安装在群集中的所有节点上。“控制台”可安装在对群集中的节点有访问权的任何网络节点上。最通用的和推荐的安装方式是在群集中的每个节点上均安装“控制台”和“代理程序”。

安装“GigaNet cLAN 管理控制台”。有关“cLAN 管理控制台”的安装说明和其他信息,请参考 *GigaNet GNN1000 User Guide*。以下是关于安装过程的附加详细信息:

- a. 将 cLAN 软件 CD 插入 CD-ROM 驱动器。
- b. 等待 CD 自动安装菜单出现。
- c. 单击“安装 cLAN 管理控制台”。
- d. 对群集中的每个剩余节点重复此安装过程。

“GigaNet cLAN 管理控制台”软件也可在 GigaNet 的 Web 站点获取:  
<http://www.giganet.com>。请参考可在 GigaNet 的 Web 站点的支持页面上找到的下载和安装指南。

“cLAN 管理控制台”的安装可能导致节点重新启动。

5. 测试 GigaNet 硬件是否正在工作。为此,可执行下列操作:
  - a. 打开 GMC。(程序->GigaNet->cLAN 管理控制台)
  - b. 出现一个对话框,显示 LAN 中所有可访问的机器。按 **ESC**。
  - c. 从菜单条选择**控制台—>本地**。
  - d. 确认显示了群集中的所有成员并且它们全部是“活动的”。
  - e. 从菜单条选择**实用程序—>VI 吞吐量**。这将运行一个吞吐量测试,以检查数据确实是通过该硬件传送。
  - f. 以大写字体输入您希望在测试中使用的两个节点的计算机名。将本地节点标识为源节点。
  - g. 单击**开始测量**。应该看到数据以至少每秒 65 MB 的速率传送。
  - h. 单击**停止测量**来停止连接测试
  - i. 对群集中的其他节点重复此测试,以测量本地节点(源节点)和其他节点(接收节点)之间的吞吐量。

若连接测试有问题,参考 *GigaNet GNN1000 User Guide* 和 *GigaNet GNX5000 User Guide* 中的故障诊断部分。

有关如何安装和实现“DB2 通用数据库 Windows NT 版”的信息,参考 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings*。

## ServerNet Interconnect 的设置过程

设置此环境所需的硬件和软件的列表包括下列产品:

- ServerNet PCI 适配器驱动程序 (SPAD), 产品标识 T0089, 版本 1.3.5 或更高版本
- ServerNet 开关 1
- ServerNet 区域网络管理器 (SANMan), 产品标识 T0087, 版本 1.1.3 或更高版本。

以下是为确保 ServerNet Interconnect 可使用“DB2 通用数据库”所必需的步骤。每个步骤只是简单介绍了该步骤所需的操作: 此处不提供与每个步骤相关的所有详细信息。您还应参考每个步骤中引用的文档, 该文档提供了详细的指示信息和指导。

下面显示的步骤还假设在群集中最多使用六 (6) 个节点。若您要求使用更多的节点, 请与 ServerNet 联系。

以下是各步骤:

1. 安装 ServerNet 网络接口卡。有关安装说明, 请参考 *ServerNet-I Virtual Interface Software Release Document, (product ID N0031)*。
2. 安装 ServerNet 开关 1。有关安装说明, 请参考 *ServerNet-I Virtual Interface Software Release Document, (product ID N0031)*。
3. 卸装先前的 ServerNet 驱动程序。(若这是首次安装 ServerNet, 则跳过此步骤。)
  - a. 打开“网络”控制面板。(开始->设置->控制面板->网络)
  - b. 单击**适配器选项卡**。
  - c. 除去“Tandem ServerNet PCI 适配器驱动程序”。
  - d. 单击**服务选项卡**。
  - e. 除去 SANMan。
  - f. 单击**协议选项卡**。
  - g. 除去 Tandem ServerNet-I VI 协议。
4. 安装“Tandem ServerNet PCI 适配器驱动程序”。若要使用由 ServerNet 提供的软件 CD 进行安装, 以下是附加详细信息:
  - a. 打开“网络”控制面板。(开始->设置->控制面板->网络)
  - b. 单击**适配器选项卡**。(出现“适配器”屏幕。)
  - c. 确保将新的 ServerNet 驱动程序存放到单独的驱动器和/或目录中。然后, 从引用正确的驱动器和/或目录的命令提示符处, 输入“ernnn.exe -d”来启动自解压程序。(“ernnn.exe”是“工程发行版”的名称, 后接一个数 - ERnnn.EXE - 它标识要安装的 ServerNet 驱动程序的特定版本。)

- d. 切换至解压的文件所在的驱动器和 / 或目录。切换至 "Spad n.n.n \ Free" 子目录 (其中 "n.n.n" 是此产品的特定版本)。(若在故障诊断环境或开发环境中工作, 应切换至 "Spad n.n.n \ Checked" 子目录而不是 "Spad n.n.n \ Free" 子目录。)
  - e. 将 "oemsetup.multi\_node" 文件重命名为 "oemsetup.inf"。
  - f. 在 "适配器选项卡" 中选择**添加**。(出现 "选择适配器" 屏幕。)
  - g. 单击**从磁盘安装...**。(出现 "插入磁盘" 屏幕。)
  - h. 输入 oemsetup.inf 文件所在的驱动器和 / 或目录。
  - i. 确保此对话框显示 "Tandem ServerNet PCI 适配器驱动程序", 然后单击**确定**。确保适配器的列表现在显示 ServerNet 适配器。单击**关闭**。
  - j. 选择**是**以重新启动计算机 或选择**否**并继续安装 SANMan 和 "VI 软件开发者工具箱 (SDK)"
5. 安装 SANMan。若要使用由 ServerNet 提供的软件 CD 进行安装, 以下是附加详细信息:
- a. 打开 "网络" 控制面板。(开始->设置->控制面板->网络)
  - b. 单击**服务选项卡**。(出现 "服务" 屏幕。)
  - c. 确保将新的 ServerNet 驱动程序存放到单独的驱动器和 / 或目录中。然后, 从引用正确的驱动器和 / 或目录的命令提示符处, 输入 "ernnn.exe -d" 来启动自解压程序。("ernnn.exe" 是 "工程发行版" 的名称, 后接一个数 - ERnnn.EXE - 它标识要安装的 ServerNet 驱动程序的特定版本。)
  - d. 在 "服务选项卡" 中选择**添加**。(出现 "选择服务" 屏幕。)
  - e. 切换至解压的文件所在的驱动器和 / 或目录。切换至 "SANMan n.n.n \Free" 子目录 (其中 "n.n.n" 是此产品的特定版本)。(若在故障诊断环境或开发环境中工作, 应切换至 "SANMan n.n.n \ Checked" 子目录而不是 "SANMan n.n.n \ Free" 子目录。)
  - f. 查看 "开关" 上的灯, 确定 "开关" 为 X 还是 Y。一个灯显示 "X", 而另一个灯显示 "Y"。
  - g. 若是 "X 开关", 则选择 X=1 和 Y=0。确保所有电缆都与网卡上的 X 端口相连。
  - h. 若是 "Y 开关", 则选择 X=0 和 Y=1。确保所有电缆都与网卡上的 Y 端口相连。
  - i. 提供与当前机器上的网卡连接的开关的端口号。
  - j. 对所有六 (6) 个端口选择 "PC"。
6. 安装 "虚拟接口协议"。若要使用由 ServerNet 提供的软件 CD 进行安装, 以下是附加详细信息:

- a. 打开“网络”控制面板。(开始->设置->控制面板->网络)
  - b. 单击**协议选项卡**。(出现“网络协议”屏幕。)
  - c. 确保将新的 ServerNet 驱动程序存放到单独的驱动器和/或目录中。然后,从引用正确的驱动器和/或目录的命令提示符处,输入“ernnn.exe -d”来启动自解压程序。(“ernnn.exe”是“工程发行版”的名称,后接一个数 - ERnnn.EXE - 它标识要安装的 ServerNet 驱动程序的特定版本。)
  - d. 在“协议选项卡”中选择**添加**。(出现“选择网络协议”屏幕。)
  - e. 单击从**磁盘安装...**。(出现“插入磁盘”屏幕。)
  - f. 输入解压的文件所在的驱动器和/或目录。
7. 测试 ServerNet 硬件正在工作。没有可用的测试程序。只须使用 DB2 来测试 ServerNet 硬件。

若该硬件有问题,参考 *ServerNet-I Virtual Interface Software Release Document*, (product ID N0031) 以获得附加的故障诊断帮助。

有关如何安装和实现“DB2 通用数据库 Windows NT 版”的信息,参考 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings*。

### Synfinity Interconnect 的设置过程

设置此环境所需的硬件和软件的列表包括下列产品:

- Synfinity PCI 网络接口卡
- Synfinity 六端口开关
- Synfinity Interconnect 电缆
- Synfinity Cluster Manager Software, 版本 1.10。

确保 Synfinity Interconnect 可使用“DB2 通用数据库”所必需的步骤如下所示。每个步骤只是简单介绍了该步骤所需的操作: 此处不提供与每个步骤相关的所有详细信息。您还应参考每个步骤中引用的文档,该文档提供了详细的指示信息和指导。

每个 Synfinity 系统都附带有一张 Synfinity Cluster Manager Software 版本 1.10 CD-ROM。该 CD-ROM 包含设置 Synfinity Interconnect 所需的所有文档和软件。另外,该 CD-ROM 还包含 *Synfinity Cluster Manager Software User Guide*。

若安装了其他 VI 硬件、软件和协议,则可能必须在安装 Synfinity interconnect 之前除去所有它们。

在安装之后, Synfinity interconnect 被认为是外来硬件,不能通过 Windows NT 控制面板查看。

步骤摘要:

1. 安装适配卡
2. 安装 Synfinity Cluster Manager 软件
3. 安装开关和电缆
4. 测试 Interconnect

以下是各步骤:

1. 安装 Synfinity PCI 网络接口卡。有关安装说明, 请参考 *Synfinity Cluster Manager Software User Guide*。
2. 在与“开关”相连的节点上安装 Synfinity Cluster Manager Software。有关安装说明, 请参考 *Synfinity Cluster User Guide*。

您选择的节点将是“群集管理器”。这是唯一一个您必须要在上面安装 CD 中的软件的节点。

在安装之后, 应运行 Synfinity Cluster Manager 软件。“群集管理器”将为您提供一个群集计划, 并帮助您执行一个按步执行的指南来配置网络, 并对最佳的路由和布线选项提出建议。在将任何电缆与 Synfinity 开关和网卡相连之前, 应完成此步骤。作为计划过程的一部分, Cluster Manager 将使用群集计划来创建可安装的软件, 以供在其他节点上使用。这将包括其他节点上的卡的驱动程序软件。有关完整的详细信息, 参考 *Synfinity Cluster Manager Software User Guide*。

3. 安装 Synfinity “开关”和“电缆”。有关安装说明, 参考 *Synfinity Cluster User Guide*。
4. 测试 Synfinity 硬件是否工作。为此, 可执行下列操作:
  - a. 在群集中的任何系统上, 在 Windows NT 中打开“命令提示”窗口。
  - b. 将目录切换至装入 Synfinity Cluster Manager 软件的“utils”子目录。
  - c. 输入“vitest”并记下显示的节点号。
  - d. 移至群集中的任何其他系统, 打开“命令提示”窗口。
  - e. 将目录切换至这个另一系统上装入 Synfinity Cluster Manager 软件的位置下面的“utils”子目录。
  - f. 输入“vitest x”, 其中 x 是来自上面步骤 c 的节点号。
  - g. 应该会显示“CONNECTION GOOD”消息。
  - h. 若显示了“NO CONNECTION”消息, 则检查布线和硬件设置, 有关排除问题的进一步信息, 参考 *Synfinity Cluster Manager Software User Guide*。并请访问支持 Web 页面中的“Tech-tips”, 地址是: <http://www.fjst.com/>



有关如何安装和实现“DB2 通用数据库 Windows NT 版”的信息，参考 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings*。

## 启用 DB2 以使用 VI 运行

可在 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings* 中找到详细的安装信息。

在完成 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings* 中描述的 DB2 的安装之后，设置下列 DB2 注册表变量并对实例中的每个数据库分区执行下列任务：

### 1. 设置 DB2\_VI\_ENABLE=ON

使用 **db2set** 命令来修改注册表变量的值。在实例中的所有数据库分区服务器上使用 **db2\_all** 命令来运行 **db2set** 命令。必须使用“管理员”组成员的用户帐户来登录，才能执行 **db2\_all** 命令。

在下列示例中，将 ; 字符放置在双引号里面，以允许在实例中的所有数据库分区服务器上并发地执行请求：

```
db2_all ";db2set DB2_VI_ENABLE=ON"
```

有关 **db2\_all** 命令的更多信息，参见《管理指南：实现》中的“对多个数据库分区服务器发出命令”。

### 2. 设置 DB2\_VI\_DEVICE=nic0

例如：

```
db2_all ";db2set DB2_VI_DEVICE=nic0"
```

**注：**对于 Synfinity Interconnect，此变量应设置为 **DB2\_VI\_DEVICE=VINIC**。设备名 (VINIC) 必须是大写的。

### 3. 设置 DB2\_VI\_VIPL=vip1.dll

例如：

```
db2_all ";db2set DB2_VI_VIPL=vip1.dll"
```

**注：**在示例中所用的值是注册表变量的缺省值。有关注册表变量的更多信息，参见《管理指南：性能》。

### 4. 在分区实例上输入 db2start。

### 5. 复查 db2diag.log 文件。对每个分区都应有一条消息，声明“已启用 VI”。

### 6. 可能需要更新“快速通信管理器”(FCM) 配置参数。万一因为涉及 FCM 的资源约束而遇到问题，则应提高 FCM 配置参数的值。若正在从另一已提高 FCM 配置参数值的高速互连环境转移，则可能需要降低这些值。并且，在 Windows

| NT上，可能需要设置 DB2NTMEMSIZE 注册表变量来覆盖 DB2 缺省值。有  
| 关注册表变量的更多信息，参考 《管理指南：性能》。

## 附录J. “轻量级目录访问协议” (LDAP) 的目录服务

“轻量级目录访问协议” (LDAP) 是一个业界标准的访问目录服务的方法。目录服务是一个关于分布式环境中的多个系统和资源的资源信息的储存库；它提供对这些资源的客户机和服务器访问。每个数据库服务器实例都将它的存在情况发布给 LDAP 服务器，并在创建数据库时向 LDAP 目录提供数据库信息。当客户机与数据库连接后，可从 LDAP 目录检索服务器的目录信息。不再要求每个客户机将目录信息以本地方式存储在每台机器上。客户机应用程序搜索 LDAP 目录，查找连接数据库所需的信息。

由于存在高速缓存机制，因此客户机仅在其本地目录编目中搜索 LDAP 目录一次。一旦检索到信息，就将它存储或高速缓存在本地机器上。对同一信息的后续访问依赖于 *dir\_cache* 数据库管理器配置参数和 DB2LDAPCACHE 注册表变量的值。

- 若 DB2LDAPCACHE=NO 且 *dir\_cache*=NO，则总是从 LDAP 读取信息。
- 若 DB2LDAPCACHE=NO 且 *dir\_cache*=YES，则从 LDAP 读取一次信息，并将该信息插入 DB2 高速缓存。
- 若 DB2LDAPCACHE=YES 或未设置，且在本地高速缓存中找不到必需的信息，则从 LDAP 目录读取该信息，并刷新本地高速缓存。

注：DB2LDAPCACHE 注册表变量仅适用于数据库和节点目录。

### 支持 LDAP 客户机和服务器配置

下表概述受支持的 LDAP 客户机和服务器配置：

表 26. 受支持的 LDAP 客户机和服务器配置

	IBM SecureWay Directory V3.1 和 V3.1.1	Microsoft 活动目录
IBM LDAP 客户机	受支持	不受支持
Microsoft LDAP/ADSI 客户机	受支持	受支持

IBM SecureWay Directory 版本 3.1 是可用于 Windows NT、AIX 和 Solaris 的 LDAP 版本 3 服务器。在 AIX 和 AS/400 上，SecureWay Directory 与“OS/390 安全性服务器”作为基本操作系统的一部分交付。

DB2 在 AIX、Solaris、Windows NT 和 Windows 98 上支持 IBM LDAP 客户机。

“Microsoft 活动目录”是一个 LDAP 版本 3 服务器，可作为“Windows 2000 服务器”操作系统的一部分获得。

下列 Microsoft 产品中包括“Microsoft LDAP 客户机”支持：

- Outlook 98、Outlook 2000 或 Outlook Express

注：Outlook Express 作为 Microsoft Internet Explorer 的一部分安装。

- Exchange Server
- Windows NT 服务器服务包 4
- Windows 98 第二版
- Windows 2000

“活动目录服务接口” (ADSI) 组件中也包含“Microsoft LDAP 客户机”支持。可从以下站点下载最新版本的 ADSI：

<http://www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/adsilinks.asp>

当在 Windows 98、Windows NT 或 Windows 2000 操作系统上运行时，DB2 支持使用 IBM LDAP 客户机或 Microsoft LDAP 客户机来访问“IBM SecureWay 目录服务器”。若 Microsoft LDAP 客户机不可用，则 DB2 尝试使用 IBM LDAP 客户机。要显式地选择 IBM LDAP 客户机，使用 **db2set** 命令来将 DB2LDAP\_CLIENT\_PROVIDER 注册表变量设置为“IBM”。

## 对 Windows 2000 活动目录的支持

DB2 按如下方式使用“活动目录”：

1. DB2 数据库服务器作为 `ibm_db2Node` 对象发布在“活动目录”中。  
`ibm_db2Node` 对象类是 `ServiceConnectionPoint (SCP)` 对象类的子类。每个 `ibm_db2Node` 对象都包含用来允许客户机应用程序连接 DB2 数据库服务器的协议配置信息。创建新数据库时，在“活动目录”中，该数据库作为 `ibm_db2Database` 对象发布在 `ibm_db2Node` 对象下面。
2. 当连接远程数据库时，DB2 客户机通过 LDAP 接口查询“活动目录”，以查找 `ibm_db2Database` 对象。用于连接数据库服务器的协议通信（绑定信息）从特定 `ibm_db2Node` 对象获取，`ibm_db2Database` 对象便在该对象下面创建。

---

## 配置 DB2，以使用活动目录

为了访问“Microsoft 活动目录”，确保符合下列条件：

1. 运行 DB2 的机器必须属于 Windows 2000 域。

2. 已安装 Microsoft LDAP 客户机。Microsoft LDAP 客户机是 Windows 2000 操作系统的一部分。对于 Windows 98 或 Windows NT，需要验证系统目录上是否存在 wldap32.dll。
3. 启用 LDAP 支持。对于 Windows 2000，LDAP 支持是通过安装程序启用的。对于 Windows 98/NT，必须通过使用 **db2set** 命令将 DB2\_ENABLE\_LDAP 注册表变量设置为 "YES" 来显式地启用 LDAP。
4. 当运行 DB2 来从“活动目录”读取信息时，登录至域用户帐户。

---

## 在 IBM LDAP 环境中配置 DB2

在可以在 IBM LDAP 环境中使用 DB2 之前，必须在每台机器上配置下列各项：

- 启用 LDAP 支持。对于 Windows 2000，LDAP 支持是通过安装程序启用的。对于 Windows 98/NT，必须通过使用 **db2set** 命令将 DB2\_ENABLE\_LDAP 注册表变量设置为 "YES" 来显式地启用 LDAP。
- LDAP 服务器的 TCP/IP 主机名和端口号。这些值可以在无人照管安装期间使用 DB2LDAPHOST 响应关键字输入，也可以在以后使用 DB2SET 命令手工设置它们：

```
db2set DB2LDAPHOST=<hostname[:port]>
```

其中，hostname 是 LDAP 服务器的 TCP/IP 主机名，而 [:port] 是端口号。若未指定端口号，则 DB2 将使用缺省 LDAP 端口 (389)。

DB2 对象位于 LDAP 基本专有名称 (baseDN) 中。若是在使用 IBM SecureWay LDAP 目录服务器版本 3.1，则不必配置基本专有名称，因为 DB2 可以动态地从服务器获取此信息。然而，若是在使用“IBM eNetwork 目录服务器”版本 2.1，则必须使用 DB2SET 命令在每台机器上配置 LDAP 基本专有名称：

```
db2set DB2LDAP_BASEDN=<baseDN>
```

其中，baseDN 是 LDAP 服务器上定义的 LDAP 后缀的名称。此 LDAP 后缀用来包含 DB2 对象。

- LDAP 用户的专有名称 (DN) 和密码。仅当您计划使用 LDAP 来存储 DB2 用户特定信息时，这些信息才是必需的。

## 创建 LDAP 用户

DB2 支持在用户级设置 DB2 注册表变量和 CLI 配置。（在 AIX 和 Solaris 平台上，此功能不可用。）用户级支持在多用户环境中提供了用户特定设置。

“Windows NT 终端服务器”便是一个示例，每个登录用户都可以定制他或她自己的环境，而不会干扰系统环境或另一用户的环境。

当使用 IBM LDAP 目录时，必须定义 LDAP 用户，然后才能在 LDAP 中存储用户级信息。可以使用下列方法之一创建 LDAP 用户：

- 创建一个 LDIF 文件来包含用户对象的所有属性，然后运行 LDIF 导入实用程序来将该对象导入到 LDAP 目录中。用于 IBM LDAP 服务器的 LDIF 实用程序是 "LDIF2DB"。
- 使用“目录管理工具”(DMT)（仅可用于“IBM SecureWay LDAP 目录服务器”版本 3.1）来创建用户对象。

包含人员对象属性的 LDIF 文件外观类似于：

文件名: newuser.ldif

```
dn: cn=Mary Burnnet, ou=DB2 UDB Development, ou=Toronto, o=ibm, c=ca
objectclass: ePerson
cn: Mary Burnnet
sn: Burnnet
uid: mburnnet
userPassword: password
telephonenumber: 1-416-123-4567
facsimiletelephonenumber: 1-416-123-4568
title: Software Developer
```

以下是使用 IBM LDIF 导入实用程序导入 LDIF 文件的 LDIF 命令的一个示例：

```
LDIF2DB -i newuser.ldif
```

注：

1. 必须从 LDAP 服务器运行 LDIF2DB 命令。
2. 必须将必需的访问权 (ACL) 授予 LDAP 用户对象，以使 LDAP 用户可以添加、删除、读取和写入他自己的对象。要授予用户对象的 ACL，使用“LDAP 目录服务器 Web 管理”工具。

## 为 DB2 应用程序配置 LDAP 用户

使用 IBM LDAP 客户机时，在运行 DB2 之前，必须配置当前登录用户的 LDAP 用户专有名称 (DN) 和密码。这可以使用 db2ldcfg 实用程序完成：

```
db2ldcfg -u <userDN> -w <password> -> set the user's DN and password
-r -> clear the user's DN and password
```

例如：

```
db2ldcfg -u "cn=Mary Burnnet,ou=DB2 UDB Development,ou=Toronto,o=ibm,c=ca"
-w password
```

---

## 在安装之后注册 DB2 服务器

必须在 LDAP 中注册每个 DB2 服务器实例，以便发布客户机应用程序用来连接 DB2 服务器实例的协议配置信息。当注册一个数据库服务器实例时，需要指定一个节点名。该节点名由客户机应用程序在连接服务器时使用。可使用 CATALOG LDAP NODE 命令编目 LDAP 节点的另一个别名。

**注：**若是在 Windows 2000 域环境中工作，则安装期间，将在“活动目录”中用下列信息自动注册 DB2 服务器实例：

```
nodename: TCP/IP hostname
protocol type: TCP/IP
```

若 TCP/IP 主机名长于 8 个字符，则它将被截断为 8 个字符。

REGISTER 命令如下所示：

```
db2 register db2 server in ldap
as <ldap_node_name>
protocol tcpip
```

protocol 子句指定与此数据库服务器连接时要使用的通信协议。

当为包含多台物理机器的“DB2 通用数据库 EEE”创建实例时，必须对每台机器调用一次 REGISTER 命令。使用 *rah* 命令在所有机器上发出 REGISTER 命令。

**注：**每台机器不能使用相同的 ldap\_node\_name，因为在 LDAP 中该名称必须是唯一的。应用每台机器的主机名替换 REGISTER 命令中的 ldap\_node\_name。例如：

```
rah ">DB2 REGISTER DB2 SERVER IN LDAP AS <> PROTOCOL TCP/IP"
```

"<>" 被替换为运行 rah 命令的每台机器的主机名。在有多个“DB2 通用数据库 EEE”实例这种很少见的情况下，可将实例与主机索引的组合用作 rah 命令中的节点名。

可以对远程 DB2 服务器发出 REGISTER 命令。为此，当注册远程服务器时，必须指定远程计算机名、实例名和协议配置参数。可以按如下所示使用该命令：

```
db2 register db2 server in ldap
as <ldap_node_name>
protocol tcpip
hostname <host_name>
svcname <tcpip_service_name>
remote <remote_computer_name>
instance <instance_name>
```

以下是计算机名的约定：

- 若配置了 TCP/IP，则计算机名必须与 TCP/IP 主机名相同。
- 若配置了 APPN，则使用伙伴 LU 名作为计算机名。

当在一个高可用性或故障转移的环境中运行，并使用 TCP/IP 作为通信协议时，必须使用群集 IP 地址。使用群集 IP 地址允许客户机连接到任何一台机器上的服务器，而不必为每台机器编目独立的 TCP/IP 节点。使用 `hostname` 子句指定群集 IP 地址，如下所示：

```
db2 register db2 server in ldap
  as <ldap_node_name>
  protocol tcpip
  hostname n.nn.nn.nn
```

其中 `n.nn.nn.nn` 是群集 IP 地址。

有关 REGISTER 命令的其他信息，参考 *Command Reference*。

---

## 更新 DB2 服务器的协议信息

LDAP 中的 DB2 服务器信息必须保持为最新信息。例如，更改协议配置参数或服务网络地址要求更新 LDAP。

要更新本地机器上的 LDAP 中的 DB2 服务器，使用以下命令：

```
db2 update ldap ...
```

可更新的协议配置参数的示例包括：

- TCP/IP 主机名和服务名或端口号参数。
- IPX 地址。
- APPC 协议信息，如 TP 名、伙伴 LU 或方式。
- NetBIOS 工作站名。

要更新远程 DB2 服务器协议配置参数，应使用带 `node` 子句的 UPDATE LDAP 命令：

```
db2 update ldap
  node <node_name>
  hostname <host_name>
  svcname <tcpip_service_name>
```

有关 UPDATE LDAP 命令的更多信息，参考 *Command Reference*。



---

## 编目节点别名以便连接

当在 LDAP 中注册服务器时，必须指定 DB2 服务器的节点名。应用程序使用该节点名连接数据库服务器。若需要另一个节点名，如在将节点名硬编码进应用程序中时，可使用 CATALOG LDAP NODE 命令进行更改。该命令类似于：

```
db2 catalog ldap node <ldap_node_name>
as <new_alias_name>
```

要取消编目 LDAP 节点，使用 UNCATALOG LDAP NODE COMMAND 命令。该命令类似于：

```
db2 uncatalog ldap node <ldap_node_name>
```

---

## 注销 DB2 服务器

从 LDAP 中注销一个实例，同时也除去了所有引用该实例的节点、别名、对象和数据库对象。

要在本地机器或远程机器上注销 DB2 服务器，要求为服务器指定 LDAP 节点名：

```
db2 deregister db2 server in ldap
node <node_name>
```

当注销了 DB2 服务器时，引用 DB2 服务器的同一个实例的任何 LDAP 节点项和 LDAP 数据库项也将被取消编目。

---

## 注册数据库

在实例中创建数据库期间，会在 LDAP 中自动注册数据库。注册允许远程客户机与数据库连接，而不必在客户机上编目该数据库和节点。当客户机试图连接数据库时，若本地机器上的数据库目录中不存在该数据库，则搜索 LDAP 目录。

若在 LDAP 目录中已存在该名称，仍然会在本地机器上创建该数据库，但是会返回一个警告消息，说明在 LDAP 目录中发生名称冲突。因此，可在 LDAP 目录中手工编目数据库。用户可使用 CATALOG LDAP DATABASE 命令在 LDAP 中注册远程服务器上的数据库。当注册远程数据库时，指定表示远程数据库服务器的 LDAP 节点的名称。在注册数据库之前，必须使用 REGISTER DB2 SERVER IN LDAP 命令在 LDAP 中注册远程数据库服务器。

要在 LDAP 中手工注册数据库，使用 CATALOG LDAP DATABASE 命令：

```
db2 catalog ldap database <dbname>
at node <node_name>
with "My LDAP database"
```

---

## 连接远程服务器

在 LDAP 环境中，可在 ATTACH 命令中使用 LDAP 节点名连接远程数据库服务器：

```
db2 attach to <ldap_node_name>
```

当客户机应用程序首次连接节点或连接数据库时，由于该节点不在本地节点目录中，DB2 就会搜索 LDAP 目录寻找目标节点项。若在 LDAP 目录中找到该项，就会检索远程服务器的协议信息。若连接的是数据库，且在 LDAP 目录中找到该项，则还检索数据库信息。使用该信息，DB2 可自动在本地机器上编目数据库项和节点项。客户机应用程序下次连接相同的节点或数据库时，可使用本地数据库目录中的信息，而不必搜索 LDAP 目录。

具体的说，由于存在高速缓存机制，因此客户机仅在其本地目录编目中搜索 LDAP 目录一次。一旦检索到信息，就将它存储或高速缓存在本地机器上。对同一信息的后续访问依赖于 *dir\_cache* 数据库管理器配置参数和 DB2LDAPCACHE 注册表变量的值。

- 若 DB2LDAPCACHE=NO 且 *dir\_cache*=NO，则总是从 LDAP 读取信息。
- 若 DB2LDAPCACHE=NO 且 *dir\_cache*=YES，则从 LDAP 读取一次信息，并将该信息插入 DB2 高速缓存。
- 若 DB2LDAPCACHE=YES 或未设置，且在本地高速缓存中找不到必需的信息，则从 LDAP 目录读取该信息，并刷新本地高速缓存。

**注：**LDAP 信息的高速缓存不适用于用户级 CLI 或 DB2 简要表注册表变量。并且，有一个用于数据库、节点和 DCS 目录的“内存中”高速缓存。然而，却没有用于节点目录这样的高速缓存。

---

## 注销数据库

当发生下列情况时，就自动从 LDAP 注销数据库：

- 删除数据库。
- 从 LDAP 中注销自己的实例。

可使用以下命令从 LDAP 中手工注销数据库：

```
db2 uncatalog ldap database <dbname>
```

---

## 刷新本地数据库和节点目录中的 LDAP 项

LDAP 信息可能会更改，因此有必要刷新本地目录和节点目录中的 LDAP 项。本地数据库和节点目录用来高速缓存 LDAP 中的项。

具体的说，由于存在高速缓存机制，因此客户机仅在其本地目录编目中搜索 LDAP 目录一次。一旦检索到信息，就将它存储或高速缓存在本地机器上。对同一信息的后续访问依赖于 *dir\_cache* 数据库管理器配置参数和 DB2LDAPCACHE 注册表变量的值。

- 若 DB2LDAPCACHE=NO 且 *dir\_cache*=NO，则总是从 LDAP 读取信息。
- 若 DB2LDAPCACHE=NO 且 *dir\_cache*=YES，则从 LDAP 读取一次信息，并将该信息插入 DB2 高速缓存。
- 若 DB2LDAPCACHE=YES 或未设置，且在本地高速缓存中找不到必需的信息，则从 LDAP 目录读取该信息，并刷新本地高速缓存。

**注：**LDAP 信息的高速缓存不适用于用户级 CLI 或 DB2 简要表注册表变量。并且，有一个用于数据库、节点和 DCS 目录的“内存中”高速缓存。然而，却没有用于节点目录这样的高速缓存。

要刷新引用 LDAP 资源的数据库项，使用以下命令：

```
db2 refresh ldap database directory
```

要刷新本地机器上引用 LDAP 资源的节点项，使用以下命令：

```
db2 refresh ldap node directory
```

在刷新过程中，会除去本地数据库和节点目录中保存的所有 LDAP 项。下次应用程序访问该数据库或节点时，它将直接从 LDAP 中读取该信息并在本地数据库或节点目录中生成新项。

要确保用一种快速的方法完成刷新，可以这样做：

- 安排定期刷新。
- 在系统启动期间运行 REFRESH 命令。
- 在所有客户机上使用一个可用的管理软件包来调用 REFRESH 命令。
- 设置 DB2LDAPCACHE="NO" 以避免将 LDAP 信息高速缓存在数据库、节点和 DCS 目录中。

---

## 搜索

DB2 在 Windows 2000 环境中搜索当前 LDAP 目录分区或当前“活动目录”域。在有多个 LDAP 目录分区或域的环境中，可设置搜索范围。例如，若在当前的分区或域中找不到信息，可请求自动搜索所有其他分区或域。另一方面，可限制搜索范围仅搜索本地机器。

搜索范围由 DB2 简要表注册表变量 DB2LDAP\_SEARCH\_SCOPE 控制。要设置 LDAP 中的全局级搜索范围值，在 `db2set` 命令中使用 `"-gl"` 选项，该选项表示“LDAP 中的全局”：

```
db2set -gl db2ldap_search_scope=<value>
```

可能的值包含：“local”、“domain”或“global”。缺省值是“domain”，它将搜索范围限制在当前目录分区。可在 LDAP 中设置搜索范围，以用作整个企业的缺省搜索范围设置。例如，可能需要在创建新的数据库后将搜索范围初始化为“global”。这就允许所有的客户机搜索所有其他分区或域，以查找在特定的分区或域中定义的数据库。任何一台客户机首次连接之后，一旦在自己的机器上记录了该项，搜索范围可更改为“local”。一旦更改为“local”，每台客户机将不搜索任何一个分区或域。

**注：**DB2 简要表注册表变量 DB2LDAP\_SEARCH\_SCOPE 是唯一支持在 LDAP 中的全局级设置该变量的注册表变量。

---

## 注册主机数据库

当在 LDAP 中注册主机数据库时，有两个可能的配置：

- 直接连接主机数据库；或者
- 通过网关连接主机数据库。

在第一种情况下，用户将在 LDAP 中注册主机服务器，然后在 LDAP 中编目主机数据库，并指定主机服务器的节点名。在第二种情况下，用户应在 LDAP 中注册网关服务器，然后在 LDAP 中编目主机数据库，并指定网关服务器的节点名。

作为显示上述两种情况的示例，请考虑：假设有一个名为 NIAGARA\_FALLS 的主机数据库。它可接受使用 APPN 和 TCP/IP 的入局连接。若客户机因没有 DB2 Connect 而不能直接连接主机，则它将使用名为“goto@niagara”的网关进行连接。

需要完成下列步骤：

1. 在 LDAP 中为 APPN 连通性注册主机数据库服务器。REMOTE 和 INSTANCE 子句是任意的。将 NODETYPE 子句设置为“DCS”，指示这是主机数据库服务器。

```
db2 register ldap as nfappn appn network CAIBMOML partnerlu NFLU
mode IBMRDB remote mvssys instance msvinst nodetype dcs
```

2. 在 LDAP 中为 TCP/IP 连通性注册主机数据库服务器。服务器的 TCP/IP 主机名是“myhost”，端口号是“446”。与步骤 1 类似，将 NODETYPE 子句设置为“DCS”，以指示这是主机数据库服务器。

```
db2 register ldap as nftcpip tcpip hostname myhost svcname 446
remote mvssys instance mvsinst nodetype dcs
```

3. 在 LDAP 中为 TCP/IP 连通性注册 DB2 Connect 网关服务器。网关服务器的 TCP/IP 主机名是 "niagara"，端口号是 "50000"。

```
db2 register ldap as whasf tcpip hostname niagara svcname 50000
remote niagara instance goto nodetype server
```

4. 使用 TCP/IP 连通性来在 LDAP 中编目主机数据库。主机数据库名是 "NIAGARA\_FALLS"，数据库别名是 "nftcpip"。使用 GWNODE 子句来指定 DB2 Connect 网关服务器的节点名。

```
db2 catalog ldap database NIAGARA_FALLS as nftcpip at node nftcpip
gwnode whasf authentication dcs
```

5. 使用 APPN 连通性来在 LDAP 中编目主机数据库。

```
db2 catalog ldap database NIAGARA_FALLS as nfappn at node nfappn
gwnode whasf authentication dcs
```

在完成上面显示的注册和编目之后，若要使用 TCPIP 连接主机，则连接 "nftcpip"。若要使用 APPN 连接主机，则连接 "nfappn"。若客户机工作站上没有 DB2 Connect，则该连接将使用 TCPIP 来通过网关，并且，从那里，根据您使用的是 "nftcpip" 还是 "nfappn"，它将分别使用 TCP/IP 或 APPN 来与主机连接。

之后，通常可在 LDAP 中手工配置主机数据库信息，以便每台客户机不必在自己的机器上以本地方式手工编目该数据库和节点。该过程如下：

1. 在 LDAP 中注册主机数据库服务器。在 REGISTER 命令中，必须使用 REMOTE、INSTANCE 和 NODETYPE 子句分别指定远程计算机名、实例名和主机数据库服务器的节点类型。REMOTE 子句可以设置为主机名或主机服务器的 LU 名。INSTANCE 子句可以设置为任何长度不超过 8 个字符的字符串。（例如，可以将实例名设置为 "DB2"。）必须将 NODE TYPE 子句设置为 "DCS"，以指示这是主机数据库服务器。
2. 使用 CATALOG LDAP DATABASE 命令在 LDAP 中注册主机数据库。可使用 PARMS 参数指定任何附加的 DRDA 子句。应将数据库认证类型设置为 "DCS"。

---

## 在用户级设置 DB2 注册表变量

在 LDAP 环境中，可在用户级设置 DB2 简要表注册表变量，这样可允许用户定制自己的 DB2 环境。要在用户级设置 DB2 简要表注册表变量，使用 -ul 选项：

```
db2set -ul <variable>=<value>
```

注：这在 AIX 和 Solaris 上不受支持。

DB2 有高速缓存机制。将用户级 DB2 简要表注册表变量高速缓存到本地机器上。若指定了 `-u1` 参数, DB2 将始终从高速缓存中读取 DB2 注册表变量。当发生下列情况时, 会刷新高速缓存:

- 更新或复位用户级 DB2 注册表变量。
- 在用户级刷新 LDAP 简要表变量的命令是:

```
db2set -ur
```

---

## 安装完成后启用 LDAP 支持

要在安装过程完成后启用 LDAP 支持, 在每台机器上使用以下过程:

- 安装 LDAP 支持二进制文件。运行安装程序并从定制安装中选择 “LDAP 目录维护” 支持。安装程序安装二进制文件并将 DB2 简要表注册表变量 `DB2_ENABLE_LDAP` 设置为 “YES”。

**注:** 对于 Windows 98/NT 和 UNIX 平台, 必须通过使用 `db2set` 命令将 `DB2_ENABLE_LDAP` 注册表变量设置为 “YES” 来显式地启用 LDAP。

- (仅在 UNIX 平台上) 使用以下命令说明 LDAP 服务器的 TCP/IP 主机名和 (可选的) 端口号:

```
db2set DB2LDAPHOST=<base_domain_name>[:port_number]
```

其中, `base_domain_name` 是 LDAP 服务器的 TCP/IP 主机名, 而 `[:port]` 是端口号。若未指定端口号, 则 DB2 将使用缺省 LDAP 端口 (389)。

DB2 对象位于 LDAP 基本专有名称 (baseDN) 中。若是在使用 IBM SecureWay LDAP 目录服务器版本 3.1, 则不必配置基本专有名称, 因为 DB2 可以动态地从服务器获取此信息。然而, 若是在使用 “IBM eNetwork 目录服务器” 版本 2.1, 则必须使用 `DB2SET` 命令在每台机器上配置 LDAP 基本专有名称:

```
db2set DB2LDAP_BASEDN=<baseDN>
```

其中, `baseDN` 是 LDAP 服务器上定义的 LDAP 后缀的名称。此 LDAP 后缀用来包含 DB2 对象。

- 使用 `REGISTER LDAP AS` 命令在 LDAP 中注册 DB2 服务器的当前实例。例如:

```
db2 register ldap as <node-name> protocol tcpip
```

- 若有要在 LDAP 中注册的数据库, 运行 `CATALOG LDAP DATABASE` 命令。例如:

```
db2 catalog ldap database <dbname> as <alias_dbname>
```

- 输入 LDAP 用户的专有名称 (DN) 和密码。仅当您计划使用 LDAP 来存储 DB2 用户特定信息时, 这些信息才是必需的。

---

## 禁用 LDAP 支持

要禁用 LDAP 支持，使用下列过程：

- 对每个 DB2 服务器实例，注销 LDAP 中的 DB2 服务器：

```
db2 deregister db2 server in ldap node <nodename>
```
- 将 DB2 简要表注册表变量 DB2\_ENABLE\_LDAP 设置为 "NO"。

---

## LDAP 支持和 DB2 Connect

若 DB2 Connect 网关上有 LDAP 支持可用，且在网关数据库目录中找不到数据库，则 DB2 将查找 LDAP，并尝试保留找到的信息。

---

## 安全性注意事项

在访问 LDAP 目录中的信息之前，LDAP 服务器要认证应用程序或用户。认证过程称为与 LDAP 服务器绑定。

对存储在 LDAP 目录中的信息进行访问控制很重要，这样可以防止匿名用户添加、删除或修改信息。

缺省情况下，将继承访问控制，并可在容器级应用。当创建了一个新对象时，它就继承父对象的相同的安全性属性。可使用 LDAP 服务器的管理工具来定义容器对象的访问控制。

缺省情况下，按如下所示定义访问控制：

- 对于 LDAP 中的数据库项和节点项，每个人（或任何匿名用户）都有读访问权。只有目录管理员以及对象的所有者或创建者具有读 / 写访问权。
- 对于用户简要表，简要表所有者和目录管理员具有读 / 写访问权。若一个用户没有“目录管理员”权限，则不能访问另一个用户的简要表。

**注：**授权检查总是由 LDAP 服务器执行，而不是由 DB2 执行。LDAP 授权检查与 DB2 授权无关。具有 SYSADM 权限的帐户或授权标识也许不能访问 LDAP 目录。

当运行 LDAP 命令或 API 时，若未指定绑定“专有名称” (bindDN) 和密码，则 DB2 使用缺省凭证绑定至 LDAP 服务器，该凭证可能没有足够的权限来执行请求的命令，将返回错误。

可使用 DB2 命令或 API 的 USER 和 PASSWORD 子句显式地指定用户的 bindDN 和密码。有关 DB2 命令的更多信息，参考 *Command Reference*，有关 DB2 API 的更多信息，参考 *Administrative API Reference*。

## Windows 2000 活动目录的安全性注意事项

DB2 数据库和节点对象在特定机器的计算机对象下创建，在该机器上，“活动目录”中安装了 DB2 服务器。要在“活动目录”中注册数据库服务器或编目数据库，您需要具有足够的权限来创建和 / 或更新计算机对象下的对象。

在缺省情况下，计算机对象下的对象可由任何经认证的用户读取，并可由管理员（属于“管理员”、“域管理员”和“企业管理员”组的用户）更新。要授予特定用户或组的访问权，使用“活动目录用户和计算机管理控制台” (MMC)，如下所示：

1. 启动活动目录用户和计算机管理工具  
(开始 —> 程序 —> 管理工具 —> 活动目录用户和计算机)
2. 在查看下面，选择高级特性
3. 选择计算机容器
4. 右键单击表示安装有 DB2 的服务器的计算机对象，并选择特性
5. 选择安全性选项卡，然后向指定的用户或组添加必需的访问权

用户级的 DB2 注册表变量和 CLI 设置是在用户对象下面的 DB2 特性对象中维护的。要在用户级设置 DB2 注册表变量或 CLI 设置，用户需要具有足够的权限来在“用户”对象下面创建对象。

在缺省情况下，只有管理员才有权在“用户”对象下面创建对象。要授权用户在用户级设置 DB2 注册表变量或 CLI 设置，使用活动目录用户和计算机“管理控制台” (MMC)，如下所示：

1. 启动活动目录用户和计算机管理工具  
(开始 —> 程序 —> 管理工具 —> 活动目录用户和计算机)
2. 在“用户”容器下面选择该用户对象
3. 右键单击该用户对象并选择特性
4. 选择安全性选项卡
5. 使用添加“按钮”将用户名添加至列表
6. 授予“写入”和“创建所有子对象”访问权
7. 使用“高级”设置，设置许可权以应用到“此对象和所有子对象”
8. 选择复选框“允许父代的可继承许可权传播至此对象”



---

## 扩展具有 DB2 对象类和属性的目录模式

“LDAP 目录模式”定义了存储在 LDAP 目录项中的信息的对象类和属性。对象类由一组必要的和可选的属性组成。LDAP 目录中的每一项都有一个相关的对象类。

在 DB2 将信息存储到 LDAP 中之前，LDAP 服务器的“目录模式”必须包含 DB2 使用的对象类和属性。向基本模式中添加新对象类和属性的过程称作扩展“目录模式”。

**注：**若使用的是“IBM SecureWay LDAP 目录”版本 3.1，则 DB2 需要的所有对象类和属性都包括在基本模式中。不必扩展具有 DB2 对象类和属性的基本模式。

### 扩展“IBM eNetwork 目录”版本 2.1 的目录模式

当使用“IBM eNetwork 目录”版本 2.1 时，必须扩展具有 DB2 使用的对象类和属性的基本模式。

使用下列步骤扩展“IBM eNetwork 目录”版本 2.1 的基本模式：

1. 将 DB2 属性定义文件 `db2.at` 和对象类定义文件 `db2.oc` 复制到包含系统属性定义文件 `slapd.at.conf` 和对象类定义文件 `slapd.oc.conf` 的相同目录下。DB2 属性和对象类定义文件位于 `sql1lib` 子目录下的 `cfg` 子目录中。系统属性和对象类定义文件位于 `%LDAPHome%` 子目录的 `etc` 子目录中。
2. 复查 DB2 属性和对象类定义文件。注释掉在当前的“LDAP 目录模式”中已定义的所有对象类和属性。
3. 在 `slapd.oc.conf` 文件尾添加一行，如下所示：  

```
include db2.oc
```
4. 在 `slapd.at.conf` 文件尾添加一行，如下所示：  

```
include db2.at
```
5. 重新启动 LDAP 服务器。

### 扩展 Windows 2000 活动目录的目录模式

在 DB2 可以在“Windows 2000 活动目录”中存储信息之前，需要扩展目录模式以包括新的 DB2 对象类和属性。向目录模式中添加新对象类和属性的过程称作“模式扩展”。

在任何作为 Windows 2000 域一部分的机器上首次安装 DB2 之前，必须运行“DB2 模式安装”程序 `db2schex` 扩展“活动目录”的模式。

**db2schex** 程序可在产品 CD-ROM 上找到。此程序在 CD-ROM 上的位置在 db2 目录的 common 子目录下面。例如:

```
x:\db2\common
```

其中, x: 是 CD-ROM 驱动器。

按如下所示使用该命令:

```
db2schex
```

此命令还有其他可选子句:

- **-b UserDN**  
指定用户“专有名称”。
- **-w Password**  
指定绑定密码。
- **-u**  
卸载模式。
- **-k**  
强制卸载继续, 忽略错误。

**注:**

1. 若未指定 UserDN 和密码, 则 **db2schex** 作为当前已登录的用户绑定。
2. 可指定 userDN 子句作为 Windows NT 用户名。
3. 要更新模式, 您必须是“模式管理员”组的成员, 或已被授予更新模式的权限。

示例:

- 要安装 DB2 模式:

```
db2schex
```

- 要安装 DB2 模式并指定绑定 DN 和密码:

```
db2schex -b "cn=A Name,dc=toronto1,dc=ibm,dc=com"  
-w password
```

或者,

```
db2schex -b Administrator -w password
```

- 要卸载 DB2 模式:

```
db2schex -u
```

- 要卸载 DB2 模式并忽略错误:

```
db2schex -u -k
```

“活动目录”的“DB2 模式安装”程序执行下列任务:

注:

1. 检测哪一个服务器是“模式主机”
2. 绑定至作为“模式主机”的“域控制器”
3. 确保该用户有足够的权限来向该模式添加类和属性
4. 确保模式主机可写(即, 除去了注册表中的安全互锁装置)
5. 创建所有新属性
6. 创建所有新对象类
7. 检测错误, 若发生错误, 则该程序将回滚对模式的任何更改。

## Windows 2000 活动目录中的 DB2 对象

DB2 在“活动目录”中的两个位置中创建对象:

1. 在安装有“DB2 服务器”的机器的计算机对象下面创建 DB2 数据库和节点对象。对于不属于 Windows NT 域的 DB2 服务器, 在“系统”容器下面创建 DB2 数据库和节点对象。
2. 用户级的 DB2 注册表变量和 CLI 设置存储在“用户”对象下面的 DB2 特性对象中。这些对象包含该用户的特定信息。

## DB2 使用的对象类和属性

下面的表描述了 DB2 使用的对象类:

表 27. *cimManagedElement*

类	<b>cimManagedElement</b>
活动目录 LDAP 显示名	不适用
活动目录公共名 (cn)	不适用
描述	提供“IBM 模式”中许多系统管理对象类的基类
SubClassOf	顶部
必需的属性	
可选属性	描述
类型	抽象
OID (对象标识符)	1.3.18.0.2.6.132
GUID (全局唯一标识符)	b3afd63f-5c5b-11d3-b818-002035559151

表 28. *cimSetting*

类	<b>cimSetting</b>
活动目录 LDAP 显示名	不适用
活动目录公共名 (cn)	不适用
描述	提供“IBM 模式”中的配置和设置的基类
SubClassOf	cimManagedElement
必需的属性	
可选属性	settingID
类型	抽象
OID (对象标识符)	1.3.18.0.2.6.131
GUID (全局唯一标识符)	b3afd64d-5c5b-11d3-b818-002035559151

表 29. *eProperty*

类	<b>eProperty</b>
活动目录 LDAP 显示名	ibm-eProperty
活动目录公共名 (cn)	ibm-eProperty
描述	用来指定用户首选项特性的任何应用程序特定设置
SubClassOf	cimSetting
必需的属性	
可选属性	propertyType cisPropertyType cisProperty cesPropertyType cesProperty binPropertyType binProperty
类型	结构
OID (对象标识符)	1.3.18.0.2.6.90
GUID (全局唯一标识符)	b3afd69c-5c5b-11d3-b818-002035559151

表 30. *DB2Node*

类	<b>DB2Node</b>
活动目录 LDAP 显示名	ibm-db2Node
活动目录公共名 (cn)	ibm-db2Node
描述	表示 DB2 服务器
SubClassOf	eSap / ServiceConnectionPoint
必需的属性	db2nodeName
可选属性	db2nodeAlias  db2instanceName  db2Type  host / dNSHostName (参见注释 2)  protocolInformation/ServiceBindingInformation
类型	结构
OID (对象标识符)	1.3.18.0.2.6.116
GUID (全局唯一标识符)	b3afd65a-5c5b-11d3-b818-002035559151
特殊注意事项	<ol style="list-style-type: none"> <li>1. <i>DB2Node</i> 类是从 IBM SecureWay 目录下面的 <i>eSap</i> 对象类和“Microsoft 活动目录”下面的 <i>ServiceConnectionPoint</i> 对象类派生的。</li> <li>2. <i>host</i> 用于 IBM SecureWay 环境。<i>dNSHostName</i> 属性在“Microsoft 活动目录”下使用。</li> <li>3. <i>protocolInformation</i> 仅在 IBM SecureWay 环境中使用。对于“Microsoft 活动目录”，从 <i>ServiceConnectionPoint</i> 类继承的 <i>ServiceBindingInformation</i> 属性用来包含协议信息。</li> </ol>

*DB2Node* 对象中的 *protocolInformation* (在“IBM SecureWay 目录”中) 或 *ServiceBindingInformation* (在“Microsoft 活动目录”中) 属性包含用来绑定 DB2 数据库服务器的通信协议信息。它由标记组成，这些标记描述受支持的网络协议。标记之间由分号隔开。标记之间没有空格。可使用星号 (\*) 来指定可选参数。

TCP/IP 的标记为:

- "TCPIP"
- 服务器主机名或 IP 地址
- 服务名 (svcname) 或端口号 (例如, 50000)

- (可选) 安全性 ("NONE" 或 "SOCKS")

APPN 的标记为:

- "APPN"
- 网络标识
- 伙伴 LU
- 事务处理程序 (TP) 名 (仅支持 "应用程序 TP", 不支持 "服务 TP" - 十六进制的 TP)
- 方式
- 安全性 ("NONE"、"SAME" 或 "PROGRAM")
- (可选) LAN 适配器地址
- (可选) 更改密码 LU

**注:** 在 DB2 Windows NT 版 (或 Windows 98 版) 客户机上, 若未在本地 SNA 堆栈上配置 APPN 信息; 并且, 若在 LDAP 中找到了 LAN 适配卡地址和可选的更改密码 LU, 则 DB2 客户机尝试使用此信息来配置 SNA 堆栈 (若它知道如何配置堆栈的话)。此支持在 DB2 AIX 版或 DB2 Solaris 版客户机上不可用。

IPX/SPX 的标记为:

- "IPXSPX"
- IPX 地址

IPX/SPX 侦听器可用于 AIX 和 Solaris 上的 DB2 服务器 (而不是在客户机上)。NetBIOS 和 NPIPE 在 AIX 和 Solaris 上不受支持。

NetBIOS 的标记为:

- "NETBIOS"
- 服务器 NetBIOS 工作站名称

"命名管道" 的标记为:

- "NPIPE"
- 服务器的计算机名
- 服务器的实例名

表 31. DB2Database

类	<b>DB2Database</b>
活动目录 LDAP 显示名	ibm-db2Database

表 31. DB2Database (续)

类	DB2Database
活动目录公共名 (cn)	ibm-db2Database
描述	表示 DB2 数据库
SubClassOf	顶部
必需的属性	db2databaseName db2nodePtr
可选属性	db2databaseAlias db2additionalParameters db2ARLibrary db2authenticationLocation db2gwPtr db2databaseRelease DCEPrincipalName
类型	结构
OID (对象标识符)	1.3.18.0.2.6.117
GUID (全局唯一标识符)	b3afd659-5c5b-11d3-b818-002035559151

表 32. db2additionalParameters

属性	db2additionalParameters
活动目录 LDAP 显示名	ibm-db2AdditionalParameters
活动目录公共名 (cn)	ibm-db2AdditionalParameters
描述	包含连接主机数据库服务器时使用的附加参数
语法	忽略大小写的字符串
最大长度	1024
多值	单值
OID (对象标识符)	1.3.18.0.2.4.426
GUID (全局唯一标识符)	b3afd315-5c5b-11d3-b818-002035559151

表 33. db2authenticationLocation

属性	db2authenticationLocation
活动目录 LDAP 显示名	ibm-db2AuthenticationLocation

表 33. *db2authenticationLocation* (续)

属性	<b>db2authenticationLocation</b>
活动目录公共名 (cn)	ibm-db2AuthenticationLocation
描述	指定执行认证的位置
语法	忽略大小写的字符串
最大长度	64
多值	单值
OID (对象标识符)	1.3.18.0.2.4.425
GUID (全局唯一标识符)	b3afd317-5c5b-11d3-b818-002035559151
注释	有效值有： CLIENT、SERVER、DCS、DCE、KERBEROS、 SVRENCRYPT 或 DCSENCRYPT

表 34. *db2ARLibrary*

属性	<b>db2ARLibrary</b>
活动目录 LDAP 显示名	ibm-db2ARLibrary
活动目录公共名 (cn)	ibm-db2ARLibrary
描述	“应用程序请求器”库的名称
语法	忽略大小写的字符串
最大长度	256
多值	单值
OID (对象标识符)	1.3.18.0.2.4.427
GUID (全局唯一标识符)	b3afd316-5c5b-11d3-b818-002035559151

表 35. *db2databaseAlias*

属性	<b>db2databaseAlias</b>
活动目录 LDAP 显示名	ibm-db2DatabaseAlias
活动目录公共名 (cn)	ibm-db2DatabaseAlias
描述	数据库别名
语法	忽略大小写的字符串
最大长度	1024
多值	多值
OID (对象标识符)	1.3.18.0.2.4.422
GUID (全局唯一标识符)	b3afd318-5c5b-11d3-b818-002035559151



表 36. *db2databaseName*

属性	<b>db2databaseName</b>
活动目录 LDAP 显示名	ibm-db2DatabaseName
活动目录公共名 (cn)	ibm-db2DatabaseName
描述	数据库名
语法	忽略大小写的字符串
最大长度	1024
多值	单值
OID (对象标识符)	1.3.18.0.2.4.421
GUID (全局唯一标识符)	b3afd319-5c5b-11d3-b818-002035559151

表 37. *db2databaseRelease*

属性	<b>db2databaseRelease</b>
活动目录 LDAP 显示名	ibm-db2DatabaseRelease
活动目录公共名 (cn)	ibm-db2DatabaseRelease
描述	数据库发行版号
语法	忽略大小写的字符串
最大长度	64
多值	单值
OID (对象标识符)	1.3.18.0.2.4.429
GUID (全局唯一标识符)	b3afd31a-5c5b-11d3-b818-002035559151

表 38. *db2nodeAlias*

属性	<b>db2nodeAlias</b>
活动目录 LDAP 显示名	ibm-db2NodeAlias
活动目录公共名 (cn)	ibm-db2NodeAlias
描述	节点别名
语法	忽略大小写的字符串
最大长度	1024
多值	多值
OID (对象标识符)	1.3.18.0.2.4.420
GUID (全局唯一标识符)	b3afd31d-5c5b-11d3-b818-002035559151

表 39. db2nodeName

属性	db2nodeName
活动目录 LDAP 显示名	ibm-db2NodeName
活动目录公共名 (cn)	ibm-db2NodeName
描述	节点名
语法	忽略大小写的字符串
最大长度	64
多值	单值
OID (对象标识符)	1.3.18.0.2.4.419
GUID (全局唯一标识符)	b3afd31e-5c5b-11d3-b818-002035559151

表 40. db2nodePtr

属性	db2nodePtr
活动目录 LDAP 显示名	ibm-db2NodePtr
活动目录公共名 (cn)	ibm-db2NodePtr
描述	指向表示拥有数据库的数据库服务器的“节点”(DB2Node) 对象的指针
语法	专有名称
最大长度	1000
多值	单值
OID (对象标识符)	1.3.18.0.2.4.423
GUID (全局唯一标识符)	b3afd31f-5c5b-11d3-b818-002035559151
特殊注意事项	此关系允许客户机检索用来连接数据库的协议通信信息。

表 41. db2gwPtr

属性	db2gwPtr
活动目录 LDAP 显示名	ibm-db2GwPtr
活动目录公共名 (cn)	ibm-db2GwPtr
描述	指向表示网关服务器的“节点”对象的指针，可从该网关服务器访问数据库
语法	专有名称
最大长度	1000
多值	单值
OID (对象标识符)	1.3.18.0.2.4.424

表 41. db2gwPtr (续)

属性	db2gwPtr
GUID (全局唯一标识符)	b3afd31b-5c5b-11d3-b818-002035559151

表 42. db2instanceName

属性	db2instanceName
活动目录 LDAP 显示名	ibm-db2InstanceName
活动目录公共名 (cn)	ibm-db2InstanceName
描述	数据库服务器实例的名称
语法	忽略大小写的字符串
最大长度	256
多值	单值
OID (对象标识符)	1.3.18.0.2.4.428
GUID (全局唯一标识符)	b3afd31c-5c5b-11d3-b818-002035559151

表 43. db2Type

属性	db2Type
活动目录 LDAP 显示名	ibm-db2Type
活动目录公共名 (cn)	ibm-db2Type
描述	数据库服务器的类型
语法	忽略大小写的字符串
最大长度	64
多值	单值
OID (对象标识符)	1.3.18.0.2.4.418
GUID (全局唯一标识符)	b3afd320-5c5b-11d3-b818-002035559151
注释	数据库服务器的有效类型是: SERVER、MPP 和 DCS

表 44. DCEPrincipalName

属性	DCEPrincipalName
活动目录 LDAP 显示名	ibm-DCEPrincipalName
活动目录公共名 (cn)	ibm-DCEPrincipalName
描述	DCE 委托人名称
语法	忽略大小写的字符串
最大长度	2048

表 44. *DCEPrincipalName* (续)

属性	<b>DCEPrincipalName</b>
多值	单值
OID (对象标识符)	1.3.18.0.2.4.443
GUID (全局唯一标识符)	b3afd32d-5c5b-11d3-b818-002035559151

表 45. *cesProperty*

属性	<b>cesProperty</b>
活动目录 LDAP 显示名	ibm-cesProperty
活动目录公共名 (cn)	ibm-cesProperty
描述	此属性的值可用来提供应用程序特定首选项配置参数。例如, 值可以包含 XML 格式化的数据。此属性的所有值在 <code>cesPropertyType</code> 属性值中都必须相似的。
语法	大小写精确的字符串
最大长度	32700
多值	多值
OID (对象标识符)	1.3.18.0.2.4.307
GUID (全局唯一标识符)	b3afd2d5-5c5b-11d3-b818-002035559151

表 46. *cesPropertyType*

属性	<b>cesPropertyType</b>
活动目录 LDAP 显示名	ibm-cesPropertyType
活动目录公共名 (cn)	ibm-cesPropertyType
描述	此属性的值可用来描述 <code>cesProperty</code> 属性的所有值的语法、语义或其他特性。例如, 值 "XML" 可用来指示 <code>cesProperty</code> 的所有值都作为 XML 语法编码。
语法	忽略大小写的字符串
最大长度	128
多值	多值
OID (对象标识符)	1.3.18.0.2.4.308
GUID (全局唯一标识符)	b3afd2d6-5c5b-11d3-b818-002035559151

表 47. *cisProperty*

属性	<b>cisProperty</b>
活动目录 LDAP 显示名	ibm-cisProperty
活动目录公共名 (cn)	ibm-cisProperty
描述	此属性的值可用于提供应用程序特定首选项配置参数。例如，值可以包含 INI 文件。此属性的所有值在其 <code>cisPropertyType</code> 属性值中都必须相似的。
语法	忽略大小写的字符串
最大长度	32700
多值	多值
OID (对象标识符)	1.3.18.0.2.4.309
GUID (全局唯一标识符)	b3afd2e0-5c5b-11d3-b818-002035559151

表 48. *cisPropertyType*

属性	<b>cisPropertyType</b>
活动目录 LDAP 显示名	ibm-cisPropertyType
活动目录公共名 (cn)	ibm-cisPropertyType
描述	此属性的值可用于描述 <code>cisProperty</code> 属性的所有值的语法、语义或其他特性。例如，值 "INI File" 可用于指示 <code>cisProperty</code> 属性的所有值都是 INI 文件。
语法	忽略大小写的字符串
最大长度	128
多值	多值
OID (对象标识符)	1.3.18.0.2.4.310
GUID (全局唯一标识符)	b3afd2e1-5c5b-11d3-b818-002035559151

表 49. *binProperty*

属性	<b>binProperty</b>
活动目录 LDAP 显示名	ibm-binProperty
活动目录公共名 (cn)	ibm-binProperty
描述	此属性的值可用于提供应用程序特定首选项配置参数。例如，值可以包含一组二进制编码的 Lotus 123 特性。此属性的所有值在其 <code>binPropertyType</code> 属性值中都必须相似的。
语法	二进制
最大长度	250000

表 49. binProperty (续)

属性	binProperty
多值	多值
OID (对象标识符)	1.3.18.0.2.4.305
GUID (全局唯一标识符)	b3afd2ba-5c5b-11d3-b818-002035559151

表 50. binPropertyType

属性	binPropertyType
活动目录 LDAP 显示名	ibm-binPropertyType
活动目录公共名 (cn)	ibm-binPropertyType
描述	此属性的值可用来描述 binProperty 属性的所有值的语法、语义或其他特性。例如, 值 "Lotus 123" 可用来指示 binProperty 属性的所有值都是二进制编码的 Lotus 123 特性。
语法	忽略大小写的字符串
最大长度	128
多值	多值
OID (对象标识符)	1.3.18.0.2.4.306
GUID (全局唯一标识符)	b3afd2bb-5c5b-11d3-b818-002035559151

表 51. PropertyType

属性	PropertyType
活动目录 LDAP 显示名	ibm-propertyType
活动目录公共名 (cn)	ibm-propertyType
描述	此属性的值描述 eProperty 对象的语义特性
语法	忽略大小写的字符串
最大长度	128
多值	多值
OID (对象标识符)	1.3.18.0.2.4.320
GUID (全局唯一标识符)	b3afd4ed-5c5b-11d3-b818-002035559151

表 52. settingID

属性	settingID
活动目录 LDAP 显示名	不适用
活动目录公共名 (cn)	不适用

表 52. *settingID* (续)

属性	<b>settingID</b>
描述	可用来标识 <code>cimSetting</code> 派生的对象项 (如 <code>eProperty</code> ) 的命名属性
语法	忽略大小写的字符串
最大长度	256
多值	单值
OID (对象标识符)	1.3.18.0.2.4.325
GUID (全局唯一标识符)	b3afd596-5c5b-11d3-b818-002035559151





---

## 附录K. 扩展控制中心

在版本 7 中，可使用新的插件体系结构来扩展 DB2 通用数据库控制中心，以提供附加功能。

插件体系结构能够为控制中心弹出菜单中的给定对象添加项目，以及向工具栏添加新按钮。这些工具附带了一组您必须实现的 Java 接口。这些接口用来与控制中心通信，包含那些附加的操作。

---

### 性能注意事项

在控制中心工具启动时，装入扩充的插件 (db2plug.zip)。这可能延长工具的启动时间，这视 ZIP 文件的大小而定；然而，我们希望对于大多数用户而言插件 ZIP 文件属于小文件，因此影响也应最小。

---

### 封装注意事项

根据 Java 归档文件的规则，必须 ZIP 扩展部分类文件。要把“控制中心”工具当作应用程序来运行，ZIP 文件 (db2plug.zip) 必须位于类路径中。要把控制中心工具当作小应用程序来运行，ZIP 文件必须位于控制中心html 文件中的 <codebase> 标记指向的位置。

应在不压缩的情况下构建 ZIP 文件，并维持所有类文件的相对路径位置 (zip -r0 db2plug.zip \*.class)。

---

### 接口描述

附带了下列接口：

- CCExtension
- CCOBJECT
- CCMenuAction
- CCToolbarAction。

下面几节将描述这些接口，然后举例说明。

## CCExtension

CCExtension 接口允许通过添加新工具栏按钮、新菜单项并覆盖现存的菜单操作来扩展控制中心用户界面。

该外部接口的定义如下所示:

```
public interface CCExtension
{
    /**
     * Get an array of CCOBJECT subclass objects which define
     * a list of objects to be inserted or overridden in the
     * Control Center
     * @return CCOBJECT[] CCOBJECT subclass objects array
     */
    public CCOBJECT[] getObjects();

    /**
     * Get an array of CCToolbarAction subclass objects which represent
     * a list of buttons to be added to the Control Center
     * main toolbar.
     * @return CCToolbarAction[] CCToolbarAction subclass objects array
     */
    public CCToolbarAction[] getToolbarActions();
}
```

要使用 CCExtension, 应创建一个 Java 类, 它导入 "com.ibm.db2.tools.cc.navigator" 软件包并实现此接口。此新类必须能够实现 getObjects() 和 getToolbarActions() 方法。

getObjects() 方法返回一组定义现存对象的 CCOBJECT, 用户可向它添加新菜单操作或除去一组预定义的菜单操作。

getToolbarActions() 方法返回一组将添加到控制中心主工具栏中的 CCToolbarAction。

可使用一个 CCExtension 子类文件或多个 CCExtension 子类文件定义控制中心扩展部分。为了使控制中心使用这些扩展部分, 使用以下设置步骤:

1. 创建一个 "db2plug.zip" 文件, 以包含所有 CCExtension 子类文件。不应压缩这些文件。例如, 若 CCExtension 文件在插件软件包中, 并位于 plugin 目录下, 则使用下列命令:

```
zip -r0 db2plug.zip plugin\*.class
```

此命令将所有插件软件包类文件置于 db2plug.zip 文件中, 并保留它们的相对路径信息。

2. 要把控制中心当作小应用程序来运行，将 db2plug.zip 文件置于控制中心.html 文件中 <codebase> 标记指向的位置。要把控制中心当作应用程序来运行，将 db2plug.zip 置于 CLASSPATH 环境变量指向的目录中。

对于支持多个存档件的浏览器，只需将 "db2plug.zip" 添加到控制中心 html 页的存档件列表中。否则，所有 CCExtension、CCObject、CCToolbarAction 和 CCMenuAction 子类文件都必须位于它们的相对目录下，这取决于它们所属的软件包。

## CCObject

CCObject 接口允许您更改现存对象的菜单操作行为。

该外部接口的定义如下所示：

```
public interface CCObject
{
    /**
     * The following static constants defines a list of object type
     * available to be added to the Control Center tree.
     */
    public static final int UDB_SYSTEMS_FOLDER           = 0;
    public static final int UDB_SYSTEM                  = 1;
    public static final int UDB_INSTANCES_FOLDER        = 2;
    public static final int UDB_INSTANCE                = 3;
    public static final int UDB_DATABASES_FOLDER        = 4;
    public static final int UDB_DATABASE                = 5;
    public static final int UDB_TABLES_FOLDER           = 6;
    public static final int UDB_TABLE                   = 7;
    public static final int UDB_TABLESPACES_FOLDER      = 8;
    public static final int UDB_TABLESPACE              = 9;
    public static final int UDB_VIEWS_FOLDER            = 10;
    public static final int UDB_VIEW                    = 11;
    public static final int UDB_ALIASES_FOLDER          = 12;
    public static final int UDB_ALIAS                   = 13;
    public static final int UDB_TRIGGERS_FOLDER         = 14;
    public static final int UDB_TRIGGER                 = 15;
    public static final int UDB_SCHEMAS_FOLDER          = 16;
    public static final int UDB_SCHEMA                  = 17;
    public static final int UDB_INDEXES_FOLDER          = 18;
    public static final int UDB_INDEX                   = 19;
    public static final int UDB_CONNECTIONS_FOLDER      = 20;
    public static final int UDB_CONNECTION              = 21;
    public static final int UDB_REPLICATION_SOURCES_FOLDER = 22;
    public static final int UDB_REPLICATION_SOURCE      = 23;
    public static final int UDB_REPLICATION_SUBSCRIPTIONS_FOLDER = 24;
    public static final int UDB_REPLICATION_SUBSCRIPTION = 25;
    public static final int UDB_BUFFERPOOLS_FOLDER     = 26;
    public static final int UDB_BUFFERPOOL              = 27;
    public static final int UDB_APPLICATION_OBJECTS_FOLDER = 28;
    public static final int UDB_USER_DEFINED_DISTINCT_DATATYPES_FOLDER = 29;
    public static final int UDB_USER_DEFINED_DISTINCT_DATATYPE = 30;
    public static final int UDB_USER_DEFINED_DISTINCT_FUNCTIONS_FOLDER = 31;
```

```

public static final int UDB_USER_DEFINED_DISTINCT_FUNCTION      = 32;
public static final int UDB_PACKAGES_FOLDER                    = 33;
public static final int UDB_PACKAGE                           = 34;
public static final int UDB_STORE_PROCEDURES_FOLDER           = 35;
public static final int UDB_STORE_PROCEDURE                   = 36;
public static final int UDB_USER_AND_GROUP_OBJECTS_FOLDER     = 37;
public static final int UDB_DB_USERS_FOLDER                   = 38;
public static final int UDB_DB_USER                           = 39;
public static final int UDB_DB_GROUPS_FOLDER                  = 40;
public static final int UDB_DB_GROUP                           = 41;
public static final int UDB_DRDA_TABLE                         = 42;

public static final int S390_SUBSYSTEMS_FOLDER                = 43;
public static final int S390_SUBSYSTEM                        = 44;
public static final int S390_BUFFERPOOLS_FOLDER              = 45;
public static final int S390_BUFFERPOOL                       = 46;
public static final int S390_VIEWS_FOLDER                     = 47;
public static final int S390_VIEW                             = 48;
public static final int S390_DATABASES_FOLDER                 = 49;
public static final int S390_DATABASE                         = 50;
public static final int S390_TABLESPACES_FOLDER               = 51;
public static final int S390_TABLESPACE                       = 52;
public static final int S390_TABLES_FOLDER                    = 53;
public static final int S390_TABLE                           = 54;
public static final int S390_INDEXES_FOLDER                   = 55;
public static final int S390_INDEX                             = 56;
public static final int S390_STORAGE_GROUPS_FOLDER            = 57;
public static final int S390_STORAGE_GROUP                    = 58;
public static final int S390_ALIASES_FOLDER                   = 59;
public static final int S390_ALIAS                             = 60;
public static final int S390_SYNONYMS_FOLDER                  = 61;
public static final int S390_SYNONYM                          = 62;
public static final int S390_APPLICATION_OBJECTS_FOLDER      = 63;
public static final int S390_COLLECTIONS_FOLDER               = 64;
public static final int S390_COLLECTION                       = 65;
public static final int S390_PACKAGES_FOLDER                  = 66;
public static final int S390_PACKAGE                           = 67;
public static final int S390_PLANS_FOLDER                     = 68;
public static final int S390_PLAN                             = 69;
public static final int S390_PROCEDURES_FOLDER                = 70;
public static final int S390_PROCEDURE                         = 71;
public static final int S390_DB_USERS_FOLDER                  = 72;
public static final int S390_DB_USER                           = 73;
public static final int S390_LOCATIONS_FOLDER                 = 74;
public static final int S390_LOCATION                         = 75;
public static final int S390_DISTINCT_TYPES_FOLDER            = 76;
public static final int S390_DISTINCT_TYPE                     = 77;
public static final int S390_USER_DEFINED_FUNCTIONS_FOLDER    = 78;
public static final int S390_USER_DEFINED_FUNCTION            = 79;
public static final int S390_TRIGGERS_FOLDER                  = 80;
public static final int S390_TRIGGER                           = 81;
public static final int S390_SCHEMAS_FOLDER                   = 82;
public static final int S390_SCHEMA                           = 83;
public static final int S390_CATALOG_TABLES_FOLDER            = 84;
public static final int S390_CATALOG_TABLE                     = 85;

```

```

public static final int DCS_GATEWAY_CONNECTIONS_FOLDER           = 86;
public static final int DCS_GATEWAY_CONNECTION                  = 87;

/**
 * Total number of object types
 */
public static final int NUM_OBJECT_TYPES                         = 88;

/**
 * Get the name of these object
 * The function returns the name of this object. This name
 * can be of three types:
 * (1) Fully qualified name
 *     Syntax: xxxxx-yyyyy-zzzzz
 *           where xxxxx-yyyyy is the fully qualified name of the
 *           parent object and zzzzz is the name of the new object.
 *     Note: Parent and child object name is separated by '-' character.
 *     If a schema name is required to identify object, the fully
 *     qualified name is represented by xxxxx-yyyyy-wwwww.zzzzz
 *     where wwwwww is the schema name.
 *     Only the behavior of the object that matches this fully
 *     qualified name will be affected.
 * (2) Parent fully qualified name
 *     Syntax: xxxxx-yyyyy
 *           where xxxxx-yyyyy is the fully qualified name of the
 *           parent object.
 *     When the object type is folder (ie. DATABASES_FOLDER), the
 *     getName() should only return the fully qualified name of the
 *     folder's parent.
 *     Only the behavior of the object that match this name
 *     and the specific type return by the getType() function will be
 *     affected.
 * (3) null
 *     Syntax: null
 *     If null is return, the CCActions returns by the getActions()
 *     call will be applied to all objects of type returns by the
 *     getType() call.
 * @return String object name
 */
public String getName();

/**
 * Get the type of this object
 * @return int return one of the static type constants defined
 * in this interface
 */
public int getType();

/**
 * Get the CCMenu Action array which defines the list of menu actions
 * to be created for the selected object
 * return CCMenuAction[] CCMenuAction array
 */
public CCMenuAction[] getMenuActions();

```

```

/**
 * Check if this object is editable.
 * If not, the Alter related menu items will be removed from
 * the object's popup menu return boolean If false, the Alter
 * menu item will be removed from the object's popup menu
 */
public boolean isEditable();

/**
 * Check if this object is configurable.
 * If not, the configuration related menu items will be
 * removed from the object's popup menu return boolean If
 * false, the Configuration related menu item will be removed
 * from the object's popup menu
 */
public boolean isConfigurable();
}

```

注: 这时, CCOBJECT 中的最后两个方法: isEditable() 和 isConfigurable() 应始终返回 **true**。

## CCMenuItem

CCMenuItem 接口允许您为控制中心对象定义一个新操作。

该外部接口的定义如下所示:

```

public interface CCMenuItem
{
/**
 * Get the name of this action
 * @return String Name text on the menu item
 */
public String getMenuText();

/**
 * Invoked when an action occurs. Use the getActionCommand()
 * method of the ActionEvent to get the fully qualified name of
 * the invoked Control Center object.
 * @param e Action event
 */
public void actionPerformed(ActionEvent e);
}

```

## CCToolbarAction

CCToolbarAction 接口允许您为控制中心工具栏定义一个新操作。

该外部接口的定义如下所示:

```

public interface CCToolbarAction
{
/**

```

```

    * Get the name of this action
    * @return String Name text on the menu item, or toolbar
    * button hover help
    */
public String getHoverHelpText();

/**
 * Get the icon for the toolbar button
 * Any toolbar CCAction should implement this function and return
 * a valid ImageIcon object. Otherwise, the button will have no icon.
 * @return ImageIcon icon to be displayed
 */
public ImageIcon getIcon();

/**
 * Invoked when an action occurs.
 * @param e Action event
 */
public void actionPerformed(ActionEvent e);
}

```

---

## 使用方案

以下示例中的代码将:

1. 更新 SAMPLE 数据库的操作 (参见第392页的『MySample.java』)
2. 更新所有数据库对象的操作 (参见第393页的『MyDatabaseActions.java』)
3. 添加新实例对象 (参见第393页的『MyInstance.java』)
4. 更新 DB2 实例的操作 (参见第394页的『MyDB2.java』)
5. 更新数据库文件夹的操作 (参见第395页的『MyDatabases.java』)
6. 更新 SYSIBM.SYSPLAN 表的操作 (参见第395页的『MySYSPLAN.java』)
7. 添加新表对象 (参见第396页的『MyTable.java』)
8. 更新应用程序对象的 DB\_User 对象的操作 (参见第397页的『MyDBUser.java』)
9. 向控制中心工具栏添加按钮 (参见第398页的『MyToolbarAction.java』)。

主要扩展部分文件是 MyExtension.java。所有类文件都存储在 plugin 目录下, 用下面命令解压:

```
zip -r0 db2plug.zip plugin
```

db2plug.zip 的输出被置于 CLASSPATH 目录下或 codebase 目录下, 这取决于控制中心是作为应用程序还是作为小应用程序来运行。

## MyExtension.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyExtension implements CCExtension
{
    public CCOBJECT[] getObjects()
    {
        CCOBJECT[] objs = new CCOBJECT[10];
        objs[0] = new MySample();
        objs[1] = new MyDatabaseActions();
        objs[2] = new MyInstance();
        objs[3] = new MyDB2();
        objs[4] = new MyDatabases();
        objs[5] = new MySYSPLAN();
        objs[6] = new MyTable();
        objs[7] = new MyDBUser();
        return objs;
    }

    public CCACTION[] getActions()
    {
        CCACTION[] actions = new CCACTION[1];
        actions[0] = new MyToolbarAction();
        return actions;
    }
}
```

## MySample.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MySample implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE";
    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }
}
```



```

        public CCAction[] getActions()
        {
            CCAction[] acts = new CCAction[2];
            acts[0] = new MyAlterAction();
            acts[1] = new MyAction();
            return acts;
        }
    }
}

```

## MyDatabaseActions.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDatabaseActions implements CCOBJECT
{
    public String getName()
    {
        return null;
    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyDropAction();
        acts[1] = new MyAction();
        return acts;
    }
}

```

## MyInstance.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyInstance implements CCOBJECT

```

```

{
    public String getName()
    {
        return "LOCAL - MyInstance";
    }

    public int getType()
    {
        return INSTANCE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return true;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return null;
    }
}

```

## MyDB2.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDB2 implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2";
    }

    public int getType()
    {
        return INSTANCE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {

```

```

        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[3];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        acts[2] = new MyCascadeAction();
        return acts;
    }
}

```

## MyDatabases.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDatabases implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - Databases";
    }

    public int getType()
    {
        return DATABASE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[1];
        acts[0] = new MyCreateAction();
        return acts;
    }
}

```

## MySYSPLAN.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

```

```

public class MySYSPLAN implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE - SYSIBM - SYSPLAN";
    }

    public int getType()
    {
        return TABLE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCACTION[] getActions()
    {
        CCACTION[] acts = new CCACTION[2];
        acts[0] = new MyALTERACTION();
        acts[1] = new MyACTION();
        return acts;
    }
}

```

## MyTable.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyTable implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - SAMPLE - SYSIBM - MyTable";
    }

    public int getType()
    {
        return TABLE;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()

```

```

    {
        return true;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return acts;
    }
}

```

## MyDBUser.java

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDBUser implements CCOBJECT
{
    public String getName()
    {
        return "LOCAL - DB2 - TEST-DB Users";
    }

    public int getType()
    {
        return DB_USER;
    }

    public javax.swing.ImageIcon getIcon()
    {
        return null;
    }

    public boolean isNew()
    {
        return false;
    }

    public CCAction[] getActions()
    {
        CCAction[] acts = new CCAction[2];
        acts[0] = new MyAlterAction();
        acts[1] = new MyAction();
        return acts;
    }
}

```

## MyToolBarAction.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;
import javax.swing.*;

public class MyToolBarAction extends CCAction
{
    public MyToolBarAction()
    {
        super("MyToolBarAction");
    }

    public ImageIcon getIcon()
    {
        return <Your icon>;
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My action performed, object name = " +
                            objectName );
        return true;
    }
}
```

## MyAlterAction.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyAlterAction extends CCAction
{
    public MyAlterAction()
    {
        super(0);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My alter action performed, object name = " +
                            objectName );
        return true;
    }
}
```

## MyAction.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyAction extends CCAction
{
    public MyAction()
    {
```

```

        super("MyAction");
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My action performed, object name = " +
            objectName );
        return true;
    }
}

```

### **MyDropAction.java**

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyDropAction extends CCAction
{
    public MyDropAction()
    {
        super(1);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My drop action performed, object name = " +
            objectName );
        return true;
    }
}

```

### **MyCascadeAction.java**

```

package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyCascadeAction extends CCAction
{
    public MyCascadeAction()
    {
        super(11,2);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My cascade action performed, object name = " +
            objectName );
        return true;
    }
}

```

## MyCreateAction.java

```
package plugin;
import com.ibm.db2.tools.cc.navigator.*;

public class MyCreateAction extends CCAction
{
    public MyCreateAction()
    {
        super(0);
    }

    public boolean actionPerformed(String objectName)
    {
        System.out.println( "My create action performed, object name = " +
            objectName );
        return true;
    }
}
```



---

## 附录L. 使用 DB2 资料库

“DB2 通用数据库”资料库由联机帮助、书籍（PDF 和 HTML）和 HTML 格式的样本程序组成。本节描述所提供的信息以及如何访问这些信息。

要访问联机产品信息，可以使用“信息中心”。有关更多信息，参见第413页的『通过“信息中心”访问信息』。可以查看任务信息、DB2 书籍、故障诊断信息、样本程序和 Web 上的 DB2 信息。

---

### DB2 PDF 文件和打印的书籍

#### DB2 信息

下表将 DB2 书籍分为四个类别：

##### DB2 指南和参考信息

这些书籍包含所有平台的公共 DB2 信息。

##### DB2 安装和配置信息

这些书籍是针对特定平台上的 DB2 的。例如，有分别针对 OS/2 平台、Windows 平台和基于 UNIX 的平台上 DB2 的《快速入门》书籍。

##### HTML 格式的跨平台样本程序

这些样本是与“应用程序开发客户机”一起安装的样本程序的 HTML 版本。样本仅供参考，并不替代实际程序。

##### 发行说明

这些文件包含 DB2 书籍中未能包括的最新信息。

HTML 格式的安装手册、发行说明和教程可直接在产品 CD-ROM 上看到。大部分书籍在产品 CD-ROM 上都有 HTML 格式以便查看，而在 DB2 出版物 CD-ROM 上则有 Adobe Acrobat (PDF) 格式以便查看和打印。还可从 IBM 订购打印的副本；请参阅第410页的『订购打印书籍』。下表列示了可订购的书籍。

在 OS/2 和 Windows 平台上，可在 `sql1lib\doc\html` 目录下安装 HTML 文件。DB2 信息被翻译成各种语言；但是，并非所有的信息都有每一种语言的翻译版本。每当信息不能以某种特定语言表示出来时，就会提供英语信息。

在 UNIX 平台上，可在 `doc/%L/html`（其中 `%L` 表示语言环境）目录下安装多种语言版本的 HTML 文件。有关更多信息，参考适当的《快速入门》书籍。

您可以各种方法来获取 DB2 书籍并访问信息:

- 第412页的『查看联机信息』
- 第416页的『搜索联机信息』
- 第410页的『订购打印书籍』
- 第409页的『打印 PDF 书籍』

表 53. DB2 信息

名称	描述	书号	HTML 目录 PDF 文件名
<b>DB2 指南和参考信息</b>			
《管理指南》	《管理指南: 计划》提供数据库概念的概述、有关设计问题（如逻辑和物理数据库设计）的信息，以及高可用性的讨论。	SB84-0219	db2d0
	《管理指南: 实现》提供有关实现问题（如实现设计、访问数据库、审核、备份和恢复）的信息。	SB84-0218	db2d2x70
	《管理指南: 性能》提供有关数据库环境以及应用程序性能评估和调整的信息。	SB84-0243	db2d3x70
	在北美，可使用书号 SBOF-8934 来订购三卷英文版的《管理指南》。		
<i>Administrative API Reference</i>	描述 DB2 应用程序编程接口 (API) 以及您可以用来管理数据库的数据结构。此书还说明如何在应用程序中调用 API。	SC09-2947	db2b0
		db2b0x70	
《应用程序构建指南》	提供环境设置信息和关于如何在 Windows、OS/2 和基于 UNIX 的平台上编译、链接和运行 DB2 应用程序的循序渐进说明。	SB84-0220	db2ax
		db2axx70	
<i>APPC, CPI-C, and SNA Sense Codes</i>	提供关于使用“DB2 通用数据库”产品时可能遇到的 APPC、CPI-C 和 SNA 检测码的一般信息。	无书号	db2ap
	仅有 HTML 格式的版本。	db2apx70	

表 53. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
<i>Application Development Guide</i>	说明如何开发使用嵌入式 SQL 或 Java (JDBC 和 SQLJ) 来访问 DB2 数据库的应用程序。讨论主题包括在分区环境或联合体系统中编写存储过程、编写用户定义函数、创建用户定义类型、使用触发器和开发应用程序。	SC09-2949 db2a0x70	db2a0
<i>CLI Guide and Reference</i>	说明如何开发使用“DB2 调用层接口”(一个与 Microsoft ODBC 规范兼容的可调用 SQL 接口)来访问 DB2 数据库的应用程序。	SC09-2950 db2l0x70	db2l0
<i>Command Reference</i>	说明如何使用“命令行处理器”，并描述可用来管理数据库的 DB2 命令。	SC09-2951 db2n0x70	db2n0
<i>Connectivity Supplement</i>	提供有关以下各项的设置和参考信息：如何将作为 DRDA 应用程序请求器的 DB2 AS/400 版、DB2 OS/390 版、DB2 MVS 版、DB2 VM 版与“DB2 通用数据库”服务器配合使用。此书还详述了如何将 DRDA 应用服务器与 DB2 Connect 应用程序请求器配合使用。  仅有 HTML 和 PDF 格式。	无书号 db2h1x70	db2h1
<i>Data Movement Utilities Guide and Reference</i>	说明如何使用 DB2 实用程序(如导入、导出、装入、自动装入程序和 DPROP)来使数据移动易于进行。	SC09-2955 db2dmx70	db2dm
《数据仓库中心管理指南》	提供有关如何使用“数据仓库中心”构建和维护数据仓库的信息。	SB84-0226 db2ddx70	db2dd
<i>Data Warehouse Center Application Integration Guide</i>	提供帮助程序员将应用程序与“数据仓库中心”和“信息目录管理器”集成的信息。	SC26-9994 db2adx70	db2ad
《DB2 Connect 用户指南》	提供 DB2 Connect 产品的概念、编程以及一般用法信息。	SB84-0221 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	提供 DB2 Query Patroller 系统的操作概述、特定操作和管理信息以及管理图形用户界面实用程序的任务信息。	SC09-2958 db2dwx70	db2dw

表 53. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
《DB2 Query Patroller 用户指南》	描述如何使用 DB2 Query Patroller 的工具和功能。	SB84-0222	db2ww
		db2wwx70	
《词汇表》	提供 DB2 及其组件中使用的术语的定义。  有 HTML 格式可用且在 <i>SQL Reference</i> 中。	无书号	db2t0
		db2t0x70	
《Image、Audio 和 Video Extenders 管理和编程》	提供有关 DB2 Extender 的一般信息，有关 Image、Audio 和 Video (IAV) Extender 的管理和配置的信息，以及有关使用 IAV Extender 进行编程的信息。它包括参考信息、诊断资料（带有消息）和样本。	SB84-0247	dmbu7
		dmbu7x70	
<i>Information Catalog Manager Administration Guide</i>	提供有关管理信息目录的指南。	SC26-9995	db2di
		db2dix70	
<i>Information Catalog Manager Programming Guide and Reference</i>	提供“信息目录管理器”的体系结构接口的定义。	SC26-9997	db2bi
		db2bix70	
《信息目录管理器用户指南》	提供有关使用“信息目录管理器”用户界面的信息。	SB84-0227	db2ai
		db2aix70	
《安装和配置补遗》	指导您了解计划、安装和设置特定于平台的 DB2 客户机。此补遗还包含关于联编、设置客户机和服务器通信、DB2 GUI 工具、DRDA AS、分布式安装、配置分布式请求和访问多机种数据源的信息。	GB84-0127	db2iy
		db2iyx70	
《消息参考》	列出由 DB2、“信息目录管理器”和“数据仓库中心”发出的消息和代码，并描述应执行的操作。  在北美，您可订购两卷英文版的《消息参考》（使用书号 SBOF-8932）。	第 1 卷 S152-0064	db2m0
		db2m1x70	
		第 2 卷 GB84-0217	
		db2m2x70	
<i>OLAP Integration Server Administration Guide</i>	说明如何使用“OLAP 集成服务器”的“管理器”组件。	SC27-0782	n/a
		db2dpx70	

表 53. DB2 信息 (续)

名称	描述	书号	HTML 目录
		<b>PDF 文件名</b>	
<i>OLAP Integration Server Metaoutline User's Guide</i>	说明如何使用标准“OLAP 元轮廓”接口（而非通过使用“元轮廓辅助程序”）创建和植入 OLAP 元轮廓。	SC27-0784 db2upx70	n/a
<i>OLAP Integration Server Model User's Guide</i>	说明如何使用标准“OLAP 模型接口”（而非使用“模型辅助程序”）来创建 OLAP 模型。	SC27-0783 db2lpx70	n/a
《OLAP 安装与用户指南》	提供 OLAP Starter Kit 的配置和设置信息。	SA40-1755 db2ipx70	db2ip
《OLAP Spreadsheet Add-in 用户指南 Excel 版》	描述如何使用 Excel 电子表格程序来分析 OLAP 数据。	SA40-1756 db2epx70	db2ep
《OLAP Spreadsheet Add-in 用户指南 Lotus 1-2-3 版》	描述如何使用 Lotus 1-2-3 电子表格程序来分析 OLAP 数据。	SA40-1757 db2tpx70	db2tp
<i>Replication Guide and Reference</i>	提供随 DB2 提供的“IBM 复制”工具的计划、配置、管理和用法信息。	SC26-9920 db2e0x70	db2e0
《Spatial Extender 用户指南和参考》	提供关于 Spatial Extender 的安装、配置、管理、编程和故障诊断的信息。还提供对空间数据概念的重要描述，并提供 Spatial Extender 特定的参考资料（消息和 SQL）。	SB84-0249 db2sbx70	db2sb
《SQL 入门》	介绍 SQL 概念，并提供许多构造和任务的示例。	SB84-0223 db2y0x70	db2y0
<i>SQL Reference, 第 1 卷和第 2 卷</i>	描述 SQL 语法、语义和语言规则。此书还包括关于发行版间的不兼容性、产品限制和目录视图的信息。  在北美，可使用书号 SBOF-8933 来订购两卷英文版的 <i>SQL Reference</i> 。	第 1 卷 SC09-2974 db2s1x70  第 2 卷 SC09-2975 db2s2x70	db2s0   卷
<i>System Monitor Guide and Reference</i>	描述如何收集关于数据库和数据库管理程序的各种信息。此书说明如何利用信息来了解数据库活动、提高性能和确定问题的原因。	SC09-2956 db2f0x70	db2f0

表 53. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
《Text Extender 管理和编程》	提供有关 DB2 Extender 的一般信息，有关 Text Extender 的管理和配置的信息，以及有关使用 Text Extender 进行编程的信息。它包括参考信息、诊断资料（带有消息）和样本。	SB84-0248	desu9
		desu9x70	
<i>Troubleshooting Guide</i>	帮助您确定错误源、从问题中恢复并向“DB2 客户服务”咨询以使用诊断工具。	GC09-2850	db2p0
		db2p0x70	
《新增内容》	描述“DB2 通用数据库的版本 7”中的新特性、函数和增强功能。	SB84-0224	db2q0
		db2q0x70	
<b>DB2 安装和配置信息</b>			
<i>DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings</i>	提供 OS/2 和 Windows 32 位操作系统上的 DB2 Connect 企业版的计划、迁移、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2953	db2c6
		db2c6x70	
<i>DB2 Connect Enterprise Edition for UNIX Quick Beginnings</i>	提供基于 UNIX 的平台上的 DB2 Connect 企业版的计划、迁移、安装、配置和任务信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2952	db2cy
		db2cyx70	
《DB2 Connect 个人版快速入门》	提供 OS/2 和 Windows 32 位操作系统上的 DB2 Connect 个人版的计划、迁移、安装、配置和任务信息。此书还包含所有受支持的客户机的安装和设置信息。	GB84-0212	db2c1
		db2c1x70	
<i>DB2 Connect Personal Edition Quick Beginnings for Linux</i>	在进行所有受支持的 Linux 分发时，提供“DB2 Connect 个人版”的计划、安装、迁移和配置信息。	GC09-2962	db2c4
		db2c4x70	
《DB2 Data Links Manager 快速入门》	提供“DB2 Data Links Manager AIX 版”和 Windows 32 位操作系统的计划、安装、配置和任务信息。	GB84-0211	db2z6
		db2z6x70	
《DB2 扩充企业版 UNIX 版快速入门》	提供在基于 UNIX 的平台上的 DB2 扩充企业版的计划、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0209	db2v3
		db2v3x70	

表 53. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
<i>DB2 Enterprise - Extended Edition for Windows Quick Beginnings</i>	提供 DB2 扩充企业版 Windows 32 位操作系统版的计划、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2963 db2v6x70	db2v6
<i>DB2 for OS/2 Quick Beginnings</i>	提供 OS/2 操作系统上的“DB2 通用数据库”的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2968 db2i2x70	db2i2
《DB2 UNIX 版快速入门》	提供在基于 UNIX 的平台上的“DB2 通用数据库”的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0214 db2ixx70	db2ix
《DB2 Windows 版快速入门》	提供 Windows 32 位操作系统上的“DB2 通用数据库”的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0215 db2i6x70	db2i6
《DB2 个人版快速入门》	提供 OS/2 和 Windows 32 位操作系统上的“DB2 通用数据库个人版”的计划、安装、迁移和配置信息。	GB84-0213 db2i1x70	db2i1
<i>DB2 Personal Edition Quick Beginnings for Linux</i>	在进行所有受支持的 Linux 分发时，提供“DB2 通用数据库个人版”的计划、安装、迁移和配置信息。	GC09-2972 db2i4x70	db2i4
《DB2 Query Patroller 安装指南》	提供有关 DB2 Query Patroller 的安装信息。	GB84-0208 db2iwx70	db2iw
《DB2 仓库管理器安装指南》	提供仓库代理程序、仓库转换程序和“信息目录管理器”的安装信息。	GB84-0122 db2idx70	db2id
HTML 格式的跨平台样本程序			
HTML 格式的样本程序	为所有受 DB2 支持的平台上的编程语言提供 HTML 格式的样本程序。提供的样本程序仅供参考。并非所有样本都有所有编程语言的版本。HTML 样本仅当安装了“DB2 应用程序开发客户机”时才可用。  有关这些程序的更多信息，参考《应用程序构建指南》。	无书号	db2hs

表 53. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
发行说明			
<i>DB2 Connect</i> 发行说明	提供 DB2 Connect 书籍中未能包括的最新信息。	参见注释 2。	db2cr
<i>DB2</i> 安装说明	提供 DB2 书籍中未能包括的最新安装特定信息。	仅在产品 CD-ROM 上提供。	
<i>DB2</i> 发行说明	提供 DB2 书籍中未能包括的、有关所有 DB2 产品和功能部件的最新信息。	参见注释 2。	db2ir

**注:**

1. 文件名第六个位置的字符 *x* 指示书籍的语言版本。例如，文件名 db2d0e70 标识英语版本的《管理指南》，而文件名 db2d0f70 标识同一本书的法语版本。下列字母用在文件名的第六个位置以指示语言版本：

语言	标识符
巴西葡萄牙语	b
保加利亚语	u
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
俄语	r
简体中文	c
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



2. DB2 书籍中未能包括的最新信息以 HTML 格式在“发行说明”中提供，或作为 ASCII 文件提供。在“信息中心”中和产品 CD-ROM 上都提供了 HTML 版本。要查看 ASCII 文件：
  - 在基于 UNIX 的平台上，请参阅 `Release.Notes` 文件。此文件位于 `DB2DIR/Readme/%L` 目录中，其中 `%L` 表示语言环境名，而 `DB2DIR` 表示：
    - 在 AIX 上，是 `/usr/lpp/db2_07_01`
    - 在 HP-UX、PTX、Solaris 和 Silicon Graphics IRIX 上，是 `/opt/IBMDB2/V7.1`
    - 在 Linux 上，是 `/usr/IBMDB2/V7.1`。
  - 在其它平台上，请参阅 `RELEASE.TXT` 文件。此文件在安装产品的目录中。在 OS/2 平台上，还可双击 **IBM DB2** 文件夹，然后双击**发行说明**图符。

## 打印 PDF 书籍

如果想要书籍的打印副本，则可打印 DB2 出版物 CD-ROM 上的 PDF 文件。使用 Adobe Acrobat Reader，可打印整本书籍或特定范围内的页。有关资料库中每本书的文件名，参见第402页的表53。

可从 Adobe Web 站点（网址 <http://www.adobe.com>）获取 Adobe Acrobat Reader 的最新版本。

这些 PDF 文件包括在 DB2 出版物 CD-ROM 上，文件扩展名为 PDF。要访问这些 PDF 文件：

1. 插入 DB2 出版物 CD-ROM。在基于 UNIX 的平台上，安装 DB2 出版物 CD-ROM。参考《快速入门》以了解安装过程。
2. 启动 Acrobat Reader。
3. 从下列位置之一打开期望的 PDF 文件：
  - 在 OS/2 和 Windows 平台上：  
`x:\doc\language` 目录，其中 `x` 表示 CD-ROM 驱动器而 `language` 表示两个字符的国家或地区代码，它表示您所用的语言（例如，EN 表示英语）。
  - 在基于 UNIX 的平台上：  
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。

还可从 CD-ROM 将 PDF 文件复制至本地或网络驱动器并从该处读取它们。

## 订购打印书籍

通过使用销售单 (SBOF) 书号, 可单独订购或成套订购已打印的 DB2 书籍 (仅限北美)。要订购书籍, 与 IBM 授权经销商或市场代表联系, 或致电 1-800-879-2755 (美国) 或 1-800-IBM-4YOU (加拿大)。还可从 Publications Web 页面 (网址为 <http://www.elink.ibm.com/pbl/pbl>) 订购这些书籍。

有两套书籍。SBOF-8935 提供了“DB2 仓库管理器”的参考和用法信息。SBOF-8931 提供了所有其他“DB2 通用数据库”产品和功能部件的参考和用法信息。每个 SBOF 的内容列示在下表中:

表 54. 订购打印书籍

SBOF 号	包括的书籍
SBOF-8931	<ul style="list-style-type: none"> <li>• 管理指南: 计划</li> <li>• 管理指南: 实现</li> <li>• 管理指南: 性能</li> <li>• Administrative API Reference</li> <li>• 应用程序构建指南</li> <li>• Application Development Guide</li> <li>• CLI Guide and Reference</li> <li>• Command Reference</li> <li>• Data Movement Utilities Guide and Reference</li> <li>• 数据仓库中心管理指南</li> <li>• Data Warehouse Center Application Integration Guide</li> <li>• DB2 Connect 用户指南</li> <li>• 安装和配置补遗</li> <li>• Image、Audio 和 Video Extenders 管理和编程</li> <li>• 消息参考, 第 1 卷和第 2 卷</li> <li>• OLAP Integration Server Administration Guide</li> <li>• OLAP Integration Server Metaoutline User's Guide</li> <li>• OLAP Integration Server Model User's Guide</li> <li>• OLAP Integration Server User's Guide</li> <li>• OLAP 安装和用户指南</li> <li>• OLAP Spreadsheet Add-in Excel 版用户指南</li> <li>• OLAP Spreadsheet Add-in Lotus 1-2-3 版用户指南</li> <li>• Replication Guide and Reference</li> <li>• Spatial Extender Administration and Programming Guide</li> <li>• SQL 入门</li> <li>• SQL Reference, 第 1 卷和第 2 卷</li> <li>• System Monitor Guide and Reference</li> <li>• Text Extender 管理和编程</li> <li>• Troubleshooting Guide</li> <li>• 新增内容</li> </ul>

表 54. 订购打印书籍 (续)

SBOF 号	包括的书籍
SBOF-8935	<ul style="list-style-type: none"> <li>• Information Catalog Manager Administration Guide</li> <li>• Query Patroller Administration Guide</li> <li>• 信息目录管理器用户指南</li> <li>• Query Patroller 用户指南</li> <li>• Information Catalog Manager Programming Guide and Reference</li> </ul>

## DB2 联机文档

### 访问联机帮助

随所有 DB2 组件都附带提供了联机帮助。下表描述了各种类型的联机帮助。

帮助类型	内容	如何访问...
命令帮助	说明命令行处理器中命令的语法。	<p>从命令行处理器，以交互方式输入： <code>? command</code></p> <p>其中 <i>command</i> 表示一个关键字或整个命令。</p> <p>例如，<code>? catalog</code> 显示所有 CATALOG 命令的帮助，而 <code>? catalog database</code> 显示 CATALOG DATABASE 命令的帮助。</p>
客户机配置辅助程序帮助	说明您可在窗口或笔记本中执行的任务。此帮助包括您需要知道的概述和先决条件信息，并描述如何使用窗口或笔记本控件。	从窗口或笔记本，单击 <b>帮助</b> 按钮或按 <b>F1</b> 键。
命令中心帮助		
控制中心帮助		
数据仓库中心帮助		
事件分析程序帮助		
信息目录管理器帮助		
卫星管理中心帮助		
脚本中心帮助		

帮助类型	内容	如何访问...
消息帮助	描述消息的起因以及您应该执行的任何操作。	<p>从命令行处理器，以交互方式输入：  <code>? XXXnnnnn</code></p> <p>其中 <code>XXXnnnnn</code> 表示有效的消息标识符。</p> <p>例如，<code>? SQL30081</code> 显示关于 <code>SQL30081</code> 消息的帮助。</p> <p>要每次查看一屏消息帮助，可输入：  <code>? XXXnnnnn   尚有</code></p> <p>要在文件中保存消息帮助，可输入：  <code>? XXXnnnnn &gt; filename.ext</code></p> <p>其中 <code>filename.ext</code> 表示想要保存消息帮助的文件。</p>
SQL 帮助	说明 SQL 语句的语法。	<p>从命令行处理器，以交互方式输入：  <code>help statement</code></p> <p>其中，<code>statement</code> 表示 SQL 语句。</p> <p>例如，<code>help SELECT</code> 显示有关 <code>SELECT</code> 语句的帮助。  <b>注：</b>在基于 <code>UNIX</code> 的平台上，SQL 帮助不可用。</p>
SQLSTATE 帮助	说明 SQL 状态及类代码。	<p>从命令行处理器，以交互方式输入：  <code>? sqlstate</code> 或 <code>? class code</code></p> <p>其中，<code>sqlstate</code> 表示有效的五位 SQL 状态，而 <code>class code</code> 表示该 SQL 状态的头两位。</p> <p>例如，<code>? 08003</code> 显示 <code>08003</code> SQL 状态的帮助，而 <code>? 08</code> 显示 <code>08</code> 类代码的帮助。</p>

## 查看联机信息

此产品中的书籍为超文本标记语言 (HTML) 软拷贝格式。软拷贝格式使您可搜索或浏览信息，并提供访问相关信息的超文本链接。它还使得在站点间共享资料库更容易。

可使用遵循 HTML 版本 3.2 规范的任何浏览器来查看联机书籍或样本程序。

要查看联机书籍或样本程序：

- 如果正在运行 `DB2` 管理工具，则使用“信息中心”。

- 从浏览器，单击**文件** → **打开页**。打开的页中包含 DB2 信息的描述和至 DB2 信息的链接：

- 在基于 UNIX 的平台上，打开以下页：

`INSTHOME/sql1lib/doc/%L/html/index.htm`

其中 %L 表示语言环境名称

- 在其它平台上，打开以下页：

`sql1lib\doc\html\index.htm`

该路径位于安装了 DB2 的驱动器上。

如果尚未安装“信息中心”，则可通过双击 **DB2 信息** 图符来打开该页。视您正在使用的系统不同，图符在主产品文件夹中或在“Windows 开始”菜单中。

### 安装 Netscape 浏览器

如果还未安装 Web 浏览器，则可从产品包装箱中的 Netscape CD-ROM 安装 Netscape。要获取如何安装它的详细指示信息，执行：

1. 插入 Netscape CD-ROM。
2. 安装 CD-ROM（仅限于在基于 UNIX 的平台上）。参考《快速入门》以了解安装过程。
3. 关于安装说明，可参考 CDNAV *nn.txt* 文件，其中 *nn* 表示两字符语言标识符。该文件位于 CD-ROM 的根目录下。

### 通过“信息中心”访问信息

“信息中心”提供对 DB2 产品信息的快速访问。在所有装有 DB2 管理工具的平台上，都提供了“信息中心”。

可通过双击“信息中心”图符来打开“信息中心”。视正在使用的系统的不同，该图符在主产品文件夹的“信息”文件夹中，或在 Windows 的**开始**菜单中。

还可通过使用工具栏和 DB2 Windows 平台上的**帮助**菜单来访问“信息中心”。

“信息中心”提供了六种类型的信息。单击适当的标签来查看提供给该类型的主题。

<b>任务</b>	可使用 DB2 执行的关键任务。
<b>参考</b>	DB2 参考信息，如关键字、命令以及 API。
<b>书籍</b>	DB2 书籍。
<b>故障诊断</b>	错误消息类别及其恢复操作。

**样本程序** 随“DB2 应用程序开发客户机”一起提供的样本程序。如果未安装“DB2 应用程序开发客户机”，则不显示此标签。

**Web** 万维网（WWW）上的 DB2 信息。要访问此信息，必须从系统连接至 Web。

当选择其中一个列表中的项时，“信息中心”启动一个查看器来显示信息。视所选择的信息种类的不同，查看器可能是系统帮助查看器、编辑器或 Web 浏览器。

“信息中心”提供了查找功能部件，因此您不用浏览这些列表就能查找特定主题。

对于全文本搜索，请遵循“信息中心”中指向**搜索 DB2 联机信息**搜索表格的超文本链接。

HTML 搜索服务器通常是自动启动的。如果 HTML 信息中的搜索不起作用，则可能必须使用下列其中一个方法来启动搜索服务器：

#### 在 Windows 上

单击开始并选择程序 → IBM DB2 → 信息 → 启动 HTML 搜索服务器。

#### 在 OS/2 上

双击 **DB2 OS/2** 版文件夹，然后双击**启动 HTML 搜索服务器**图符。

如果在搜索 HTML 信息时遇到任何其它问题，可参考发行说明。

**注：**搜索功能在 Linux、PTX 和 Silicon Graphics IRIX 环境中不可用。

## 使用 DB2 向导

向导通过让您一次一步地完成每一个任务来协助您完成特定管理任务。可通过“控制中心”和“客户机配置辅助程序”来获取向导。下表列出了这些向导并描述了它们的用途。

**注：**“创建数据库”、“创建索引”、“配置多站点更新”和“性能配置”向导对分区数据库环境可用。

向导	帮助您...	如何访问...
添加数据库	在客户机工作站上编目数据库。	从“客户机配置辅助程序”单击添加。
备份数据库	确定、创建并调度备份计划。	从“控制中心”，右键单击想要备份的数据库并选择 <b>备份</b> → <b>数据库</b> （使用向导）。

向导	帮助您...	如何访问...
配置多站点更新	配置多站点更新、分布式事务或两阶段落实。	从“控制中心”，右键单击数据库文件夹并选择多站点更新。
创建数据库	创建数据库并执行一些基本配置任务。	从“控制中心”，右键单击数据库文件夹，并选择创建 → 数据库（使用向导）。
创建表	选择基本数据类型并创建表的主键。	从“控制中心”，右键单击表图符，并选择创建 → 表（使用向导）。
创建表空间	创建新的表空间。	从“控制中心”，右键单击表空间图符，并选择创建 → 表空间（使用向导）。
创建索引	建议对于所有查询要创建和删除哪些索引。	从“控制中心”，右键单击索引图符，并选择创建 → 索引（使用向导）。
性能配置	通过更新配置参数来调整数据库性能以满足您的业务需求。	<p>从“控制中心”，右键单击想要调整的数据库并选择使用向导配置性能。</p> <p>对于分区数据库环境，从“数据库分区”视图，右键单击想要调整的首个数据库分区并选择使用向导配置性能。</p>
复原数据库	在故障之后恢复数据库。它帮助您了解要使用的备份及要重放的纪录。	从“控制中心”，右键单击想要复原的数据库并选择恢复 → 数据库（使用向导）。

## 设置文档服务器

在缺省情况下，DB2 信息安装在本地系统上。这表示需要访问 DB2 信息的每个人都必须安装相同的文件。要将 DB2 信息存储在单个位置中，执行下列步骤：

1. 将所有文件和子目录从本地系统上的 \sql1lib\doc\html 复制至 Web 服务器。每一本书都有其自己的子目录，该子目录包含构成该书的所有必需的 HTML 和 GIF 文件。确保目录结构仍相同。
2. 配置 Web 服务器以查找新位置中的文件。有关信息，可参考《安装和配置补遗》中的 NetQuestion 附录。
3. 如果正在使用“信息中心”的 Java 版本，可为所有 HTML 文件指定基本的 URL。您应将该 URL 用于书籍列表。
4. 当能够查看书籍文件时，可将经常查看的主题做成书签。您可能想把下列各页做成书签：
  - 书籍列表

- 经常使用的书籍的目录
- 经常引用的文章，如 ALTER TABLE 主题
- 搜索格式

有关如何从中央机器处理“DB2 通用数据库”联机文档文件的信息，参考《安装和配置补遗》中的 NetQuestion 附录。

## 搜索联机信息

要查找 HTML 文件中的信息，使用下列方法之一：

- 在顶部框中单击**搜索**。使用搜索格式来查找特定的主题。此功能在 Linux、PTX 和 Silicon Graphics IRIX 环境中不可用。
- 在顶部框中单击**索引**。使用索引来查找书中的特定主题。
- 显示帮助或 HTML 书籍的目录或索引，然后使用 Web 浏览器的查找功能查找书中的特定主题。
- 使用 Web 浏览器的书签功能来快速返回至特定的主题。
- 使用“信息中心”的搜索功能来查找特定的主题。请参见第413页的『通过“信息中心”访问信息』以获取详细信息。



---

## 附录M. 声明

IBM 可能未在所有国家或地区中提供本文档中讨论的产品、服务或功能部件。关于您所在区域目前可用的产品及服务的信息，请向当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并不明示或默示只能使用 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，都可以用来代替 IBM 产品、程序或服务。然而，评估和验证任何非 IBM 产品、程序或服务均由用户自行负责。

IBM 可能已经申请或正在申请与本文档有关的各项专利权。提供本文档并不表示允许您使用这些专利。 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，请与您的国家或地区的“IBM 知识产权部”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

以下段落不适用于联合王国或该条款与当地法律不一致的任何国家或地区：国际商业机器公司以“仅此状态”的基础提供此出版物，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）不侵犯、适销性或适用于某特定用途的默示保证或条件。一些国家或地区允许否认某些事务中的明示或默示保证，因此，此声明可能不适用于您。

本资料可能会包含技术错误或印刷错误。此处的信息会定期得到更改；这些更改将编入本出版物的新版本中。IBM 可能随时对此出版物中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

在此信息中对非 IBM Web 站点的任何引用仅是出于方便起见，不以任何方式提供对这些 Web 站点的保证。这些 Web 站点中的资料不是此 IBM 产品资料的一部分，使用这些 Web 站点时风险自负。

IBM 对于您以任何方式提供的任何信息，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。

为了以下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换（ii）允许对已经交换的信息进行相互使用，而希望获取本程序有关信息的被许可方请与以下地址联系：

IBM Canada Limited  
Office of the Lab Director  
1150 Eglinton Ave. East  
North York, Ontario  
M3C 1H7  
CANADA

只要遵守适当的条款和条件，包括某些情形下的一定数量的付款，都可获取这方面的信息。

本资料中描述的许可程序和它可用的全部许可证材料均由 IBM 根据“IBM 客户协议”、“IBM 国际程序许可证协议”或任何与客户之间的等效协议中的条款提供。

此处包含的所有性能数据都是在受控环境中确定的。因此，在其他操作环境中获得的结果可能与之相差很大。某些测量可能是在开发级的系统上进行的，不能保证这些测量方法在通用系统上同样可用。此外，某些测量方法可能是通过外推法归纳来估计的。实际结果可能会有所不同。此文档的用户应针对他们的特定环境验证数据是否适用。

关于非 IBM 产品的信息是从那些产品的供应商、他们发布的声明或其他公用来源获得的。IBM 未测试那些产品，不能确认与非 IBM 产品相关的性能、兼容性或任何其他声明的准确性。如有关于非 IBM 产品的功能的问题，应向那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能随时更改或撤销，而不作任何通知，并且仅代表发展目标。

本资料中可能包含用于日常业务运作的的数据或报表的示例。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相似，纯属巧合。

版权许可证：

本资料可能包含源语言的样本应用程序，它们举例说明各种操作平台上的编程技术。为了开发、使用、市场营销或分发符合编写这些样本程序所针对的操作系统的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

以星号 (\*) 标出的下列各项是国际商业机器公司在美国和 / 或其他国家或地区的商标。

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extender	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational	SystemView
Database Architecture	VisualAge
DRDA	VM/ESA
eNetwork	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WIN-OS/2

下列各项是其他公司的商标或注册商标:

Microsoft、Windows 和 Windows NT 是 Microsoft Corporation 的商标或注册商标。

Java 或所有基于 Java 的商标和徽标以及 Solaris 是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

Tivoli 和 NetView 是 Tivoli Systems Inc. 在美国和 / 或其他国家或地区的商标。

UNIX 是经 X/Open Company Limited 唯一许可的在美国和 / 或其他国家或地区的注册商标。

以双星号 (\*\*) 标出的其他公司、产品或服务名称, 可能是其他公司的商标或服务标记。

# 索引

## [ A ]

### 安全性

- 分布式计算环境 (DCE) 的目录服务 292
- 服务器选项 208
- 服务, Windows NT 326
- 计划 193
- 联合服务器认证示例 209
- 联合数据库的标识和密码处理 207
- 联合系统的 APPC 设置 209
- 认证, 联合数据库 206
- 用户映射 207
- CLIENT 级 197
- DCS 处理, 联合系统 207

### 安装

- Netscape 浏览器 413

## [ B ]

### 绑定

- 重新绑定无效软件包 226
- 命令行处理器 96
- 数据库实用程序 96

### 包装器, 创建 132

### 备份数据库向导 5, 414

### 备份域控制器

- 配置 DB2 326, 327

### 本地数据库目录

- REORG实用程序 93

### 编目数据库 97

### 标量 UDF 122

### 标识别名 140

### 标识列 114

- 改变 177

### 标准输入 311

### 表

- 标识列 114
- 重命名 179
- 定义参考约束 109

### 表 (续)

- 定义检查约束 112
- 定义唯一约束 108
- 分配到节点组 95
- 改变 165
- 更改分区键 176
- 更改属性 176
- 检索具有访问权的名称 235
- 临时 91
- 命名 105
- 取消特权 225
- 删除 180
- 生成列 112, 172
- 添加参考约束 168
- 易失的 175
- 在分区数据库中创建 119
- ALTER TABLE 语句 166
- CREATE TABLE 语句 104

### 表达式

- NEXTVAL 116
- PREVVAL 116

### 表检查约束

- 定义 112
- 删除 172
- 添加 170

### 表空间

- 重命名 162
- 创建 97
- 创建数据库时的缺省值 91
- 调整容器的大小 160
- 分开不同类型的数据, 示例 118
- 更改 159
- 检查可用空间 (DMS) 38
- 扩充容器 160
- 删除 163
- 删除系统临时 163
- 删除用户临时 164
- 设备容器示例 99
- 设置为 ONLINE 状态 162
- 特权 220
- 添加容量 38

### 表空间 (续)

- 添加容器 160
- 文件容器示例 98
- 文件系统容器示例 98
- 系统临时 100
- 用户临时 100
- 在节点组中 101

### 表 UDF 122

### 别名

- 创建 139
- 权限 131
- 软件包特权处理 228
- 使用 131
- 数据源的视图 229
- 特权 222

### 别名 (DB2 MVS/ESA 版) 132

### 别名, 创建 131

### 并行性

- 启用 49

### 并行性, 分区内

- 启用 49

### 不可信的客户机 197

## [ C ]

### 参考约束

- 定义 109
- 添加至表 168
- FOREIGN KEY 子句,  
CREATE/ALTER TABLE 语句 109
- PRIMARY KEY 子句,  
CREATE/ALTER TABLE 语句 109
- REFERENCES 子句,  
CREATE/ALTER TABLE 语句 109

### 层次结构表 117

- 删除 181

### 查看

- 联机信息 412

- 查询重写
  - 摘要表 129
- 查找对象 16
- 重命名表 179
- 重命名表空间 162
- 重新分布数据
  - 跨节点 159
- 触发器
  - 创建 120
  - 从属性 121
  - 删除 182
  - 优点 120
- 创建包装器 132
- 创建表 104
- 创建表空间 97
- 创建表空间向导 5, 415
- 创建表向导 5, 415
- 创建别名 131, 139
- 创建触发器 120
- 创建服务器 133
- 创建函数模板 124
- 创建函数映射 123
- 创建类型表 117
- 创建类型视图 129
- 创建类型映射 126
- 创建模式 103
- 创建视图 127
- 创建数据库向导 5, 415
- 创建索引 141
- 创建索引规范 141
- 创建索引扩充 141
- 创建用户定义单值类型 125
- 创建用户定义函数 122
- 创建用户定义结构化类型 126
- 创建用户定义类型 125
- 创建 LDAP 用户 357
- 从机器列表中删除重复项 316
- 从类型表中删除行 179
- 存储
  - 管理 37

## [ D ]

- 打开日志 20
- 打印 PDF 书籍 409

- 大对象 (LOB)
  - 列注意事项 107
- 大小
  - 估计 37
- 单元目录服务 (CDS) 283
- 调度
  - 保存的命令脚本 19
- 调用层接口 (CLI)
  - 与数据库绑定 96
- 定义表检查约束 112
- 定义参考约束 109
- 定义唯一约束 108
- 动态 SQL
  - 数据库访问的 EXECUTE 特权 228
- 对象
  - 显示相关 7
- 对象类属性
  - DB\_Authentication (DAU) 289
  - DB\_Comment (DCO) 289
  - DB\_Communication\_Protocol 289
  - DB\_Database\_Locator\_Name 290
  - DB\_Database\_Protocol 290
  - DB\_Native\_Database\_Name 290
  - DB\_Object\_Type 290
  - DB\_Principal (DPR) 289
  - DB\_Product\_Name 291
  - DB\_Product\_Release 291
  - DB\_Target\_Database\_Info 291
- 多个实例 47

## [ F ]

- 发行说明 409
- 访问控制 196
  - 表的视图 229
  - 认证 196
  - 数据库对象 224
  - 数据库管理器 224
- 非唯一索引
  - 删除 189
- 非主索引
  - 删除 189
  - 删除应用程序的蕴含项 189
- 分布式计算环境 (DCE)
  - 安全性服务 201

- 分布式计算环境 (DCE) (续)
  - 临时替换 DCE 目录信息 303
  - 目录服务的概述 94
  - 目录服务的限制 306
  - 目录服务任务 303
  - 配置参数和注册表变量 294
  - 认证 201
    - 如何搜索目录 300
    - 设置 DB2 服务器 203
    - 设置 DB2 客户机实例 205
    - 设置 DB2 用户 202
    - 使用目录服务 304
    - 限制 205
  - ATTACH 命令 295, 300
  - CATALOG GLOBAL DATABASE 295
  - CDS 283
  - CONNECT 语句 295, 301
  - GDS 283
- 分区键
  - 表的注意事项 119
  - 更改 176
  - 依据分区键分区的索引 143
- 分区内并行性
  - 启用 49
- 服务器
  - 创建 133
  - 特权 222
- 服务器选项
  - 安全性详细信息 208
  - 密码 208
  - collating\_sequence 136
  - comm\_rate 136
  - connectstring 136
  - cpu\_ratio 136
  - dbname 137
  - fold\_id 137, 208
  - fold\_pw 137, 208
  - io\_ratio 137
  - node 138
  - password 138
  - plan\_hints 138
  - pushdown 138
  - varchar\_no\_trailing\_blanks 138
- 复原数据库向导 5
- 复原向导 415

复制 269  
    配置 87  
复制数据 39

## [ G ]

改变表 165  
改变表空间 159  
改变别名 187  
改变服务器 186  
改变节点组 159  
改变结构化类型 179  
改变列 166  
改变视图 183  
改变约束 168  
改变摘要表特性 178  
改变 IDENTITY 列 167  
高速通信 343  
跟踪  
    审计 239  
更改表属性 176  
更改分区键 176  
更改环境变量 156  
更改节点配置文件 157  
更改数据库配置 157  
更改注册表变量 156  
更新类型表 179  
更新实例列表 83  
更新 DAS 配置 84  
公共互连 344  
故障诊断 39  
管理  
    使用 GUI 工具 3  
管理服务器 13, 69  
管理工具  
    概述 4  
    脚本中心 18  
    命令中心 18  
过滤器 8

## [ H ]

函数  
    DECRYPT 232  
    ENCRYPT 232  
    GETHINT 232

函数调用  
    选择性 151  
函数模板, 创建 124  
函数映射, 创建 123  
行  
    除去 167  
环境变量 61  
    更改 156  
    在 OS/2 上设置 65  
    在 UNIX 上的设置 67  
    在 Windows 95 上的设置 65  
    在 Windows NT 上的设置 65  
rah 317  
RAHDOTFILES 318  
恢复  
    创建数据库期间分配日志 96  
    概述 273, 307  
恢复不起作用的视图 184  
恢复不起作用的摘要表 186  
恢复日志 96  
活动目录 355  
    安全性 368  
    对象 371  
    扩展目录模式 369  
    正在配置 356  
    支持 356

## [ J ]

记录  
    审计 239  
    原始设备 101  
加密  
    数据 232  
监视  
    rah 进程 312  
检索数据  
    索引 143  
简要表注册表 61  
将标识和密码传送到数据源 207  
将数据分区 50  
脚本中心 18  
    使用现存的脚本 19  
节点号 85  
节点级简要表注册表 62  
节点间通信 343

节点配置文件  
    创建 84  
    更改 157  
节点组  
    表的注意事项 119  
    初始定义 90  
    创建 95  
    分区键, 更改 176  
    改变 159  
    IBMDEFAULTGROUP, 在缺省情  
    况下创建的表 119  
结构化类型  
    改变 179  
静态SQL  
    数据库访问的 EXECUTE 特权  
    228  
聚合函数 122

## [ K ]

可信的客户机  
    认证 197  
    CLIENT 级安全性 197  
客户机  
    不可信的 197  
    管理 4  
    可信的 197  
空值  
    列定义 105  
控制中心  
    定制 11  
    显示系统 13  
控制中心作为 Java 小应用程序 41  
控制 rah 命令 317  
块结构化设备 99

## [ L ]

类型表  
    层次结构表 117  
    创建 117  
    更新行 179  
    删除行 179  
    填充 117  
类型视图, 创建 129  
类型映射, 创建 126

- 联合数据库
  - 包装器, 创建 132
  - 别名, 标识 140
  - 别名, 创建 139
  - 别名, 使用 140
  - 服务器选项, 安全性 208
  - 服务器, 创建 133
  - 函数模板, 创建 124
  - 函数映射, 创建 123
  - 将标识和密码传送到数据源 207
  - 类型映射, 创建 126
  - 区分大小写的名称 280
  - 认证 206
  - 认证示例 209
  - 索引规范, 创建 141
  - 引用别名 140
  - 用户映射, 创建 207
  - APPC 设置 209
  - DCS 设置 207
- 联机帮助 411
- 联机重组
  - 索引 143
- 联机信息
  - 查看 412
  - 搜索 416
- 列
  - 定义 105
  - 改变 166
  - 添加 165
- 列选项
  - 数字串 188
  - varchar\_no\_trailing\_blanks 188
- 列 UDF 122
- 临时表
  - 删除用户定义 182
  - 用户定义 113
- 路由信息对象
  - 创建 286
  - 示例 286
- 逻辑节点
  - 多个 341

**[ M ]**

- 命令
  - 以并行方式运行 312

- 命令 (续)
  - CATALOG GLOBAL DATABASE 295
- 命令行处理器
  - 与数据库绑定 96
- 命令中心 18
- 命名约定
  - 一般 277
  - Windows NT 限制 326
- 模式
  - 创建 103
  - 删除 164
  - REORG实用程序 48
  - SESSION 182
- 目录
  - 本地数据库目录 93
  - 节点目录 94
  - 系统数据库目录 93
- 目录对象
  - 创建 283
  - 对象类属性 288
- 目录高速缓存
  - 对编目数据库的影响 97
- 目录节点
  - 描述 51

**[ P ]**

- 配置参数
  - 分布式计算环境 (DCE) 294
  - 分区数据库 51
- 配置多站点更新向导 5, 414
- 配置 LDAP 357

**[ Q ]**

- 启动 DB2 46
- 前缀序列 315
- 轻量级目录访问协议 95
  - 安全性 367
  - 编目节点项 361
  - 对象类和属性 371
  - 更新协议信息 360
  - 禁用 367
  - 扩展目录模式 369
  - 配置主机数据库 364

- 轻量级目录访问协议 (续)
  - 启用 366
  - 设置注册表变量 365
  - 刷新项 362
  - 搜索 363
  - 远程连接 362
  - 注册数据库 361
  - 注销服务器 361
  - 注销数据库 362
  - DB2 Connect 367
  - IBM eNetwork 目录 369
  - Windows 2000 活动目录 369
- 轻量级目录访问协议 (LDAP) 40, 355
  - 区分大小写的名称, 联合数据库 280
  - 取消权限和特权 34
  - 全局级简要表注册表 62
  - 全局目录服务 (GDS) 283
  - 权限 213
    - 从 SYSADM 中除去 DBADM 214
    - 从 SYSCTRL 中除去 DBADM 214
  - 级别 211
  - 取消 34
  - 任务和必需的权限 233
  - 授权 34
  - 数据库管理 (DBADM) 215, 218
  - 系统控制 (SYSCTRL) 214
  - 系统维护 (SYSMAINT) 215
- 缺省属性规范 105
- 确定rah的问题 319

**[ R ]**

- 认证 196
  - 定义 196
  - 分布式计算环境 (DCE) 的目录服务 292
  - 分区数据库注意事项 201
  - 联合数据库处理 206
  - 域安全性 328
  - 远程客户机 200
  - 组 328
    - DCE 安全性服务 201
  - 认证类型 196



## 认证类型 (续)

- CLIENT 197
- DCE 199
- DCE\_SERVER\_ENCRYPT 199
- DCS 198
- DCS\_ENCRYPT 198
- KERBEROS 199
- KRB\_SERVER\_ENCRYPT 199
- SERVER 196
- SERVER\_ENCRYPT 196

## 日志

- 审计 239

## 容器

- 添加 (至 DMS 表空间) 160
- 添加至 SMS 表空间 162
- 修改 (对 DMS 表空间) 160

## 软件包

- 不起作用 189
- 从属性 189
- 取消特权 226
- 删除 189
- 所有者 227
- 特权 223
- 添加外键之后无效 169
- SQL 的访问特权 227

## [ S ]

- 删除包装器 186
- 删除表 180
- 删除表检查约束 172
- 删除别名 187
- 删除触发器 182
- 删除服务器 186
- 删除模式 164
- 删除视图 183
- 删除数据库 159
- 删除索引 188
- 删除索引规范 188
- 删除索引扩充 188
- 删除唯一约束 171
- 删除系统临时表空间 163
- 删除用户表空间 163
- 删除用户定义表 182
- 删除用户定义函数 182
- 删除用户定义类型 183

## 删除用户临时表空间 164

- 删除约束 171
- 删除摘要表 185
- 设计, 实现 45
- 设置
  - rah 的缺省环境简要表 319
- 设置模式 104
- 设置文档服务器 415
- 设置 VARCHAR 166
- 审计跟踪 239
- 审计活动 239
- 审计设施

### 参数描述 243

- 操作 239
- 错误处理 241
- 行为 241
- 记录条目的布局 246
- 记录条目的同步写入 241
- 记录条目的异步写入 241
- 检查事件表 247
- 控制活动 262
- 权限 / 特权 239
- 审计事件表 247
- 使用方案 242
- 事件 240
- 示例 262
- 提示和技巧 261
- 消息 246
- 语法 242
- CONTEXT 事件表 260
- ERRORTYPE 参数 241
- OBJMAINT 事件表 251
- SECMAINT 事件表 253
- SYSADMIN 事件表 257
- VALIDATE 事件表 259

## 生成列 112, 172

## 实例

- 除去 156
- 创建 54
- 定义 54
- 分区服务器, 更改 337
- 分区服务器, 删除 338
- 分区服务器, 添加 335
- 改变 153
- 更新 154
- 列出 60, 154

## 实例 (续)

- 列示数据库分区服务器 335
- 目录 54
- 启动 46
- 缺点 54
- 缺省值 54
- 设置当前 60
- 使用的原因 54
- 所有者 56
- 停止 52
- 显示 13
- 运行多个 61
- 自动启动 61
- add 57
- REORG实用程序 47
- 实例级简要表注册表 62
- 实例简要表注册表 62
- 实例所有者 56
- 实例用户
  - 设置环境 55
- 视图
  - 不起作用 185
  - 创建 127
  - 对表的访问控制 229
  - 访问特权, 示例 229
  - 改变 183
  - 行访问 229
  - 恢复不起作用的 184
  - 列访问 229
  - 删除 183
  - 删除系统目录的蕴含项 183
  - 数据安全性 127
  - 数据完整性 127
  - 特权信息 237
  - 限制 183
  - CHECK OPTION 子句, CREATE VIEW 语句 128
- 授权
  - 定义 211
  - 可信的客户机 197
  - 系统管理 (SYSADM) 213
- 授权名称
  - 检索具有表访问权限的名称 235
  - 检索具有 DBADM 权限的全部名称 235
  - 检索授予的特权 236

- 授权名称 (续)
    - 检索特权信息 234
    - 为特权信息创建视图 237
  - 授予权限和特权 34
  - 书籍 401, 410
  - 数据
    - 更改分布 159
    - 移动 35
  - 数据安全性
    - 保密系统目录 236
    - 重要性 193
    - 控制数据库访问 193
  - 数据传送
    - REORG 实用程序 269
  - 数据定义语言 (DDL)
    - 生成 7
  - 数据复制 269
  - 数据加密 232
  - 数据库 45
    - 编目 97
    - 创建 89
    - 创建之前 45
    - 改变节点组 159
    - 更改 158
    - 更改数据的分布 159
    - 更改之前的注意事项 153
    - 恢复日志 96
    - 启用数据分区 50
    - 软件包从属性 189
    - 删除 159
    - 远程, 管理 32
    - 在所有节点上创建 51
    - 注意事项, 创建 53
  - 数据库的设计
    - 改变 153
  - 数据库定位器对象
    - 创建 285
    - 示例 285
  - 数据库对象
    - 创建 284
    - 访问控制 224
    - 示例 284
  - 数据库访问
    - 控制 193
    - 通过带 SQL 的软件包授予的特权 227
  - 数据库分区服务器
    - 发出命令 309
    - Windows 2000 335
    - Windows NT 335
  - 数据库管理器
    - 绑定实用程序 96
    - 访问控制 224
    - 启动 46
    - 索引 143
    - 停止 52
  - 数据库管理员 (DBADM) 权限
    - 检索名称 235
    - 特权 215
  - 数据库配置
    - 创建的文件 86
    - 更改 157
  - 数据类型
    - 多字节字符集 105
    - 列定义 105
  - 数据完整性
    - 唯一索引 141
  - 数字串列选项 188
  - 刷新摘要表中的数据 179
  - 双字节字符集用户
    - 数据类型 105
  - 搜索
    - 联机信息 414, 416
  - 索引
    - 创建 141
    - 定义 142
    - 非唯一 145
    - 非主 189
    - 更改 188
    - 联机重组 143, 146
    - 如何使用 144
    - 特权 223
    - 唯一的 145
    - 选择性 151
    - 用户定义扩充 147
    - 优化数量 142
    - 主 109
    - 主的与用户定义的 142
    - CREATE INDEX 语句 145
    - CREATE UNIQUE INDEX statement 145
    - DROP INDEX 语句 188
  - 索引键, 定义 142
  - 索引扩充 141
  - 索引向导 5, 415
- [ T ]**
- 特权
    - 表 220
    - 表空间 220
    - 别名 222
    - 层次结构 212
    - 定义 211
    - 对软件包的隐式 213
    - 服务器 222
    - 个别的 213
    - 间接特权, 别名 228
    - 检索名称 236
    - 检索授权名 234
    - 模式 218
    - 取消 34
    - 任务和必需的权限 233
    - 软件包 223
    - 视图 220
    - 授权 34
    - 授予和取消权限 216
    - 数据库管理器 216
    - 所有权 (CONTROL) 213
    - 为信息创建视图 237
    - 系统目录列表 234
    - 有别名的视图 229
    - 摘要 211
    - ALTER 220
    - BINDADD 216
    - CONNECT 216
    - CONTROL 220
    - CREATETAB 216
    - CREATE\_NOT\_FENCED 216
    - DELETE 220
    - GRANT 语句 224
    - IMPLICIT\_SCHEMA 216
    - INDEX 223
    - INSERT 220
    - PUBLIC 218
    - REFERENCES 220
    - REVOKE 语句 225
    - SELECT 220

特权 (续)

- USAGE 224
- 添加表检查约束 170
- 添加数据库向导 414, 415
- 添加唯一约束 168
- 添加约束 168
- 添加作用域 166
- 填充类型表 117
- 停止 DB2 52
- 通信
  - 高速 343
- 通信协议
  - VI 体系结构 345
- 同义词 (DB2 MVS/ESA 版) 132

## [ W ]

外键

- 删除必需的特权 171
- 添加 168
- 外键定义的规则 111
- 约束名 111
- 装入实用程序, 参考完整性含意 111
- 组合 111
- “导入”实用程序, 参考完整性含意 112
- DROP FOREIGN KEY 子句,  
ALTER TABLE语句 171
- 网关连接 15
- 唯一约束
  - 定义 108
  - 删除 171
  - 添加 168
- 为应用程序配置 LDAP 用户 358

## [ X ]

- 稀疏文件分配 108
- 系统管理 (SYSADM) 权限 213
  - 概述 213
  - 特权 213
- 系统临时表空间 100
- 系统目录
  - 安全性 236
  - 检索具有表访问权限的名称 235

系统目录 (续)

- 检索具有特权的授权名 234
- 检索具有 DBADM 权限的全部名称 235
- 检索授予名称的特权 236
- 删除表 180
- 删除视图蕴含项 184
- 设置 92
- 特权列表 234
- 添加新列 165
- 系统目录表
  - 存储在数据库目录节点上 51
- 系统数据库目录

REORG实用程序 93, 94

显式模式使用 48

显示相关对象 7

显示 SQL 语句 6

限定目标名 48

限制

Windows NT 命名 326

向导 4

备份数据库 5, 414

创建表 5, 415

创建表空间 5, 415

创建数据库 5, 415

复原数据库 5, 415

配置多站点更新 5, 414

索引 5, 415

添加数据库 414, 415

完成任务 414

性能配置 5, 157, 415

消息

审计设施 246

信息中心 413

性能

访问远程信息 333

复位值 334

目录信息, 减少争用 51

显示信息 332

允许远程访问信息 332

摘要表 129

性能监视器

Windows NT 331

性能配置向导 5, 157, 415

修改表 165

修改列 166

虚拟接口 (VI) 体系结构 345, 346

虚拟远程通信访问方法 (VTAM) 294

许可证管理 61

许可证信息

改变 153

许可证中心 20

序列 117

创建 115

改变 177

删除 178

特权 224

选择性 151

## [ Y ]

样本程序

跨平台 407

HTML 407

移动数据 269

已处理的设备 99

隐式模式使用 48

用户

管理 34

用户定义单值类型, 创建 125

用户定义函数 (UDF)

创建 122

创建未保护的特权 217

类型 122

删除 182

用户定义结构化类型, 创建 126

用户定义类型 (UDT)

创建 125

删除 183

用户定义临时表 113, 182

用户临时表空间 100

用户认证

Windows NT 326

用户映射

创建 207

语言标识符

书籍 408

域安全性

认证 328

原始日志 101

原始设备 99

原始 I/O 101

- 远程管理 75
- 远程系统 33
- 约束
  - 定义唯一的 108
  - 更改 168
  - 删除 171
  - 添加 168
- 约束名
  - 定义表检查约束 112
  - 定义外键 111
  
- [ Z ]
- 在多个表空间中创建表 118
- 摘要表
  - 创建 129
  - 改变特性 178
  - 恢复不起作用的 186
  - 删除 185
  - 刷新数据 179
  - 自动 130
- 主键
  - 何时创建 109
  - 删除必需的特权 171
  - 添加 168
  - 主索引 109
  - 主索引, 创建 142
  - DROP PRIMARY KEY 子句,  
ALTER TABLE 语句 171
- 主索引
  - 删除 189
  - 主键的唯一性 109
- 注册表变量 61
  - 分布式计算环境 (DCE) 294
  - 更改 156
- 专用互连 345, 346
- 资源访问控制设施 (RACF) 294
- 自动摘要表 130
- 字符串行设备 99
- 最新信息 409
- 作用域
  - 添加 166

## [ 特别字符 ]

- “装入”实用程序
  - 概述 269

## A

- ALTER 特权
  - 定义 220
- ALTER COLUMN 166
- ALTER NICKNAME 语句, 示例 187
- ALTER SERVER 语句, 示例 186
- ALTER TABLE 语句
  - 删除检查约束的示例 172
  - 删除键的示例 172
  - 删除唯一约束的示例 171
  - 添加检查约束的示例 170
  - 添加键的示例 169
  - 添加列的示例 166
  - 添加唯一约束的示例 168
  - 添加约束的提示 168
- ALTER TABLESPACE 语句
  - 示例 160
- ALTER VIEW 语句; 示例 184
- ATTACH 命令
  - 指定分布式计算环境 (DCE) 信息 295
  - REORG实用程序 48
- audit\_buf\_sz 241

## B

- BIND 命令
  - OWNER 选项 227
- BIND 特权
  - 定义 223
- BINDADD 特权
  - 定义 217

## C

- CATALOG DATABASE
  - 示例 97

- CATALOG GLOBAL DATABASE
  - 指定分布式计算环境 (DCE) 信息 295
- CDS 283
- CLIENT 级安全性 197
- CLIENT, 认证类型 197
- collating\_sequence 服务器选项 136
- comm\_rate 服务器选项 136
- CONNECT 特权
  - 定义 217
- CONNECT 语句
  - 指定分布式计算环境 (DCE) 信息 295
- connectstring 服务器选项 136
- CONTROL 特权
  - 定义 220
  - 软件包特权 223
  - 隐式发出 227
- cpu\_ratio 服务器选项 136
- CREATE ALIAS 语句
  - 使用 131
  - 示例 132
- CREATE DATABASE 命令
  - 示例 89
- CREATE INDEX 语句
  - 联机重组 143, 146
  - 示例 145
  - 唯一索引 145
- CREATE NICKNAME 139
- CREATE SERVER 133
- CREATE TABLE 语句
  - 定义参考约束 110
  - 定义检查约束 112
  - 使用多个表空间 118
  - 示例 106
- CREATE TABLESPACE 语句
  - 示例 98
- CREATE TRIGGER 语句
  - 示例 121
- CREATE VIEW 语句
  - 更改列名 128
  - 示例 128
- CREATE WRAPPER 132
- CREATETAB 特权
  - 定义 217

CREATE\_NOT\_FENCED 特权  
    定义 217  
CURRENT\_SCHEMA 104  
CURRENT\_SCHEMA 专用寄存器 48

## D

DataPropagator Relational (DPROPR)  
    概述 269  
DAU (DB\_Authentication) 289  
DB2  
    在 Windows NT 上启动 47  
DB2 管理服务器  
    更新配置 84  
    更新实例列表 83  
    使用“客户机配置辅助程序”和  
    “控制中心” 83  
DB2 管理服务器 (DAS) 75  
    安全性 79  
    安全性注意事项 73  
    除去 74  
    创建 70  
    对分区数据库系统进行设置 75  
        示例 75  
    分区数据库系统 (UNIX) 中的节点  
    间管理通信 78  
    分区数据库系统 (Windows NT) 中  
    的节点间管理通信 79  
    服务端点 78  
    概述 69  
    更新 74  
    环境 80  
    节点间管理通信 78  
    列出 73  
    配置 77  
    启动和停止 72  
    所有权规则 68  
    通信 77  
    允许查找 80  
    正在配置 73  
    注册表变量 80  
    注册表变量注意事项 80  
        “控制中心”通信 77  
    UNIX EEE 服务器 78  
    Windows NT EEE 服务器 79  
DB2 资料库  
    查看联机信息 412  
    打印 PDF 书籍 409  
    订购打印书籍 410  
    结构 401  
    联机帮助 411  
    设置文档服务器 415  
    书籍 401  
    书籍的语言标识符 408  
    搜索联机信息 416  
    向导 414  
    信息中心 413  
    最新信息 409  
DB2 Connect 269  
DB2 OS/390 版  
    对象, 管理 14  
    子系统, 添加 14  
DB2 Windows NT 版性能计数器  
    331  
db2audit 242  
db2audit.log 239  
db2dmnbckctlr  
    使用 326, 327  
db2gncol 实用程序 173  
db2icrt 命令 57  
db2idrop 156  
db2ilist 154  
DB2INSTANCE 环境变量  
    定义缺省实例 47  
db2iupdt 154  
DB2LDAP\_CLIENT\_PROVIDER 356  
db2ldcfg 实用程序 358  
db2nchg 337  
db2ncrt 335  
db2ndrop 338  
db2nlist 335  
db2nodes.cfg 文件 84  
db2perfc 334  
db2perfi 331  
db2perfr 332  
db2set 命令 62, 63  
db2start 命令 46  
db2stop 命令 52  
db2\_all 309, 310, 311  
db2\_call\_stack 310  
DB2\_INDEX\_2BYTEVARLEN 142  
db2\_kill 310  
dbname 服务器选项 137  
DB\_Authentication (DAU) 289  
DB\_Comment (DCO) 289  
DB\_Communication\_Protocol  
    (DCP) 289  
DB\_Database\_Locator\_Name  
    (DLN) 290  
DB\_Database\_Protocol (DDP) 290  
DB\_Native\_Database\_Name  
    (DNN) 290  
DB\_Object\_Type (DOT) 290  
DB\_Principal (DPR) 289  
DB\_Product\_Name (DPN) 291  
DB\_Product\_Release (DRL) 291  
DB\_Target\_Database\_Info (DTI) 291  
DCE 网络数据库  
    创建 297  
    连接 298, 299  
DCE, 认证类型 199  
DCE\_SERVER\_ENCRYPT, 认证类  
    型 199  
DCO (DB\_Comment) 289  
DCP  
    (DB\_Communication\_Protocol) 289  
DCS  
    联合数据库处理 207  
    认证类型 198  
DCS\_ENCRYPT, 认证类型 198  
DDP (DB\_Database\_Protocol) 290  
DECLARE GLOBAL TEMPORARY  
    TABLE 113  
DELETE 特权  
    定义 220  
DETACH 命令  
    REORG实用程序 48  
Discovery  
    配置 84  
    设置参数 82  
    隐藏服务器实例 82  
DLN  
    (DB\_Database\_Locator\_Name) 290  
DMS 表空间  
    创建 99  
DNN  
    (DB\_Native\_Database\_Name) 290

DOT (DB\_Object\_Type) 290  
DPN (DB\_Product\_Name) 291  
DPR (DB\_Principal) 289  
DPROPR 269  
DRL (DB\_Product\_Release) 291  
DROP DATABASE 命令  
    示例 159  
DROP INDEX 语句; 示例 188  
DROP NICKNAME 语句, 示例 187  
DROP SERVER 语句, 示例 187  
DROP TABLE 语句  
    示例 180  
DROP TABLESPACE 语句; 示例  
    163  
DROP VIEW 语句; 示例 184  
DTI (DB\_Target\_Database\_Info) 291

## E

EXECUTE 特权  
    定义 223  
    动态 SQL 的数据库访问 228  
    静态 SQL 的数据库访问 228

## F

FCM 通信 87  
fold\_id 服务器选项 137  
fold\_pw 服务器选项 137  
FOREIGN KEY 子句  
    参考约束 111  
    外键定义的规则 111

## G

GDS 283  
GRANT  
    示例 224  
GRANT 语句  
    安全性 294  
    使用 224  
    隐式发出 227  
GUI 工具  
    管理使用 3

## H

HTML  
    样本程序 407

## I

IBM eNetwork 目录  
    对象类和属性 371  
    扩展目录模式 369  
IBMCATGROUP 节点组 90  
IBMDEFAULTGROUP 节点组 90  
IBMTEMPGROUP 节点组 90  
IDENTITY 列 117  
IMPLICIT\_SCHEMA 权限 103  
IMPLICIT\_SCHEMA 特权  
    定义 217  
IMPORT 实用程序  
    参考完整性意义 112  
    与数据库绑定 96  
    LOAD 111  
INDEX 特权  
    定义 220  
INSERT 特权  
    定义 220  
io\_ratio 服务器选项 137

## J

Java 小应用程序 41

## K

Kerberos 安全性协议 199  
KERBEROS, 认证类型 199  
KRB\_SERVER\_ENCRYPT, 认证类型  
    199

## L

LDAP 40, 95, 355  
LDAP 配置  
    支持 355  
LOAD 权限 216

## M

MINPCTUSED 子句 146

## N

Netscape 浏览器  
    安装 413  
NEXTVAL 116  
node 49  
    编目 51  
    在节点组中更改 159  
    在所有上创建数据库 51  
node 服务器选项 138

## P

password 服务器选项 138  
PDF 409  
plan\_hints 服务器选项 138  
PRECOMPILE 命令  
    OWNER 选项 227  
PREVVAL 116  
PRIMARY KEY 子句  
    添加主键 168  
    限制 109  
PUBLIC  
    特权 218  
pushdown 服务器选项 138

## R

rah 310  
    环境变量 317  
    监视进程 312  
    介绍 309  
    控制 317  
    设置缺省环境简要表 319  
    以并行方式运行命令 312  
    指定机器列表 316  
RAHDOTFILES 318  
RAHOSTFILE 316  
RAHOSTLIST 316  
RAHWAITTIME 312  
RAHCHECKBUF 312  
RAHTREETHRESH 313

## REFERENCES 特权

定义 220

## REFERENCES 子句

参考约束 111

删除规则 111

使用 111

添加外键 168

## REORG 实用程序

与数据库绑定 96

## REVOKE 语句

安全性 294

使用 225

示例 225

隐式发出 227

## S

### Search Discovery

附加设置 81

### SELECT 特权

定义 220

### SELECT 语句

选择视图 128

### SERVER, 认证类型 196

### SERVER\_ENCRYPT

认证类型 196

### SET ENCRYPTION

PASSWORD 232

### SIGTTIN 311

### SmartGuide

向导 414

### SMS 表空间

创建 98

添加容器 162

### SQL 语句

不起作用 189

显示 6

### SWITCH ONLINE 子句 162

### SYSCAT 视图 234

### SYSCATSPACE 表空间 91

## T

### TCP/IP 344

### TEMPSPACE1 表空间 91

## U

### UPDATE 特权

定义 220

### USAGE 特权 224

### USERSPACE1 表空间 91

## V

### varchar\_no\_trailing\_blanks 服务器选项

138

### varchar\_no\_trailing\_blanks 列选项

188

### VI

设置以与 DB2 一起使用 353

### VI 体系结构 345

## W

### Windows 2000 活动目录

对象 371

扩展目录模式 369

### Windows NT 活动目录

对象类和属性 371

### Windows NT 性能监视器 331

注册 DB2 331

### \$RAHBUFDIR 312

### \$RAHBUFNAME 312

### \$RAHCHECKBUF 312

### \$RAHENV 318





---

## 与 IBM 联系

如果有技术问题，请在与“DB2 客户支持中心”联系之前复查并执行 *Troubleshooting Guide* 所建议的操作。本指南提出了一些建议，指导您收集一些信息从而帮助“DB2 客户支持中心”更好地为您服务。

要获取信息或订购任何“DB2 通用数据库”产品，与当地分支机构的 IBM 代表联系或与任何授权的 IBM 软件经销商联系。

您如果住在美国，请致电下列其中一个号码：

- 1-800-237-5511，可获得客户支持
- 1-888-426-4343，可了解所提供的服务项目

---

## 产品信息

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) 或 1-800-3IBM-OS2 (1-800-342-6672)，可订购产品或获取一般信息。
- 1-800-879-2755，可订购出版物。

**<http://www.ibm.com/software/data/>**

DB2 万维网网页提供关于新闻、产品描述和培训计划等等的当前 DB2 信息。

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

“DB2 产品和服务技术库”可供您访问常见问题、修正、书籍以及最新的 DB2 技术信息。

注：此信息可能只有英文版。

**<http://www.elink.ibm.com/pbl/pbl/>**

“国际出版物” Web 订购站点提供关于如何订购书籍的信息。

**<http://www.ibm.com/education/certify/>**

IBM Web 站点中的“专业证书程序”提供各种 IBM 产品（包括 DB2）的证书测试信息。

**<ftp://software.ibm.com>**

以匿名形式登录。可在目录 /ps/products/db2 中找到有关 DB2 和许多其他产品的演示、修正、信息和工具。

**comp.databases.ibm-db2, bit.listserv.db2-l**

这些因特网新闻组可供用户来讨论使用 DB2 产品的经验。

**On Compuserve: GO IBMDB2**

输入此命令来访问 IBM DB2 系列论坛。这些论坛支持所有的 DB2 产品。

有关如何在美国以外的地区与 IBM 联系的信息，请参阅 *IBM Software Support Handbook* 的附录 A。要访问此文档，访问以下 Web 页面：<http://www.ibm.com/support/>，然后选择该页面底部附近的 IBM Software Support Handbook 链接。

**注：**在某些国家或地区，IBM 授权的经销商应与他们的经销商支持机构联系，而不是与“IBM 支持中心”联系。





SB84-0218-01

