

IBM[®] DB2[®] 通用数据库



管理指南：计划

版本 7

IBM[®] DB2[®] 通用数据库



管理指南：计划

版本 7

在使用本资料 and 它支持的产品之前，请参阅第383页的『附录F. 注意事项』中的一般信息。

本文档包含 IBM 的专利信息。它在许可协议下提供，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散发，而不必对您负任何责任。

© Copyright International Business Machines Corporation 1993, 2000. All rights reserved.

目录

关于本书	vii	DB2 安全性概述	45
谁应使用本书	viii	认证	45
本书的结构	viii	授权	46
“管理指南”的其他卷的简要概述	ix	联合体数据库认证和授权概述	47
管理指南: 实现	ix	第3章 联合体系统	49
管理指南: 性能	x	启用联合体系统	51
第1部分 “DB2 通用数据库”的世界 1		第4章 并行数据库系统	53
第1章 管理 DB2 通用数据库	3	节点组和数据分区	53
第2部分 数据库概念	5	并行性的类型	55
第2章 基本关系数据库概念	7	I/O 并行性	55
数据库对象概述	7	查询并行性	55
实例	8	实用程序并行性	57
数据库	9	硬件环境	58
节点组	9	单处理器上的单分区	59
表	9	具有多个处理器的单分区	60
视图	9	多分区配置	61
索引	10	最适合每种硬件环境的并行性的摘要	65
模式	11	第5章 关于数据入库	67
系统目录表	11	什么是数据入库?	67
恢复对象概述	12	主题区	67
恢复日志文件	12	仓库源	68
恢复历史文件	12	仓库目标	68
存储器对象概述	13	仓库代理程序和代理点	68
表空间	13	步骤和过程	68
容器	16	入库任务	70
缓冲池	17	第6章 关于 Spatial Extender	73
系统对象概述	18	Spatial Extender 的用途	73
配置参数	18	表示地理面貌的数据	73
数据的商业规则	20	数据如何表示地理面貌	74
恢复数据库	23	空间数据的性质	75
恢复概述	23	空间数据从哪里来	75
影响恢复的因素	27	第3部分 数据库设计	79
灾难恢复的考虑事项	40	第7章 逻辑数据库设计	81
降低媒体故障的影响	41	决定数据库中要记录的数据	81
降低事务故障的影响	43	为每种类型的关系定义表	83
分区数据库系统中的系统时钟同步	43		
重组数据库中的表	44		

一对多和多对一关系	83
多对多关系	84
一对一关系	85
为所有表提供列定义	85
标识一列或多列作为主关键字	87
标识入选关键字列	88
定义标识列	89
确保相等的值表示相同的实体	90
考虑使表规范化	90
第一种规范形式	91
第二种规范形式	91
第三种规范形式	93
第四种规范形式	94
制定约束实施计划	95
唯一约束	95
参考完整性	96
表检查约束	100
触发器	100
数据库设计的其他考虑事项	101
第8章 物理数据库设计	103
数据库目录	103
数据库文件	104
估计表的空间需求	105
系统目录表	106
用户表数据	106
长整数字段数据	108
大对象 (LOB) 数据	108
索引空间	109
附加空间需求	111
日志文件空间	112
临时工作空间	112
设计节点组	113
节点组设计考虑事项	114
设计和选择表空间	120
系统管理空间	122
数据库管理空间的表空间	126
表空间的设计考虑事项	127
联合体数据库设计考虑事项	137
第9章 设计分布式数据库	139
在一个事务中使用单个数据库	139
在单个事务中使用多个数据库	140
更新单个数据库	140
更新多个数据库	142
其他配置考虑事项	145

在多站点更新中存取基于 LAN 的 DB2 通 用数据库服务器的主机或 AS/400 应用程序	146
了解两阶段落实过程	147
在两阶段落实期间校正问题	150
若 AUTORESTART=OFF, 则再同步不确 定事务	151

第10章 针对事务管理器的设计 153

X/Open 分布式事务处理模型	154
应用程序 (AP)	154
事务管理器 (TM)	155
资源管理程序 (RM)	156
将数据库设置为资源管理程序	157
xa_open 和 xa_close 字符串的用法	157
DB2 版本 7 的新 xa_open 字符串格式	157
TPM 和 TP_MON_NAME 值	159
较早版本的 DB2 的 xa_open 字符串格式	161
更新主机或 AS/400 数据库服务器	162
数据库连接考虑事项	162
作试探性决定	163
安全性考虑事项	165
配置考虑事项	166
受支持的 XA 函数	166
XA 接口的问题确定	168
配置 XA 事务管理器以使用 DB2 UDB	169
配置 IBM TXSeries CICS	169
配置 IBM TXSeries Encina	170
配置 BEA Tuxedo	172
配置 Microsoft 事务服务器	174

第11章 针对高可用性的设计 181

热备份	182
示例	182
相互替换	185
示例	185
故障恢复后的重新连接	188
资源	188

第4部分 高可用性 189

第12章 高可用性群集多重处理增强可缩放性

(HACMP ES) AIX 版	191
群集配置	192
配置 DB2 数据库分区	196
热备份配置示例	197
相互替换配置示例	198

NFS 服务器节点的配置	198
NFS 服务器替换配置的示例	199
配置 SP 交换时的考虑事项	200
DB2 HACMP 配置示例	201
DB2 HACMP 启动建议	209
HACMP ES 事件监控和用户定义事件	210
HACMP ES 脚本文件	213
使用 HACMP ES 执行 DB2 恢复脚本操作	215
其他脚本实用程序	217
监控 HACMP 群集	217
DB2 SP HACMP ES 安装	219
首次安装 DB2 SP HACMP ES	219
DB2 SP HACMP ES 迁移	220
DB2 SP HACMP ES 工作表	221
第13章 Windows NT 环境中的高可用性	231
故障恢复配置	232
热备份配置	232
相互替换配置	233
使用 DB2MCS 实用程序	234
指定 DB2MCS.CFG 文件	235
为单分区数据库系统设置故障恢复	239
为两个单分区数据库系统设置相互替换配置	239
为分区数据库系统设置多个 MCS 群集	240
维护 MCS 系统	241
回退考虑事项	242
在分区数据库环境中为相互替换配置注册数据库驱动器映射	242
协调数据库驱动器映射	244
示例 - 为相互替换设置两个单分区实例	245
初步任务	245
运行 DB2MCS 实用程序	246
示例 - 为相互替换设置四节点分区数据库系统	247
初步任务	248
运行 DB2MCS 实用程序	249
注册 ClusterA 的数据库驱动器映射	250
注册 ClusterB 的数据库驱动器映射	250
在 MCS 环境中管理 DB2	251
启动和停止 DB2 资源	251
运行脚本	252
数据库考虑事项	255
用户和组支持	256
通信考虑事项	256
系统时间考虑事项	257
分区数据库环境中的管理服务器和控制中心考虑事项	257

限制	259
第14章 Sun Cluster 2.2 上的 DB2 和高可用性	261
高可用性	261
故障容错和连续可用性	263
Sun Cluster 2.2	264
受支持的系统	264
代理程序	264
逻辑主机	265
逻辑网络接口	265
磁盘组和文件系统	266
控制方法	268
磁盘和文件系统配置	269
HA-NFS	270
cconsole 和 ctelnet 实用程序	270
校园分群和大陆分群	270
常见问题	270
DB2 考虑事项	271
连接 HA 实例的应用程序	271
EE 和 EEE 实例的磁盘布局	272
EE 和 EEE 实例的主目录布局	273
逻辑主机和 DB2 UDB EEE	274
DB2 安装位置和选项	275
数据库和数据库管理程序配置参数	275
应急恢复	276
通过数据复制的高可用性	276
DB2 高可用性代理程序	276
注册 hadb2 服务	276
hadb2tab 文件	277
控制方法	277
用户脚本	279
其他考虑事项	281
故障监控程序	281
EEE 考虑事项	281
HA.config 文件	283
控制方法是如何运行 DB2 命令的设置	284
公共安装步骤	285
DB2 UDB 企业版上的设置	285
DB2 UDB 扩充企业版上的设置	285
hadb2_setup 命令	286
故障恢复时间	289
故障排除	291

第5部分 附录及附属资料 295

附录A. 使用 DB2 资料库	297	“DB2 通用数据库” 版本 7 的不兼容性 . . .	329
DB2 PDF 文件和打印的书籍	297	应用程序设计	329
DB2 信息	297	SQL	331
打印 PDF 书籍.	305	实用程序和工具	332
订购打印书籍	306	连通性和共存	332
DB2 联机文档	307	“DB2 通用数据库版本 6” 的不兼容性. . .	332
存取联机帮助	307	系统目录视图	333
查看联机信息	309	应用程序设计	338
使用 DB2 向导	311	SQL	341
设置文档服务器	312	数据库安全性和协调	343
搜索联机信息	313	实用程序和工具	344
		连通性和共存	344
		配置参数.	345
附录B. 命名规则	315	附录E. 国家语言支持 (NLS)	347
数据库名称	315	国家代码和代码页支持	347
数据库名和数据库别名	315	派生代码页值	362
用户 ID 和口令	316	字符集	362
模式名	317	标识符的字符集	363
组名和用户名	317	编码 SQL 语句	363
对象名	318	双向 CCSID 支持.	363
联合体数据库对象名	319	整理顺序.	367
如何在联合体系统中保留区分大小写的值	320	日期时间值	369
附录C. 数据库迁移计划.	323	DB2 UDB 中的 Unicode/UCS-2 和 UTF-8	
迁移考虑事项	323	支持	375
迁移限制.	323	介绍	375
安全性与授权	324	DB2 UDB 中的 UCS-2/UTF-8 实现. . .	376
存储器需求	324	附录F. 注意事项	383
发行版间的不兼容性	324	注册商标.	385
迁移数据库	325	索引	387
附录D. 发行版间的不兼容性	327	与 IBM 联系	395
“DB2 通用数据库” 计划的不兼容性 . . .	328	产品信息.	395
“DB2 通用数据库” 未来版本中的只读视			
图	328		
“DB2 通用数据库” 未来版本中的			
PK_COLNAMES 和 FK_COLNAMES . . .	328		
在“DB2 通用数据库” 未来版本中			
COLNAMES 不再可用	328		

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接口来执行：

- **命令处理器**，它允许您从图形界面存取和操纵数据库。从此接口，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的是此接口的使用。有关使用命令处理器的详情，参见 *Command Reference*。
- **应用程序设计接口**，它允许您在应用程序内执行 DB2 实用程序功能。有关使用应用程序设计接口的详情，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。控制中心也包含复制管理，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符，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**性能监控程序**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。
- **警报中心**，用于监控其他 DB2 操作产生的信息。
- **工具设置**，用于更改“控制中心”、“警报中心”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、安全管理员和系统操作员，他们需要设计、实施和维护本地或远程客户机要存取的数据库。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：

DB2 通用数据库的世界

- 第1章 管理 DB2 通用数据库，提供了“DB2 通用数据库”的介绍和概述。

数据库概念

- 第2章 基本关系数据库概念，提供了数据库对象（包括恢复对象、存储器对象和系统对象）的概述。
- 第3章 联合体系统，讨论联合体系统，联合体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。
- 第4章 并行数据库系统，提供有关 DB2 可使用的并行性类型的介绍。
- 第5章 关于数据入库，提供数据入库和数据入库任务的概述。
- 第6章 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

- 第7章 逻辑数据库设计，讨论逻辑数据库设计的概念和准则。
- 第8章 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考虑事项。
- 第9章 设计分布式数据库，讨论如何在单个事务中存取多个数据库。
- 第10章 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。
- 第11章 针对高可用性的设计，提供 DB2 提供的高可用性故障恢复支持的概述。

高可用性系统

- 第12章 高可用性群集多重处理增强可缩放性 (HACMP ES) AIX 版，讨论 DB2 在 AIX 上对高可用性故障恢复的支持。
- 第13章 Windows NT 环境中的高可用性，讨论 DB2 在 Windows NT 上对高可用性故障恢复的支持。

- 第14章 Sun Cluster 2.2 上的 DB2 和高可用性，讨论 DB2 在 Sun Solaris 操作系统上对高可用性故障恢复的支持。

附录

- 附录B. 命名规则，介绍命名数据库和对象时需遵循的规则。
- 附录C. 数据库迁移计划，提供关于将数据库迁移至版本 7 的信息。
- 附录D. 发行版间的不兼容性，介绍不同发行版（最多到版本 7）之间的不兼容性。
- 附录E. 国家语言支持 (NLS)，介绍“DB2 国家语言支持”，包括关于国家、语言和代码页的信息。
- 附录A. 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、信息和书籍。

“管理指南”的其他卷的简要概述

管理指南：实现

*管理指南：实现*主要讨论数据库设计的实现。下面简要描述该卷中的特定章节和附录：

使用“控制中心”进行管理

- “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

- “在创建数据库之前”讨论创建数据库的前提条件。
- “创建数据库”讨论那些与数据库及相关数据库对象的创建相关联的任务。
- “改变数据库”讨论在改变数据库之前必须执行的工作，并讨论与修改和卸下数据库或相关数据库对象相关联的任务。

数据库安全性

- “控制数据库存取”描述如何控制对数据库资源的存取。
- “审核 DB2 活动”描述如何检测和监控对数据的不需要的存取或不期望的存取。

移动数据

- “移动数据实用程序”是一个只有一页纸的介绍，它说明移动数据的多种方法并指导您参考 *Data Movement Utilities Guide and Reference* 一书。

恢复

- “恢复数据库”讨论当选择数据库和表空间恢复方法（包括备份和复原数据库或表空间）以及使用前滚恢复方法时要考虑的因素。

附录

- “使用分布式计算环境 (DCE) 目录服务”提供有关如何使用“DCE 目录服务”的信息。
- “用于数据库恢复的用户出口”讨论如何结合数据库日志文件来使用用户出口程序，并描述一些样本用户出口程序。
- “向多数据库分区服务器发出命令”讨论如何使用 *db2_all* 和 *rah* 外壳脚本来将命令发送至一个分区数据库环境中的所有分区。
- “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。
- “使用 Windows NT 性能监控程序”提供有关向“Windows NT 性能监控程序”注册 DB2 以及使用性能信息的信息。
- “使用 Windows NT 或 Windows 2000 数据库分区服务器”提供有关可用来使用 Windows NT 或 Windows 2000 上的数据库分区服务器的实用程序。
- “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。
- “高速节点间通信”描述如何启用“虚拟接口体系结构”以用于“DB2 通用数据库”。
- “轻量级目录存取协议 (LDAP) 目录服务”提供关于如何使用“LDAP 目录服务”的信息。
- “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。

管理指南: 性能

*管理指南: 性能*主要讨论性能问题，即，那些主题和论点讨论建立、测试和改进应用程序及“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 性能的概念和考虑事项。
- “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

- “应用程序考虑事项”描述在设计应用程序时用于改善数据库性能的一些技术。

- “环境考虑事项” 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 编译程序为存取数据所做的选择。

调整和配置系统

- “运行性能” 概述数据库管理程序如何使用内存以及影响运行期性能的其他考虑事项。
- “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” 介绍与增加数据库系统的大小相关的一些考虑事项和任务。
- “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” 讨论数据库管理程序、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” 列出简要表的注册表值和环境变量。
- “解释表和定义” 提供有关“DB2 解释”设施使用的表以及如何创建那些表的信息。
- “SQL 解释工具” 提供有关使用 DB2 解释工具（db2expln 和 dynexpln）的信息。
- “db2exfmt — 解释表格式工具” 提供有关使用 DB2 解释工具来格式化解释表数据的信息。

第1部分 “DB2 通用数据库” 的世界

第1章 管理 DB2 通用数据库

DB2 很灵活，它允许您运行各种硬件配置。它允许您选择如何使用特定的 DB2 产品配置来最好地满足您的硬件需求和应用程序需求。

由于 DB2 支持的数据库环境的复杂程度有所不同，因此针对每个环境有着特定的考虑事项和任务。这些在**管理指南**和 DB2 库中的其他书籍中有详细的讨论（参见第297页的『附录A. 使用 DB2 资料库』）。在某些情况下，这些书籍的全部章节只适合于某个特定的环境。在阅读本书的前言（“关于本书”）之后，您将了解本书以及其他卷的**管理指南**（**管理指南：实现**和**管理指南：性能**）中的哪些章节适合于您的商业需要。

如果您不熟悉关系数据库管理系统 (RDBMS) 或 DB2，您将会发现标题为“基本关系数据库概念”的那一节对您很有帮助。如果您熟悉这些概念，或者不需要进行回顾，则可跳过本节，并直接转到描述更高级主题的那些节中，如：

- **联合体系统**。本节讨论支持提交特定 SQL 语句的应用程序和用户的数据库管理系统 (DBMS)，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**。本节提供有关 DB2 可使用的并行性类型的介绍。任务（如数据库查询）的各个部分可并行运行以大幅提高性能。
- **分布式事务处理**。本节讨论如何在单一事务中存取多个数据库，以及如何在分布式事务处理环境中使用数据库。
- **高可用性系统**。本节提供 DB2 提供的高可用性故障恢复支持的概述。故障恢复能力允许在发生硬件故障时，自动将工作负荷从一个处理器转移至另一个处理器。

DB2 可以满足您最特殊的数据管理需要，如：

- **复制**，它允许您定期将数据复制至多个远程数据库。若需要将主数据库中的更新自动复制至其他数据库，则可使用 DB2 的复制功能部件来指定应复制哪些数据，应将数据复制到哪些数据库表，以及应以何种频率复制更新。若要使用 DB2 的复制功能部件，请参考 *Replication Guide and Reference*。该书介绍 DB2 数据复制的概念，并描述如何规划、配置和管理复制环境。
- **数据入库**，在此操作中，您可以存储“资料式数据”，即从运行数据中抽取出来然后进行变换以供最终用户进行决策的数据。例如，数据入库工具可以从运行数据库复制所有销售数据，执行计算以对该数据进行摘要，并将摘要数据写至单独数据库中的目标。您可以查询单独的数据库（仓库），而不会对运行数据库有所影响。有关数据入库的详情，参考**数据仓库中心管理指南**。

- **地理信息系统 (GIS)**, 它可以通过 **Spatial Extender** 创建。GIS 是对象、数据和应用程序的复合体, 它允许您生成和分析关于地理面貌的空间信息。在 **Spatial Extender** 中, 地理面貌可以由表或视图中的一行表示, 也可以由这样的行的一部分表示。有关使用 **Spatial Extender** 的详情, 参考 *Spatial Extender 用户指南* 和参考。

管理指南: 计划还讨论了数据库设计, 包括 **DB2** 的逻辑数据库设计和物理数据库设计考虑事项。还讨论了其他计划问题, 如计划数据库迁移、标识可能会对应用程序有所影响的不兼容性 (不兼容性是 **DB2** 通用数据库的一部分, 它的工作方式与它在 **DB2** 先前发行版中的工作方式不同; 若在现有的应用程序中使用, 则会生成预期不到的结果, 或者需要更改应用程序, 或者降低性能) 以及利用国家语言支持 (NLS)。

管理指南: 实现阐述了与实现数据库设计有关的细节。本书包括有关创建和改变数据库、数据库安全性、数据库恢复以及使用“控制中心”(这是一个 **DB2** 图形用户界面) 管理 **DB2** 的主题。

管理指南: 性能阐述了与建立、测试以及改进应用程序及 **DB2** 本身的性能相关的主题和问题。

第2部分 数据库概念

第2章 基本关系数据库概念

本节包括下列主题:

- 『数据库对象概述』
- 第12页的『恢复对象概述』
- 第13页的『存储器对象概述』
- 第18页的『系统对象概述』
- 第20页的『数据的商业规则』
- 第23页的『恢复数据库』
- 第44页的『重组数据库中的表』
- 第45页的『DB2 安全性概述』

数据库对象概述

本节提供了下列关键数据库对象的概述:

- 实例
- 数据库
- 节点组
- 表
- 视图
- 索引
- 模式
- 系统目录表

第8页的图1举例说明了这些对象的某一些之间的关系。它也显示了表、索引和长整数数据存储于表空间中的情况。

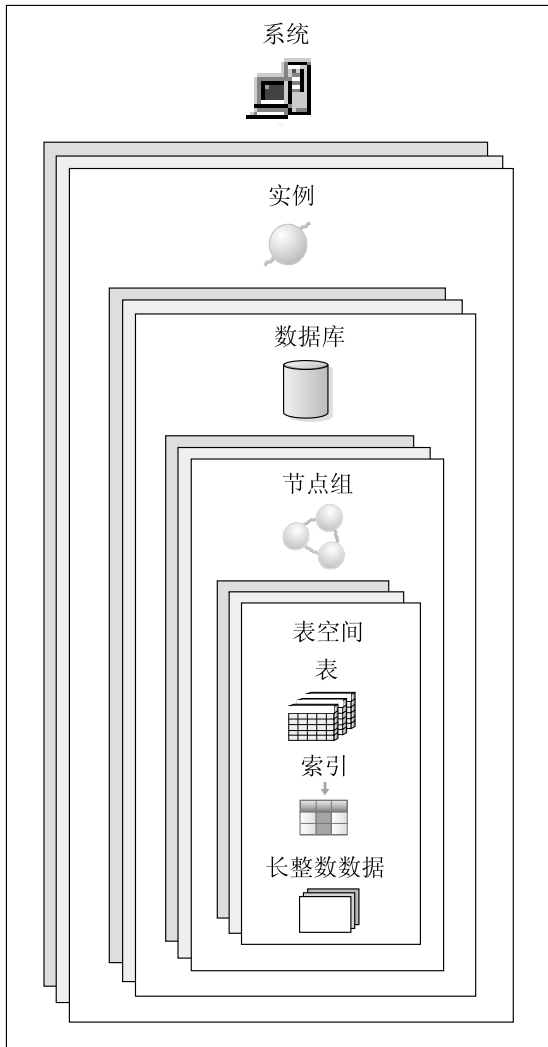


图 1. 一些数据库对象之间的关系

实例

实例（有时称为数据库管理程序）是管理数据的 DB2 代码。它控制可对数据执行的操作，并管理分配给它的系统资源。每一个实例都是一个完整的环境。它包含为一个给定的并行数据库系统定义的所有数据库分区（参见第53页的『第4章 并行数据库系统』）。一个实例有它自己的数据库（其他实例不能存取它），并且它的全部数据库分区共享相同的系统目录。它还有独立的安全性，不受同一机器（系统）上其他实例的影响。

数据库

关系数据库将数据表示成表的集合。表由数目已定的列和任意数目的行组成。每个数据库都包括一组描述数据的逻辑和物理结构的系统目录表，一个包含为该数据库分配的参数值的配置文件以及一个带有正在进行的事务和可归档事务的恢复日志。

节点组

节点组是一个或多个数据库分区组成的集合。当您想为数据库创建表时，首先创建节点组以存储表空间，然后创建表空间以存储表。有关节点组的详情，参见第53页的『节点组和数据分区』。有关数据库分区的定义，参见第53页的『第4章 并行数据库系统』。有关表空间的详情，参见第13页的『表空间』。

表

关系数据库将数据表示成表的集合。表由逻辑排列的行和列数据组成。所有数据库和表数据都被存储在表空间中。有关表空间的详情，参见第13页的『表空间』。表中的数据在逻辑上是相关的，且可以定义表与表之间的关系。根据数学规则和关系运算来查看和操纵数据。

表数据通过“结构化查询语言”（SQL，参见 *SQL Reference*）存取，SQL 是一种标准化语言，用于定义和操纵关系数据库中的数据。应用程序或用户使用查询从数据库中检索数据。查询使用 SQL 来创建如下格式的语句

```
SELECT <data_name> FROM <table_name>
```

视图

视图是高效率的数据呈现方法（无需维护数据）。视图不是实际的表，不需要永久性存储器。创建并使用一个“虚拟表”。

视图可以包括它所基于的表中的所有或某些列或行。例如，可以在视图中连接一个部门表和一个雇员表，以便可以列出特定部门中的所有雇员。

第10页的图2显示了表与视图之间的关系。

数据库

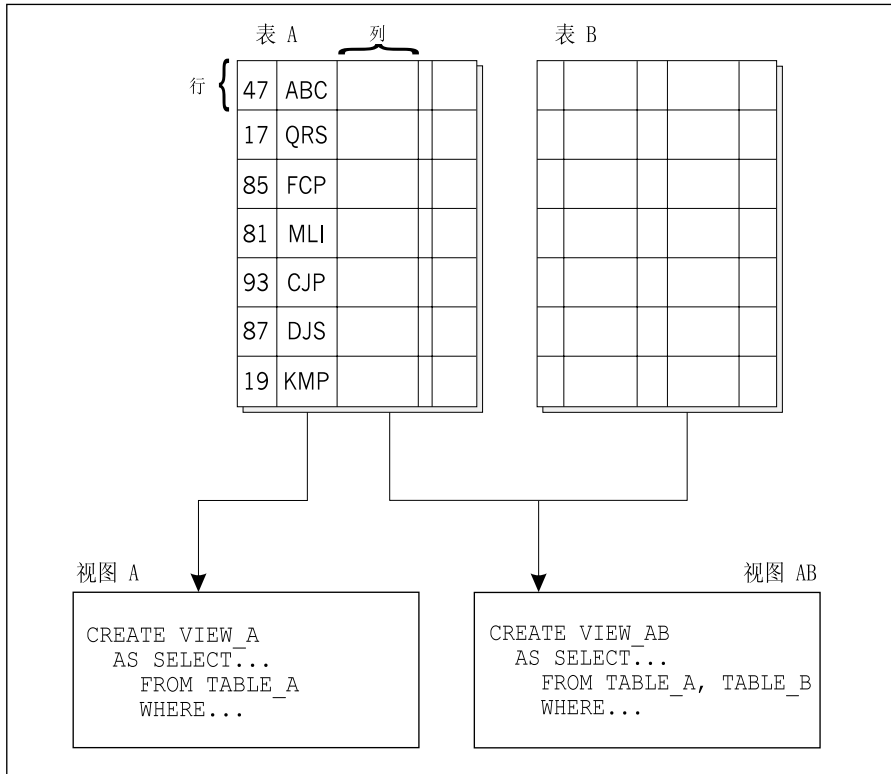


图 2. 表与视图之间的关系

索引

索引是一组关键字，其每一个都指向一些表行。例如，在第11页的图3中，表 A 的一个索引基于表中的雇员号码。此关键字值提供指向表行的指针：雇员号码 19 指向雇员 **KMP**。通过使用指针创建指向数据的直接路径，索引使更有效率地存取表行成为可能。

SQL 优化器自动选择最有效率的存取表中数据的方法。当确定最快速的数据存取路径时，优化器会将索引考虑在内。

可创建唯一索引以确保索引关键字的唯一性。索引关键字是定义了一个列或一些列的有序集合。使用唯一索引将确保在编入索引的列中，每个索引关键字的值都是唯一的。第20页的『数据的商业规则』更详细地描述了关键字和索引。

第11页的图3显示了索引与表之间的关系。

数据库

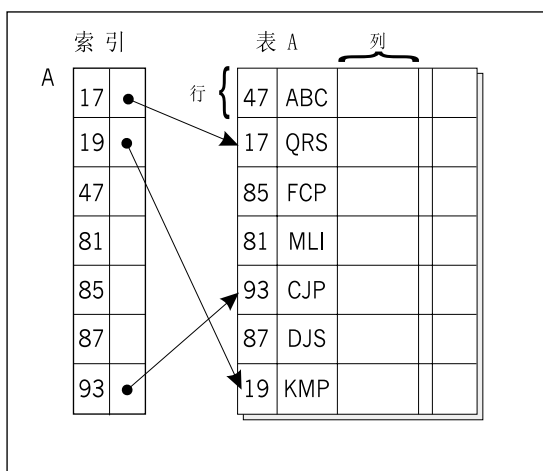


图 3. 索引与表之间的关系

模式

模式是一个标识符，如用户 ID，它帮助分组表和其他数据库对象。模式可以归个人拥有，拥有者可以控制对数据以及其中的对象的存取。

模式也可以是数据库中的对象。它可以在创建模式中的第一个对象时自动创建。这样的对象可以是任何可以由模式名限定的对象，如表、索引、视图、程序包、单值类型、函数或触发器。若要自动创建模式，则您必须拥有 `IMPLICIT_SCHEMA` 权限，也可以隐式地创建模式。

模式名用作两部分对象名的第一部分。创建一个对象时，可将其分配给特定模式。若不指定模式，则它被分配给缺省模式，缺省模式通常是创建该对象的人员的用户 ID。名称的第二部分是对象名。例如，名为 `Smith` 的用户可以有一个名为 `SMITH.PAYROLL` 的表。

系统目录表

每个数据库都包括一组描述数据的逻辑和物理结构的系统目录表。DB2 为每个数据库维护一大组系统目录表。这些表包含有关数据库对象（例如，用户表、视图和索引）的定义的信息，以及用户对这些对象所拥有的权限的安全性信息。它们在数据库创建时被创建，并在常规操作期间得到更新。不能显式地创建或卸下它们，但是可以使用目录视图查询和查看它们的内容。

恢复对象概述

日志文件和恢复历史文件是在数据库创建时自动创建的（图4）。您不能直接修改日志文件或恢复历史文件；然而，万一您需要使用数据库备份映象来恢复丢失或损坏的数据，则这两个文件非常重要。

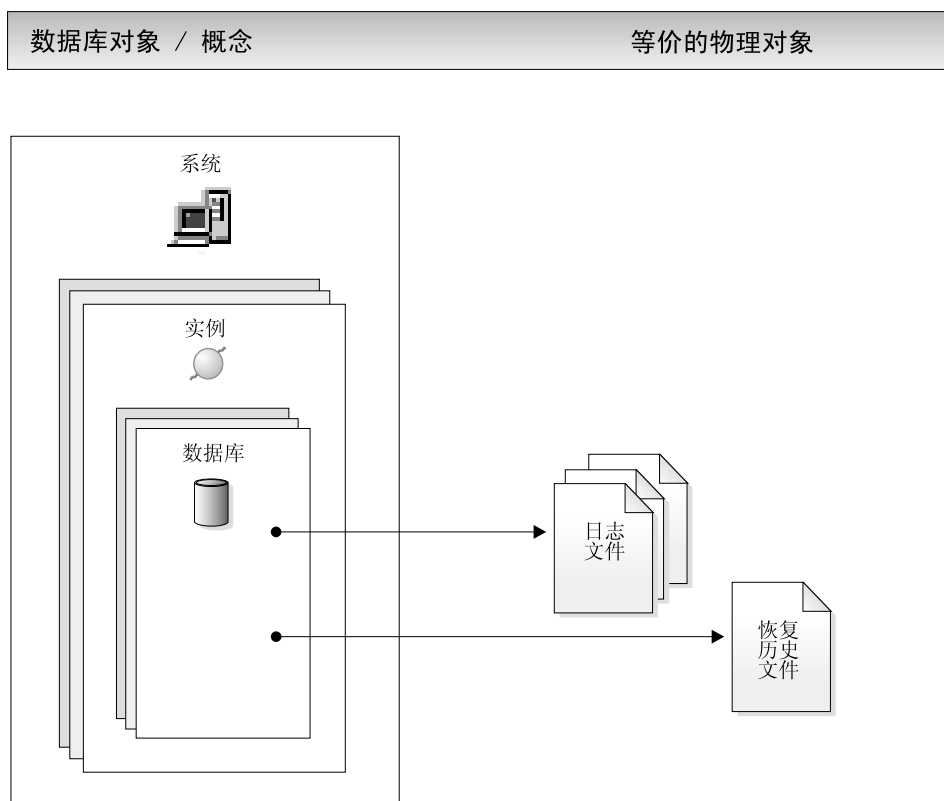


图 4. 日志文件和恢复历史文件

恢复日志文件

每个数据库都包括恢复日志，它们用来从应用程序或系统错误中恢复。与数据库备份一起，它们用来将数据库的一致性恢复到出错时的时间点。第23页的『恢复数据库』更详细地讨论数据库恢复。

恢复历史文件

恢复历史文件包含特定备份信息的摘要，当必须将整个数据库或数据库的一部分恢复至给定时间点时，可以使用这些信息。恢复历史文件用来跟踪与恢复相关的

事件，如备份、复原和装入操作。第23页的『恢复数据库』描述了备份和复原数据库的过程。*Data Movement Utilities Guide and Reference* 描述了装入实用程序。

存储器对象概述

下列数据库对象用来定义在系统上存储数据的方式以及改进（与存取数据相关的）性能的方法：

- 表空间
- 容器
- 缓冲池

表空间

数据库由称为表空间的部件组成。表空间是用来存储表的位置。当创建表时，您可以决定将特定对象（如索引和大对象 (LOB)）数据与其余表数据分开存放。表空间也可以分布在一个或多个物理存储设备上。下图显示了在表空间之间分布数据时具有的一些灵活性。

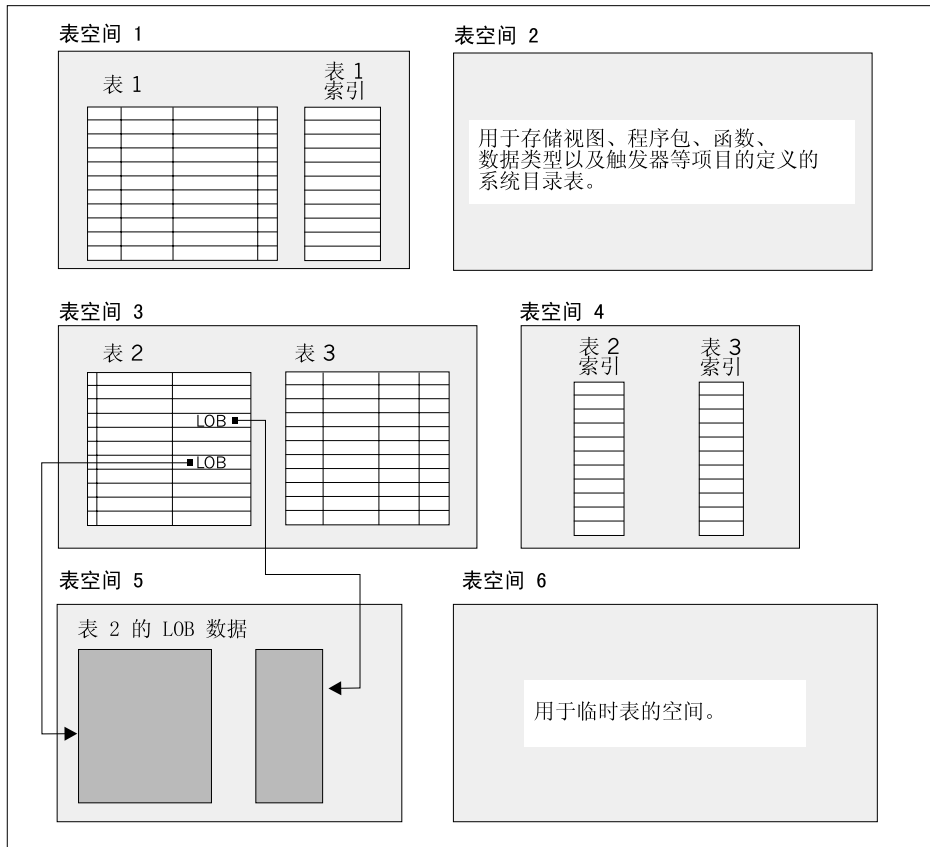


图 5. 表空间

表空间驻留在节点组中（参见第9页的『节点组』）。表空间定义和属性记录在数据库系统目录中（参见第11页的『系统目录表』）。

将容器分配给表空间。容器是分配的物理存储器（如文件或设备）。

表空间可以是系统管理空间 (SMS) 或数据库管理空间 (DMS)。对于 SMS 表空间，每个容器都是操作系统的文件空间中的一个目录，由操作系统的文件管理器控制存储空间。对于 DMS 表空间，每个容器或者是固定大小的预分配文件，或者是物理设备如磁盘，由数据库管理程序控制存储空间。

第15页的图6举例说明了表、表空间以及两种类型的空间之间的关系。它还显示了表、索引和长整数数据存储表空间中的情况。

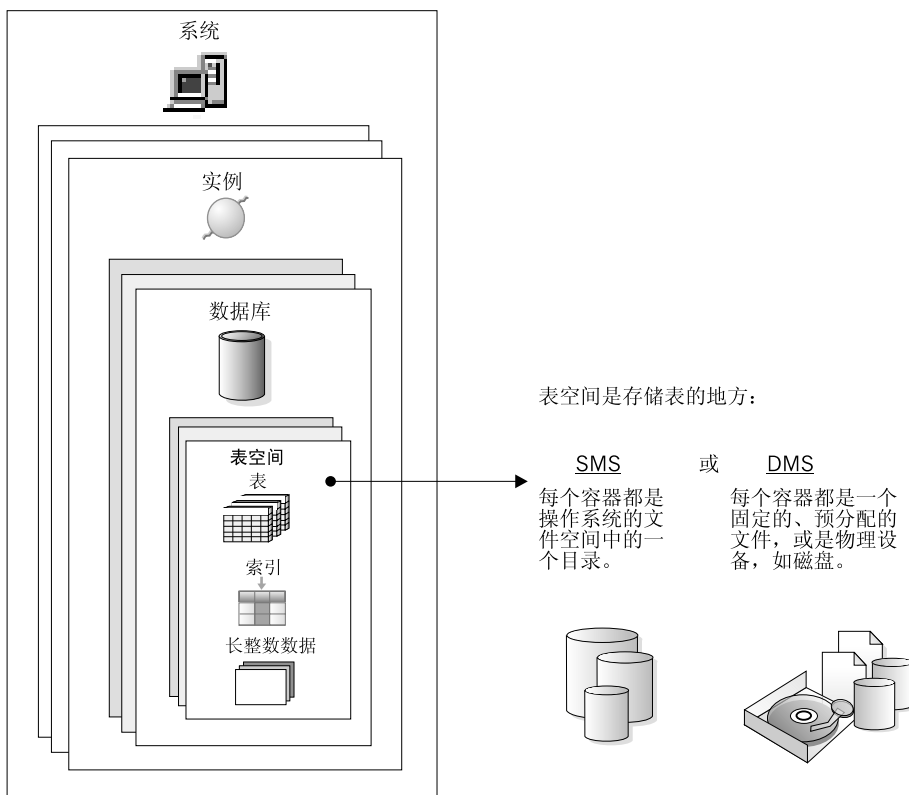


图 6. 表空间和表

第16页的图7显示了三种表空间类型：*规则*、*临时*和*长整数*。

包含用户数据的表存放在规则表空间中。缺省用户表空间名为 `USERSPACE1`。索引也存储在规则表空间中。系统目录表存放在规则表空间中。缺省系统目录表空间名为 `SYSCATSPACE`。

包含长整数字段数据或长整数对象数据（如多媒体对象）的表存放在长整数表空间中。

*临时表空间*分为系统临时表空间或用户临时表空间。系统临时表空间用来存储 SQL 操作（如排序、重组表、创建索引和连接表）期间所需的内部临时数据。虽然可以创建任意数目个系统临时表空间，但建议您只使用大多数表所使用的页大小创建一个。缺省系统临时表空间名为 `TEMPSPACE1`。用户临时表空间用来存储已说

明全局临时表（已说明全局临时表存储的是应用程序临时数据）。用户临时表空间不是在数据库创建时缺省创建的。

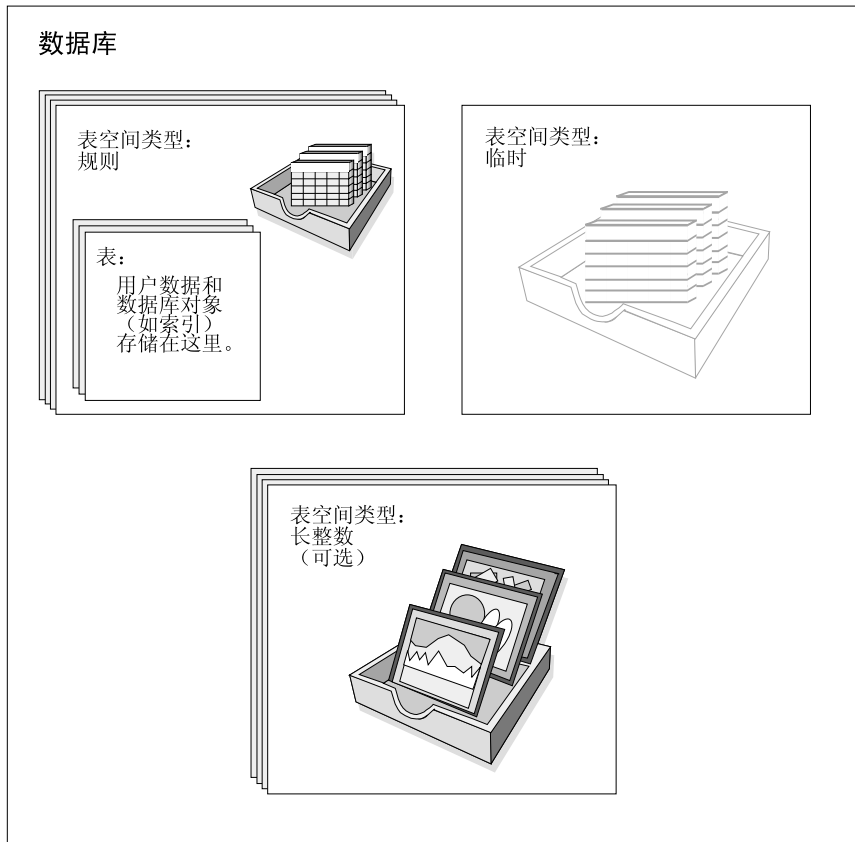


图7. 三种表空间类型

容器

容器是物理存储设备。它可以由目录名、设备名或文件名标识。

将为表空间分配容器。单个表空间可以横跨多个容器，但每个容器只能属于一个表空间。

第17页的图8举例说明了表与数据库中的表空间、相关联的容器和磁盘之间的关系。

数据库

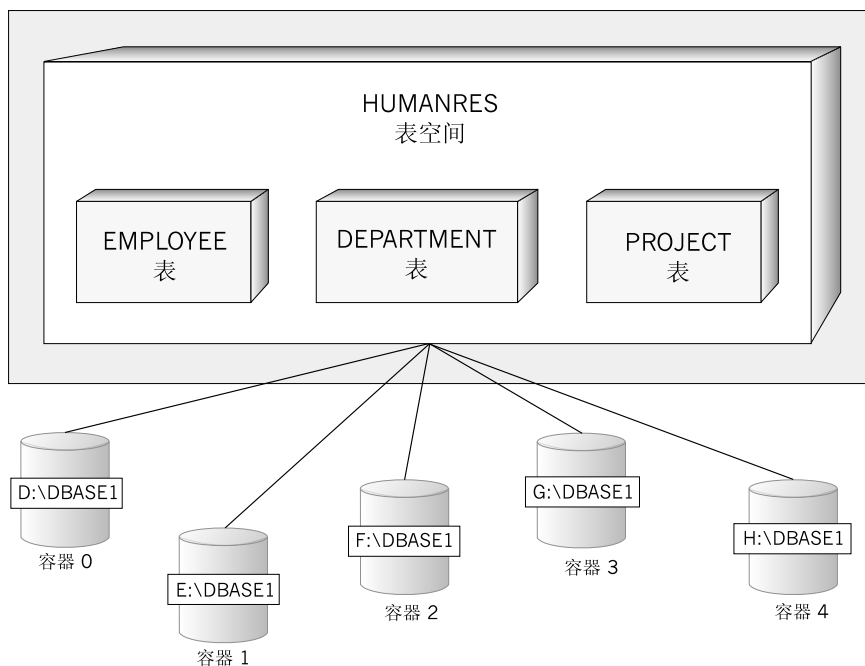


图 8. 数据库中的表空间和表

EMPLOYEE、DEPARTMENT 和 PROJECT 表在 HUMANRES 表空间中，该表空间横跨容器 0、1、2、3 和 4。此示例显示每个容器存在于不同的磁盘上。

任何表的数据都以循环方式存储在表空间中的所有容器中。这能在属于给定表空间的容器之间平衡数据。数据库管理程序在使用另一个容器之前写入一个容器的页数称为数据块大小。

缓冲池

缓冲池指的是从磁盘读取高速缓存表和索引数据页时或修改它们时分配给它们的主存储器。缓冲池的目的是改进系统性能。从内存存取数据要比从磁盘存取数据快得多；因此，数据库管理程序需要读写磁盘 (I/O) 的次数越少，性能也越好。(可以创建多个缓冲池，虽然在大多数情况下只需要一个。)

因为可以缩短慢速 I/O 所造成的延迟，所以缓冲池的配置是最为重要的调整项目。

第18页的图9举例说明了缓冲池与容器之间的关系。

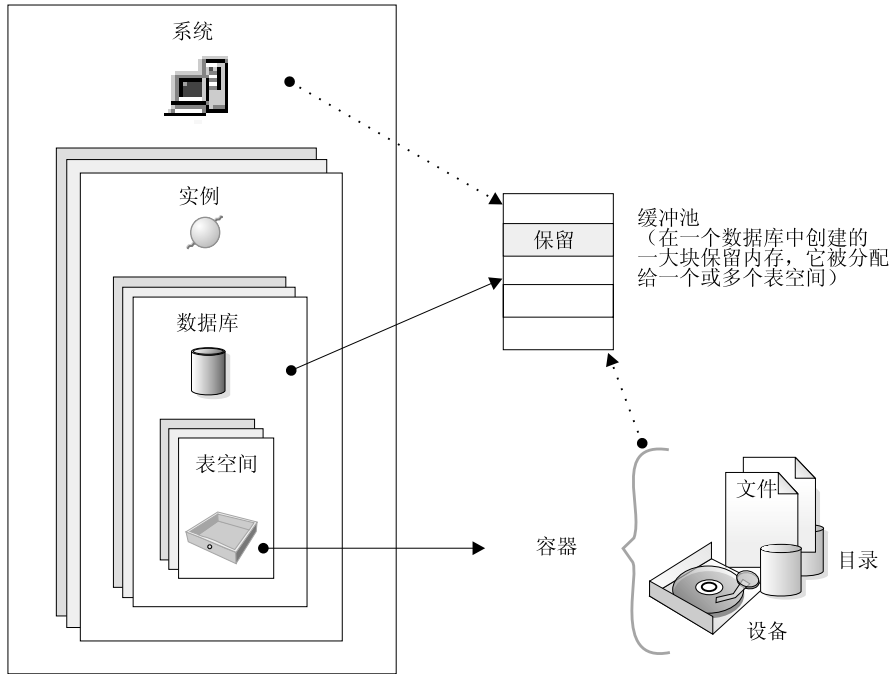


图9. 缓冲池和容器

系统对象概述

当创建 DB2 实例或数据库时, 将使用缺省参数值创建相对应的配置文件。可修改这些参数值以改进性能。

配置参数

配置文件包含特定的参数, 这些参数定义诸如分配给 DB2 产品和个别数据库的资源以及诊断级别之类的值。有两种类型的配置文件: 用于每个 DB2 实例的数据库管理程序配置文件, 以及用于每个个别数据库的数据库配置文件 (参见第19页的图10)。

数据库管理程序配置文件是在创建 DB2 实例时创建的。它包含的参数在实例级影响系统资源，不受任何一个作为该实例一部分的数据库的影响。根据系统的配置，可将这些参数中的许多参数的值更改为非系统缺省值，以改进性能或增加容量。

每个客户机安装也有一个数据库管理程序配置文件。此文件包含关于特定工作站的客户机使能器的信息。在可用于服务器的参数中，有一个子集可用于客户机。

数据库配置文件是在创建数据库时创建的，它驻留在数据库所驻留的地方。每个数据库都有一个配置文件。其参数指定要分配给该数据库的资源量以及其他事项。可更改许多参数的值，以改进性能或增加容量。根据特定数据库中活动类型的不同，可能需要进行不同的更改。

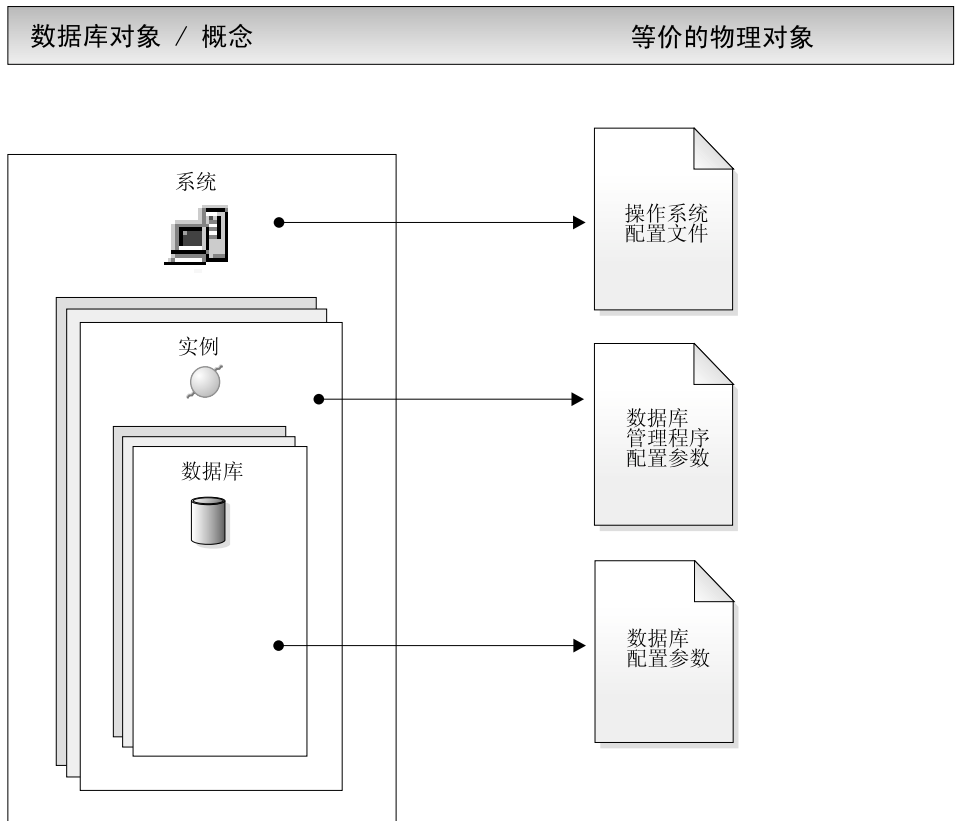


图 10. 配置参数文件

数据的商业规则

在任何商业中，数据通常必须符合特定限制或规则。例如，雇员号码必须是唯一的。DB2 提供了约束作为实施这样的规则的方法。

DB2 提供了下列约束类型：

- NOT NULL 约束
- 唯一约束
- 主关键字约束
- 外部关键字约束
- 检查约束

NOT NULL 约束

NOT NULL 约束防止空值进入一个列。

唯一约束

唯一约束确保一组列中的值对于表中的所有行都是唯一的，且不为空。例如，DEPARTMENT 表中的典型唯一约束可以是：部门号是唯一的，且不为空。

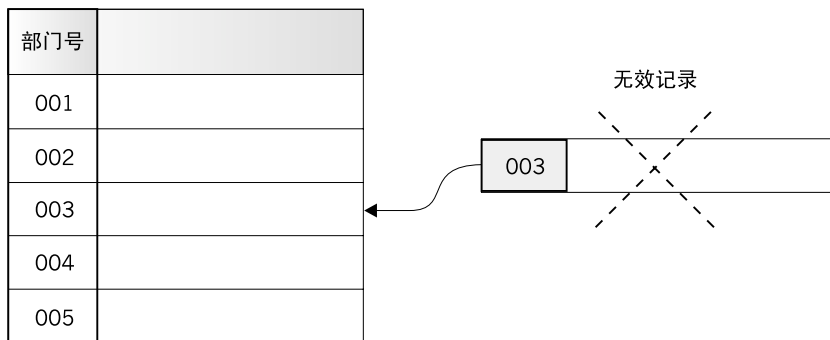


图 11. 唯一约束防止数据重复

数据库管理程序在插入和更新操作期间强制执行此约束，以确保数据完整性。

主关键字约束

每个表都可以有一个主关键字。主关键字是与唯一约束具有相同特性的一个列或列的组合。可使用主关键字和外部关键字约束来定义表之间的关系。

因为主关键字用来标识表中的一行，所以它应该是唯一的，并且只进行非常少的添加或删除。一个表不能有多个主关键字，但可以有多个唯一关键

字。主关键字是可选的，可以在创建或改变表时定义。当调出或重组数据时，主关键字可以对数据进行排序，所以它们也是有益的。

在下面的表中，DEPTNO 和 EMPNO 是 DEPARTMENT 表和 EMPLOYEE 表的主关键字。

表 1. DEPARTMENT 表

DEPTNO (主关键字)	DEPTNAME	MGRNO
A00	Spiffy Computer Service Division	000010
B01	Planning	000020
C01	Information Center	000030
D11	Manufacturing Systems	000060

表 2. EMPLOYEE 表

EMPNO (主关键字)	FIRSTNAME	LASTNAME	WORKDEPT (外部关键字)	PHONENO
000010	Christine	Haas	A00	3978
000030	Sally	Kwan	C01	4738
000060	Irving	Stern	D11	6423
000120	Sean	O'Connell	A00	2167
000140	Heather	Nicholls	C01	1793
000170	Masatoshi	Yoshimura	D11	2890

外部关键字约束

外部关键字约束（也称为参考完整性约束）使您能够定义表间以及表内必需的关系。

例如，典型的外部关键字约束可能规定 EMPLOYEE 表中的每个雇员必须是一个现有部门的成员，该部门在 DEPARTMENT 表中定义。

要建立此关系，应将 EMPLOYEE 表中的部门号定义成外部关键字，并将 DEPARTMENT 表中的部门号定义成主关键字。

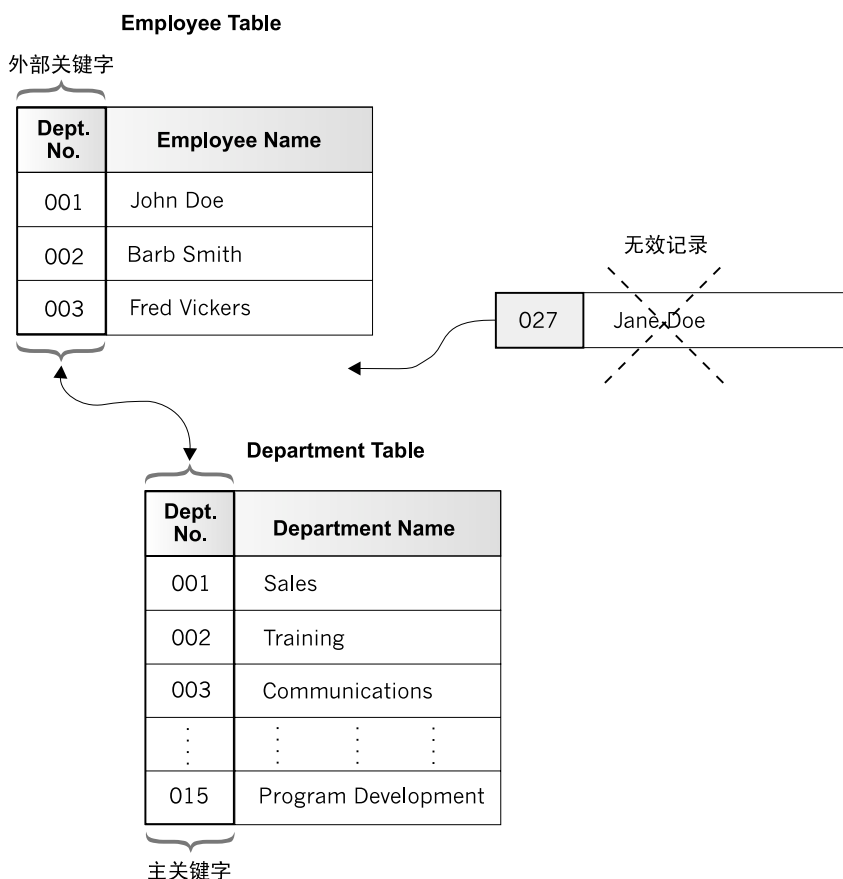


图 12. 外部关键字约束和主关键字约束定义关键并保护数据

检查约束

检查约束是指定每个表行的一列或多列所允许的值的数据库规则。

例如，在 **EMPLOYEE** 表中，可将“工作类型”列定义为“推销员”、“经理”或“职员”。有了此约束，在“工作类型”列中具有不同的值的任何记录都是无效的，将被拒绝，并实施关于表中允许的数据类型的规则。

还可在数据库中使用触发器。与约束相比，触发器更为复杂，且潜在功能更加强大。它们定义一组操作，这组操作与用于指定基表的 **INSERT**、**UPDATE** 或 **DELETE** 子句一起执行或由这些子句触发。可使用触发器支持一般形式的完整性或商业规则。例如，在接受订单之前，可使用触发器来检查客户的信用额度，也可以在银行业务应用程序中使用触发器，以便在帐户提款未符合客户的标准提款模式时提高警惕。有关触发器的详情，参考 *Application Development Guide*。

恢复数据库

数据库可能由于硬件和 / 或软件故障而变得不能使用，不同的故障情形可能需要不同的恢复操作。应该有一个适当的预演策略，以防止数据库可能出现故障。

本节讨论不同的恢复方法，并显示了如何确定哪一种恢复方法最适合于您的业务环境。包括下列主题：

- 『恢复概述』
- 第27页的『影响恢复的因素』
- 第40页的『灾难恢复的考虑事项』
- 第41页的『降低媒体故障的影响』
- 第43页的『降低事务故障的影响』
- 第43页的『分区数据库系统中的系统时钟同步』

恢复概述

您需要知道当数据库中存在问题时可使用的策略。这些问题包括媒体和存储器问题、电源中断和应用程序故障。您可以备份数据库或个别表空间，一旦它们以某种方式被损坏或破坏就可重建它们。数据库备份的概念与任何其他数据备份的概念相同：复制一份数据，并将其存储在另一媒体上，以防原始媒体发生故障或损坏。最简单的备份案例涉及关闭数据库（以确保不发生更多的事务），然后简单地对其进行备份。

数据库的重建称为恢复。应急恢复在发生故障后自动尝试恢复数据库。有两种方法来恢复损坏的数据库：版本恢复和前滚恢复。

在不可恢复数据库中禁用了 *logretain* 和 *userexit* 数据库配置参数。这表明只有保存的日志才是应急恢复所必需的。这些日志称为活动日志，它们包含当前事务数据。使用脱机备份的版本恢复是解决不可恢复数据库的恢复问题的主要手段。（脱机备份表示当备份操作正在进行时，其他应用程序无法使用该数据库。）这样的数据库只能进行脱机复原。它被复原为创建备份映象时它所处的状态。

可恢复数据库将 *logretain* 数据库配置参数设置为 "RECOVERY" 和 / 或启用 *userexit* 数据库配置参数。活动日志仍可用于应急恢复，但您还有已归档日志，它包含已提交的事务数据。这样的数据库只能进行脱机复原。它被复原为创建备份映象时它所处的状态。然而，对于前滚恢复，可通过使用活动日志和已归档日志来将数据库正向滚动（即，越过创建备份映象的时间）至一个特定的时间点，或者滚动至活动日志末尾。

可恢复数据库备份操作可以脱机执行，也可以联机执行（联机表示在备份操作期间其他应用程序可与该数据库连接）。数据库复原和前滚操作总是必须以脱机方式执行。联机备份操作期间，前滚恢复确保捕捉所有表更改，且在复原该备份时重新应用这些更改。

若有一个可恢复数据库，则可备份、复原和前滚个别表空间，而不是对整个数据库操作。当联机备份表空间时，它仍然可用，同时发生的更新被记录在日志中。当对表空间执行联机复原或前滚操作时，在该操作完成之前，该表空间本身不可用，但不阻止用户存取其他表空间中的表。

应急恢复防止数据库处于不一致或不可用状态。对数据库执行的事务（也称工作单元）可能被意外中断。若在作为工作单元一部分的所有更改完成和落实之前发生故障，则该数据库就会处于不一致和不可用的状态。

然后，需要将该数据库转移为一致和可用的状态。为此，回滚未完成的事务，并完成当发生崩溃时仍在内存中的已落实事务（图13）。

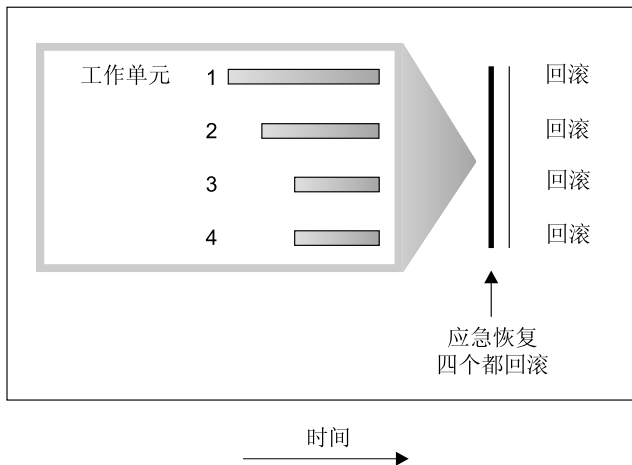


图 13. 回滚工作单元

当数据库处于一致和可用状态时，它处于一种称为“一致点”的状态。脱机数据库备份表示一个一致点。达到一致点时，所有事务都已处理完成，并且数据可由其他用户或应用程序使用。

发生崩溃之后，可通过调用 `RESTART DATABASE` 命令转移至某个一致点（参见 *Command Reference*）。若希望在每次发生故障时都执行此命令，应考虑使用启用 `自动重新启动 (autorestart)` 配置参数。此数据库配置参数的缺省行为是每当需要时都调用 `RESTART DATABASE`。当启用 `autorestart` 时，在故障后对数据库的下一个连接请求导致调用 `RESTART DATABASE` 命令。

应急恢复将数据库转变为一致和可用的状态。但是，若对允许正向恢复的数据库应用应急恢复（即，将 *logretain* 配置参数设置为 "RECOVERY"，或启用 *userexit* 配置参数），且在应急恢复期间因个别表空间而发生错误，则必须使该表空间脱机，直到修复后才能存取它。应急恢复继续进行。在应急恢复完成时，该数据库中的其他表空间仍是可使用的，且可以与该数据库建立连接。（对于有临时表或系统目录表的表空间，存在例外情况。这将在前滚恢复中进行讨论。）

正如前面所提到的那样，DB2 提供了两种方法来恢复损坏的数据库：

- 版本恢复指的是使用备份操作期间创建的映象来复原数据库的先前版本。

数据库复原操作使用早先建立的数据库的备份来重建整个数据库。数据库的备份允许您将数据库复原至建立该备份时所处的状态。从备份时间至故障时间之间的每个工作单元都丢失（参见图14）。

使用版本恢复方法，必须定期调度和执行完整数据库备份。

在一个分区数据库环境中，数据库位于许多数据库分区服务器（或节点）上。必须复原所有分区，而且用于复原数据库操作的所有备份映象必须是同时建立的。（每个数据库分区是单独备份和复原的。）同时建立的每个数据库分区的备份称为版本备份。

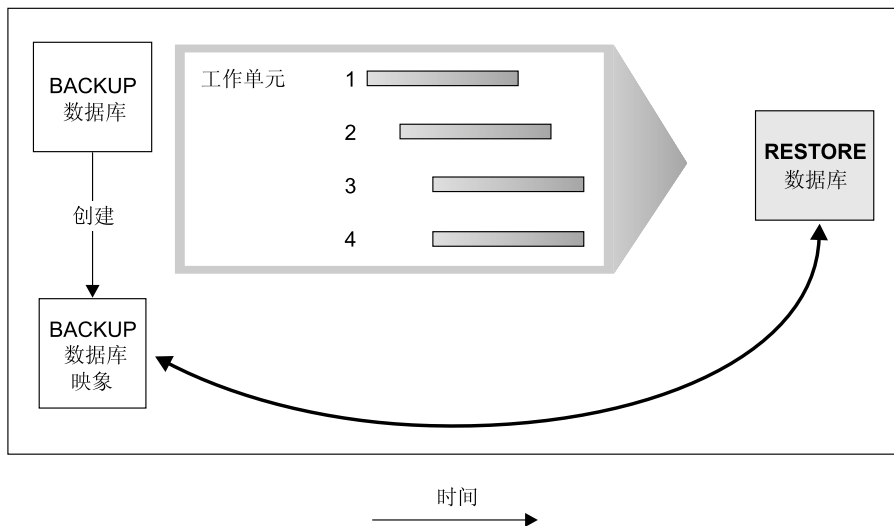


图 14. 复原数据库

- 要使用前滚恢复方法，必须已建立了数据库的备份，并且已将日志归档（通过启用 *logretain* 和 / 或 *userexit* 数据库配置参数）。有关您必须对所用的记录过程作出的决策的详情，参见第29页的『数据库日志』。复原数据库并指定 **WITHOUT ROLLING FORWARD** 选项等效于使用版本恢复方法。此数据库被

复原到创建脱机备份映象时的状态。若复原数据库，但没有对复原数据库操作指定 `WITHOUT ROLLING FORWARD` 选项，则该数据库在复原操作结束时将处于前滚暂挂状态。这允许进行前滚恢复。

要考虑的两类前滚恢复是：

- 数据库前滚恢复。在此类前滚恢复中，在数据库复原操作后应用数据库日志中记录的事务（参见图15）。该数据库日志记录了对该数据库所作的所有更改。这种方法将数据库恢复到在某特定时间点的状态，或恢复到故障前的状态（即，恢复到活动日志的末尾）。

在一个分区数据库环境中，数据库位于许多数据库分区上。若执行时间点前滚恢复，则必须前滚所有数据库分区，以确保所有分区都在同一级别。若需要复原单个数据库分区，则可以执行前滚恢复至日志末尾，以将它复原至与数据库中的其他分区相同的级别。

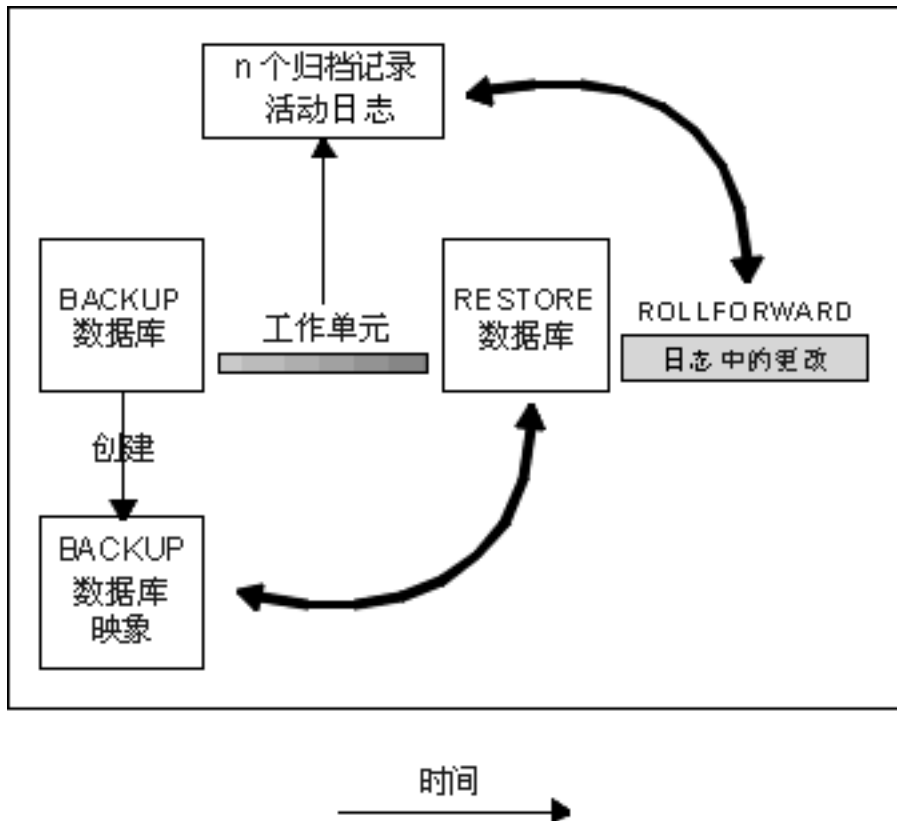


图 15. 数据库前滚恢复

- 表空间复原和前滚。若允许数据库前滚恢复，则也可能备份、复原和前滚表空间。要执行表空间复原和前滚操作，需要整个数据库（即所有表空间）或一个或多个个别表空间的备份映象。还需要影响要恢复的表空间的运行记录。可在日志中前滚至以下两点之一：
 - 日志末尾；或
 - 一个特定时间点（称为时间点恢复）。

注：

1. 执行备份操作时未选择的表空间将与复原后的表空间的状态不同。
2. 当对表空间使用前滚恢复方法时，必须标识数据库中要恢复的“关键”表空间，并定期调度和执行数据库（或“关键”表空间）的备份。

可在下列两种情况下使用表空间前滚恢复：

- 在一个表空间复原操作后，该表空间始终处于前滚暂挂状态，且必须将它前滚。调用 `ROLLFORWARD DATABASE` 命令（参考 *Command Reference*）来将日志应用于表空间，使其恢复到某个时间点或日志末尾。
- 若一个或多个表空间在应急恢复后处于前滚暂挂状态，首先应校正该表空间的问题。某些情况下，校正表空间的问题不包括执行复原数据库操作。例如，掉电可能导致表空间处于前滚暂挂状态。若在应急恢复之前校正此问题，应急恢复将足以使数据库恢复到一致可用的状态。在这种情况下，复原数据库操作并不是必需的。一旦校正了该表空间的问题，可使用 `ROLLFORWARD DATABASE` 命令将日志应用于表空间，使其恢复到某个时间点或日志末尾。

注：若出错的表空间包含系统目录表，将不能启动数据库。必须复原 `SYSCATSPACE` 表空间，然后执行前滚恢复直至日志末尾。

在分区数据库环境中，若要将表空间前滚至一个时间点，不必提供该表空间所驻留的节点（数据库分区）的列表。DB2 将前滚请求提交至所有分区。这意味着必须在表空间驻留的所有数据库分区上恢复表空间。

在分区数据库环境中，若要将表空间前滚至日志末尾，而又不希望在所有的分区上前滚表空间，则必须提供数据库分区的列表。若要在所有分区上将所有处于前滚暂挂状态的表空间前滚到日志末尾，则不必提供数据库分区的列表。缺省情况下，会将前滚数据库请求发送至所有分区。

影响恢复的因素

要确定使用哪种数据库恢复方法，必须考虑下列关键因素：

- 该数据库将是可恢复的，还是不可恢复的？
- 需要将数据库恢复至离故障时间多近的位置（恢复点）？

- 恢复数据库可能花费多长时间？这包括：
 - 备份之间的时间（将影响前滚恢复）
 - 该数据库可用或可存取的时间（根据数据可用性需要，以联机或脱机方式备份）
- 可以为副本和归档日志分配多少存储空间？
- 将使用表空间级备份，还是完整的数据库级备份？

通常，数据库维护和恢复策略应确保所有的信息在需要恢复数据库时都是可用的。该策略应包括用于执行数据库备份的定期调度，以及当创建数据库时的预定备份，或者，对于一个分区数据库系统，当通过添加或卸下数据库分区服务器（节点）来调整系统时的预定备份。除这些基本要求外，好的策略还会包括降低数据库故障的可能性并减少影响的元素。

下列主题提供其他信息：

- 『可恢复的和不可恢复的数据库』
- 第29页的『数据库日志』
- 第33页的『减少工作表的记录』
- 第35页的『恢复点』
- 第35页的『备份的频率和必需的时间』
- 第36页的『必需的恢复时间』
- 第36页的『存储器考虑事项』
- 第37页的『将相关的数据保存在一起』
- 第38页的『使用不同操作系统的限制』
- 第38页的『损坏的表空间的恢复』
- 第39页的『恢复性能考虑事项』。

虽然本节的整体重点集中在数据库上，但是整体恢复计划应该包括恢复：

- 操作系统和 DB2 可执行程序
- 操作系统库中的应用程序、UDF 和存储过程代码
- 用于创建 DB2 实例和非 DB2 资源的命令
- 操作系统安全性
- 通过装入操作来装入副本（若在 LOAD 命令上指定 COPY YES 的话）

可恢复的和不可恢复的数据库

若可以很容易地重新创建数据，则存放该数据的数据库可以是不可恢复的数据库。例如：

- 保存只读应用程序的外部源数据的表（且该数据不与现存数据混合）应考虑放在不可恢复的数据库中。
- 带有少量数据的表。对于这种情况，恢复是没有问题的。问题在于，没有对该数据做足够的记录，而无法协调在复原之后管理日志文件和进行前滚所增加的复杂性。
- 定期添加少量行的大表。同样，没有足够的变更率，来协调在复原操作后管理日志文件和进行前滚。

若重新创建数据并不简单，则存放该数据的数据库应该是可恢复数据库。以下是应作为可恢复数据库一部分的数据的示例：

- 不能重新创建的数据。这包括装入数据后它的源已损坏的数据，和以人工方式输入表中的数据。
- 装入数据库之后，应用程序或工作站用户修改的数据。

数据库日志

所有数据库都有相关的日志。这些日志保存了有关数据库更改的记录。若需要将数据库复原至上一完全、脱机备份之前的一个点，则需要日志才能将数据前滚至故障点。

有两类 DB2 记录：*循环*和*归档*，它们提供了不同级别的恢复能力。

当创建新数据库时，*循环*记录是缺省行为。对于这种类型的记录，只有完全、脱机数据库备份才有效。正如其名称所暗示的那样，循环记录使用一个联机日志“环”来提供从事务故障和系统崩溃的恢复。仅使用和保留日志到确保当前事务的完整性这样一个程度。循环记录不允许将数据库从上一完全备份前滚穿过先前事务。通过从完全、脱机备份进行复原来完成从媒体故障和灾难的恢复。在上次备份之后的所有更改都将丢失。进行完全备份时，数据库必须脱机（用户不可存取）。因为这种类型的复原将数据恢复至完全备份的特定时间点，所以它称为版本恢复。

第30页的图16显示当循环记录活动时，活动日志使用一个日志文件环。

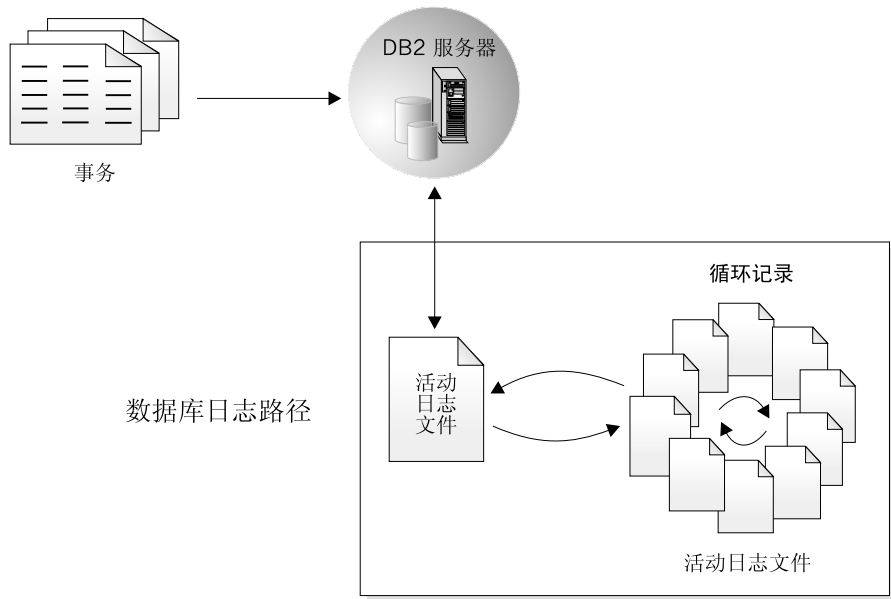


图 16. 循环记录

应急恢复期间，使用活动日志来防止故障（系统电源或应用程序错误）使数据库处于不一致的状态。**RESTART DATABASE** 命令在需要时使用活动日志来将数据库转变为一致和可用状态。在应急恢复期间，这些日志中所记录的因故障而未落实的更改被回滚。重新执行虽已落实但未从内存（缓冲池）实际写入磁盘（数据库容器）的那些更改。这些操作确保了数据库的完整性。若需要的话，**ROLLFORWARD DATABASE** 命令也可以在时间点恢复或至日志末尾的恢复期间使用活动日志。活动日志位于数据库日志路径目录中。

归档日志专门用于前滚恢复。它们可以是：

联机归档日志 当活动日志中的更改不再需要用于正常处理时，该日志就会关闭，然后变成归档日志。当归档日志存储在数据库日志路径目录中时，就称它为联机的（参见第31页的图17）。

脱机归档日志 当在数据库日志路径目录中再找不到归档日志时，就称该归档日志为脱机的（参见第32页的图18）。也可以使用用户出口程序将归档日志存储在数据库日志路径目录之外的位置中。（有关其他信息，参见管理指南：实现中的“用于数据库恢复的用户出口”。）

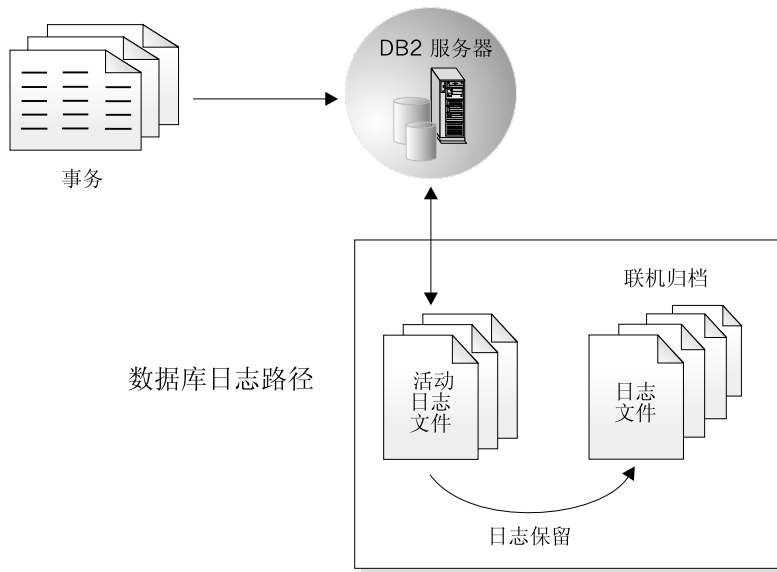


图 17. 归档记录

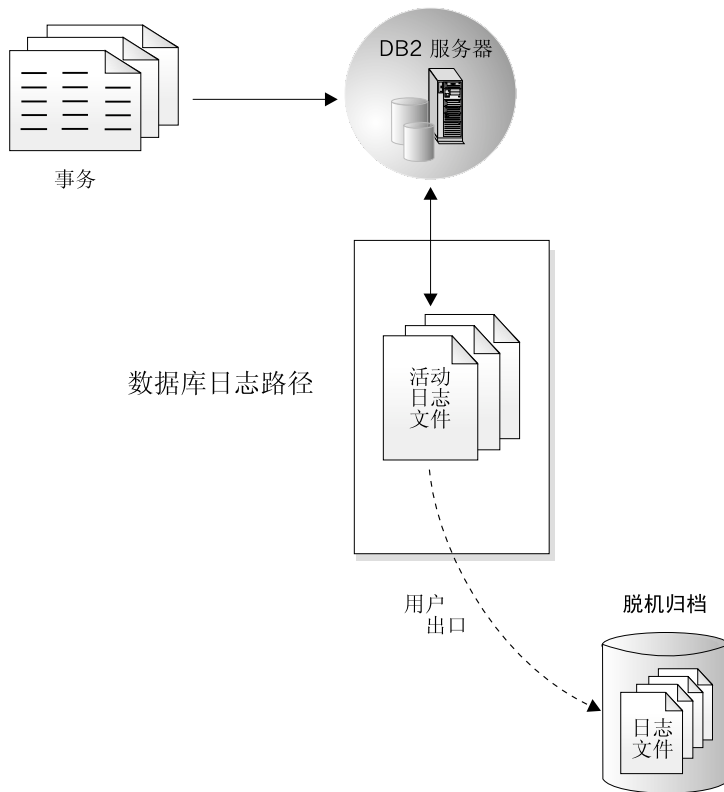
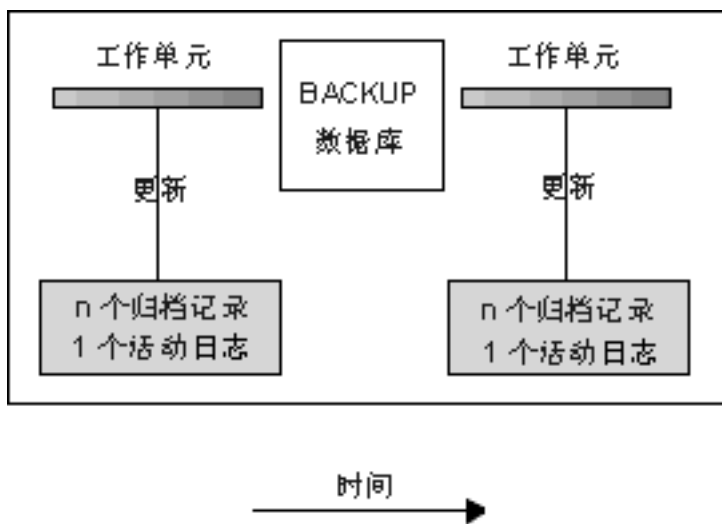


图 18. 脱机归档日志

前滚恢复可使用归档日志和活动日志来将数据库重建至日志末尾或特定时间点。为此，前滚功能将在归档日志和活动日志中发现的已落实更改重新应用于复原的数据库。

前滚恢复还能使用日志重建表空间，方法是重新应用在归档日志和活动日志中已落实的更新。可以将表空间恢复至日志末尾或特定的时间点。

联机备份期间，所有针对数据库的活动都被记录下来。复原联机备份时，必须至少将日志前滚至完成备份的时间点。为此，必须将日志归档并使它们在复原数据库时可用。在备份操作完成后，备份时使用的日志文件可以继续保持打开。**BACKUP DATABASE** 命令上用于联机备份的 **FLUSH LOG** 选项将在联机备份完成时关闭活动日志。这使您能够将活动日志归档，以便您将有一个完整的备份以及复原该备份所需的所有日志。



在备份之间使用记录跟踪数据库的更改。

图 19. 前滚恢复中活动的和归档的数据库日志

有两个数据库配置参数可用来更改归档日志的存储位置：*newlogpath* 参数和 *userexit* 参数。更改 *newlogpath* 参数还会影响活动日志的存储位置。有关这些配置参数的详情，参考管理指南：性能。

要确定数据库日志路径目录中的哪些日志数据块（参见第16页的『容器』）是归档日志，检查 *loghead* 数据库配置参数的值。此参数指示活动的最低编号的日志。那些序号小于 *loghead* 的日志是归档日志且可移动。可使用控制中心检查此参数的值；也可使用命令行处理器和 `GET DATABASE CONFIGURATION` 命令来查看“首个活动日志文件”。有关此配置参数的详情，参考管理指南：性能。

注：

1. 若擦除了一个活动日志，则该数据库就会成为不可使用的，且必须将它复原之后，才能再次使用。只能前滚至擦除的第一个日志为止。
2. 若担心活动日志可能已损坏（由于磁盘失效导致），应该考虑为存储这些日志的卷建立镜像。

减少工作表的记录

若您的应用程序根据主表创建和填充了工作表，且您不担心这些工作表的恢复性（因为它们可以从主表很容易地重新创建），则您可能希望在 `CREATE TABLE` 语句上指定 `NOT LOGGED INITIALLY` 参数创建工作表。使用 `NOT LOGGED INITIALLY` 参数的优点是，不记录在创建该表的同一个工作单元中对该表所作的任

何更改（包括插入、删除、更新或创建索引操作）。这不仅减少了记录工作量，也可能提高应用程序的性能。还可以对现存表使用带 NOT LOGGED INITIALLY 参数的 ALTER TABLE 语句，来获得同样的结果。

注：

1. 可以在同一个工作单元中使用 NOT LOGGED INITIALLY 参数创建多个表。
2. 仍会记录对目录表和其他用户表的更改。

因为不记录表的更改，所以当决定使用 NOT LOGGED INITIALLY 参数时应该考虑下列事宜：

- 在落实时，必须将对表的所有更改写到磁盘。这意味着该落实可能占用更长的时间。
- 在创建该表的工作单元中执行任何操作而返回的错误将导致整个工作单元回滚 (SQLCODE -1476, SQLSTATE 40506)。
- 当前滚时，不能恢复这些表。若前滚操作遇到使用 NOT LOGGED INITIALLY 选项创建的表，会将该表标记为不可用。在恢复了数据库之后，存取该表的任何尝试都将返回 SQL1477N。

注：当创建了一个表时，将会保持对目录表的行锁定，直到执行 COMMIT 为止。要利用不记录行为，必须在创建该表的同一个工作单元中填充该表。这就存在并发情况。有关详情，参考管理指南：性能中的“并行性”。

有关创建表的详情，参考 *SQL Reference*。

若计划使用已说明临时表作为工作表，注意下列各项：

- 已说明临时表不是在目录中创建的；因此，不挂起锁定。
- 不对已说明临时表执行记录，甚至在第一个 COMMIT 之后也如此。
- 使用 ON COMMIT PRESERVE 选项来使各行在 COMMIT 之后仍留在表中；否则，将删除所有行。
- 只有创建已说明临时表的应用程序才能存取表的那个实例。
- 当卸下应用程序与数据库的连接时，隐式卸下该表。
- 工作单元中使用一个已说明临时表的操作中的错误不会导致该工作单元完全回滚。然而，更改已说明临时表内容的语句中的操作错误将删除该表中的所有行。回滚工作单元（或保存点）将删除已说明临时表中被该工作单元（或保存点）修改过的所有行。

有关已说明临时表及其限制的详情，参考 *SQL Reference* 中的 DECLARE GLOBAL TEMPORARY TABLE 语句。

恢复点

版本恢复方法和前滚恢复方法提供不同的恢复点。版本方法涉及到在预定时间为数据库建立一个脱机的、完整的数据库副本。使用此方法，恢复的数据库将仅恢复到已复原的副本所处的状态。例如，若在每天结束时建立一个副本，并在第二天的中间时间丢失该数据库，将会丢失相当于半天的更改。

在前滚恢复方法中，对数据库的更改保留在日志中。使用此方法，先使用副本来复原数据库或表空间；然后使用这些日志来重新应用自创建该副本以来对数据库所做的更改。

启用前滚恢复后，可以利用联机备份和表空间级备份。要进行完整的数据库和表空间的前滚恢复，可以选择恢复至日志末尾或指定的时间点。例如，若一个应用程序破坏了数据库，可从该数据库的一个复原副本开始，将更改前滚至启动该应用程序之前的那一刻。不重新应用在指定时间之后写至日志的工作单元。

还可以将表空间前滚至日志末尾或特定的时间点。

备份的频率和必需的时间

恢复计划应该允许定期的预定备份，因为备份数据库需要时间和系统资源。

即使已将日志归档，也应定期执行完整的数据库备份（以便允许前滚恢复）。若恢复策略包括前滚恢复，一个最新的完整数据库备份意味着只需将更少的归档日志应用于数据库，这样减少了 `ROLLFORWARD` 实用程序恢复数据库所需的时间。

还应该考虑不要覆盖备份和日志，而应该保存多个完整的数据库备份及其相关的日志作为额外的预防措施。

当恢复和前滚非常活跃的数据库时，若应用归档日志所需的时间是主要关注的问题，则请考虑进行更频繁的数据库备份的成本。这减少了前滚时需要应用的归档日志数目。

可以在数据库联机或脱机时执行备份。若它是联机的，则在运行备份操作的同时，其他应用程序或进程可以继续与该数据库连接并继续读取和修改数据。若脱机执行该备份，则只有备份操作可与数据库连接；当备份任务正在运行时，您机构中的其余任务都不能与数据库连接。

要缩短数据库不可用的持续时间，考虑使用联机备份。仅当启用前滚恢复时，才支持联机备份。若启用前滚恢复且有一组完整的日志，可以在需要时重建数据库。

注:

1. 若有一个或多个数据库日志跨越备份操作的持续时间, 则只能使用联机备份。
2. 脱机备份比联机备份快。

若一个数据库包含大量的长整数字段和 LOB 数据, 则备份该数据库可能会占用非常多的时间。BACKUP 命令能够备份选择的表空间。若使用 DMS 表空间, 则可以将不同类型的数据存储在它们各自的表空间中, 以减少备份操作所需的时间。可以将表数据保存在一个表空间中, 将长整数字段和 LOB 数据保存在另一个表空间中, 再将索引保存在一个表空间中。通过将长整数字段和 LOB 数据存储在单独的表空间中, 不选择备份包含该长整数字段和 LOB 数据的表空间, 可以减少完成备份所需的时间。若长整数字段和 LOB 数据对于您的业务很重要, 应参照完成这些表空间的复原操作所需的时间考虑备份这些表空间。若可从单独的源复制 LOB 数据, 则当创建或改变一个表以包括 LOB 列时, 选择 NOT LOGGED 选项。

若重组一个表, 应该在该操作完成后备份受影响的表空间。若不得不复原这些表空间, 将不必通过数据重组来前滚。

注: 若备份未包含所有表数据的一个表空间, 则不能对该表空间执行时间点前滚恢复。包含一个表的任何类型数据的所有表空间都必须同时前滚至同一个时间点。

必需的恢复时间

恢复一个数据库所需的时间由两部分组成: 复原备份所需的时间; 以及, 若允许数据库进行前滚恢复, 在前滚操作期间应用日志所需的时间。当确定恢复计划时, 应考虑这些恢复成本和它们对商业运作的影响。测试整体恢复计划有助于确定恢复数据库所需的时间是否适合给定的业务需求。在每次测试之后, 可能要增加建立备份的频率。若前滚恢复是策略的一部分, 这将会减少备份之间归档的日志数, 因此也减少了复原操作之后前滚数据库所需的时间。

注: 设置“启用分区内并行性” (*intra_parallel*) 数据库管理程序配置参数不影响备份或复原操作的性能。不管 *intra_parallel* 参数的设置如何, 都将使用多个进程来完成这两个操作。

存储器考虑事项

当确定要使用哪种恢复方法时, 考虑必需的存储空间。

版本恢复方法需要空间来容纳数据库的副本和复原的数据库。前滚恢复方法需要空间来容纳数据库或表空间的副本、复原的数据库和归档的数据库日志。

若一个表包含长整数字段或大对象 (LOB) 列，应考虑将此数据置于单独的表空间中。这将会影响存储空间考虑事项，并影响恢复计划。如果使用单独的表空间来存储长整数字段和 LOB 数据，并知道备份长整数字段和 LOB 数据所需的时间，那么可以决定使用这样一个恢复计划，它以较低的频率保存此表空间的备份。也可选择在创建或改变一个表以包括 LOB 列时，不记录对那些列的更改。这将减少必需的日志空间和对应的日志归档空间的大小。

包含 LOB 的 SMS 表空间的备份大小可能比原始表空间的大小要大。根据表空间中的 LOB 数据大小，该备份可以增大 40%。例如，若建立了一个 1GB 的 SMS 表空间（含有 LOB），当复原它时将需要大于 1GB 的磁盘空间。此只在支持稀疏分配的文件系统（例如，基于 UNIX 的操作系统）上才会出现。

要防止媒体故障损坏数据库且使您无法重建它，应该在不同的设备上保存数据库备份、数据库日志和数据库本身。鉴于此原因，极力建议您使用 *newlogpath* 配置参数，这样一旦创建了该数据库，就将数据库日志置于单独的设备中。（此配置参数和与记录相关的其他配置参数在 *管理指南：实现中的“在数据库中前滚更改”* 中有讨论。）

数据库日志可能会占用大量的存储器。若计划使用前滚恢复方法，则必须确定管理归档日志的方法。以下是您的选择：

- 在数据库日志路径目录中提供足够的空间来保存这些日志。
- 当某些日志不再出现在活动的日志集中后，以人工方式将这些日志复制到存储设备或非数据库日志路径目录的一个目录中。
- 使用用户出口程序来将这些日志复制到您环境中的另一个存储设备中。（有关详情，参见 *管理指南：实现中的“用于数据库恢复的用户出口”*。）

注：在 OS/2 上，DB2 支持用户出口程序将数据库备份映象和数据库日志存储在标准和非标准的设备上。有关详情，参见 *管理指南：实现中的“用于数据库恢复的用户出口”*。

将相关的数据保存在一起

作为数据库设计的一部分，您将了解表之间存在的关系。这些关系可以在应用程序级表示（当事务更新多个表时），也可以在数据库级表示（表之间存在参考完整性，或者一个表上的触发器影响另一个表）。当制定恢复计划时，应该考虑这些关系。您将希望把相关的数据集备份在一起。可以在表空间级或数据库级建立这样的集合。通过将相关的数据集保存在一起，可以恢复至所有数据都一致的时间点。若您希望能够对表空间执行时间点前滚恢复，这一点尤其重要。

使用不同操作系统的限制

在具有多个操作系统的—个环境中工作时，必须注意备份和恢复计划是不能集成的。也就是说，可能在一个操作系统上不能使用 `BACKUP DATABASE` 命令，而在另一个操作系统上不能使用 `RESTORE DATABASE` 命令。应使每个操作系统的恢复计划分开并使它们互不相关。

若必须将表从一个操作系统移动至另一个操作系统，可使用 `db2move` 命令；或使用 `EXPORT` 与 `IMPORT` 或 `LOAD` 命令。有关详情，参考 *Data Movement Utilities Guide and Reference*。

损坏的表空间的恢复

损坏的表空间带有一个或多个不能存取的容器。这通常是由媒体问题引起的：这些问题或者是永久性的（例如，坏的磁盘），或者是临时性的（例如，脱机磁盘或未安装的文件系统）。

若损坏的表空间是系统目录表空间，则不能重新启动数据库。若不能修复容器问题但要使原始数据不受影响，可用的选项包括：

- 复原此数据库；或
- 复原目录表空间。

注：表空间复原仅对可恢复的数据库有效，因为必须前滚该数据库。

若损坏的表空间不是系统目录表空间，DB2 将试图使该数据库尽可能多的部分变得可用；在这种情况下，成功与否取决于记录策略。

若损坏的表空间是唯一的临时表空间，则一旦与该数据库建立了连接，便应创建一个新的临时表空间。创建后，可使用这个新的临时表空间，并且需要临时表空间的正常数据库操作也可以继续执行。若希望的话，可卸下脱机的临时表空间。使用系统临时表空间的表重组有特殊的考虑事项：

- 若数据库或数据库管理程序配置参数 `indexrec` 设置为 "RESTART"，在数据库活动期间必须重建所有无效的索引；这包括构建阶段期间发生故障的重组中的索引。
- 若在损坏的临时表空间中有未完成的重组请求，可能必须将 `indexrec` 配置参数设置为 "ACCESS" 以避免重新启动失败。

可恢复的数据库的表空间恢复：损坏的表空间被置于脱机和不可存取状态，并且，因为必须执行应急恢复，所以还处于前滚暂挂状态。若没有其他问题，则重新启动操作将成功。可以再次使用损坏的表空间的条件是：

- 修复损坏的容器而不丢失原始数据，然后完成表空间前滚操作。（前滚操作将首先尝试使该表空间由脱机状态转移至正常状态。）
- 修复损坏的容器（丢失或不丢失原始数据）之后，执行表空间复原操作，然后执行前滚操作。

不可恢复的数据库的表空间恢复： 因为应急恢复是必需的，且日志不会无限期保存，因此仅当用户希望卸下损坏的表空间时重新启动操作才会成功。（恢复成功完成意味着将损坏的表空间恢复到一致状态所必需的运行记录不再存在；因此，对这类表空间可以采取的唯一有效操作是卸下它们。）

可通过调用未限定的重新启动数据库操作来做到这一点。若没有损坏的表空间，它将成功。若它失败 (SQL0290N)，则可查看 db2diag.log 文件，以获得当前已损坏的表空间的完整列表。

- 如果您愿意在重新启动数据库操作完成时卸下所有这些表空间，则可启动另一重新启动数据库操作，使用 `DROP PENDING TABLESPACES` 选项列示所有已损坏的表空间。若损坏的表空间包括在 `DROP PENDING TABLESPACES` 列表中，则表空间进入卸下暂挂状态，在恢复之后，唯一的选项是卸下该表空间。重新启动操作继续，而不恢复此表空间。若损坏的表空间未包括在 `DROP PENDING TABLESPACES` 列表中，则重新启动数据库操作失败，并返回 SQL0290N。
- 若您不想卸下这些表空间并因此丢失其中的数据，则有以下选项：
 - 等待并修复损坏的容器（没有丢失原始数据），然后再次尝试重新启动操作
 - 执行数据库复原操作。

注： 将表空间名放入 `DROP PENDING TABLESPACES` 列表中并不表示该表空间将处于卸下暂挂状态。仅当在重新启动操作期间发现表空间损坏时才会是这样。重新启动操作成功后，应发出 `DROP TABLESPACE` 语句来卸下每个处于卸下暂挂状态的表空间（调用 `LIST TABLESPACES` 命令来了解哪些表空间处于此状态）。这样，可收回该空间，或者可重新创建那些表空间。

恢复性能考虑事项

当考虑恢复性能时，应注意下列各项：

- 可以将日志置于单独的设备上，以提高频繁更新的数据库的性能。在联机事务处理 (OLTP) 环境中，更常见的是需要 I/O 来将数据写至日志而不是存储数据行。将日志置于单独的设备上，可将在日志和数据库文件之间进行移动所需的磁盘臂移动最小化。

还应该考虑该磁盘上的其他文件。例如，将日志移至一个系统中用于系统分页的磁盘，而该磁盘没有足够的实内存，这样会破坏调整。

- 要缩短完成一次复原操作所需的时间:
 - 调整复原缓冲区大小。缓冲区大小必须是在备份操作期间使用的缓冲区大小的倍数。
 - 增加缓冲区的数量。

若使用多个缓冲区和 I/O 通道，应该使用至少两倍于通道数量的缓冲区，以确保通道不必等待数据。所用的缓冲区大小也影响复原操作的性能。理想的复原缓冲区大小应该是表空间的数据块大小的倍数。

若有多个具有不同数据块大小的表空间，则指定最大数据块大小的一个倍数的一个值。

建议的最小缓冲区数目是媒体设备或容器的数目加上对 PARALLELISM 选项指定的数目。
 - 使用多个源设备。
 - 对于复原操作，将 PARALLELISM 选项设置为至少比源设备数大 1。
- 若一个表包含大量的长整数字段和 LOB 数据，则复原它可能会占用非常多的时间。若允许数据库进行前滚恢复，则 RESTORE 命令能够复原选择的表空间。若该长整数字段和 LOB 数据对于您的业务很重要，应参照完成这些表空间的备份任务所需的时间考虑复原这些表空间。通过将长整数字段和 LOB 数据存储在单独的表空间中，不选择复原包含该长整数字段和 LOB 数据的表空间，可以减少完成复原操作所需的时间。若可从单独的源复制 LOB 数据，则当创建或改变一个表以包括 LOB 列时选择 NOT LOGGED 选项。若选择不复原包含长整数字段和 LOB 数据的表空间，但需要复原包含该表的表空间，则必须前滚到日志末尾，以便所有包含表数据的表空间都是一致的。

注：若备份包含表数据的一个表空间，而未备份相关的长整数或 LOB 字段，则不能对该表空间执行时间点前滚恢复。必须将一个表的所有表空间同时前滚至同一个时间点。
- 下列说明适用于备份和复原操作:
 - 应使用多个 I/O 缓冲区和设备。
 - 分配至少两倍于正在使用的设备数量的缓冲区。
 - 不要使 I/O 设备控制器带宽超载。
 - 使用数量多的小缓冲区，而不使用数量少的大缓冲区。
 - 根据系统资源调整缓冲区的数量和大小。

灾难恢复的考虑事项

术语灾难恢复用于描述在火灾、地震、恶意破坏或其他大灾害事件中复原数据库所需要执行的活动。灾难恢复的计划可以包括下列其中一项或多项:

- 在紧急情况中要使用的场地
- 用于恢复数据库的另一台机器
- 数据库备份和归档日志的非现场存储器。

若灾难恢复计划是在另一台机器上恢复整个数据库，则至少需要一个完整的数据库备份和该数据库的所有归档日志。可选择通过将归档日志应用于数据库，以备备用数据库保持最新。或者，可选择将数据库备份和日志归档保存在备用场地，并只在发生灾难之后执行复原和前滚操作。（在这种情况下，最新的数据库备份无疑是所希望的。）但是，当发生灾难时，将所有事务恢复至灾难发生时的状态一般是不可能的。

灾难恢复的表空间备份的有用性取决于发生故障的范围。通常，灾难恢复需要复原整个数据库，因此，应在备用场所保存完全数据库备份。即使有每个表空间的单独备份映象，也不能使用它们来恢复该数据库。若该灾难是磁盘损坏，则可以使用该磁盘上每个表空间的表空间备份来恢复。若由于磁盘故障（或任何其他原因）而无法存取一个容器，可以将该容器复原至不同的位置。有关其他信息，参见**管理指南：实现中的“在 RESTORE 期间重新定义表空间容器”**。

表空间备份和完整的数据库备份都可以在任何灾难恢复计划中起到一定的作用。可用于备份、复原和前滚数据的 DB2 设施为灾难恢复计划提供了一个基础。应确保已测试了恢复过程，它可以保护您的业务。

降低媒体故障的影响

要降低媒体故障的可能性，并简化从此类故障的恢复：

- 镜象或复制保存重要数据库的数据和日志的磁盘。
- 在分区数据库环境中，要设置一个严格的过程，来处理该目录节点上的数据和日志。因为此节点是维护数据库的关键：
 - 确保它驻留在可靠的磁盘上
 - 复制它
 - 频繁地进行备份
 - 不要在此节点上存放用户数据。

防止磁盘故障

若您担心由于磁盘故障而使数据或日志损坏的可能性，则请考虑使用某种形式磁盘故障容错。通常，这通过使用**磁盘阵列**来实现。磁盘阵列由一组磁盘驱动器组成，对于一个应用程序，磁盘阵列表现为单一的大容量磁盘驱动器。

磁盘阵列包含磁盘条纹化（即将一个文件分布至多个磁盘）、磁盘的镜象和数据奇偶校验。

磁盘阵列有时简单地称为 RAID（独立磁盘冗余阵列）。特定术语 RAID 一般只应用于硬件磁盘阵列。磁盘阵列也可以通过处于操作系统级或应用程序级的软件来提供。硬件磁盘阵列和软件磁盘阵列的不同点在于处理 I/O 请求的 CPU 处理的方式。对于硬件磁盘阵列，I/O 活动由磁盘控制器管理，对于软件磁盘阵列，这由操作系统或应用程序完成。

硬件磁盘阵列 (RAID): 在 RAID 磁盘阵列中，一个磁盘控制器可使用和管理多个磁盘，且使用它自己的 CPU 来完成。管理构成此阵列的磁盘所需的所有逻辑都包含在磁盘控制器中；因此，此实现与操作系统无关。

RAID 体系结构有五种类型：RAID-1 至 RAID-5，它们每一种都提供了磁盘故障容错。其每一种的功能和性能均有所不同。通常，RAID 指冗余阵列。这里不讨论 RAID-0，它只提供数据条纹化（而未提供故障容错冗余）。虽然 RAID 说明定义了五种体系结构，但是现在通常只使用 RAID-1 和 RAID-5。

RAID-1 也称为磁盘镜象或磁盘双工。磁盘镜象使用单个磁盘控制器将数据（一个完整的文件）从一个磁盘复制到第二个磁盘。磁盘双工，除磁盘与第二个磁盘控制器连接（象两个 SCSI 适配器）外，与磁盘镜象相同。数据可以得到较好的保护。任何一个磁盘都可能发生故障，但仍可从另一个磁盘存取数据。使用双工，磁盘控制器也可能发生故障，但仍能提供数据保护。使用 RAID-1 的性能也较好，但是此实现中的代价是所需的磁盘容量是实际数据量的两倍，因为数据是在一对驱动器上复制。

RAID-5 涉及到在所有磁盘上按扇区分布数据和奇偶性条纹化。奇偶性与数据信息交错在一起，而不是存储在单独的驱动器中。数据可以得到较好的保护。若任何磁盘发生故障，仍可以使用其他磁盘的信息以及条纹化奇偶性信息来存取该数据。读性能较好，但写性能比 RAID-1 或普通磁盘差很多。RAID-5 配置需要最少三个完全相同的磁盘。额外开销所需的额外磁盘空间容量随阵列中的磁盘数改变。对于配置了 5 个磁盘的 RAID-5，空间的额外开销是 20%。

使用 RAID（但不是 RAID-0）磁盘阵列时，故障磁盘不会使您存取不了该阵列上的数据。当在该阵列中使用可热插入的或可热交换的磁盘时，可以在该阵列正在使用时将替换磁盘与故障磁盘交换。对于 RAID-5，若两个磁盘同时发生故障，则会丢失所有的数据（但是同时发生磁盘故障的可能性非常小）。

可以考虑使用 RAID-1 或软件镜象磁盘（参见第43页的『软件磁盘阵列』）来存储日志，因为这可使您恢复至故障点且提供较好的写性能，这一点对于日志很重要。若您的情况是可靠性很关键（不能将时间浪费在发生磁盘故障之后的恢复数

据上)，而写性能不那么关键，则应考虑使用 RAID-5 磁盘。另外，若写性能很关键，且您愿意使用附加的磁盘空间来达到此目的，则考虑使用 RAID-1 来存储数据和日志。

软件磁盘阵列：软件磁盘阵列在许多方面与硬件磁盘阵列相同（参见第42页的『硬件磁盘阵列 (RAID)』），但是它的磁盘通信的管理是由操作系统任务或在服务器上运行的应用程序来执行。软件阵列象其他程序一样，必须争用 CPU 和系统资源。对于 CPU 受约束的系统，它不是一个好的选项，同时要记住磁盘阵列的总体性能取决于服务器的 CPU 负荷和能力。

典型的软件磁盘阵列提供磁盘镜象（参见第42页的『硬件磁盘阵列 (RAID)』）。虽然冗余磁盘是必需的，但是由于不需要高成本的 RAID 磁盘控制器，所以软件磁盘阵列的实现相对便宜。

注：如果操作系统引导驱动器在磁盘阵列中，则当该驱动器发生故障时，系统将无法启动。若在磁盘阵列运行之前该驱动器发生故障，则磁盘阵列就不允许存取该驱动器。引导驱动器应与磁盘阵列分开。

降低事务故障的影响

要降低事务故障的影响，应尽量确保：

- 不间断电源
- 有足够的磁盘空间用于存储数据库日志
- 在分区数据库环境中的数据库分区服务器之间有可靠的通信链路
- 在分区数据库环境中的系统时钟同步（参见『分区数据库系统中的系统时钟同步』）。

分区数据库系统中的系统时钟同步

应该使所有数据库分区服务器的系统时钟保持相对同步，以确保数据库操作顺利进行以及正向可恢复性不受限制。数据库分区服务器之间的时间差加上一个事务的任何潜在的操作和通信延迟，应小于对 *max_time_diff*（节点之间的最大时间差）数据库管理程序配置参数指定的值。

为了确保运行记录的时间戳记反映分区数据库系统中的事务的顺序，DB2 使用每台机器上的系统时钟作为运行记录中时间戳记的基准。但是，若将系统时钟设置得提前，也自动将日志时钟设置得提前。虽然可以将系统时钟设置得落后，但是日志时钟却不能这样设置，它会保持*相同*的超前时间，直至系统时钟与此时间相匹配为止。于是，这两个时钟便同步了。这表示一个数据库节点上的短期系统时钟错误可能对数据库日志的时间戳记产生长期的影响。

例如，假定数据库分区服务器 A 上的系统时钟被错误地设置为 1999 年 11 月 7 日，而当前年份是 1997 年，并假定在数据库分区服务器上的分区中落实了更新事务之后将该错误校正。若继续使用该数据库，且过一段时间便定期更新它，则 1997 年 11 月 7 日至 1999 年 11 月 7 日之间的任何点实际上是不可通过前滚恢复达到的。当完成数据库分区服务器 A 上的 COMMIT 时，数据库日志中的时间戳记被设置为 1999，而日志的时钟会停留在 1999 年 11 月 7 日，直到系统时钟与此时间相匹配为止。若试图前滚至此时间范围内的一个时间点，则操作将在超过指定的停止点（即 1997 年 11 月 7 日）的第一个时间戳记处停止。

虽然 DB2 不能控制对系统时钟的更新，但是 *max_time_diff* 数据库管理程序配置参数降低了发生此类机会：

- 此参数的可配置值的范围是 1 分钟至 24 小时。有关设置 *max_time_diff* 的详情，参考管理指南：性能。
- 当对非目录节点发出第一个连接请求时，数据库分区服务器会将它的时间发送至该数据库的目录节点。该目录节点就会检查请求该连接的节点上的时间以及它自己的时间是否在 *max_time_diff* 参数指定的范围之内。若超过此范围，则拒绝该连接。
- 涉及到数据库中两个以上数据库分区服务器的更新事务，必须先验证参与的数据库分区服务器上的时钟是否同步，然后才可落实该更新。若两个或多个数据库分区服务器的时差超过了 *max_time_diff* 允许的限制，则会回滚该事务，以防止将不正确的时间传播至其他数据库分区服务器。

要校正一个数据库日志中不正确的的时间戳记，并防止它进一步传播：

1. 将系统时钟校准为正确的时间。
2. 使用在不正确地设置时间之前所建立的备份，在适当的数据库分区服务器上复原数据库分区。
3. 将数据库分区的更改前滚至日志末尾。
4. 在前滚了更改之后，立即建立该数据库分区的副本。

当完成这些操作之后，将校准日志时间，且不会传播不正确的的时间戳记，而且您将可以使用在该数据库分区上建立的上一个备份来执行时间点恢复。

重组数据库中的表

在许多次的更新之后，表会成为碎片状，这导致性能严重下降。若收集了统计信息，但看不出有明显的性能改进，则重组表数据可能会有所帮助。重组表数据时，根据指定的索引重新安排数据的物理顺序，并除去碎片数据中的空闲空间。这使该数据可以更快速地被存取，从而改进性能。

在重组表之前，建议您调用 `REORGCHK` 命令，并收集该表的统计信息。运行此命令将帮助您确定重组表数据是否适当。关于 `REORGCHK` 命令的详情，参考 *Command Reference*。

DB2 安全性概述

要保护与一个数据库服务器相关的数据和资源，DB2 同时使用外部安全服务与内部存取控制信息。要存取一个数据库服务器，必须在被允许存取数据库数据或资源之前通过一些安全性检查。数据库安全性中的第一步称为认证，在此步骤，您必须证明您的身份是真实的。第二步称为授权，在此步骤，数据库管理程序决定是否允许已经验证的用户执行请求的操作或存取请求的数据。

认证

使用 DB2 之外的安全性设施来完成用户认证。此安全性设施可以是操作系统的一部分，可以是一个独立的产品，或者，在特定情况下，可能根本不存在。例如，在基于 UNIX 的系统上，安全性设施包含在操作系统之内。“DCE 安全服务”是一个独立的产品，它为分布式环境提供安全性设施。而在 Windows 95 或 Windows 3.1 操作系统上没有安全性设施。

安全性设施需要两个项目来认证用户：用户 ID 和口令。用户 ID 向安全性设施标识用户。通过提供正确的口令（只有该用户和安全性设施才知道的信息）来验证该用户的身份（与该用户 ID 相对应）。

一旦认证：

- 就必须使用一个 SQL 授权名或 *authid* 来向 DB2 标识该用户。此名称可以与用户 ID 或一个映像值相同。例如，在基于 UNIX 的系统上，可将符合 DB2 命名约定的一个 UNIX 用户 ID 转换为大写字体，来获得一个 DB2 *authid*。在“DCE 安全服务”产品中，DB2 *authid* 包含在 DCE 注册表中，一旦认证成功完成，则可从注册表中将它提取出来。
- 获取用户所属的组的列表。在授权用户时，可使用组成员资格。组是安全性设施实体，它们也必须映射至 DB2 授权名。执行此映射的方法与映射用户 ID 的方法类似。

DB2 将获得最多 64 组的组列表。若一个用户所属的组超过 64 组，只将映射至有效 DB2 授权名的前 64 组添加到 DB2 组列表。不会返回任何错误，而且 DB2 将忽略前 64 组之后的所有组。

DB2 通过下列两种方式之一，使用安全性设施来认证用户：

- DB2 使用成功的安全性系统注册作为身份证明，并允许：
 - 使用本地命令存取本地数据

- 使用远程连接，在这种情况下服务器信赖客户机认证。
- DB2 接受用户 ID 和口令的组合。它使用安全性设施对此组合的成功验证作为身份证明，并允许：
 - 使用远程连接，在这种情况下服务器需要认证的证明
 - 使用操作，在这种情况下用户希望以某个不同于注册时所用的身份来运行命令。

DB2 管理员可允许其他人在 AIX 和 Windows NT EEE 系统上，通过简要表注册表变量 DB2CHGPWD_EEE 来更改口令。此变量的缺省值是 NOT SET（禁用）。DB2CHGPWD_EEE 接受由其他 DB2 简要表变量使用的标准布尔值。

DB2 管理员负责在 Windows NT 上使用“Windows NT 域控制器”或在 AIX 上使用 NIS，来确保集中维护所有节点的口令。

注：若没有集中维护口令，启用 DB2CHGPWD_EEE 变量可能造成口令在所有节点上不一致。即，若使用“更改口令”功能部件，则您的口令仅在您连接的节点上被更改。

AIX 上的 DB2 UDB 可记录对操作系统的失败口令尝试次数，并检测客户机何时超过允许的注册尝试次数，该值由 LOGINRETRIES 参数指定。

有关系统进入验证检查的其他信息（若您有存取数据库的远程客户机，该信息就更有参考价值），参见管理指南：实现中的“为服务器选择认证方法”。

授权

授权是一个过程，通过这个过程 DB2 获取有关一个已认证的 DB2 用户的信息，此信息指示该用户可执行的数据库操作，以及可以存取的数据对象。执行每个用户请求时，根据所涉及的对象和操作，可能有多个授权检查。

使用 DB2 设施来执行授权。DB2 表和配置文件用于记录与每个授权名相关的许可权。将一个已认证的用户的授权名和此用户所属的那些组的授权名，与记录的许可权比较。根据这个比较，DB2 决定是否允许执行请求的存取。

DB2 记录了两种类型的许可权：特权和权限级。特权为授权名定义单个许可权，它使用户能够创建或存取数据库资源。特权存储在数据库目录中。权限级提供将特权分组的一个方法，并控制更高级别的数据库管理程序的维护和实用程序操作。数据库特定的权限存储在数据库目录中；系统权限与组成员资格相关联，并存储在一个给定实例的数据库管理程序配置文件中。

组提供了一个简便的方法，来对一组用户执行授权，而不必单独对每个用户授予或撤消特权。除非另有指定，否则，组授权名可以用在为了授权而使用授权名的

任何地方。通常，对于动态 SQL 和非数据库对象授权（如实例级命令和实用程序），考虑使用组成员资格，但对于静态 SQL，则不考虑使用它。这种一般情况的例外是，当授予 PUBLIC 特权时：在处理静态 SQL 时要考虑它们。DB2 文档中的适当地方提到了组成员资格不适用的特殊情况。

有关详情，参见*管理指南：实现中的“特权、权限和授权”*。

联合体数据库认证和授权概述

因为 DB2 联合体数据库系统可存取多个数据库管理系统中的信息，所以可能需要附加的步骤来保密数据。

当计划认证方法时，应考虑用户可能需要通过数据源和 DB2 的认证检查这一事实。在联合体系统中，认证可在 DB2 客户机工作站、DB2 服务器、数据源（DB2、DB2 OS/390 版、其他 DRDA 服务器、Oracle）或在 DB2（客户机或 DB2 服务器）和数据源的组合中进行。甚至在 DCE 环境中，若数据源需要用户 ID 和口令，也可能需要特定的步骤。有关详情，参见*管理指南：实现中的“联合体数据库认证处理”*。

类似地，用户必须通过数据源和 DB2 的授权检查。每个数据源（DB2、Oracle、DB2 OS/390 版等）在其控制下维护对象的安全性。当用户根据译名执行操作时，用户必须通过该译名引用的表或视图的授权检查。

第3章 联合体系统

联合体数据库系统或联合体系统是一个数据库管理系统 (DBMS)，它支持提交特定 SQL 语句的应用程序或用户，这些 SQL 语句在单条语句中引用两个或更多个 DBMS 或数据库。一个示例是两个不同的 DB2 数据库中的表之间的连接。这种类型的语句称为分布式请求。

“DB2 通用数据库”联合体系统支持跨数据库和 DBMS 的分布式请求。例如，可在 DB2 表和 Oracle 视图之间执行一个 UNION 操作。受支持的 DBMS 包括 DB2、DB2 系列的成员（如 DB2 OS/390 版和 DB2 AS/400 版）和 Oracle。

DB2 联合体系统为数据库对象提供位置透明性。若（表和视图中的）信息被移动，则不必对请求该信息的应用程序进行任何更改，便可更新对该信息的引用（称为混名）。DB2 联合体系统还弥补了不支持所有 DB2 SQL 语言或特定优化能力的 DBMS 的不足。不能在这样的 DBMS 下执行的操作（如递归 SQL）可以在 DB2 下运行。

DB2 联合体系统以一种半自动方式运行：当 Oracle 应用程序存取同一个服务器时，可提交引用 Oracle 对象的 DB2 查询。DB2 联合体系统不独占或限制对 Oracle 或其他 DBMS 对象的存取权（不包括完整性 / 锁定约束）。

一个 DB2 联合体系统由一个 DB2 UDB 实例、一个充当联合体数据库的数据库以及一个或多个数据源组成。联合体数据库包含标识数据源及其特征的目录项。数据源由一个 DBMS 和数据组成。应用程序与联合体数据库连接就如同与任何其他 DB2 数据库连接一样。有关联合体数据库环境的形象化表示，参见第50页的图 20。

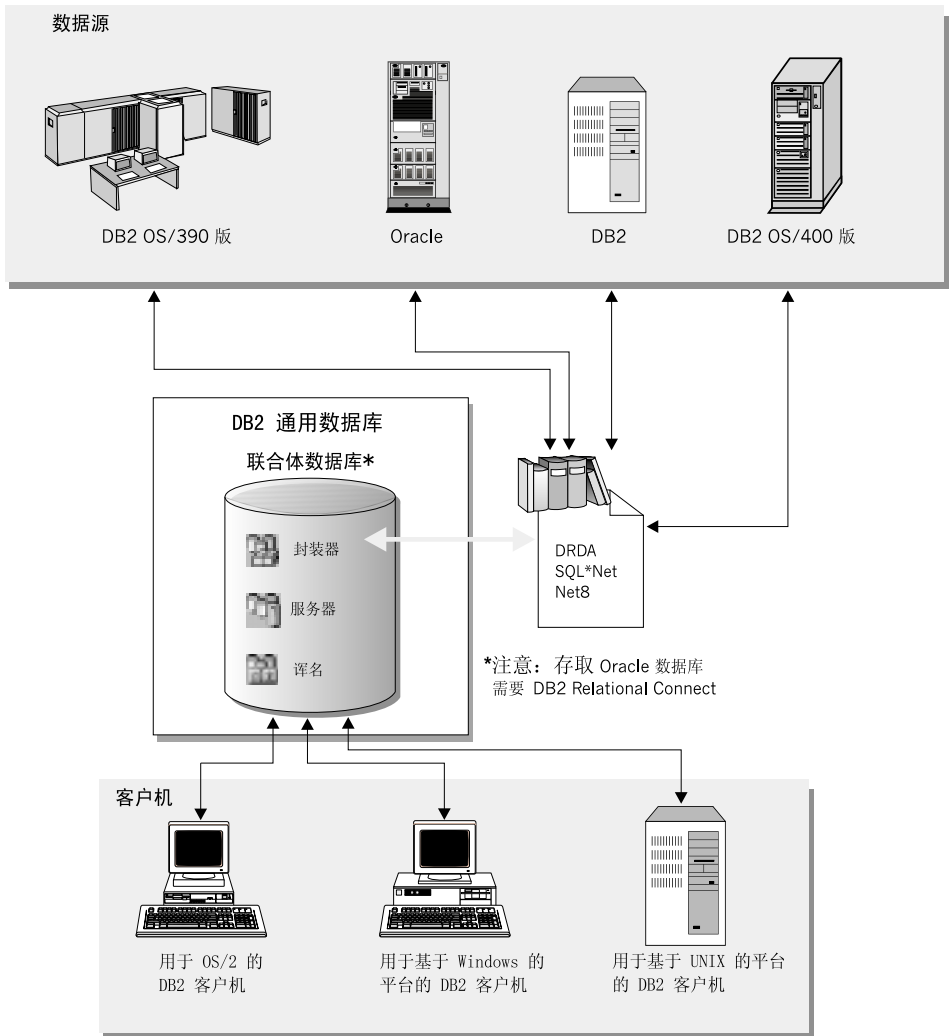


图 20. 联合体数据库系统

DB2 联合体数据库目录项包含有关数据源对象的信息：它们的名称、所包含的信息以及使用它们的条件。因此 DB2 目录将有关对象的信息存储在多个 DBMS 中，因此它称为全局目录。也将对象属性存储在该目录中。引用的实际 DBMS、用于与数据源通信的模块以及将存取的 DBMS 数据对象（如表）在数据库之外。（一个例外是：联合体数据库可以是联合体系统的一个数据源。）可使用控制中心或 SQL DDL 语句创建联合体对象。必需的联合体数据库对象包括：

封装器 标识存取特定类或类别的数据源所用的模块（DLL、库等）。

服务器 定义数据源。服务器数据包括封装器名、服务器名、服务器类型、服务器版本、授权信息和服务器选项。

浑名 存储在联合体数据库中的标识符，它们引用特定的数据源对象（表、别名、视图）。应用程序在查询中引用浑名就象引用表和视图一样。

可创建附加对象，这取决于您的特定需要：

- 用户映射，解决认证问题
- 数据类型映射，定制数据源类型和 DB2 类型之间的关系
- 函数映射，将本地函数映射为数据源函数
- 索引规范，改进性能。

设置联合体系统后，可以象在一个大型数据库中一样存取数据源中的信息。用户和应用程序将查询发送到一个联合体数据库，然后该数据库根据需从 DB2 系列和 Oracle 系统中检索数据。用户和应用程序在查询中指定浑名；这些混名提供对数据源中的表和视图的引用。从最终用户的角度来看，浑名与别名是相似的。

有许多因素影响联合体系统性能。最关键的因素是确保将关于数据源的准确的、最新的信息存储在联合体数据库的全局目录中。DB2 优化器使用此信息，该信息可影响是否下推操作以便在数据源求值的决定。有关联合体系统性能的其他信息，参考管理指南：性能。

DB2 联合体系统运行时受某些限制的约束。分布式请求只限于只读操作。此外，不能对浑名执行实用程序操作（LOAD、REORG、REORGCHK、IMPORT、RUNSTATS 等）。

但可使用通过设施并使用与数据源相关的 SQL 语言将 DDL 和 DML 语句直接提交给数据库管理程序。

联合体系统支持并行环境。将联合体数据库按语义细分为本地对象（表、视图）引用和浑名引用的范围限制了性能改善的程度。对浑名数据的请求按顺序处理；本地对象可并行处理。例如，假设有查询 SELECT * FROM A, B, C, D, 其中 A 和 B 是本地表，而 C 和 D 是引用位于 Oracle 数据源的表的浑名，一个可能的方案是将表 A 和 B 并行连接。结果是按顺序与浑名 C 和 D 连接。

启用联合体系统

“DB2 企业版” (EE) 和 “DB2 扩充企业版” (EEE) 可以支持联合体数据库。要启用联合体系统：

1. 在安装期间，选择 DB2 EE 或 EEE 的 DB2 数据库的分布式连接安装选项。

2. 若联合体系统中包括 Oracle 数据库，则安装 DB2 Relational Connect。有关详情，参考[安装和配置补遗](#)。
3. 将数据库管理程序配置参数 *federated* 设置为 "YES"。
4. 创建封装器、服务器和诨名（有关详情，参见[管理指南：实现中的“创建数据库”](#)）。
5. 按要求创建附加的对象或设置选项（有关详情，参见[管理指南：实现中的“实现设计”](#)）。

第4章 并行数据库系统

DB2 将数据库管理程序扩展到了并行多节点环境中。数据库分区是数据库的一部分，它由自己的数据、索引、配置文件和事务日志组成。数据库分区有时称为节点或数据库节点。（“节点”是在“DB2 并行版 AIX 版的版本 1”产品中使用的术语。）

单分区数据库是只有一个数据库分区的数据库。该数据库中的所有数据都存储在此分区中。对于这种情况，节点组（若存在的话）不提供任何附加功能（参见第9页的『节点组』）。

分区数据库是有两个或更多个数据库分区的数据库。表可位于一个或多个数据库分区中。当一个表在由多个分区组成的节点组中时，它的某些行存储在一个分区中，而其他行存储在其他分区中。

通常，每个物理节点都存在单个数据库分区，而在每个数据库分区中，数据库管理程序使用每个系统上的处理器来管理该数据库中的全部数据中属于该分区的那一部分。

由于数据分布在各数据库分区上，所以可在多个物理节点上使用多个处理器的能力来满足对信息的请求。数据检索和更新请求被自动分解成子请求，并在适用的数据库分区中并行执行。数据库分布在各数据库分区中的这个事实对于发出 SQL 语句的用户是透明的。

用户交互作用通过一个数据库分区发生，该数据库分区称为该用户的协调程序节点。协调程序与应用程序在相同的数据库分区上运行，或者对于远程应用程序，协调程序在该应用程序连接的数据库分区上运行。任何数据库分区都可用作协调程序节点。

节点组和数据分区

可以在数据库中定义一个或多个数据库分区组成的命名子集。您定义的每个子集称为节点组。包含多个数据库分区的每个子集称为多分区节点组。多分区节点组只能用属于相同实例的数据库分区来定义。

第54页的图21给出了一个含五个分区的数据库示例，在这个示例中：

- 一个节点组横跨除一个数据库分区外的所有其他分区（节点组 1）。
- 一个节点组包含一个数据库分区（节点组 2）。

- 一个节点组包含两个数据库分区。
- 节点组 2 中的数据库分区与节点组 1 共享并与之相交。
- 节点组 3 中只有一个数据库分区与节点组 1 共享并与之相交。

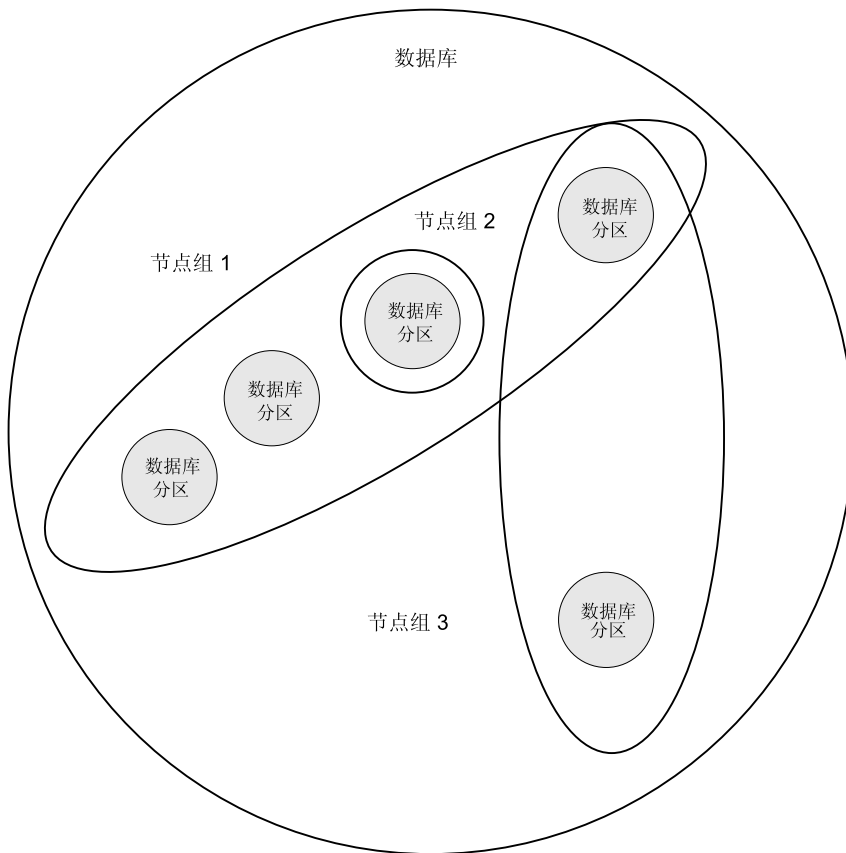


图 21. 数据库中的节点组

使用 `CREATE NODEGROUP` 语句可以创建一个新节点组。有关详情，参考 *SQL Reference*。数据分布在一个节点组中的所有分区上。若使用多分区节点组，则必须注意几个节点组设计考虑事项。有关详情，参见第113页的『设计节点组』。

并行性的类型

任务（如数据库查询）的各个部分可并行运行以大幅增强性能。任务的性质、数据库配置和硬件环境确定了 DB2 将如何以并行方式执行任务。这些考虑事项是相关的，在决定数据库的物理和逻辑设计方案时，应当一起考虑它们。本节描述受 DB2 支持的下列类型的并行性：

- I/O
- 查询
- 实用程序

I/O 并行性

当一个表空间有多个容器时，数据库管理程序可以利用并行 I/O。并行 I/O 指的是同时处理对两个或多个 I/O 设备的写入或读取；这能够显著地改进吞吐量。

I/O 并行性是第58页的『硬件环境』中描述的每个硬件环境的一个组成部分。第65页的表3列出了最适合 I/O 并行性的硬件环境。

查询并行性

有两种查询并行性：查询间并行性和查询内并行性。

查询间并行性是指多个应用程序同时查询一个数据库的能力。每个查询都独立于其他查询执行，但是 DB2 同时执行所有查询。DB2 始终支持这种类型的并行性。

查询内并行性是指使用分区内并行性和 / 或分区间并行性来同时处理单个查询的各部分。

术语查询并行性将在全书中使用。

分区内并行性

分区内并行性是指将一个查询分为多个部分的能力。（有些实用程序也采用这种类型的并行性。查看第57页的『实用程序并行性』。

分区内并行性将通常认为的单个数据库操作（如索引创建、数据库装入或 SQL 查询）细分成多个部分，其中的大部分或全部操作可以在单个数据库分区内以并行方式运行。

图22显示了一个查询，它被分为可并行运行的四个部分，返回结果的速度比按串行方式运行该查询的速度快得多。这几部分互为副本。要使用分区内并行性，必须适当地配置数据库。您可以选择并行度，或由系统为您选择。并行度表示一个查询中并行运行的部分数。

第65页的表3列出了最适合分区内并行性的硬件环境。

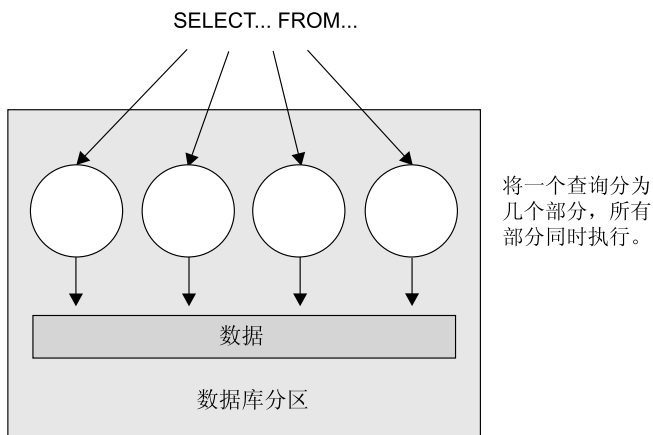


图 22. 分区内并行性

分区间并行性

分区间并行性是指将一个查询分为多个部分并将这几部分置于一个分区数据库的多个分区（位于一台或多台机器上）上的能力。该查询以并行方式运行。（有些实用程序也采用这种类型的并行性。查看第57页的『实用程序并行性』。

分区间并行性将通常认为的单个数据库操作（如索引创建、数据库装入或 SQL 查询）细分成多个部分，其中的大部分或全部操作可以在一台或多台机器上的一个分区数据库的多个分区中以并行方式运行。

第57页的图23显示了一个查询，它被分为可并行运行的四个部分，返回结果的速度比在单个分区上按串行方式运行该查询的速度快得多。

并行度在很大程度上取决于您创建的分区数和您定义节点组的方式。

第65页的表3列出了最适合分区间并行性的硬件环境。

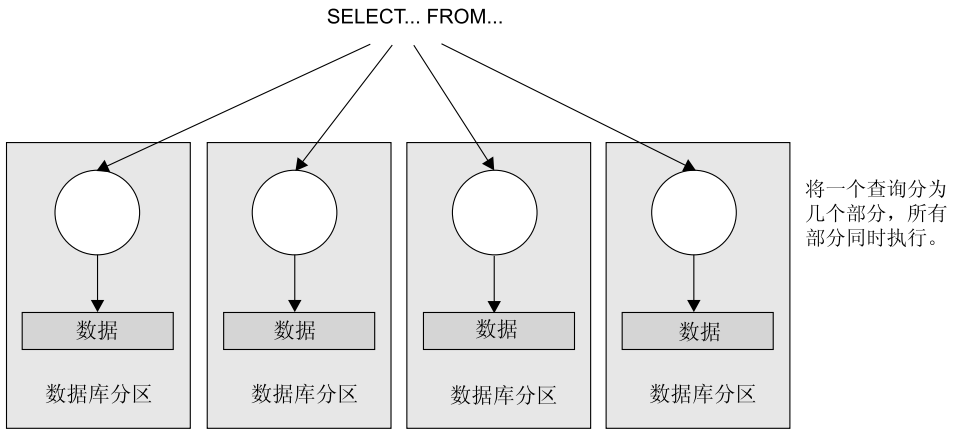


图 23. 分区间并行性

同时使用分区内和分区间并行性

可以同时使用分区内并行性和分区间并行性。此组合提供两种并行性程度，这也使处理查询的速度显著加快。

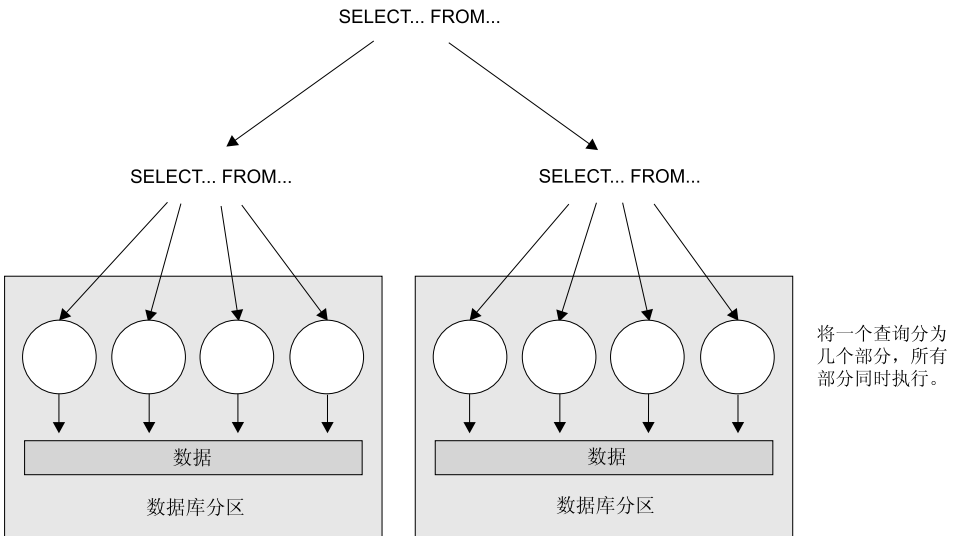


图 24. 同时使用分区内和分区间并行性

实用程序并行性

DB2 实用程序可以利用分区内并行性。它们还可以利用分区间并行性；前提是存在多个数据库分区，而实用程序在每个分区内并行执行。

装入实用程序可以利用分区内并行性和 I/O 并行性。装入数据是一个大量使用 CPU 的任务。装入实用程序利用多处理器来执行如语法分析和格式化数据这类任务。它也可使用并行 I/O 服务器来以并行方式将数据写入容器中。有关如何对装入实用程序启用并行性的详情，参考 *Data Movement Utilities Guide and Reference*。

在分区数据库环境中，“自动装入”实用程序通过在该表驻留的每个数据库分区上并行调用 LOAD 命令，来利用分区内、分区间或 I/O 并行性。有关“自动装入”实用程序的详情，参考 *Data Movement Utilities Guide and Reference*。

在创建索引期间，可并行执行数据的扫描和后续排序。在创建索引时，DB2 既利用 I/O 并行性又利用分区内并行性。这有助于在发出 CREATE INDEX 语句时、重新启动期间（若一个索引标记为无效）及数据重组期间加快索引创建的速度。

备份和复原数据任务会涉及到繁重的涉及 I/O 的任务。当执行备份和复原操作时，DB2 既利用 I/O 并行性又利用分区内并行性。备份操作通过以并行方式读取多个表空间容器并以并行方式异步写入多备份媒体，来利用 I/O 并行性。有关如何对 BACKUP DATABASE 命令和 RESTORE DATABASE 命令启用并行性的详情，参考 *Command Reference* 中对这些实用程序的介绍。

硬件环境

本节对下列硬件环境提供了一个概述：

- 单处理器上的单分区
- 具有多处理器的单分区 (SMP)
- 多分区配置
 - 具有一个处理器的分区 (MPP)
 - 具有多处理器的分区 (SMP 群集)
 - 逻辑数据库分区（在“DB2 并行版 AIX 版，版本 1”中又称为“多逻辑节点”，或 MLN）

下面讨论每种环境的容量和可扩充性。容量是指能存取数据库的用户数和应用程序数。这很大程度上取决于内存、代理程序数、锁定数、I/O 和存储器管理。可扩充性是指一个数据库随着其增长而继续显示出相同的操作特征和响应时间的能力。

单处理器上的单分区

此环境由内存和磁盘组成，但仅包含一个 CPU（参见图25）。可使用许多不同的名称来称呼它，包括独立数据库、客户机 / 服务器数据库、串行数据库、单处理器系统和单节点或非并行环境。

此环境中的数据库可满足一个部门或小办公室的需要，其中的数据和系统资源（包括一个单处理器或 CPU）由单数据库管理程序来管理。

第65页的表3列出了此硬件配置最适合的并行性类型。

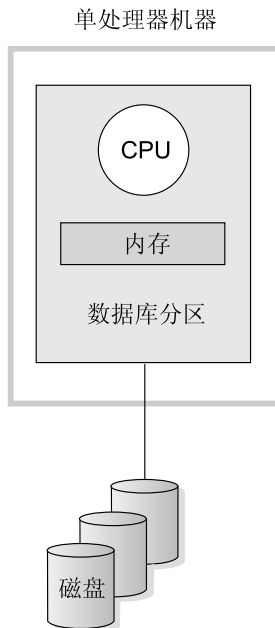


图 25. 单处理器上的单分区

容量和可扩充性

在此环境中可以添加更多的磁盘。如果让每个磁盘拥有一个或多个 I/O 服务器，则多个 I/O 操作可同时执行。也可以向此环境添加更多的硬盘空间。

单处理器系统受处理器可处理的磁盘空间容量的限制。然而，无论可以添加的其他部件（如内存或磁盘）如何，随着工作负荷的增加，单个 CPU 可能无法更快地处理用户请求。若已达到最大容量或最大可扩充性，则可考虑移到一个有多处理器的单分区系统中。

具有多个处理器的单分区

此环境通常由同一台机器中几个能力相等的处理器组成（参见图26），称为对称多处理器（SMP）系统。象磁盘空间和内存这类资源是共享的。

利用可用的多处理器，可以更快地完成不同的数据库操作。DB2 还可把单个查询的工作分布在可用的处理器中，以提高处理速度。其他数据库操作，如装入数据、备份和复原表空间以及在现存数据上创建索引，都可以利用多个处理器。

第65页的表3列出了此硬件配置最适合的并行性类型。

SMP 机器

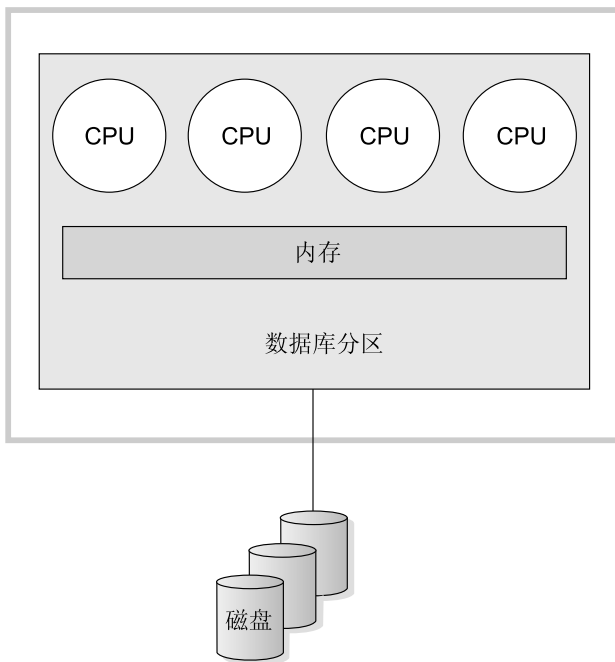


图 26. 单分区数据库对称多处理器系统

容量和可扩展性

在此环境中可以添加更多的处理器。然而，由于不同的处理器可能会试图存取同一数据，所以随着您公司业务量的增加，此环境也可能遇到限制。利用共享内存和共享磁盘，可有效地共享所有数据库数据。

可以通过增加磁盘数来增加与处理器相关的数据库分区的 I/O 容量。可以建立 I/O 服务器以专门处理 I/O 请求。如果让每个磁盘拥有一个或多个 I/O 服务器，则多个 I/O 操作可同时执行。

若已达到最大容量或最大可扩充性，可考虑移到有多个分区的系统中。

多分区配置

可以将一个数据库分布在多个分区中，每个分区在它自己的机器上。可将有多个数据库分区的多台机器编组在一起。本节描述下列分区配置：

- 具有一个处理器的系统上的分区
- 具有多个处理器的系统上的分区
- 逻辑数据库分区

具有一个处理器的分区

在此环境中，有许多数据库分区。每个分区都驻留在它自己的机器上，而且它有自己的处理器、内存和磁盘（第62页的图27）。所有机器通过通信设施连接在一起。可使用许多不同的名称来称呼此环境，包括群集、单处理器群集、大规模并行处理（MPP）环境和不共享配置。最后一个名称准确地反映了此环境中的资源安排。与 SMP 环境不同，MPP 环境没有共享的内存或磁盘。MPP 环境消除了由共享内存和磁盘带来的限制。

一个分区数据库环境允许一个数据库保持逻辑上的整体性，但它实际上分布于多个分区中。数据是分区的这一事实对大多数用户是透明的。可以在数据库管理程序之间划分工作；每个分区中的每个数据库管理程序都只为该数据库中它自己的那部分工作。

第65页的表3列出了此硬件配置最适合的并行性类型。

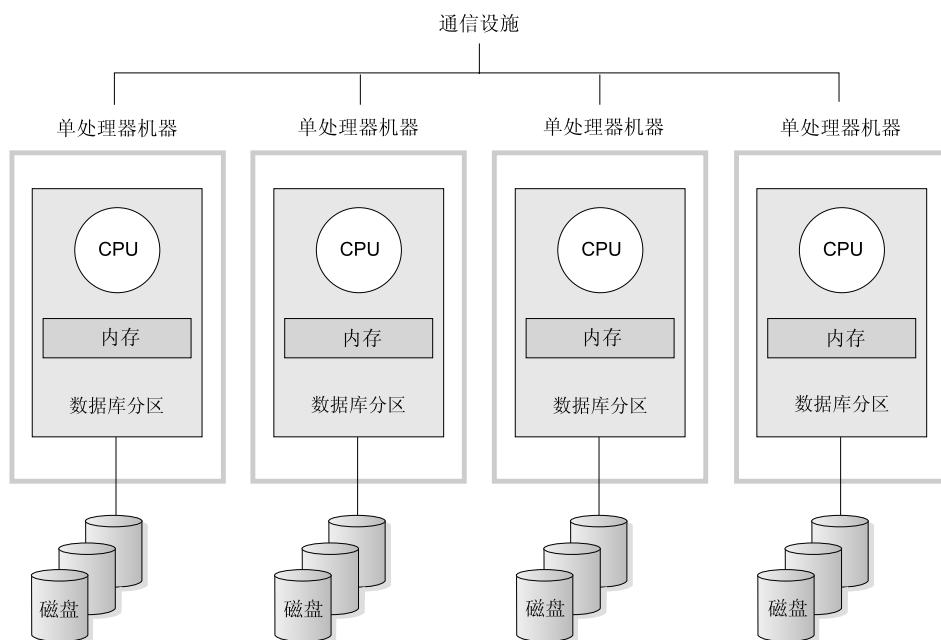


图 27. 大规模并行处理系统

容量和可扩充性： 在此环境中可以向您的配置添加更多的数据库分区（节点）。在某些平台上，例如 RS/6000 SP，最大的节点数为 512。然而，在管理很多机器和实例时，可能存在实际的限制。

若已达到最大容量或最大可扩充性，可考虑移到每个分区都有多个处理器的系统中。

具有多个处理器的分区

每个分区具有单处理器的配置的替代配置是，一个分区具有多个处理器的配置。这称为 *SMP* 群集（第63页的图28）。

此配置结合了 *SMP* 和 *MPP* 并行性的优点。这表示一个查询可以在跨多个处理器的单个分区中执行。亦即一个查询可以用并行方式在多个分区中执行。

第65页的表3列出了此硬件配置最适合的并行性类型。

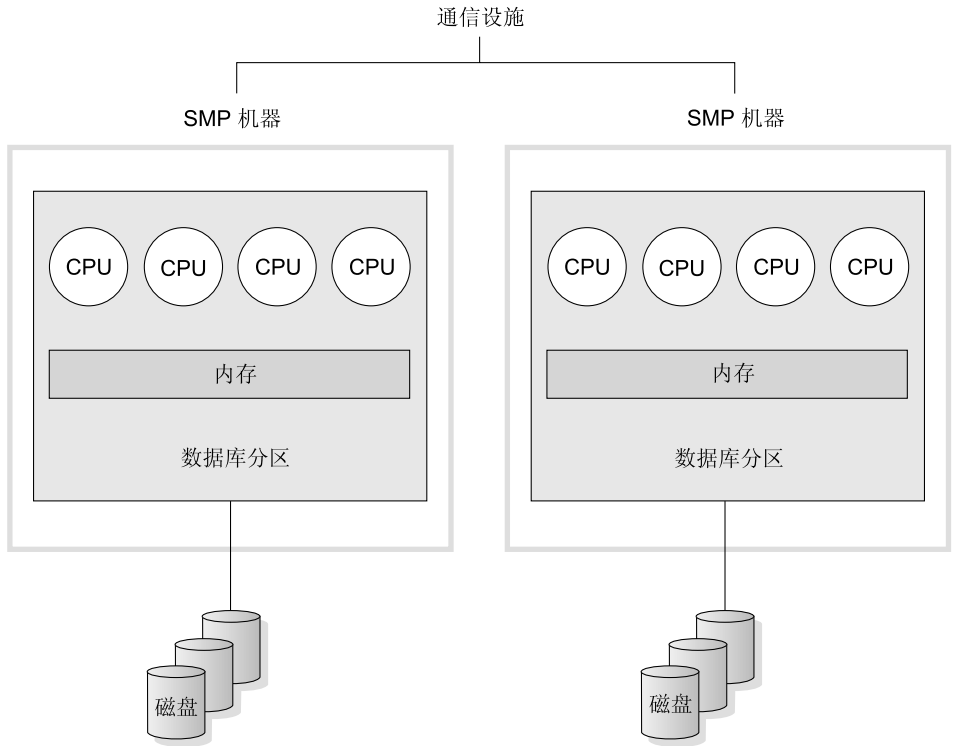


图 28. SMP 群集

容量和可扩充性: 在此环境中，可以添加更多的数据库分区，并可以向现存数据库分区添加更多的处理器。

逻辑数据库分区

逻辑数据库分区与物理分区的不同之处在于逻辑分区未被授予对整台机器的控制权。虽然机器已共享资源，但是数据库分区不共享资源。处理器是共享的，但磁盘和内存却不共享。

逻辑数据库分区提供了可扩充性。在多个逻辑分区上运行的多个数据库管理程序可以比一个单数据库管理程序更能充分利用可用的资源。第64页的图29举例说明通过添加更多的分区可以在一台 SMP 机器上获得更大的可扩充性；特别是对那些具有许多处理器的机器而言更是如此。通过对数据库分区，可以分别对每个分区进行管理和恢复。

大的 SMP 机器

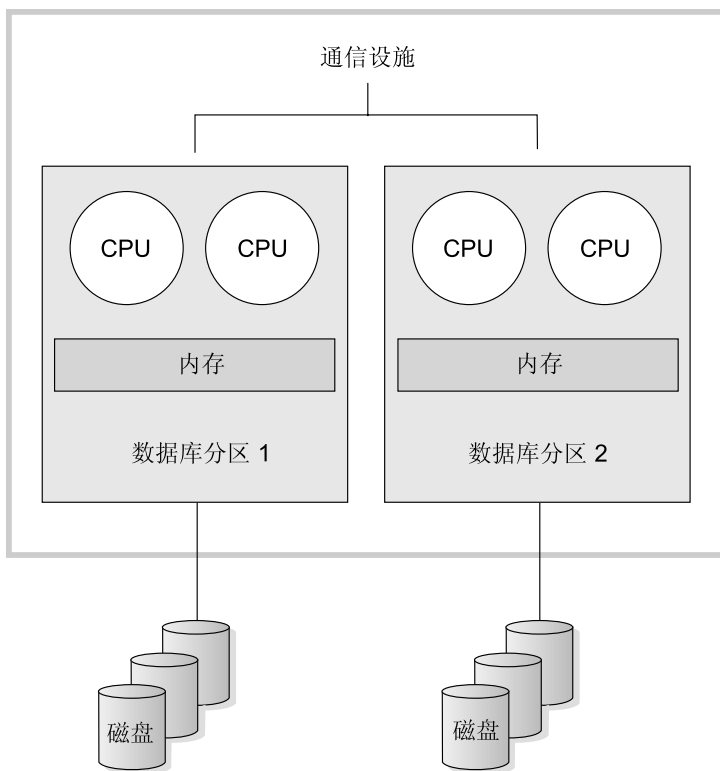


图 29. 分区数据库对称多处理器系统

第65页的图30举例说明可以扩大在图29中显示的配置以增强处理能力的事实。

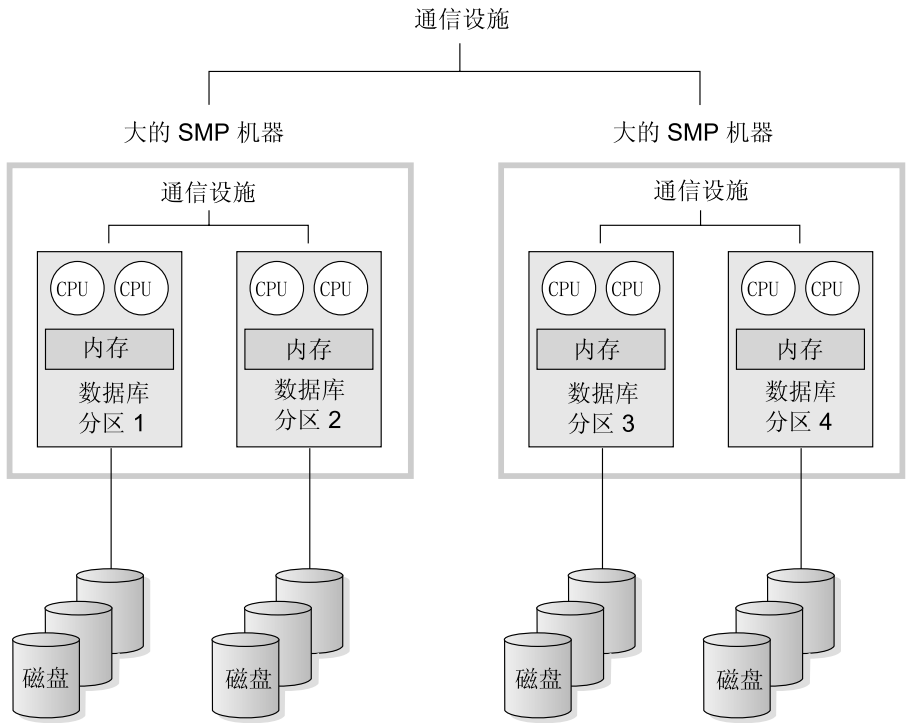


图 30. 分区数据库对称多处理器系统群集在一起

表3列出了此硬件环境最适合的并行性类型。

注：两个以上的分区同时存在于同一台机器上的能力（不考虑处理器的数量），使得可以更灵活地设计高可用性配置和故障恢复策略。机器发生故障之后，可以将一个数据库分区自动移至已包含同一数据库的另一个分区的另一台机器上，然后重新启动该分区。有关详情，参见第181页的『第11章 针对高可用性的设计』。

最适合每种硬件环境的并行性的摘要

下表总结了最适合于利用各种硬件环境的并行性的类型。

表 3. 每种硬件环境中的可能并行性类型

硬件环境	I/O 并行性	查询内并行性	
		分区内并行性	分区间并行性
单分区，单处理器	是	否 (1)	否
单分区，多处理器 (SMP)	是	是	否

表 3. 每种硬件环境中的可能并行性类型 (续)

硬件环境	I/O 并行性	查询内并行性	
		分区内并行性	分区间并行性
多分区, 一个处理器 (MPP)	是	否 (1)	是
多分区, 多处理器 (SMP 群集)	是	是	是
逻辑数据库分区	是	是	是

注: (1) 甚至是在单处理器系统中, 使用一个配置参数将并行度设置为大于 1 的值也可能获得好处, 特别是在执行的查询并未充分利用 CPU 时 (例如, 若它们受 I/O 约束)。

第5章 关于数据入库

“DB2 通用数据库”提供了“数据仓库中心”，这是一个使数据仓库处理自动化的部件。可使用“数据仓库中心”来定义要包括在仓库中的数据。然后，可使用“数据仓库中心”来自动调度仓库中数据的刷新。

本节提供数据入库和数据入库任务的概述。有关入库的详情，以及有关使用“数据仓库中心”的详情，参考*数据仓库中心管理指南*和“数据仓库中心”联机帮助。

什么是数据入库？

包含运行数据 - 运作业务的日常事务的数据 - 的系统包含对业务分析员非常有用的信息。例如，分析员可使用有关特定产品在一年中的特定时间在特定地区的销售信息来了解异常现象或预测将来的销售量。

然而，直接存取运行数据的分析员会遇到几个问题：

- 他们可能没有查询运行数据库的技术。例如，查询 IMS 数据库要求应用程序使用特殊类型的数据操作语言。通常，具有查询运行数据库技术的那些程序员的专职工作是维护数据库及其应用程序。
- 对于许多运行数据库（如银行的数据库），性能非常关键。系统不能处理进行特殊查询的用户。
- 通常，运行数据未处于适合业务分析员使用的最佳格式。例如，与原始数据相比，按产品、地区和季节归纳的销售数据对分析员更有用。

数据入库解决了这些问题。在数据入库中，将存储资料式数据 - 从运行数据中抽取的数据，经过变换之后供最终用户进行决策时使用。例如，数据入库工具能从运行数据库复制所有销售数据，执行计算来对数据进行摘要，然后将摘要数据从运行数据写至单独数据库中的一个目标。最终用户可以查询这个单独的数据库（仓库），而不会影响运行数据库。

下面各节描述您创建和维护数据仓库时将会使用到的对象（主题区、仓库源、仓库目标、代理程序、代理点、步骤和过程）。

主题区

主题区标识与商业的逻辑领域相关的过程，并对其进行分组。例如，若是在构建市场营销和销售数据的仓库，则可以定义一个“销售”主题区和一个“市场营

销”主题区。然后，可以在“销售”主题区下面添加与销售相关的过程。同样，可以在“市场营销”主题区下面添加与市场营销数据相关的定义。

仓库源

仓库源标识将要向仓库提供数据的表和文件。“数据仓库中心”使用仓库源中的说明来存取和选择数据。这些源几乎可以是任何与仓库相连接的关系或非关系源（表、视图或文件）。

仓库目标

仓库目标是包含特定数据的数据库表或文件，这些数据已经过变换，最终用户可以使用。与仓库源相同，仓库目标也可以向“数据仓库中心”步骤提供数据。

仓库代理程序和代理点

“数据仓库中心”代理程序管理数据源与目标仓库之间的数据流。在 Windows NT、AIX、OS/2、OS/390、OS/400 和 SUN Solaris 操作系统上有代理程序可用。代理程序使用“开放式数据库链接”（ODBC）驱动程序或 DB2 CLI 来与不同数据库通信。

有好几个代理程序可以处理源与目标仓库之间的数据传送。您使用的代理程序的数目取决于现存连通性配置以及您计划通过仓库移动的数据量。若同时运行多个需要同一代理程序的过程，则可以生成一个代理程序的附加实例。

代理程序可以是本地的或远程的。本地仓库代理程序是与仓库服务器安装在同一台机器上的代理程序。远程仓库代理程序是安装在与仓库服务器相连的另一台机器上的代理程序。

代理点是安装有代理程序软件的工作站的逻辑名。代理点名称与 TCP/IP 主机名不同。单一物理机器只能有一个 TCP/IP 主机名。然而，可以在单一机器上定义多个代理点。每个代理点都由一个逻辑名标识。

缺省代理点（名为“缺省 VW 代理点”）是“数据仓库中心”在初始化仓库控制数据库期间在 Windows NT 上定义的一个本地代理程序。

步骤和过程

步骤是“数据仓库中心”中的逻辑实体，它定义：

- 输出表或文件的结构。
- 用于填充输出表或文件的机制（SQL 或程序）。

- 用来填充输出表或文件的调度。

步骤通过使用 SQL 语句或通过调用程序来移动数据和变换数据。运行步骤时，将在仓库源与仓库目标之间传送数据，并对该数据执行任何变换。

过程包含执行变换和移动任务的一系列步骤。通常，一个过程通过从一个或多个仓库源（这可以是数据库表或文件）抽取数据来填充仓库数据库中的仓库目标。然而，也可定义用来启动程序，而未指定任何仓库源或目标的过程。

可以按需要运行步骤，也可以调度步骤，以在设置好的时间运行。可以将步骤调度为只运行一次，也可以将其调度为重复运行，如在每个星期五运行。还可以将步骤调度为按顺序运行，以便当一个步骤完成运行时，下一个步骤开始运行。可以将步骤调度为在另一个步骤（成功或不成功）完成后运行。若调度过程，则该过程中的第一个步骤在调度的时间运行。

当一个步骤或过程运行时，它可以使用下列方法保存数据：

- 用新数据替换仓库目标中的所有数据。
- 将新数据追加至现存数据。
- 追加数据的单独版本。

假设您想要让“数据仓库中心”执行下列任务：

1. 从不同的数据库抽取数据。
2. 将数据转换为单一格式。
3. 将数据写至数据仓库中的一个表。

您应创建一个包含个别步骤的过程。每个步骤执行单独的任务，如从数据库抽取数据，或将其转换为正确的格式。然后，应使用另一步骤来填充目标表，该表包含变换后的数据。

下列各节描述您可以在“数据仓库中心”中找到的各种步骤类型。有关步骤的详情，参考*数据仓库中心管理指南*。

SQL 步骤

SQL 步骤使用 SQL SELECT 语句来从仓库源抽取数据，并生成 INSERT 语句来将该数据插入到仓库目标表中。

程序步骤

有许多种类型的程序步骤：“DB2 AS/400 版程序”、“DB2 OS/390 版程序”、“DB2 UDB 版程序”、“Visual Warehouse 5.2 DB2 程序”、“OLAP 服务器程序”、“文件程序”和“复制”。这些步骤运行预定义的程序和实用程序。

变换器步骤

变换器步骤是特定的存储过程和用户定义函数，它们指定可以用来变换数据的统计或仓库变换器。可使用变换器来清除、反转和旋转数据；生成主关键字和周期表；以及计算各种统计信息。

在变换器步骤中，应指定其中一个统计或仓库变换器。运行过程时，该变换器步骤将数据写至一个或多个仓库目标。

用户定义程序步骤

用户定义程序步骤是“数据仓库中心”中的一个逻辑实体，它表示您想要让“数据仓库中心”启动的应用程序。仓库代理程序可以启动用户定义程序步骤：

- 在填充仓库目标期间启动。
- 在填充仓库目标之后启动。
- 自行启动。

例如，您可以编写一个执行下列过程的用户定义程序：

1. 从一个表调出数据。
2. 操作该数据。
3. 将该数据写至过渡输出资源或仓库目标。

入库任务

创建数据仓库包括下列任务：

- 定义一个主题区，它标识仓库中将要使用的过程并对它们进行分组。
- 检测源数据（或运行数据），并定义仓库源。
- 创建要用作仓库的数据库，并定义仓库目标。
- 通过定义一个过程，指定如何移动源数据以及将其变换成适合仓库数据库使用的格式。
- 测试您定义的步骤，并调度它们，以自动运行。
- 通过定义安全性，并监控数据库的使用情况来管理仓库。

- 如果有“DB2 仓库管理程序”程序包，则创建仓库中数据的信息目录。信息目录是一个包含商业元数据的数据库。元数据帮助用户标识和找出机构中可供他们使用的数据和信息。仓库的最终用户可搜索目录，以确定要查询哪些表。
- 为仓库中的数据定义星形模式模型。星形模式是一种具有专门用途的设计，它由多个维度表（描述商业的各个方面）和一个事实表（包含关于商业的事实）组成。例如，如果您生产软包装饮料，则一些维度表是产品、市场和时间。事实表包含有关每个地区每个季节订购的产品的事务信息。
- 连接事实表和维表，以将维表中的详细信息与订单信息组合在一起。例如，可以连接产品维与事实表，以将有关如何包装每一种产品的信息添加至订单。

通过使用 *Business Intelligence Tutorial*、查看 *DB2 Universal Database Quick Tour* 或阅读 *数据仓库中心管理指南*，您可以了解更多关于这些任务以及其他任务的详情。

第6章 关于 Spatial Extender

本节通过说明 Spatial Extender 的用途以及讨论它所处理的数据对它作了介绍。有关使用 Spatial Extender 的详情，参考 *Spatial Extender 用户指南和参考*。

Spatial Extender 的用途

使用 Spatial Extender 来创建地理信息系统 (GIS): 对象、数据和应用程序的一个复合体，它允许您生成和分析关于地理面貌的空间信息。地理面貌包括构成地球表面的物体和占据地球表面的物体。它们组成自然环境（例如，河流、森林、山峰和沙漠）和文化环境（城市、住宅区、办公大楼、路标，等等）。

空间信息包括下列各项:

- 地理面貌相对于其环境的位置（例如，一座城市中医院和门诊部所在的点，或城市的居住区与当地地震带的距离）
- 地理面貌相互相关的方式（例如，有关特定河流系统包含在特定地区内的信息，或该地区中的特定桥梁横跨该河流系统的支流的信息）
- 适用于一个或多个地理面貌的度量单位（例如，办公大楼与其红线之间的距离，或鸟类保护区周界的长度）

空间信息，无论是其本身还是与传统数据库输出的组合，都可以帮助您设计项目和进行商业和政策决策。例如，假设一个福利区域经理需要核实他所服务的地区中实际生活着多少福利申请人和受领人。Spatial Extender 可根据服务的地区的位置以及申请人和受领人的地址生成此信息。

或者，假设连锁饭店老板想在邻近的城市做生意。为了确定在哪里开新饭店，这位老板需要诸如此类的问题的答案：在这些城市中，将会经常光顾我的饭店的那些类型的人主要集中在什么位置？主高速公路的位置在哪里？哪里的犯罪率最低？我的竞争者的饭店的位置在哪里？Spatial Extender 可生成直观显示的空间信息来回答这样的问题，下层数据库管理系统可生成标签和文本来解释这些显示。

表示地理面貌的数据

本节提供为了获得空间信息而生成、存储和操作的数据的概述。包括下列主题:

- 数据如何表示地理面貌
- 空间数据的性质
- 生成空间数据的方法

数据如何表示地理面貌

在 Spatial Extender 中，地理面貌可以由一个表或视图中的一行表示，也可以由这样的行的一部分表示。例如，考虑下面这两个地理面貌：办公大楼和住宅区。在图 31 中，BRANCHES 表的每一行都表示一间银行的一个分部。不同的是，CUSTOMERS 表的每一行共同表示银行的一个客户；但每一行的一部分 - 明确的说，包含客户地址的单元 - 可被视为表示的是客户的住所。

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	CA	95141	A	A

图 31. 表示地理面貌的表行；其地址数据表示地理面貌的表行。BRANCHES 表中的数据行表示银行的分部。CUSTOMERS 表中地址数据的单元表示客户的住所。

这些表包含标识和描述银行分部和客户的数据。这样的数据被称为属性数据。

可以将属性数据的子集（表示分部和客户地址的值）转换成用来生成空间信息的值。例如，如图所示，一个分部的地址是 92467 Airzone Blvd., San Jose CA 95141。一个客户的地址是 9 Concourt Circle, San Jose CA 95141。Spatial Extender 可以将这些地址转换成特定的值，这些值指示该分部以及客户的住址相对于其各自环境的位置。图 32 显示了具有包含这样的值的新列的 BRANCHES 和 CUSTOMERS 表。

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	CA	95141		A	A

图 32. 添加了空间列的表。在每个表中，LOCATION 列都将包含与地址相对应的坐标。

当将地址和类似的标识符用作空间信息的起始点时，它们被称为源数据。因为起源于它们的值生成空间信息，所以这些生成的值被称为空间数据。下一节描述空间数据并介绍其相关联的数据类型。

空间数据的性质

许多空间数据由坐标组成。坐标是一个数字，它指示相对于参考点的位置。例如，纬度是指示相对于赤道的位置的坐标。经度是指示相对于格林威治子午线的位置的坐标。因此，黄石国家公园的位置由其纬度（北纬 44.45 度）及其经度（西半球 110.40 度）定义。

纬度、经度、它们的参考点以及其他相关参数共同称为坐标系。坐标系基于除纬度和经度外同时存在的值。这些坐标系有它们自己的位置量度标准、参考点和其他相异参数。

最简单的空间数据项由两个坐标组成，这两个坐标定义单个地理面貌的位置。术语数据项是指在关系表中占用一个单元的一个或多个值。更大的空间数据项由数个定义线性路径（如道路或河流）的坐标组成。第三种类型的空间数据项由定义区域周界（如小岛或水淹地带的边界）的坐标组成。

每个空间数据项都是空间数据类型的一个实例。标记一个位置的两个坐标的数据类型是 `ST_Point`；定义线性路径的坐标的数据类型是 `ST_LineString`；定义周界的坐标的数据类型是 `ST_Polygon`。这些类型与用于空间数据的其他数据类型一起，是属于单一层次结构的结构化类型。

空间数据从哪里来

可以通过三种方法获取空间数据：

- 根据属性数据生成
- 根据其他空间数据生成
- 调入

使用属性数据作为源数据

根据属性数据（如地址）生成空间数据称为地理编码。第74页的图32显示了用于空间数据的两个列，一个在 `BRANCHES` 表中，另一个在 `CUSTOMERS` 表中。想象 `Spatial Extender` 对这些表中的地址进行地理编码，并将生成的输出（与地址相对应的坐标）放到这两个列中。第76页的图33举例说明了此结果。

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	1653 3094

CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourc Circle	San Jose	CA	95141	953 1527	A	A

图 33. 包括从源数据生成的空间数据的表。CUSTOMERS 表中的 LOCATION 列包含地理编码器根据 ADDRESS、CITY、STATE 和 ZIP 列中的地址生成的坐标。通常，BRANCHES 表中的 LOCATION 列包含地理编码器根据此表的 ADDRESS、CITY、STATE 和 ZIP 列中的地址生成的坐标。

Spatial Extender 使用一个名为 *geocoder* 的函数来对属性数据进行地理编码，并将生成的空间数据放入列中。

使用其他空间数据作为源数据

空间数据不仅可以从属性数据生成，还可以从其他空间数据生成。例如，假定在 BRANCHES 表中定义了其分部的银行想知道每个分部附近 5 英里内有多少客户。在银行可以从数据库中获取此信息之前，它必须向数据库提供一个地带的定义，该地带位于每个分部半径 5 英里之内。Spatial Extender 函数 *ST_Buffer* 可以创建这样的定义。通过使用每个分部的坐标作为输入，此函数可以生成一系列坐标，这些坐标划分出了期望的地带的周界。图34显示了带有 *ST_Buffer* 提供的信息的 BRANCHES 表。

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	SALES_AREA
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	1653 3094	1002 2001, 1192 3564, 2502 3415, 1915 3394, 1002 2001

图 34. 包括从现存空间数据生成的新空间数据的表。SALES_AREA 列中的坐标是由 *ST_Buffer* 函数根据 LOCATION 列中的坐标生成的。

除 *ST_Buffer* 外，Spatial Extender 还提供了几种其他功能，它们可根据现存空间数据生成新的空间数据。

调入空间数据

获取空间数据的第三种方法是从 Spatial Extender 支持的其中一种格式的文件中调用它。这些文件包含通常适用于地图的数据：人口普查追踪、水淹地带、地震断

| 层，等等。通过将这样的数据与您生成的空间数据组合使用，可以增加您可用的
| 地理信息。例如，若公共建设工程部需要确定一个住宅社区的危险程度，可使用
| `ST_Buffer` 来在社区周围定义一个地带，然后调入有关水淹地带和地震断层的数据
| 来了解这些问题区与该地带的重叠情况。

第3部分 数据库设计

第7章 逻辑数据库设计

本节描述数据库设计的步骤:

- 『决定数据库中要记录的数据』
- 第83页的『为每种类型的关系定义表』
- 第85页的『为所有表提供列定义』
- 第87页的『标识一列或多列作为主关键字』
- 第90页的『确保相等的值表示相同的实体』
- 第90页的『考虑使表规范化』
- 第95页的『制定约束实施计划』
- 第101页的『数据库设计的其他考虑事项』。

设计数据库的目标是为您的环境建立一种表示, 它易于理解且可以扩充。此外, 您还希望数据库设计有助于维护数据的一致性和完整性。要达到此目的, 可以设计一个减少冗余度并消除在更新数据库期间可能发生的异常的数据库。

上述步骤是逻辑数据库设计的一部分。数据库设计不是一个线性过程; 进行该设计时, 可能还会重新执行那些步骤。

数据库设计的物理实现在第103页的『第8章 物理数据库设计』和 *管理指南: 实现中的“实现设计”* 中有描述。

决定数据库中要记录的数据

数据库设计的第一步是标识要存储在数据库表中的数据的类型。一个数据库包括一个机构或企业中的实体及其他它们之间的关系的信息。在关系数据库中, 将实体表示为表。

一个实体是您想要存储其信息的一个人、一个对象或一个概念。样本表中描述的某些实体是雇员、部门和项目。(有关样本数据库的说明, 参考 *SQL Reference*。)

在 *Employee* 样本表中, 实体“雇员”具有属性或特性, 如雇员号、姓名、工作部门和薪水额。那些特性显示为 *EMPNO*、*FIRSTNME*、*LASTNAME*、*WORKDEPT* 和 *SALARY* 列。

实体“雇员”的具体值由一个雇员的所有列中的值组成。每个雇员有一个唯一的雇员号 (*EMPNO*), 该号码可用于标识实体“雇员”的具体值。一个表中的每一行

表示一个实体或关系的一个具体值。例如，在下表中，第一行中的值描述名为 Haas 的雇员。

表 4. 雇员实体及其属性的具体值

EMPNO	FIRSTNME	LASTNAME	WORKDEPT	JOB
000010	Christine	Haas	A00	President
000020	Michael	Thompson	B01	Manager
000120	Sean	O'Connell	A00	Clerk
000130	Dolores	Quintana	C01	Analyst
000030	Sally	Kwan	C01	Manager
000140	Heather	Nicholls	C01	Analyst
000170	Masatoshi	Yoshimura	D11	Designer

随着支持非传统数据库应用程序（如多媒体）的需要日益增长。可能要考虑一些属性以便支持多媒体对象，如文档、视频或混合媒体、图象和语音。

在一个表中，某一行的每一列都以某种方式与该行的所有其他列相关。样本表中表达的一些关系如下：

- 雇员被分配至部门
 - Dolores Quintana 被分配至部门 C01
- 雇员从事一个工作
 - Dolores 从事分析员的工作
- 部门要向其他部门报告
 - 部门 C01 向部门 A00 报告
 - 部门 B01 向部门 A00 报告
- 雇员从事某个项目的工作
 - Dolores 和 Heather 都在从事项目 IF1000 的工作
- 雇员管理部门
 - Sally 管理部门 C01。

“雇员”和“部门”是实体；Sally Kwan 是“雇员”的一个具体值的一部分，而 C01 是“部门”的一个具体值的一部分。相同的关系应用于一个表的每一行中的相同列。例如，一个表的某行表示的关系是 Sally Kwan 管理部门 C01；而另一行表示的关系是 Sean O'Connell 是部门 A00 中的职员。

一个表中包含的信息取决于要表示的关系、需要的灵活度和期望的数据检索速度。

除了标识企业中的实体关系外，还需标识其他类型的信息，如应用于该数据的商业规则。

为每种类型的关系定义表

可以在数据库中定义几种类型的关系。考虑雇员和部门之间的可能关系。一个雇员只能在一个部门工作；对于雇员，此关系是相异值的。另一方面，一个部门可有许多雇员；对于部门，此关系是多值的。雇员（相异值的）和部门（多值的）之间的关系是一对多的关系。本节讨论下列类型的关系：

- 『一对多和多对一关系』
- 第84页的『多对多关系』
- 第85页的『一对一关系』

一对多和多对一关系

要为每个一对多和每个多对一关系定义表：

1. 将关系的“多”方是相同实体的所有关系分组。
2. 为该组中的所有关系定义单个表。

在以下示例中，第一个和第二个关系的“多”方是“雇员”，因此我们定义一个雇员表 EMPLOYEE。

表 5. 多对一关系

实体	关系	实体
雇员	被分配到	部门
雇员	从事	工作
部门	报告给	(管理)部门

在第三个关系中，“部门”是“多”方，因此我们定义一个部门表 DEPARTMENT。

下面显示如何在表中表示这些关系：

EMPLOYEE 表:

EMPNO	WORKDEPT	JOB
000010	A00	President
000020	B01	Manager
000120	A00	Clerk
000130	C01	Analyst
000030	C01	Manager
000140	C01	Analyst
000170	D11	Designer

DEPARTMENT 表:

DEPTNO	ADMRDEPT
C01	A00
D01	A00
D11	D01

多对多关系

两个方向都是多值的关系是多对多关系。一个雇员可以开发多个项目，而一个项目可以有多个雇员。问题“Dolores Quintana 在做什么？”和“谁在开发项目 IF1000？”都得到多个答案。在一个表中，对每个实体（“雇员”和“项目”）使用一列，可以表示多对多关系，如下面示例所示。

下面显示如何在一个表中表示多对多关系（一个雇员可以参与多个项目，而一个项目可以有多个雇员在参与）：

雇员活动 (EMP_ACT) 表:

EMPNO	PROJNO
000030	IF1000
000030	IF2000
000130	IF1000
000140	IF2000
000250	AD3112

一对一关系

一对一关系在两个方向都是相异的。一个经理管理一个部门；一个部门只有一个经理。问题“谁是部门 C01 的经理？”和“Sally Kwan 管理哪个部门？”都有单个答案。可将该关系指定给 DEPARTMENT 表或 EMPLOYEE 表。因为所有部门都有经理，但不是所有雇员都是经理，因此将经理添加至 DEPARTMENT 表是最合乎逻辑的，如以下示例中所示。

下面显示如何在一个表中表示一对一关系：

DEPARTMENT 表：

DEPTNO	MGRNO
A00	000010
B01	000020
D11	000060

为所有表提供列定义

要在一个关系表中定义一列：

1. 为该列选择一个名称。

一个表的每一列都必须具有对于该表唯一的名称。第315页的『附录B. 命名规则』中详细地描述了如何选择列名。

2. 说明对该列有效的数据类型。

数据类型和长度指定对该列有效的数据类型和最大长度。可以从数据库管理程序提供的那些类型中选择数据类型；也可以选择创建自己的用户定义类型。有关 DB2 提供的数据类型和用户定义类型的详情，参考 *SQL Reference*。

数据类型的示例有：数字、字符串、双字节（或图形）字符串、日期时间和二进制字符串。

大对象 (LOB) 数据类型支持多媒体对象，如文档、视频、图象和语音。这些对象是使用下列数据类型实现的：

- 二进制大对象 (BLOB) 字符串。BLOB 的示例是雇员的相片、语音和视频。
- 字符大对象 (CLOB) 字符串，它的字符序列可以是单字节字符或多字节字符，或这两者的组合。CLOB 的一个示例是雇员的简历。
- 双字节字符串大对象 (DBCLOB) 字符串，它的字符序列是双字节字符。DBCLOB 的一个示例是日语简历。

要更好地了解大对象支持，参考 *SQL Reference*。

用户定义类型 (UDT)，是从现存类型派生的一种类型。您可能需要定义从现存类型派生并与现存类型共享特性，但仍被视为单独的和 incompatible 的类型。

结构化类型是一种用户定义类型，其结构在数据库中定义。它包含一系列属性，每个属性都有一个数据类型。可将结构化类型定义为称为超类型的另一种结构化类型的子类型。子类型继承它的超类型的所有属性，并可定义附加属性。与一个公共超类型相关的结构化类型集合称为类型分级结构，没有任何超类型的超类型称为该类型分级结构的根类型。

可以将结构化类型用作表或视图的类型。结构化类型的属性的名称和数据类型，与对象标识符一起，成为这个类型表或类型视图的列的名称和数据类型可以将类型表或类型视图的行当作结构化类型的实例。

不能将结构化类型用作表或视图的列的数据类型。也不支持将整个结构化类型的实例作为应用程序中的一个主变量来检索。

引用类型是结构化类型的辅助类型。类似于单值类型，引用类型是一个标量类型，它与一种内部数据类型共享一个公共的表示法。在类型分级结构中的所有类型都共享这同一个表示法。引用类型的表示法是在创建类型分级结构的根类型时定义的。当使用引用类型时，将结构化类型指定为该类型的参数。此参数称为该引用的目标类型。

引用的目标始终是类型表或视图中的行。当使用引用类型时，可能需要定义作用域。作用域标识一个表（称为目标表）或视图（称为目标视图），它包含引用值的目标行。目标表或视图的类型必须与引用类型的目标类型相同。定义了作用域的引用类型的一个实例会唯一地标识类型表或类型视图中的一行，该行称为它的目标行。

由于许多原因，可能会使用用户定义函数 (UDF)，包括调用那些允许在用户定义类型之间比较或转换的例程。UDF 扩展并增强了内部 SQL 函数提供的支持，并可在可使用内部函数的任何地方使用。有两种类型的 UDF:

- 外部函数，它是用一种程序设计语言编写的
- 有源函数，它将用于调用其他 UDF

例如，两个数字数据类型是“欧洲鞋码”和“美国鞋码”。这两种类型表示的都是鞋码，但是它们不兼容，因为度量基数不同且不能比较。可调用用户定义函数将一种鞋码转换为另一种。

要更好地了解用户定义类型、结构化类型、引用类型和用户定义函数，参考 *SQL Reference*。

3. 说明哪些列可能需要缺省值。

某些列不可能在所有行的值都是有意义的，因为:

- 列值不适用于该行。

例如，包含雇员中间姓名首字母的一列不适用于没有中间名首字母的雇员。

- 值适用，但尚未知道该值。

例如，MGRNO 列可能包含无效的经理号，因为该部门的上一个经理已调离，新经理尚未任命。

在这两种情况下，可以选择允许 NULL 值（指示列值是未知的或不适用的特殊值），或允许数据库管理程序或应用程序指定的非 NULL 缺省值。

在 *SQL Reference* 中详细描述了 NULL 值和缺省值。

标识一列或多列作为主关键字

关键字是一组可用来标识或存取特定行的列。在表、索引或参考约束的说明中标识关键字。同一个列可以是多个关键字的一部分。

唯一关键字是约束为其任何两个值都不相等的关键字。唯一关键字的列不能包含 NULL 值。例如，可将雇员号码列定义为唯一关键字，因为该列中的每个值只标识唯一一个雇员。不能有相同雇员号码的两个雇员。

用来强制关键字唯一性的机制称为唯一索引。一个表的唯一索引是一列或若干列的有序集合，而每个值标识（在功能上确定）这些列的唯一行。唯一索引可以包含 NULL 值。

主关键字是在一个表上定义的唯一关键字中的一个，而且该关键字被选为最重要的关键字。一个表上只能有一个主关键字。

会自动为主关键字创建主索引。数据库管理程序使用主索引来有效地存取表行，且主索引允许数据库管理程序实现主关键字的唯一性。（也可以在非主关键字列上定义索引，以便在处理查询时高效率地存取数据。）

若一个表没有“自然”的唯一关键字，或者到达顺序是用于区分唯一行的方法，则使用时间戳记作为关键字的一部分可能有帮助。（另见第89页的『定义标识列』。）

某些样本表的主关键字是：

表	关键字列
Employee 表	EMPNO
Department 表	DEPTNO
Project 表	PROJNO

以下示例显示 PROJECT 表的一部分，包括其主关键字。

表 6. PROJECT 表上的主关键字

PROJNO (主关键字)	PROJNAME	DEPTNO
MA2100	Weld Line Automation	D01
MA2110	Weld Line Programming	D11

若一个表中的每一列都包含重复的值，则不能只用一列来定义主关键字。具有多列的关键字是组合关键字。列值的组合应定义一个唯一的实体。若定义组合关键字不太容易，可以考虑创建具有唯一值的新列。

以下示例显示包含多列的一个主关键字（组合关键字）：

表 7. EMP_ACT 表上的组合主关键字

EMPNO (主关键字)	PROJNO (主关键字)	ACTNO (主关键字)	EMPTIME	EMSTDATE (主关键字)
000250	AD3112	60	1.0	1982-01-01
000250	AD3112	60	.5	1982-02-01
000250	AD3112	70	.5	1982-02-01

标识入选关键字列

要标识入选关键字，选择定义唯一实体的最少数目的列。可能有多个人选关键字。在第21页的表2中，有多个人选关键字。EMPNO、PHONENO 和 LASTNAME 这三列都唯一地标识该雇员。

从一组入选关键字中选择一个主关键字的标准应是持久性、唯一性和稳定性。

- 持久性表示总是存在每一行的主关键字值。
- 唯一性表示每一行的关键字值与所有其他行不同。
- 稳定性表示主关键字值始终不变。

在该示例的三个人选关键字中，只有 EMPNO 满足全部这些标准。一个雇员在进入一家公司时，可能没有电话号码。姓可能改变，尽管它们在某个时候可能是唯一的，但不能保证始终如此。雇员号码列是主关键字的最佳选择。一个雇员只被赋予一次一个唯一的号码，而且只要该雇员在该公司内任职，通常不会更新该号码。因为每个雇员必须有一个号码，因此雇员号码列中的值是持久的。

定义标识列

标识列向 DB2 提供了一种为一个表中的每一行自动生成唯一数值的方法。一个表只能有一个定义了标识属性的列。标识列的示例包括订单号码、雇员号码、证券号码和事故号码。

标识列的值可以“始终生成”或“在缺省情况下生成”。

- DB2 保证定义为始终生成的标识列是唯一的。它的值总是由 DB2 生成；不允许应用程序提供显式的值。
- 定义为在缺省情况下生成的标识列向应用程序提供了显式地为标识列提供一个值的方法。若未给出任何值，则 DB2 生成一个，但在此情况下，不能保证该值的唯一性。DB2 只对它生成的那些值保证唯一行。“在缺省情况下生成”用于数据填充（复制现存表的内容），或用于一个表的卸载和重新装入。

标识列特别适合于生成唯一主关键字值这一任务。应用程序可使用标识列来避免当一个应用程序在数据库外部生成它自己的唯一计数器时可能会导致的并行性和性能问题。例如，一种常见的应用程序级实现是维护一个只有一行的表，它包含一个计数器。每个事务都锁定此表，增大该数字，然后落实；即，每次只有一个事务可以增大计数器。相反，若通过标识列维护该计数器，因为事务不锁定该计数器，所以可以获得更高级别的并行性。一个未落实的已增大该计数器的事务不会阻止后续事务也增大该计数器。

标识列的计数器的增大（或减小）独立于事务。若给定的事务两次增大一个标识计数器，则该事务可能会在生成的两个数字之间看到一个间隙，因为当前可能有其他事务正在增大同一标识计数器（即，将行插入同一个表中）。若一个应用程序必须要有连续范围的数字，则该应用程序应对带有标识列的表进行互斥锁定。因为会造成丢失并行性，所以必须对此决定作权衡。此外，有可能因为生成标识列的值的事务已回滚，或因为存放值系列的数据库在指定所有存放的值之前被停用，从而导致给定的标识列出现为数字之间生成有间隙。

标识列生成的序号具有下列附加特性：

- 值可以是任何小数位为零的精确数字数据类型；即，小数位为零的 SMALLINT、INTEGER、BIGINT 或 DECIMAL。（单精度和双精度浮点类型被认为是适当的数字数据类型。）
- 连续值之间可以有任何指定的整数增量。缺省增量是 1。
- 标识列的计数器值是可恢复的。若发生故障，则从日志重构计数器值，因此可以保证继续生成唯一的值。
- 可以将标识列值存入高速缓存，以获得更好的性能。

确保相等的值表示相同的实体

您可以有多个表，它们描述同一组实体的属性。例如，EMPLOYEE 表显示雇员被分配至的部门的编号，DEPARTMENT 表显示分配到每个部门的经理。要同时检索这两组属性，可用匹配的列将这两个表连接在一起，如以下示例中所示。WORKDEPT 和 DEPTNO 中的值表示相同的实体，并表示 DEPARTMENT 和 EMPLOYEE 表之间的连接路径。

DEPARTMENT 表:

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
D21	Administration Support	000070	D01

EMPLOYEE 表:

EMPNO	FIRSTNAME	LASTNAME	WORKDEPT	JOB
000250	Daniel	Smith	D21	Clerk

当从多个表检索一个实体的信息时，确保相等的值表示相同的实体。连接列可有不同的名称（象上一个示例中的 WORKDEPT 和 DEPTNO），也可以有相同的名称（象 Department 表和 Project 表中的 DEPTNO 列）。

考虑使表规范化

规范化帮助消除表数据中的冗余和不一致。它是将表精简为一组列的过程，在这组列中，所有非关键字列都依赖于主关键字列。若不是这样，则在更新期间该数据可能变得不一致。

本节简要回顾第一种、第二种、第三种和第四种规范形式的规则。表的第五种规范形式在有关数据库设计的许多书籍中都有说明，在此就不再赘述。

形式 说明

第一种 表中的每一行和每一列均有一个值，永远不会是一组值（参见第91页的『第一种规范形式』）。

第二种 每个不是关键字一部分的列都依赖于关键字（参见第91页的『第二种规范形式』）。

第三种 每个非关键字列都独立于其他非关键字列，并依赖于关键字（参见第93页的『第三种规范形式』）。

第四种 没有行包含有关一个实体的两个或更多个独立的多值事实（参见第94页的『第四种规范形式』）。

第一种规范形式

若每个单元中都只有一个值，永远不会有一组值，则该表使用的是第一种规范形式。使用第一种规范形式的表不必满足更高的规范形式的标准。

例如，下表违反了第一种规范形式，因为对于 PART 的每个具体值，WAREHOUSE 列都包含了几个值。

表 8. 违反第一种形式的表

PART (主关键字)	WAREHOUSE
P0010	Warehouse A, Warehouse B, Warehouse C
P0020	Warehouse B, Warehouse D

以下示例显示使用第一种规范形式的同一个表。

表 9. 符合第一种规范形式的表

PART (主关键字)	WAREHOUSE (主关键字)	QUANTITY
P0010	Warehouse A	400
P0010	Warehouse B	543
P0010	Warehouse C	329
P0020	Warehouse B	200
P0020	Warehouse D	278

第二种规范形式

若每个不是关键字一部分的列都依赖于整个关键字，则该表使用的是第二种规范形式。

当一个非关键字列依赖于组合关键字的一部分时，违反了第二种规范形式，如下示例所示：

表 10. 违反第二种规范形式的表

PART (主关键字)	WAREHOUSE (主关键字)	QUANTITY	WAREHOUSE_ADDRESS
P0010	Warehouse A	400	1608 New Field Road
P0010	Warehouse B	543	4141 Greenway Drive
P0010	Warehouse C	329	171 Pine Lane

表 10. 违反第二种规范形式的表 (续)

PART (主关键字)	WAREHOUSE (主关键字)	QUANTITY	WAREHOUSE_ADDRESS
P0020	Warehouse B	200	4141 Greenway Drive
P0020	Warehouse D	278	800 Massey Street

主关键字是一个组合关键字，它由 PART 和 WAREHOUSE 列组成。因为 WAREHOUSE_ADDRESS 列只依赖于 WAREHOUSE 的值，因此该表违反了第二种规范形式的规则。

此设计存在下列问题：

- 对于该仓库中存储的一个部件，在每个记录中重复了该仓库地址。
- 若一个仓库的地址变更，则必须更新引用存储在该仓库中的一个部件的每一行。
- 由于存在这种冗余，该数据可能变得不一致，表现为不同的记录对相同的仓库显示不同的地址。
- 若某个时候一个仓库中没有存储部件，就可能没有哪一行记录了该仓库地址。

解决方案是将该表分割成下面两个表：

表 11. 符合第二种规范形式的 PART_STOCK 表

PART (主关键字)	WAREHOUSE (主关键字)	QUANTITY
P0010	Warehouse A	400
P0010	Warehouse B	543
P0010	Warehouse C	329
P0020	Warehouse B	200
P0020	Warehouse D	278

表 12. 符合第二种规范形式的 WAREHOUSE 表

WAREHOUSE (主关键字)	WAREHOUSE_ADDRESS
Warehouse A	1608 New Field Road
Warehouse B	4141 Greenway Drive
Warehouse C	171 Pine Lane
Warehouse D	800 Massey Street

若有两个表使用第二种规范形式，就需要考虑性能。根据部件位置生成报告的应用程序必须将这两个表连接起来以检索相关信息。

要更好地了解性能考虑事项，参考管理指南：性能中的“调整应用程序性能”。

第三种规范形式

若每个非关键字列都独立于其他非关键字列，且只依赖于关键字，则该表使用的是第三种规范形式。

以下示例中的第一个表包含 EMPNO 列和 WORKDEPT 列。假定添加了 DEPTNAME 列（参见表14）。这个新列依赖于 WORKDEPT，但主关键字是 EMPNO。该表现在违反了第三种规范形式。更改雇员 John Parker 的 DEPTNAME 不会更改该部门中其他雇员的部门名。注意，现在部门号 E11 中有两个不同的部门名。产生的不一致显示在该表的更新版本中。

表 13. 更新前的不规范 EMPLOYEE_DEPARTMENT 表

EMPNO (主关键字)	FIRSTNAME	LASTNAME	WORKDEPT	DEPTNAME
000290	John	Parker	E11	Operations
000320	Ramlal	Mehta	E21	Software Support
000310	Maude	Setright	E11	Operations

表 14. 更新后的不规范 EMPLOYEE_DEPARTMENT 表. 表中的信息变得不一致。

EMPNO (主关键字)	FIRSTNAME	LASTNAME	WORKDEPT	DEPTNAME
000290	John	Parker	E11	Installation Mgmt
000320	Ramlal	Mehta	E21	Software Support
000310	Maude	Setright	E11	Operations

可创建一个含有 WORKDEPT 列和 DEPTNAME 列的新表，将该表规范化。诸如更改部门名之类的更新现在更加简单；只需要更新这个新表。

返回部门名和雇员名的 SQL 查询编写起来更复杂，因为它需要连接两个表。它运行的时间可能比对单个表查询的时间长。因为 WORKDEPT 列必须出现在两个表中，所以需要附加的存储空间。

下面的表被定义为规范化的结果。

表 15. 将 EMPLOYEE_DEPARTMENT 表规范化之后的 EMPLOYEE 表

EMPNO (主关键字)	FIRSTNAME	LASTNAME	WORKDEPT
000290	John	Parker	E11

表 15. 将 *EMPLOYEE_DEPARTMENT* 表规范化之后的 *EMPLOYEE* 表 (续)

EMPNO (主关键字)	FIRSTNAME	LASTNAME	WORKDEPT
000320	Ramlal	Mehta	E21
000310	Maude	Setright	E11

表 16. 将 *EMPLOYEE_DEPARTMENT* 表规范化之后的 *DEPARTMENT* 表

DEPTNO (主关键字)	DEPTNAME
E11	Operations
E21	Software Support

第四种规范形式

若没有一行包含有关一个实体的两个或更多个独立的多值事实，则该表使用的是第四种规范形式。

考虑下列实体：雇员、技能和语言。一个雇员可有几种技能，并通晓几种语言。有两个关系，一个关系是雇员和技能之间的关系，另一个关系是雇员和语言之间的关系。若一个表表示了这两种关系，则它未使用第四种规范形式，如以下示例所示：

表 17. 违反第四种规范形式的表

EMPNO (主关键字)	SKILL (主关键字)	LANGUAGE (主关键字)
000130	Data Modelling	English
000130	Database Design	English
000130	Application Design	English
000130	Data Modelling	Spanish
000130	Database Design	Spanish
000130	Application Design	Spanish

应改用两个表表示这种关系：

表 18. 符合第四种规范形式的 *EMPLOYEE_SKILL* 表

EMPNO (主关键字)	SKILL (主关键字)
000130	Data Modelling
000130	Database Design
000130	Application Design

表 19. 符合第四种规范形式的 EMPLOYEE_LANGUAGE 表

EMPNO (主关键字)	LANGUAGE (主关键字)
000130	English
000130	Spanish

但是，若这些属性是独立的（即，雇员只将特定的语言应用于特定的技能），则不应将该表分割。

设计数据库的一个良好策略是将所有数据安排在使用第四种规范形式的表中，然后决定该结果是否提供了一个可接受的性能级别。如果没有，则可将数据重新安排在使用第三种规范形式的表中，然后重新评价性能。

制定约束实施计划

约束是数据库管理程序实施的规则。本节讨论四种类型的约束处理：

唯一约束	确保表中的关键字值是唯一的。检查对组成主关键字的列的任何更改，以保证唯一性。
参考完整性	在插入、更新和删除操作上实现参考约束。所有外部关键字的所有值都有效才是一个数据库的正确状态。
表检查约束	验证更改后的数据有无违反当创建或更改表时指定的条件。
触发器	定义要执行的一组操作，当对指定的表执行更新、删除或插入操作时要调用这组操作。

唯一约束

唯一约束是一个规则，它确保关键字值在表中是唯一的。在唯一约束中组成该关键字的每一列必须定义为 NOT NULL。在使用 PRIMARY KEY 子句或 UNIQUE 子句的 CREATE TABLE 或 ALTER TABLE 语句中定义唯一约束。

一个表可有任意多个唯一约束；但是，只能定义一个唯一约束作为一个表的主关键字。此外，一个表在相同的一组列上不能有多个唯一约束。

当定义一个唯一约束时，数据库管理程序就会创建一个唯一索引（如果需要的话），并将它指定为系统必需的主索引或唯一索引。此约束通过唯一索引来实现。一旦在某列上建立了唯一约束，则在多行更新期间对唯一性的检查将延迟到更新结束后进行。

唯一约束也可用作参考约束中的父关键字。

参考完整性

数据库管理程序通过参考约束维护参考完整性，参考完整性要求给定属性或表列的所有值也存在于某些其他表或列中。例如，参考约束可能要求 EMPLOYEE 表中的每个雇员属于 DEPARTMENT 表中的一个部门。不能有雇员属于不存在的部门。

可在一个数据库中构建参考约束，以确保维持参考完整性，并允许优化程序通过了解这些特殊关系使查询的执行更有效率。当制定参考完整性计划时，标识数据库表之间的所有关系。可定义一个主关键字和参考约束来标识一个关系。

考虑相关的下列表：

表 20. DEPARTMENT 表

DEPTNO (主关键字)	DEPTNAME	MGRNO
A00	Spiffy Computer Service Div.	000010
B01	Planning	000020
C01	Information Center	000030
D11	Manufacturing Systems	000060

表 21. EMPLOYEE 表

EMPNO (主关键字)	FIRSTNAME	LASTNAME	WORKDEPT (外部关键字)	PHONENO
000010	Christine	Haas	A00	3978
000030	Sally	Kwan	C01	4738
000060	Irving	Stern	D11	6423
000120	Sean	O'Connell	A00	2167
000140	Heather	Nicholls	C01	1793
000170	Masatoshi	Yoshimura	D11	2890

下面许多概念的讨论都涉及到这些表（这些概念对于您理解参考完整性很有帮助）。

唯一关键字是值在所有行中都不重复的一个列或一组列。可将一个唯一关键字定义为表的主关键字。当唯一关键字被一个外部关键字引用时，也称为父关键字。

主关键字是属于表定义一部分的唯一关键字。每个表只能有一个主关键字。在前面的表中，DEPTNO 和 EMPNO 分别是 DEPARTMENT 表和 EMPLOYEE 表的主关键字。

外部关键字是一个表中的某一列或一组列，它们引用同一个表或另一个表的唯一关键字或主关键字。外部关键字用于建立与唯一关键字或主关键字的关系，以便在所有表中实现参考完整性。EMPLOYEE 表中的 WORKDEPT 列是外部关键字，因为它引用 DEPARTMENT 表中的 DEPTNO 主关键字。

组合关键字是有多列的关键字。主关键字和外部关键字均可以是组合关键字。例如，若部门是由分部号和部门号的组合来唯一地标识，则需要两个列来创建 DEPARTMENT 表的关键字。

父关键字是一个参考约束的主关键字或唯一关键字。主关键字约束是未指定一组父关键字列时参考约束的缺省父关键字。

父表是包含一个父关键字的表，该父关键字与同一个表或另一个表中的至少一个外部关键字相关。一个表可以是任意多个关系中的父代。例如，具有主关键字 DEPTNO 的 DEPARTMENT 表是包含外部关键字 WORKDEPT 的 EMPLOYEE 表的父代。

父行是父表中的一行，该表的父关键字值匹配一个从属表中的至少一个外部关键字值。一个父表中的行不必是父行。DEPARTMENT 表中的第四行 (D11) 是 EMPLOYEE 表中的第三行和第六行的父行。DEPARTMENT 表的第二行 (B01) 不是任何其他行的父行。

从属表是包含一个或多个外部关键字的一个表。从属表也可以是父表。一个表可以是任意多个关系中的一个从属表。EMPLOYEE 表包含外部关键字 WORKDEPT，该表从属于主关键字为 DEPTNO 的 DEPARTMENT 表。

从属行是一个从属表中的一行，该表有一个与父关键字值相匹配的非 NULL 外部关键字值。该外部关键字值表示从属行对父行的引用。因为外部关键字可以接受 NULL 值，因此一个从属表中的一行不一定是从属行。

若一个表是从属表，或者它是从属表的子代，则该表就是一个表的子代。子表包含可以追溯到某个表的父关键字的外部关键字。

参考循环是将一个表与它自身连接的一个路径。当一个表与它自己直接连接时，它是自参考表。若 EMPLOYEE 表有另一列 MGRID，它包含每个雇员的经理的 EMPNO，则 EMPLOYEE 表就是一个自参考表。MGRID 就是 EMPLOYEE 表的外部关键字。

自参考表在同一个关系中既是父表，又是从属表。自参考行既是它本身的父行，又是它本身的从属行。在这种情况下存在的约束称为自参考约束。例如，若在一个自参考表某行中的外部关键字值与该行中唯一关键字的值匹配，则该行是自参考行。

参考约束是指一个指定的外部关键字的非 NULL 值仅当作为指定表的唯一关键字值出现时才有效。参考约束的目的是保证数据库关系得以维持，并遵守数据输入规则。

SQL 操作的意义

对于取决于该表是父表还是从属表的某些 SQL 操作，参考约束的实现有特殊的意义。本节描述维护参考完整性对 INSERT、DELETE、UPDATE 和 DROP 等 SQL 操作的影响。

DB2 不会自动在整个系统中实施参考约束。因此，若您想要在整个系统中实施参考约束，应用程序必须包含必要的逻辑。

讨论下列主题：

- 『INSERT 规则』
- 『DELETE 规则』
- 第99页的『UPDATE 规则』。

INSERT 规则： 可以随时在父表中插入一行，而不必在从属表中执行任何操作。但是，不能在从属表中插入一行，除非在该父表中有一行的父关键字值等于要插入的行的外部关键字值，除非该外部关键字值是 NULL。若一个组合外部关键字值的所有部件都为 NULL，则该值也是 NULL。

当指定一个外部关键字时，此规则是隐式的。

若尝试将一行插入有参考约束的表中，且在父关键字中不存在任何非 NULL 外部关键字值，则不允许 INSERT 操作。若在试图插入多行期间某行的 INSERT 操作失败，则不插入任何行。

DELETE 规则： 当从一个父表中删除一行时，DB2 检查在从属表中是否有任何从属行具有匹配的外部关键字值。若找到这样的从属行，则可执行几项操作。当创建从属表时，通过指定删除规则，可确定将执行哪些操作。

当删除主关键字时从属表（包含该外部关键字的表）的删除规则是：

RESTRICT

若找到任何从属行，则阻止删除父表中的任何行。若需要除去父行和从属行，则首先删除从属行。首先删除父行会违反参考约束，这是不允许的。

NO ACTION

当应用了所有参考约束后，对于每个子行强制使其父行存在。

CASCADE

表示删除父表中的一行将自动删除从属表中的任何相关行。若从属表中的一行在父表中没有父行存在时它本身也没有意义，则可使用此规则。

首先删除父行将自动删除引用主关键字的从属行。无需首先删除从属行。若某些从属行有它们自己的从属行，则对那些关系应用删除规则。

DB2 管理级联删除。

SET NULL

确保删除父表中某行时将任何从属行中外部关键字的值设置为 NULL。该行的其他部分保持不变。

若当创建表时没有明确定义删除规则，则应用 NO ACTION 规则。

在一个删除操作中可能涉及的任何表称为删除连接的表。下列限制应用于删除连接的关系。

- 在多个表组成的一个参考循环中，一个表不能与它自己是删除连接关系。
- 当一个表通过多个从属关系与另一个表形成删除连接关系时，这些关系必须有相同的删除规则 - CASCADE 或 NO ACTION。
- 当一个自参考表是 CASCADE 关系中另一个表的从属表时，自参考关系的删除规则也必须是 CASCADE。

可以随时从一个从属表中删除行，而不用对父表执行任何操作。例如，在部门-雇员关系中，一个雇员可能退休，要从雇员表中删除表示他的那一行，但不影响部门表。（在雇员 - 部门的反向关系中，部门经理 ID 是引用雇员表的父关键字的外部关键字。若一个经理退休，则会影响部门表。）

UPDATE 规则： DB2 不允许更新父行的唯一关键字。当更新一个从属表中的外部关键字，而该外部关键字不为 NULL 时，它必须与该关系中的父表的父关键字的某个值相匹配。若 UPDATE 操作违反任何参考约束，则发生错误，且不更新任何行。

当更新父关键字某列中的值时：

- 若从属表中的任何行与该关键字的原始值匹配，当更新规则是 RESTRICT 时拒绝该更新。
- 若更新语句完成时（触发后的情况除外）从属表中的任何行没有对应的父关键字，当更新规则是 NO ACTION 时拒绝该更新。

要更新父行中父关键字的值，必须通过下列操作首先除去与从属表中任何子行的关系：

- 删除子行；或，
- 更新从属表中的外部关键字，以包括另一个有效的关键字值。

若与该行中的关键字值没有从属关系，该行不再是参考关系中的父行，且可更新。

若一个外部关键字的一部分要更新，而该外部关键字值的任何一部分都不为 NULL，则该外部关键字的新值必须显示为父表中的唯一关键字值。若没有外部关键字从属于给定的唯一关键字；即，包含该唯一关键字的行不是父行，则可更新该唯一关键字的某部分。但是，在这种情况下，不能选择多行来进行更新，因为使用的是唯一关键字，不允许有重复行。

表检查约束

在设计中给出的商业规则可通过表检查约束来实施。表检查约束指定应用于一个表的每一行的搜索条件。当对表应用更新或插入语句时，自动激活这些约束。它们是通过 CREATE TABLE 或 ALTER TABLE 语句定义的。

可使用表检查约束来验证。例如，一个部门号的值必须在范围 10 至 100 之间；一个雇员的职务只能是 "Sales"、"Manager" 或 "Clerk"；或者，在该公司工作超过八年的雇员的收入必须超过 \$40,500。

有关表检查约束对 IMPORT 和 LOAD 命令的影响的详情，参考 *Data Movement Utilities Guide and Reference*。

触发器

触发器是每当对指定的表执行删除、插入或更新操作时执行的一组定义的操作。可以定义触发器来帮助支持商业规则。还可使用触发器来自动更新摘要或审查数据。因为触发器存储在数据库中，所以不必在每个应用程序中编写操作码。触发器只编码一次并存储在数据库中，当应用程序使用该数据库时，会在需要时由 DB2 自动调用它。这确保始终实施与该数据相关的商业规则。若商业规则更改，则只需修改触发器。

可在一个已触发的 SQL 语句内调用用户定义函数 (UDF)。这样，当该触发器被触发时，被触发的操作执行非 SQL 操作。例如，可将电子邮件作为警报机制发送。有关触发器的详情，参见 *管理指南：实现中的“创建触发器”* 并参考 *Application Development Guide*。

数据库设计的其他考虑事项

当设计一个数据库时，重要的是考虑用户应该能够存取哪些表。通过授权，授予或撤消对表的存取权。最高权限级别是系统管理权限 (SYSADM)。具有 SYSADM 权限的用户可指定其他授权，包括数据库管理员权限 (DBADM)。

在您的设计中，您可能要考虑其他问题，如审查活动、历史数据、摘要表、安全性、确定数据的类型和并行处理能力。

为了进行审查，可能必须在指定的时间段内记录每次对数据所做的更新。例如，您可能希望在每次雇员工资改变时更新和审查表。若定义了适当的触发器，则可自动更新此表。也可通过 DB2 审查设施执行审查活动。有关详情，参见管理指南：实现中的“审查 DB2 活动”。

考虑到性能方面的原因，可能只想存取选择的数据量，同时将基本数据作为历史维护。应在设计中加入对维护此历史数据的要求，如数据可以保留几个月或几年才可以清除。

您可能还想使用摘要信息。例如，您可能有一个表，它具有所有雇员的信息。但是，您希望将此信息按分部或部门划分为单独的表。在这种情况下，以原表中的数据为基础的每个分部或部门的摘要表将对您有帮助。有关摘要表的详情，参见管理指南：实现中的“创建摘要表”。

在设计中还应标识隐含的安全性考虑事项。例如，您可能决定通过安全性表支持用户存取特定类型的数据。可定义各种数据的存取级别以及谁可存取此数据。机密数据，如雇员和工资单数据，将具有严格的安全性限制。有关安全性和权限的详情，参见管理指南：实现中的“控制数据库存取”。

可以创建具有相关的结构化类型的表。利用这种类型表，可以使用表之间定义的关系建立一个分级结构，称为类型分级结构。该类型分级结构由单个的根类型、超类型和子类型组成。

引用类型表示法是在创建类型分级结构的根类型时定义的。引用的目标始终是类型表或视图中的行。

有关实现包括类型行和表的设计的详情，参见管理指南：实现中的“实现设计”。有关在处于分级结构的类型表之间移动数据的详情，参考 *Data Movement Utilities Guide and Reference*。

随着业务扩展，可能需要提高“DB2 扩充企业版”提供的容量和性能。在此环境中，数据库分区分布在几台机器或几个系统中，分别负责整个数据库的部分存储和检索。每个分区（或节点）并行工作以处理 SQL 或实用程序操作。

本书包括与并行操作相关的问题和考虑事项。

第8章 物理数据库设计

在完成逻辑数据库设计（参见第81页的『第7章 逻辑数据库设计』）之后，应当考虑关于数据库和表将驻留其中的物理环境的诸多因素。这包括了解为支持和管理数据库而要创建的文件、了解存储数据需要多少空间，以及确定如何使用存储数据所需的表空间。

包括下列主题：

- 『数据库目录』
- 第105页的『估计表的空间需求』
- 第111页的『附加空间需求』
- 第113页的『设计节点组』
- 第120页的『设计和选择表空间』
- 第137页的『联合体数据库设计考虑事项』

数据库目录

当创建一个数据库时，DB2 会创建一个单独的子目录来存储控制文件（如日志头文件）并将容器分配给缺省表空间。与该数据库相关的对象并非始终存储在该数据库目录中；它们可以存储在不同的位置（包括设备）中。

该数据库是在 DB2INSTANCE 环境变量定义的实例中创建的，或是在已显式连接（使用 ATTACH 命令）的实例中创建的。有关实例的介绍，参见管理指南：实现中的“使用数据库管理程序的多个实例”。

在基于 UNIX 的系统上使用的命名模式是：

```
specified_path/$DB2INSTANCE/NODEnnnn/SQL00001
```

在 OS/2 和 Windows 操作系统上使用的命名模式：

```
D:\$DB2INSTANCE\NODEnnnn\SQL00001
```

其中

- specified_path 是用来安装该实例的可选的、用户指定的位置。
- NODEnnnn 是分区数据库环境中的节点标识符。第一个节点是 NODE0000。
- "D:" 是标识根目录所在的卷的“盘符”。

SQL00001 包含与创建的第一个数据库相关的对象，并给予后续数据库更大的数字：SQL00002，依此类推。

子目录在与数据库管理程序实例同名的目录中创建，该实例是在创建数据库时连接的。（在 OS/2 和 Windows 操作系统上，子目录在由“盘符”标识的卷的根目录下创建。）这些实例和数据库子目录在 CREATE DATABASE 命令指定的路径中创建，且数据库管理程序自动维护它们。每个实例可能由一个实例所有者拥有（这取决于您使用的平台），该所有者对属于该实例的数据库具有系统管理员 (SYSADM) 权限。

要避免潜在的问题，不要创建使用相同命名模式的目录，不要操纵已经由数据库管理程序创建的目录。

数据库文件

与数据库相关的文件包括：

文件名	说明
SQLDBCON	此文件存储数据库的调整参数和标志。有关更改数据库配置参数的详情，参考管理指南：性能。
SQLLOGCTL.LFH	此文件有助于跟踪和控制所有数据库日志文件。
Syyyyyyy.LOG	<p>数据库日志文件，编号从 0000000 至 9999999。这些文件的编号由 <i>logprimary</i> 和 <i>logsecond</i> 数据库配置参数控制。个别文件的大小由 <i>logfilesiz</i> 数据库配置参数控制。</p> <p>通过使用循环记录，可重新使用这些文件，且保持相同的编号。通过使用归档记录，当将日志归档并分配新日志时，文件编号将依次增加。当达到 9999999 时，该编号绕回。</p> <p>缺省情况下，这些日志文件存储在 SQLLOGDIR 目录中。SQLLOGDIR 在 SQLnnnnn 子目录中。</p>
SQLINSLK	此文件帮助确保数据库只由数据库管理程序的一个实例使用。
SQLTMLPK	此文件帮助确保数据库只由数据库管理程序的一个实例使用。
SQLSPCS.1	此文件包含该数据库中所有表空间的定义和当前状态。
SQLSPCS.2	此文件是 SQLSPCS.1 的副本。若没有这两个文件中的任何一个，则不能存取您的数据库。
SQLBP.1	此文件包含数据库中所使用的所有缓冲池的定义。

SQLBP.2 此文件是 SQLBP.1 的副本。若没有这两个文件中的任何一个，则不能存取您的数据库。

DB2RHIST.ASC

此文件是数据库历史文件。它保存对数据库执行的管理操作的历史记录，如备份和复原操作。

DB2RHIST.BAK

此文件是 DB2RHIST.ASC 的副本。

注:

1. 不要直接更改这些文件。只能使用记载的 API 和实现那些 API 的工具（包括命令行处理器和 控制中心）来间接存取它们。
2. 不要移动这些文件。
3. 不要除去这些文件。
4. 备份数据库或表空间唯一受支持的方法是通过 **sqlubkp**（备份数据库）API 来进行，包括该 API 的命令行处理器和 控制中心实现。

估计表的空间需求

数据库对象的大小估计不可能做到很精确。磁盘碎片、空闲空间以及使用变长列所造成的额外开销都使大小估计变得十分困难，因为可能的列类型和列长度的范围实在是太广了。在最初估计数据库大小后，创建一个测试数据库，并用有代表性的数据填充它。

在“控制中心”中，可存取许多专门辅助您确定各种数据库对象的大小需求的实用程序：

- 可选择对象，然后使用“估计大小”实用程序。此实用程序可告诉您现存对象（例如，表）的当前大小。在更改该对象之后，该实用程序可以对该对象计算出新的估计值。该实用程序可以帮助您估计存储器需求（将未来的增长考虑在内）。它不仅仅是只能估计对象的大小。它还提供对象可能的大小范围：最小大小（基于当前值）和可能的最大大小。
- 可通过使用“显示相关的”对话来确定对象之间的关系。
- 可选择实例中的任何一个数据库对象，然后请求“生成 DLL”。此功能使用 **db2look** 实用程序来生成数据库的数据定义语句。有关此实用程序的详情，参考 *Command Reference*。

在这两种情况下，可使用“显示 SQL”或“显示命令”按钮。也可将生成的 SQL 语句或命令作为脚本文件保存，以便稍后使用。所有这些实用程序都有联机帮助可向您提供帮助。

在制订物理数据库需求计划时，应考虑这些实用程序。

估计数据库的大小时，必须考虑下列各项的影响：

- 『系统目录表』
- 『用户表数据』
- 第108页的『长整数字段数据』
- 第108页的『大对象 (LOB) 数据』
- 第109页的『索引空间』

不讨论与下列各项相关的空间需求：

- 本地数据库目录文件
- 系统数据库目录文件
- 操作系统所需的文件管理额外开销，包括：
 - 文件块大小
 - 目录控制空间

系统目录表

创建数据库时，会创建系统目录表。当将数据库对象和特权添加至该数据库时，这些系统表将增大。最初，这些系统表使用大约 3.5 MB 的磁盘空间。

为目录表分配的空间容量取决于表空间的类型和包含这些目录表的表空间的数据块大小。例如，若使用数据块大小为 32 的 DMS 表空间，则目录表空间最初将被分配 20MB 的空间。有关详情，参见第120页的『设计和选择表空间』。

注：对于含多个分区的数据库，目录表只驻留在发出 CREATE DATABASE 命令的分区上。目录表的磁盘空间仅对于该分区才是必需的。

用户表数据

在缺省情况下，表数据存储在 4KB 页上。每一页（不管页大小如何）都包含 76 个字节的数据库管理程序 额外开销。这留下了 4020 字节来存放用户数据（即行），虽然 4 KB 页上的行的长度不能超过 4005 字节。一行不会横跨多页。当使用 4KB 的页大小时，最多可有 500 列。

表数据页不包含用 LONG VARCHAR、LONG VARGRAPHIC、BLOB、CLOB 或 DBCLOB 数据类型定义的列的数据。但是，一个表数据页中的行的确包含这些列的描述符。（有关估计确实包含这些数据类型的表对象的空间所需的详情，参见第108页的『长整数字段数据』和第108页的『大对象 (LOB) 数据』。）

通常，以“首个合适”次序将行插入到表中。（使用空闲空间映象）搜索文件，查找大小足以存放该新行的第一个可用空间。当更新一行时，除非该页上所剩的空间不足以包含它，否则在原位置进行更新。若所剩空间不足以包含新行，则在原始行位置创建一个记录，以指向更新后的行在表文件中的新位置。

若调用 `ALTER TABLE APPEND ON` 语句，则将一直追加数据，且不保留关于数据页上任何空闲空间的信息。有关此语句的详情，参考 *SQL Reference*。

对于数据库中的每个用户表，可以通过如下计算公式来估计 4KB 页数：

```
ROUND DOWN(4020/(平均行大小 + 10)) = records_per_page
```

然后，将结果插入：

```
(number_of_records/records_per_page) * 1.1 = number_of_pages
```

其中，平均行大小是平均列大小的总和（有关每一列的大小的详情，参考 *SQL Reference* 中的 `CREATE TABLE` 语句。），而乘数“1.1”表示额外开销。

注：此公式只是提供一个估计值。若记录长度因碎片和溢出记录而改变，则估计的准确性将降低。

也可以选择创建具有 8 KB、16 KB 或 32 KB 页大小的缓冲池或表空间。在特定大小的表空间中创建的所有表都将具有匹配的页大小。假设使用 32 KB 的页大小，则单个表或索引对象的最大大小可达 512 GB。当使用 8 KB、16 KB 或 32 KB 的页大小时，最多可有 1012 列。对于 4 KB 的页大小，最大列数为 500。最大行宽也随页大小的不同而不同：

- 当页大小是 4 KB 时，行长度最大可为 4005 字节。
- 当页大小是 8 KB 时，行长度最大可为 8101 字节。
- 当页大小是 16 KB 时，行长度最大可为 16 293 字节。
- 当页大小是 32 KB 时，行长度最大可为 32 677 字节。

拥有更大的页大小有助于减小任何索引中的级别数。若使用执行随机行读写的 OLTP（联机事务处理）应用程序，则小一点的页大小会更好，这样，因不期望的行而浪费的缓冲区空间更少。若使用一次存取大量连续行的 DSS（决策支持系统）应用程序，则大一点的页大小会更好，这样可以减少读取特定行数所需的 I/O 请求数。当行大小小于页大小除以 255 的值时，会发生异常。在这种情况下，每个页上都有浪费的空间。（记住，每页最多只能有 255 行。）为减少浪费的空间，小一点的页大小可能更合适。

不能将备份复原为另一种页大小。

不能调入超过 755 列的 IXF 数据文件。有关将数据调入表和 IXF 数据文件的详情，参考 *Data Movement Utilities Guide and Reference*。

已说明临时表只能在它们自己的“用户临时”表空间类型的表空间中创建。没有缺省用户临时表空间。临时表不能含有 LONG 数据。当应用程序与数据库断开连接时，该表被隐式卸下，估计它们的空间需求时应将这一点考虑在内。

长整数字段数据

长整数字段数据存储在一个单独的表对象中，它的结构与其他数据类型不同（参见第106页的『用户表数据』和『大对象 (LOB) 数据』）。

数据存储于 32 KB 区域中，该区域被分成大小为 512 字节的“2 的幂”倍的段。（因此，这些段可以是 512 字节、1024 字节、2048 字节，以此类推，直至 32 700 字节。）

长整数字段数据类型（LONG VARCHAR 或 LONG VARGRAPHIC）以使空闲空间易于收回的方式存储。有关分配和空闲空间的信息存储在 4KB 分配页中，它在整个对象中不经常出现。

对象中未使用的空间量取决于长整数字段数据的大小以及此大小是否在该数据的所有出现之处都是不变的。对于大于 255 字节的数据项，这个未使用的空间最大可为该长整数字段数据大小的 50%。

若该字符数据的长度小于页大小，且它适合含有该数据其余部分的记录，则应该使用 CHAR、GRAPHIC、VARCHAR 或 VARGRAPHIC 数据类型，而不要使用 LONG VARCHAR 或 LONG VARGRAPHIC。

大对象 (LOB) 数据

大对象 (LOB) 数据存储于两个单独的表对象中，这两个对象的结构与其他数据类型不同（参见第106页的『用户表数据』和『长整数字段数据』）。

要估计 LOB 数据所需的存储空间，需要考虑用来存储使用这些数据类型定义的数据的两个表对象：

- **LOB 数据对象**

数据存储于 64 MB 区域中，该区域被分成大小为 1024 字节的“2 的幂”倍的段。（因此，这些段可以是 1024 字节、2048 字节、4096 字节，以此类推，直至 64MB。）

要减少 LOB 数据所用的磁盘空间量，可在 CREATE TABLE 和 ALTER TABLE 语句上的 LOB 选项子句上使用 COMPACT 参数。COMPACT 选项将所需的磁

盘空间量减至最小，方法是将 LOB 数据分成更小的段。此过程不涉及数据压缩，只是使用最接近 1 KB 极限的最小空间量。使用 COMPACT 选项可能导致在追加 LOB 值时性能下降。

包含在 LOB 数据对象中的空闲空间量将受到更新和删除活动量，以及要插入的 LOB 值的大小的影响。

• LOB 分配对象

有关分配和空闲空间的信息存储在与实际数据分离的 4 KB 分配页中。这些 4KB 页的数目取决于数据量，包括为大对象数据分配的未使用空间。额外开销的计算如下：每 64 GB 一个 4 KB 页加上每 8 MB 一个 4KB 页。

若该字符数据的长度小于页大小，且它适合含有该数据其余部分的记录，则应该使用 CHAR、GRAPHIC、VARCHAR 或 VARGRAPHIC 数据类型，而不要使用 BLOB、CLOB 或 DBCLOB。

索引空间

对于每个索引，可以按如下公式估计所需的空间：

$$(\text{平均索引关键字大小} + 8) * \text{行数} * 2$$

其中：

- “平均索引关键字大小”是索引关键字中每列的字节计数。有关如何计算不同类型的数据类型的列的字节计数的详情，参考 *SQL Reference* 中的 CREATE TABLE 语句。（估计 VARCHAR 和 VARGRAPHIC 列的平均列大小时，使用当前数据大小的平均值加一个字节。不要使用最大说明大小。）
- 乘数 “2” 表示额外开销，如非叶子页和空闲空间。

注：对于允许 NULL 的每个列，添加一个额外的字节以表示该空值指示符。

创建索引时，临时空间是必需的。在创建索引期间所需的最大临时空间可以按如下公式估计：

$$(\text{平均索引关键字大小} + 8) * \text{行数} * 3.2$$

其中，乘数 “3.2” 表示索引额外开销，以及索引创建期间进行排序所需的空间。

注：对于非唯一索引，存储重复的关键字项只需四个字节。上面显示的估计是假定没有重复项。因此以上公式可能会过多地估计存储索引所需的空间。

可使用下面两个公式来估计叶子页的数目（第二个公式提供更准确的估计）。这些估计的准确度很大程度上取决于平均值反映实际数据的准确程度。

注：对于 SMS，所需的最小空间为 12 KB。对于 DMS，最小值是一个数据块。

- 每个叶子页的平均关键字数的粗略估计是:

$$\frac{(.9 * (U - (M*2))) * (D + 1)}{K + 6 + (4 * D)}$$

其中:

- U (一页上的可用空间) 大约等于页大小减 100。对于 4096 的页大小, U 是 3996。
- M = U / (8 + 最小关键字大小)
- D = 每个关键字值的平均重复项数
- K = 平均关键字大小

记住, 最小关键字大小和平均关键字大小必须有一个额外字节, 表示每个可空关键字部分; 还必须有一个额外的字节, 表示每个变长关键字部分的长度。

若存在包括列, 则在计算最小关键字大小和平均关键字大小时应将它们考虑在内。

若在创建索引期间指定了非缺省值 10% 的一个空闲百分比, 则可以使用任何 (100 - pctfree)/100 替换 .9。

- 每个叶子页的平均关键字数的更准确估计是:

$$L = \text{叶子页数} = X / (\text{叶子页上的平均关键字数})$$

其中, X 是表中的总行数。

可按如下方法估算索引的原始大小:

$$(L + 2L/(\text{叶子页上的平均关键字数})) * \text{页大小}$$

对于 DMS 表空间, 将一个表上所有索引的大小加在一起, 然后四舍五入为该索引所在表空间的数据块大小的一个倍数。

应该为 INSERT/UPDATE 活动所引起的索引增长提供附加空间, 这种增长可能导致分页。

使用下列计算方法来获得更精确的原始索引大小的估算值, 以及该索引中级别数的估算值。(若索引定义中使用包括列, 可能要引起特别注意。) 每个非叶子页的平均关键字数大约是:

$$\frac{(.9 * (U - (M*2))) * (D + 1)}{K + 12 + (8 * D)}$$

其中:

- U (一页上的可用空间) 大约等于页大小减 100。对于 4096 的页大小, U 是 3996。
- D 是在非叶子页上每个关键字值的重复值的平均数目 (这将比在叶子页上的小很多, 您可能想将该值设置为 0 以便简化计算)。
- $M = U / (8 + \text{非叶子页的最小关键字大小})$
- $K = \text{非叶子页的平均关键字大小}$

只要没有包括列, 非叶子页与叶子页的最小关键字大小和平均关键字大小将是相同的。包括列不存储在非叶子页上。

除非 $(100 - \text{pctfree})/100$ 大于 .9, 否则不应使用它来替换 .9, 因为在创建索引期间会在非叶子页上留下 10% 的空闲空间。

可用如下所示的方法估算非叶子页数:

```

if L > 1 then {P++; Z++;}
While (Y > 1)
{
    P = P + Y
    Y = Y / N
    Z++
}

```

其中:

- P 是页数 (最初为 0)。
- L 是叶子页数。
- N 是每个非叶子页的关键字数。
- $Y = L / N$
- Z 是索引树中的级别数 (最初为 1)。

总页数是:

$$T = (L + P + 2) * 1.0002$$

附加的 0.02% 表示额外开销, 包括空间映象页。

创建索引所需的空间容量估算为:

$$T * \text{页大小}$$

附加空间需求

以下两项需要附加空间:

- 第112页的『日志文件空间』
- 第112页的『临时工作空间』

日志文件空间

日志文件所需的空间量（以字节数计），其范围可以为：

$$(\logprimary * (\logfilesiz + 2) * 4096) + 8192$$

至：

$$((\logprimary + \logsecond) * (\logfilesiz + 2) * 4096) + 8192$$

其中：

- *logprimary* 是在数据库配置文件中定义的主日志文件数
- *logsecond* 是在数据库配置文件中定义的辅助日志文件数
- *logfilesiz* 是在数据库配置文件中定义的每个日志文件中的页数
- 2 是每个日志文件所需的标题页的数目
- 4096 是一页中的字节数
- 8192 是日志控制文件的大小（以字节计）。

有关这些配置参数的详情，参考管理指南：性能。

注：总的活动日志空间不能超过 32 GB。

日志文件空间的上限取决于数据库管理程序在运行时所需要的辅助日志文件的实际数目。此上限可能永远不需要，或可能只在很偶然的大量活动期间才需要。

若允许对该数据库执行前滚恢复，应该考虑特殊的日志空间需求：

- 当启用 *logretain* 配置参数时，日志文件将被归档在日志路径目录中。除非将日志文件移至另一个位置，否则，联机磁盘空间最终将会填满。
- 当启用 *userexit* 配置参数时，用户出口程序会将归档的日志文件移至另一个位置。要允许下列情况，附加的日志空间仍是必需的：
 - 等待用户出口程序移动的联机归档日志
 - 要格式化以供将来使用的新日志文件。

临时工作空间

某些 SQL 语句需要临时表来进行处理（如使用一个工作文件来进行不能在内存中执行的排序）。这些临时表需要磁盘空间；所需的空间量取决于查询以及返回的表的大小，无法估计。

可以使用数据库系统监控程序和查询表空间 API 来跟踪在正常操作期间所用的工作空间量。

设计节点组

节点组是定义为属于某个数据库的一个或多个节点的命名集合。作为数据库系统配置一部分的每个数据库分区，必须已在称为 *db2nodes.cfg* 的分区配置文件中定义。一个节点组可以只包含一个数据库分区，也可以包含为该数据库系统定义的所有数据库分区。

使用 `CREATE NODEGROUP` 语句可创建新的节点组，可使用 `ALTER NODEGROUP` 语句修改它。可以在一个节点组中添加或卸下一个或多个数据库分区。在修改节点组之前，数据库分区必须已在 *db2nodes.cfg* 文件中定义。表空间驻留在节点组中。表驻留在表空间内。

当创建或修改一个节点组时，会有一个分区映象与它相关。分区映象，连同分区关键字和散列算法，被数据库管理员用于确定该节点组中哪个数据库分区将存储给定行的数据。有关分区映象的详情，参见第115页的『分区映射』。有关分区关键字的详情，参见第116页的『分区关键字』。

使用非分区数据库时，不需要任何分区关键字或分区映象。若使用非分区数据库，则不存在节点组设计考虑事项。数据库分区是数据库的一部分，与用户数据、索引、配置文件和事务日志组成。创建数据库时创建的缺省节点组由数据库管理员使用。`IBMCATGROUP` 是包含系统目录的表空间的缺省节点组。`IBMTEMPGROUP` 是系统临时表空间的缺省节点组。`IBMDEFAULTGROUP` 是包含用户定义表的表空间的缺省节点组，可选择将那些表置于该处。已说明临时表的临时表空间可在 `IBMDEFAULTGROUP` 或用户创建的任何节点组中创建，但不能在 `IBMTEMPGROUP` 中创建。

若使用的是多分区节点组，应考虑下列设计要点：

- 在多分区节点组中，若索引是该分区关键字的超集，则只能创建唯一的索引。
- 根据该数据库中的数据库分区数，可能有一个或多个单分区节点组，且存在一个或多个多分区节点组。
- 必须给每个数据库分区指定唯一的分区号。在一个或多个节点组中可能会发现相同的数据库分区。
- 要确保快速恢复包含系统目录表的数据库分区，则须避免将用户表放在同一个数据库分区上。这可通过将用户表放在不包括 `IBMCATGROUP` 节点组中的那些数据库分区的节点组中来实现。

除非要与一个更大的表并置，否则应该将小表放在单分区节点组中。并置是将不同表中包含相关数据的行放置在同一个数据库分区中。并置的表允许 DB2 利用更有效的连接策略。并置的表可以驻留在单个分区节点组中。若表驻留在多分区节点组中，且在分区关键字中具有相同数目的列，对应列的数据类型是分区兼容

的，则将这些表视为是并置的。在并置的表中具有相同分区关键字值的行被放在同一个数据库分区上。这些表可以位于同一个节点组中单独的表空间中，且仍被认为是并置的。

应该避免将中等大小的表扩展到太多数据库分区上。例如，100 MB 的表在 16 个分区的节点组中可能比在 32 个分区的节点组中执行得更好。

可以使用节点组将联机事务处理 (OLTP) 表与决策支持 (DSS) 表分离，以确保 OLTP 事务的性能不受负面影响。

节点组设计考虑事项

逻辑数据库设计和要处理的数据量将决定数据库是否需要分区。本节包括下列与数据库分区相关的主题：

- 『数据分区』
- 第115页的『分区映射』
- 第116页的『分区关键字』
- 第118页的『表并置』
- 第118页的『分区兼容性』
- 第119页的『复制的摘要表』

数据分区

DB2 支持分区存储模型，这允许您将数据存储于数据库的多个数据库分区中。这意味着该数据是以物理方式存储在多个数据库分区中，同时仍可以看作数据是位于同一个位置来进行存取。存取一个分区数据库中的数据的应用程序和用户不需要知道该数据的物理位置。

该数据在物理上是分离的，但在逻辑上将它作为一个整体来使用和管理。用户可以选择如何通过说明分区关键字来将数据分区。用户还可以通过选择应存储数据的表空间和相关的节点组，以确定表数据可以分布在哪些数据库分区以及多少个数据库分区上。另外，将可更新的分区映象与散列算法一起使用，来指定分区关键字值至数据库分区的映射，以确定每行数据的位置和检索。因此，可以将大表的工作负荷分布在整个分区数据库上，并将较小的表存储在一个或多个数据库分区上。每个数据库分区具有它存储的数据的本地索引，这可以提高本地数据存取的性能。

并不是必须将所有表都分布到数据库中的所有数据库分区上。DB2 支持部分撤消群集，这意味着可以将表及其表空间分布到系统（即，节点组）中的数据库分区的一个子集上。

当您想要将表放在每个数据库分区上时，可考虑的替代方法是使用摘要表，然后复制这些表。可创建包含所需信息的摘要表，然后将它复制至每个节点。有关详情，参见第119页的『复制的摘要表』。

分区映射

在分区数据库环境中，数据库管理员必须设法弄清一个表的哪些行存储哪些数据库分区上。数据库管理程序必须知道到哪里去查找所需的数据，并使用一个称为分区映射的映射来查找数据。

分区映射是一个内部生成的数组，对于多分区节点组，它包含 4 096 项，对于单分区节点组，只包含一项。对于单分区节点组，分区映射只有一个项，该项包含用于存储一个数据库表所有行的数据库分区的分区号。对于多分区节点组，以循环方式指定节点组的分区号。正如使用网格将城市地图划分为区一样，数据库管理员使用分区关键字来确定存储数据的位置（数据库分区）。

例如，假定您将一个数据库创建在四个数据库分区（编号为 0-3）上。此数据库的 IBMDEFAULTGROUP 节点组的分区映射将是：

```
0 1 2 3 0 1 2 ...
```

若已使用数据库分区 1 和 2 在该数据库中创建了一个节点组，则该节点组的分区映射将是：

```
1 2 1 2 1 2 1 ...
```

若要装入数据库的一个表的分区关键字是一个可能在范围 1 至 500 000 之间的整数，则会将分区关键字散列至 0 至 4 095 之间的一个分区号。将该编号用作分区映射中的索引，以选择用于该行的数据库分区。

第116页的图35显示如何将具有分区关键字值 (c1, c2, c3) 的行映射至分区 2，然后引用数据库分区 n5。

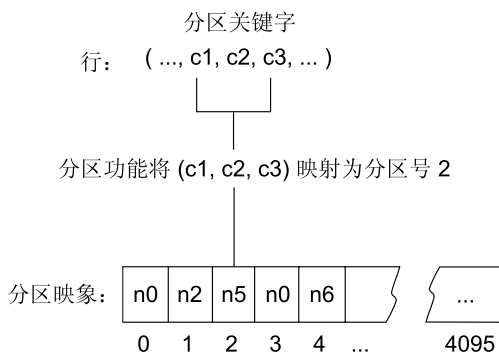


图 35. 使用分区映射的数据分布

分区映射用于灵活控制将数据存储在一个分区数据库中的位置。若将来需要更改数据库中各数据库分区上的数据分布，可以使用数据重新分布实用程序。此实用程序允许重新协调或调整数据分布。有关此实用程序的详情，参考管理指南：性能中的“跨数据库分区重新分布数据”。

可以使用“获取表分区信息” (**sqlugtpi**) API，来获取您可以查看的分区映象的副本。有关此 API 的详情，参考 *Administrative API Reference*。

分区关键字

分区关键字是一列（或一组列），用于确定将某行数据存储到什么分区。分区关键字是使用 **CREATE TABLE** 语句在表上定义的。若没有为一个表空间（该表空间被划分为一个节点组中的多个数据库分区）中的表定义分区关键字，缺省情况下会根据主关键字的第一列创建一个。若未指定主关键字，则缺省分区关键字是在该表中定义的第一个非长整数字段列。（长整数包括所有长整数数据类型和所有大对象 (LOB) 数据类型）。若正在与单个分区节点组相关的表空间中创建一个表，且您希望有一个分区关键字，则必须明确地定义该分区关键字。缺省情况下，不会创建它。

若没有列满足缺省分区关键字的要求，则会不带关键字创建该表。仅在单个分区节点组中才允许不带分区关键字的表。以后可以使用 **ALTER TABLE** 语句添加或卸下分区关键字。只能对表空间与单分区节点组相关的表更改分区关键字。

选择好的分区关键字很重要。应考虑下列各项：

- 如何存取表
- 查询工作负荷的性质
- 数据库系统所使用的连接策略。

若并置不是主要的考虑事项，则一个好的表分区关键字就是要考虑的主要事项，它可以将数据均匀地分布在节点组中的所有数据库分区上。与一个节点组相关的表空间中每个表的分区关键字确定这些表是否已并置。下列情况中，表被认为是并置的：

- 表被置于同一个节点组中的表空间内
- 每个表中的分区关键字具有相同数量的列
- 对应列的数据类型是分区兼容的。

这确保并置的表中具有相同分区关键字值的那些行位于同一个分区上。有关分区兼容性的详情，参见第118页的『分区兼容性』。有关表并置的详情，参见第118页的『表并置』。

不适当的分区关键字会导致数据分布不均匀。数据分布不均匀的列和含有少数相异值的列不应选作分区关键字。相异值的数目必须足够大，以确保将行均匀地分布在节点组中的所有数据库分区上。应用分区散列算法的成本与分区关键字的大小是成正比的。分区关键字不能超过 16 列，而且列越少，性能越好。不应将不需要的列包括在分区关键字中。

当定义分区关键字时，应该考虑下列几点：

- 不支持创建只包含长整数数据类型（LONG VARCHAR、LONG VARCHARIC、BLOB、CLOB 或 DBCLOB）的多分区表。
- 不能改变分区关键字定义。
- 不能更新表中某一行的分区关键字列值。
- 只能删除或插入分区关键字列值。
- 分区关键字应该包括最频繁连接的列。
- 分区关键字应该由经常参与 GROUP BY 子句的列组成。
- 任何唯一关键字或主关键字必须包含所有分区关键字列。
- 在联机事务处理 (OLTP) 环境中，分区关键字中的所有列都应该使用带常量或主变量的等于 (=) 谓词来参与该事务。例如，假定有一个在事务中经常使用的雇员号 *emp_no*，如：

```
UPDATE emp_table SET ... WHERE  
emp_no = host-variable
```

在此情况下，EMP_NO 列对于 EMP_TABLE 而言是一个不错的单列分区关键字。

散列分区是确定分区表中每一行的位置的方法。该方法的原理如下：

1. 将散列算法应用于分区关键字的值，并生成介于 0 与 4095 之间的分区号。

2. 在创建节点组时创建分区映射。每个分区号依次以循环方式重复，以填写该分区映射。有关分区映射的详情，参见第115页的『分区映射』。
3. 该分区号用作分区映射中的一个索引。分区映射中该位置处的编号是存储该行的数据库分区的编号。

表并置

您可能会发现，作为对特定查询的响应，两个或多个表频繁地提供数据。在此情况下，您会希望这样的表中的相关数据的位置尽可能地靠近。在数据库被物理地划分为两个或多个数据库分区的环境中，必须有一种方法可将划分的表的相关碎片尽可能地靠近。完成此过程的功能称为表并置。

当表存储在同一个节点组中且它们的分区关键字兼容时，这些表是已并置的。将两个表置于同一个节点组中，确保存在一个公共的分区映射。这些表可能位于不同的表空间，但是这些表空间必须与同一个节点组相关。每个分区关键字中对列的数据类型必须是分区兼容的。有关分区兼容性的详情，参见『分区兼容性』。

当存取用于连接或子查询的多个表时，DB2 能够识别要连接的数据是否位于同一个数据库分区上。当发生这种情况时，DB2 可以选择在存储该数据的数据库分区上执行连接或子查询，而不必在数据库分区之间移动数据。这种在数据库分区上执行连接或子查询的能力具有显著的性能优点。有关详情，参考管理指南：性能中的“并置连接”。

分区兼容性

对分区关键字的对应列的基本数据类型进行比较，并可将它们说明为是分区兼容的。分区兼容的数据类型具有如下特性：具有相同值但有不同类型的两个变量会按相同的分区算法映射至同一个分区号。

分区兼容性具有下列特征：

- 基本数据类型与另一个相同的基本数据类型兼容。
- 内部格式用于 DATE、TIME 和 TIMESTAMP 数据类型。它们彼此都不兼容，且都不与 CHAR 兼容。
- 分区兼容性不受带有 NOT NULL 或 FOR BIT DATA 定义的列的影响。
- 对兼容数据类型的空值的处理是完全相同的；对不兼容数据类型的空值的处理可能不相同。
- 用户定义类型的基本数据类型用于分析分区兼容性。
- 对分区关键字中值相同的小数的处理是完全相同的，即使它们的标度和精度不同。

- 字符串中 (CHAR、VARCHAR GRAPHIC 或 VARGRAPHIC) 的尾部空格会被散列算法忽略。
- BIGINT、SMALLINT 和 INTEGER 是兼容的数据类型。
- REAL 和 FLOAT 是兼容的数据类型。
- 不同长度的 CHAR 和 VARCHAR 是兼容的数据类型。
- GRAPHIC 和 VARGRAPHIC 是兼容的数据类型。
- 分区兼容性不适用于 LONG VARCHAR、LONG VARGRAPHIC、CLOB、DBCLOB 和 BLOB 数据类型，因为不支持它们作为分区关键字。

复制的摘要表

摘要表是由查询定义的表，也用于确定表中的数据。摘要表可用于改进查询的性能。若 DB2 确定查询的一部分可用摘要表来解决，则数据库管理程序可重写该查询以使用摘要表。

在分区数据库环境中，可复制摘要表。可使用复制的摘要表来改进查询性能。复制的摘要表基于一个可能已在单分区节点组中创建，但您想要在节点组中的所有数据库分区中进行复制的表。要创建复制的摘要表，调用带 REPLICATED 关键字的 CREATE TABLE 语句。只能对使用 REFRESH DEFERRED 选项定义的摘要表指定 REPLICATED 关键字。

有关摘要表的详情，参见管理指南：实现中的“创建摘要表”。

通过使用复制的摘要表，可将通常未并置的表并置。对于一个大的事实表和一个小的维表的连接，复制的摘要表特别有用。要将所需的额外存储器以及必须更新每个副本所带来的影响降至最小，要复制的表应较小，且更新不频繁。

注：还应考虑复制那些不常更新的更大的表：复制的一次性成本可通过并置获得的性能效益来抵消。

通过在定义复制表所用的子查询子句中指定适当的谓词，可以复制选择的列和 / 或选择的行。

有关复制的摘要表的详情，参考 *SQL Reference* 中的 CREATE TABLE 语句。有关并置连接的详情，参考管理指南：实现中的“并置连接”。

设计和选择表空间

表空间是一种存储模型，它提供一个数据库和存储在该数据库内的表之间的一个间接层。表空间驻留在节点组中。它们允许将数据库和表数据的位置直接指定到容器上。（容器可以是目录名、设备名或文件名。）这可提供改善的性能、更灵活的配置和更好的完整性。

有关创建或改变表空间的详情，参见“创建表空间”或*管理指南：实现中的“改变表空间”*。

因为表空间驻留在节点组中，所以选择保存表的表空间时要定义如何将该表的数据分布至节点组中的数据库分区上。单个表空间可跨多个容器。在同一个物理磁盘（即驱动器）上创建多个容器（一个或多个表空间）是可能的。为提高性能，每个容器应使用不同的磁盘。图36举例说明了一个数据库内的表和表空间与该数据库相关的容器之间的关系。

数据库

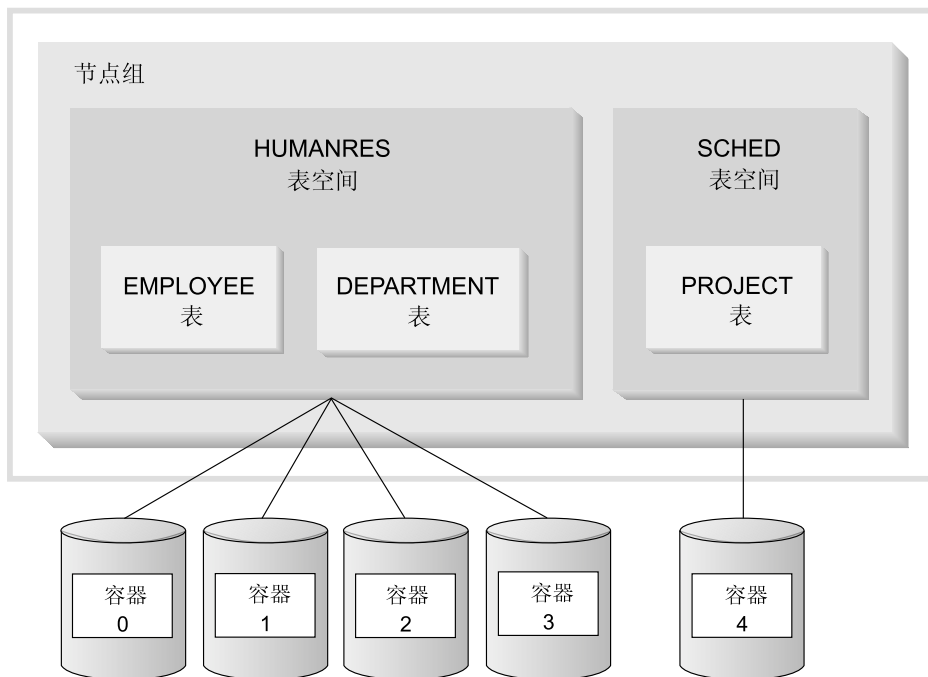


图 36. 数据库中的表空间和表

EMPLOYEE 和 DEPARTMENT 表在 HUMANRES 表空间中，该表空间跨越容器 0、1、2 和 3。PROJECT 表在容器 4 的 SCHED 表空间里。此示例显示一个独立磁盘上已有的每个容器。

数据库管理程序会尝试平衡分布在所有容器中的数据负荷。因此，所有容器都将用于存储数据。数据库管理程序在使用另一个容器之前写入一个容器的页数称为数据块大小。数据库管理程序并非始终从第一个容器开始存储表数据。

图37显示具有两个 4KB 页数据块大小的 HUMANRES 表空间，它有四个容器，每个容器有少量已分配的数据块。DEPARTMENT 和 EMPLOYEE 表都有 7 页，且横跨全部四个容器。

HUMANRES 表空间

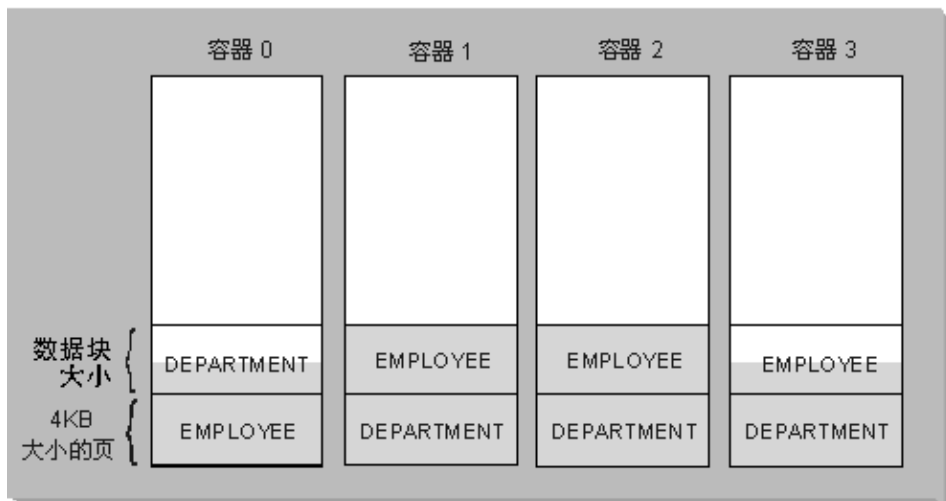


图 37. 容器和数据块

一个数据库至少必须包含三个表空间:

- 一个目录表空间，它包含该数据库的所有系统目录表。此表空间称为 SYSCATSPACE，它不能被卸下。IBMCATGROUP 是此表空间的缺省节点组。
- 一个或多个用户表空间，它们包含所有用户定义表。缺省情况下，会创建一个表空间 USERSPACE1。IBMDEFAULTGROUP 是此表空间的缺省节点组。

当创建一个表时，应指定表空间名，否则，结果可能不是您所期望的。若不指定表空间名，则根据下列规则放置该表：若存在用户创建的表空间，则选择页大小最小，但对此表而言足够大的表空间。否则，若 USERSPACE1 的页大小对此表而言足够大，则使用它。若不存在具有足够大页大小的表空间，则不创建该表。

表的页大小或者由行大小确定，或者由列数确定。一行中允许的最大长度取决于创建此表所在的表空间的页大小。可能的页大小的值为 4 KB（缺省值）、8 KB、16 KB 和 32 KB。可使用具有一种页大小的表空间作为基表，而使用具有另一种页大小的另一个表空间来存储长整数或 LOB 数据。（记住，SMS 不支持跨表空间的表，而 DMS 却支持。）若列数或行大小超过表空间页大小的限制，则返回一个错误 (SQLSTATE 42997)。

- 一个或多个临时表空间，它们包含临时表。临时表空间可以是系统临时表空间或用户临时表空间。数据库必须有至少一个系统临时表空间；在缺省情况下，创建数据库时会创建一个名为 TEMPSPACE1 的系统临时表空间。IBMTEMPGROUP 是此表空间的缺省节点组。用户临时表空间不是在数据库创建时缺省创建的。

若一个数据库使用多个临时表空间，则以循环方式将临时对象分配给这些临时表空间。

如果对用 4 KB 以上的页大小定义的表空间中的表运行查询（例如，对 1012 列使用 ORDER BY），则某些查询可能会失败。若没有使用更大页大小定义的临时表空间，则会发生这种情况。可能需要创建具有更大页大小（8 KB、16 KB 或 32 KB）的临时表空间。若没有与用户表空间中的最大页大小具有相同页大小的临时表空间，任何一种 DML（数据操作语言）语句都可能失败。

应定义一个 SMS 临时表空间，使它的页大小等于大多数用户表空间所使用的页大小。这对于典型的环境和工作负荷应已足够。另见第132页的『对临时表空间的建议』。

在分区数据库环境中，目录节点将包含全部三个缺省表空间，而其他每个数据库分区将只包含 TEMPSPACE1 和 USERSPACE1。

有两种类型的表空间，它们都可以在单个数据库中使用：

- 『系统管理空间』：操作系统的文件管理器控制该存储空间。
- 第126页的『数据库管理空间的表空间』：数据库管理程序控制该存储空间。

系统管理空间

在 SMS（系统管理空间）表空间中，操作系统的文件系统管理器分配和管理用于存储表的空间。该存储模型通常由存储在文件系统空间中的多个文件组成，这些文件表示表对象。用户决定这些文件的位置，DB2 控制它们的名称，而文件系统负责管理它们。通过控制写入每个文件的数据量，数据库管理程序均匀地将数据分布在所有表空间容器中。SMS 表空间是缺省表空间。

每个表至少有一个与它相关的 SMS 物理文件。有关这些文件的列表及其内容的说明，参见第124页的『SMS 物理文件』。

在 SMS 表空间中，当对象增大时，文件每次扩展一页。若需要提高插入性能，可以考虑启用多页文件分配。这允许系统一次为文件分配或扩展多页。必须运行 **db2empfa**，才可启用多页文件分配。在分区数据库环境中，必须对每个数据库分区运行此实用程序。一旦启用了多页文件分配，就不能禁用它。有关 **db2empfa** 的详情，参考 *Command Reference*。

应该在 CREATE DATABASE 命令上或 CREATE TABLESPACE 语句上使用 MANAGED BY SYSTEM 选项来显式地定义 SMS 表空间。当设计 SMS 表空间时，必须考虑两个关键要素：

- 表空间的容器。

必须指定要用于表空间的容器数。标识要使用的所有容器是非常重要的，因为您不能在创建了 SMS 表空间之后添加或删除容器。在分区数据库环境中，当将一个分区添加至 SMS 表空间的节点组时，可以使用 ALTER TABLESPACE 语句为新的分区添加容器。

用于一个 SMS 表空间的每个容器都标识一个绝对或相对的目录名。其中每一个目录都可以位于不同的文件系统（物理磁盘）上。表空间的最大大小可以按以下方法估计：

容器数 * (操作系统支持的最大文件系统大小)

此公式假定有一个唯一的文件系统映射至每个容器，且每个文件系统都具有大量的可用空间。实际上，情况可能不是这样，表空间最大大小可能小得多。

注：定义容器时必须很小心。若容器上已有文件或目录，将返回一个错误 (SQL0298N)。

- 表空间的数据块大小。

数据块大小只能在创建表空间时指定。因为以后不能更改它，因此为数据块大小选择一个适当的值就很重要。有关详情，参见第131页的『选择数据块大小』。

当创建表空间时，若不指定数据块大小，则数据库管理程序将使用缺省数据块大小来创建表空间，该缺省大小由 *dft_extent_sz* 数据库配置参数定义（有关此参数的详情，参考管理指南：性能）。此配置参数最初是根据创建该数据库时提供的信息设置的。若未在 CREATE DATABASE 命令上指定 DFT_EXTENT_SZ 参数，则会将缺省数据块大小设置为 32。

要为表空间的容器数和数据块大小选择适当的值，必须了解：

- 操作系统对逻辑文件系统的大小施加的限制。

例如，某些操作系统有 2GB 的限制。因此，若想要一个 64GB 的表对象，则在此类型的系统上将需要至少 32 个容器。

当创建该表空间时，可以指定驻留在不同文件系统上的容器，以便增加可以存储在该数据库中的数据量。

- 数据库管理程序如何管理与一个表空间相关的数据文件和容器。

在为该表空间指定的第一个容器中创建第一个表数据文件 (SQL00001.DAT)，并允许此文件增大至该数据块大小。当它达到此大小之后，数据库管理程序将数据写入下一个容器中的 SQL00001.DAT。此过程会继续，直到所有容器都包含 SQL00001.DAT 文件为止，在那时，数据库管理程序会返回至第一个容器。此过程（称为条形化分布）会继续在容器目录中运行，直到一个容器装满为止 (SQL0289N) 或操作系统中已无空间可分配为止（磁盘已满错误）。条纹化也用于索引 (SQLnnnnn.INX)、长整数字段 (SQLnnnnn.LF) 和 LOB (SQLnnnnn.LB 和 SQLnnnnn.LBA) 文件。

注：只要任何一个容器一满，SMS 表空间就满了。因此，为每个容器分配相同容量的空间是很重要的。

为了有助于将数据更加均匀地分布至这些容器中，数据库管理程序根据将表标识符（上例中为 1）除以容器数所得的模，来确定首先使用的容器。容器从 0 开始依次编号。

有关在 SMS 表空间中使用的文件的详情，参见『SMS 物理文件』。

SMS 物理文件

下列文件可在 SMS 表空间目录容器中找到：

文件名	说明
-----	----

SQLTAG.NAM

每个容器的子目录中都存在这样一个文件，而且当连接数据库以验证该数据库是完整和一致时，数据库管理程序要使用它们。

SQLxxxx.DAT

表文件。它存储所有表行，但 LONG VARCHAR、LONG VARGRAPHIC、BLOB、CLOB 或 DBCLOB 数据除外。

SQLxxxx.LF 包含 LONG VARCHAR 或 LONG VARGRAPHIC 数据（也称为“长整数字段数据”）的文件。仅当表中存在 LONG VARCHAR 或 LONG VARGRAPHIC 列时，才创建此文件。

SQLxxxx.LB 包含 BLOB、CLOB 或 DBCLOB 数据（也称为“LOB 数据”）的文件。仅当表中存在 BLOB、CLOB 或 DBCLOB 列时，才创建这些文件。

SQLxxxxx.LBA

包含有关 SQLxxxxx.LB 文件的分配和空闲空间信息的文件。

SQLxxxxx.INX

表的索引文件。对应表的所有索引都存储在此单个文件中。仅当定义了索引时，才创建该文件。

注：当卸下一个索引时，索引 (.INX) 文件并未实际释放该空间，除非删除该索引文件。若表上的所有索引都被卸下（和被落实），或若该表被重组，则会删除该索引文件。若未删除该索引文件，则一旦落实了卸下操作，就会将该空间标记为空闲的并将重新使用它，以便将来创建索引或维护索引。

SQLxxxxx.DTR

用于重组 DAT 文件的临时数据文件。当重组一个表时，REORG 实用程序（通过 REORG TABLE 命令调用）会在其中一个系统临时表空间中创建一个表。可以定义这些临时表空间使用与用户定义表所用的不同的容器。

SQLxxxxx.LFR

用于重组 LF 文件的临时数据文件。当重组一个表时，REORG 实用程序（通过 REORG TABLE 命令调用）会在其中一个系统临时表空间中创建一个表。可以定义这些临时表空间使用与用户定义表所用的不同的容器。

SQLxxxxx.RLB

用于重组 LB 文件的临时数据文件。当重组一个表时，REORG 实用程序（通过 REORG TABLE 命令调用）会在其中一个系统临时表空间中创建一个表。可以定义这些临时表空间使用与用户定义表所用的不同的容器。

SQLxxxxx.RBA

用于重组 LBA 文件的临时数据文件。当重组一个表时，REORG 实用程序（通过 REORG TABLE 命令调用）会在其中一个系统临时表空间中创建一个表。可以定义这些临时表空间使用与用户定义表所用的不同的容器。

注：

1. 不要直接更改这些文件。只能使用记载的 API 和实现那些 API 的工具（包括命令行处理器和控制中心）来间接存取它们。
2. 不要移动这些文件。
3. 不要除去这些文件。

4. 备份数据库或表空间唯一受支持的方法是通过 **sqlubkp**（备份数据库）API 来进行，包括该 API 的命令行处理器和 控制中心实现。

数据库管理空间的表空间

在 DMS（数据库管理空间）表空间中，数据库管理程序控制存储空间。该存储模型由有限数量的设备组成，这些设备的空间由 DB2 来管理。管理员决定要使用哪个设备，而 DB2 管理那些设备上的空间。此表空间实质上是为了最好地满足数据库管理程序的需要而设计的特殊目的的文件系统。该表空间定义包括一些设备或文件的列表，它们属于可以用来存储数据的表空间。

包含用户定义的表和数据的 DMS 表空间可以定义为：

- 存储正常表和索引数据的规则表空间
- 存储长整数字段或 LOB 数据的长整数表空间。

在设计 DMS 表空间和容器时，应该考虑下列事项：

- 数据库管理程序使用“拆开”来确保数据均匀地分布在所有容器上。
- 对于 4 KB 页，规则表空间的最大大小是 64 GB；对于 8 KB 页，是 128 GB；对于 16 KB 页，是 256 GB；对于 32 KB 页，是 512 GB。长整数表空间的最大大小是 2 TB。
- 与 SMS 表空间不同，组成一个 DMS 表空间的容器不必大小相同；然而，通常不建议这样做，因为会导致在容器间不均匀地进行条纹化，并会降低性能。若任何容器已满，DMS 表空间会使用其他容器中的可用空闲空间。
- 因为空间是预分配的，所以在能够创建表空间之前，该空间必须是可用的。当使用设备容器时，该设备也必须足够的空间以便存储容器定义。每个设备上只能定义一个容器。为避免浪费空间，设备的大小应该等于容器的大小。例如，若分配给该设备 5 000 页，而将设备容器定义为分配 3 000 页，则设备上的 2 000 页将是不可用的。
- 每个容器上有一页是为额外开销保留的，而其余页将一次使用一个数据块。只使用整个数据块，因此为了对空间进行最优管理，可以使用如下公式来帮助您确定当分配容器时要使用的适当大小：

$$(\text{extent_size} * n) + 1$$

其中，*extent_size* 是表空间中每个数据块的大小，而 *n* 是您要在该容器中存储的数据块数。

- 表空间中有三个数据块是保留给额外开销使用的。
- 要存储任何用户表数据，至少需要两个数据块。（这些数据块是一个表的规则数据所必需的，但不是任何索引、长整数字段或大对象数据所需的，它们需要自己的数据块。）

- 设备容器必须使用带“字符型特殊接口”的逻辑卷，而不是物理卷。
- 在 DMS 表空间中可以使用文件以代替设备。文件和设备之间不存操作性区别；但是，由于存在与文件系统相关的运行期额外开销，文件可能效率较低。在下列情况下，适合使用文件：
 - 设备不直接受支持
 - 设备不可用
 - 不需要最大性能
 - 不想设置设备。
- 若工作负荷涉及 LOB 或 LONG VARCHAR 数据，则可通过文件系统高速缓存改进性能。注意，DB2 的缓冲池不缓冲 LOB 和 LONG VARCHAR。
- 某些操作系统允许拥有大小超过 2GB 的物理设备。应该考虑将该物理设备分区为多个逻辑设备，以使任何容器都不超过操作系统允许的大小。

将容器添加至 DMS 表空间

ALTER TABLESPACE 语句用来向一个现存表空间添加容器，以增加它的存储容量。然后，该表空间的内容要在所有容器中重新平衡。在重新平衡期间，不限制对该表空间的存取。若需要添加多个容器，应该在一个 ALTER TABLESPACE 语句中或在同一个事务中同时添加它们，以免数据库管理程序不得不多次重新平衡这些容器。

应使用 LIST TABLESPACE CONTAINERS 或 LIST TABLESPACES 命令来检查一个表空间的容器的满载程度。添加新容器应在现存容器几乎满载或完全满载之前进行。要在重新平衡完成之后，所有容器上的新空间才可用。

添加比现存容器小的容器会导致数据分布不均匀。这可能导致并行 I/O 操作，如预读取数据，它们的执行效率比在大小相同的容器上执行的效率低。

表空间的设计考虑事项

本节包括下列主题：

- 第128页的『表空间输入和输出 (I/O) 的考虑事项』
- 第129页的『将表空间映射至缓冲池』
- 第130页的『将表空间映射至节点组』
- 第130页的『将表映射至表空间』
- 第131页的『选择数据块大小』
- 第132页的『对临时表空间的建议』
- 第133页的『对目录表空间的建议』

- 第134页的『工作负荷考虑事项』
- 第135页的『选择 SMS 或 DMS 表空间』
- 第136页的『当将数据存放在 RAID 设备中时优化性能』。

表空间输入和输出 (I/O) 的考虑事项

表空间的类型和设计决定了对该表空间执行的 I/O 的效率。在进一步考虑关于表空间的设计和使用的问题之前，您应了解下列概念。

- 大块读取** 在单个请求中检索多页（通常为一个数据块）的一种读取。一次读取几页比分别读取每页更有效。
- 预读取** 在一个查询引用页之前，对那些页的读取。总的目的是为缩短响应时间。若页的预读取可以与查询的执行异步发生，就能够达到此目的。当一个或多个 CPU 或 I/O 子系统以最大能力运行时，就达到最佳响应时间。
- 页清除** 当读取和修改页时，它们会累积在数据库缓冲池中。当读入一页时，便将其读入到缓冲池页中。若该缓冲池已充满修改的页，则必须将这些修改的页之一写出至磁盘，然后才能再读入新的页。为避免缓冲池变满，页清除程序代理程序的任务就是写出修改的页，以保证缓冲池页可用于将来的读取请求。

无论何时，只要是有利的，DB2 就会执行大块读取。当检索顺序排列或本质上是部分顺序排列的数据时，通常会发生这种情况。在一个读取操作中读取的数据量取决于数据块大小 - 数据块大小越大，一次可以读取的页就越多。

数据块存储在磁盘上的方式影响 I/O 效率。在使用设备容器的 DMS 表空间中，数据往往在磁盘上是连续的，且可以在最短的搜索时间和磁盘延迟内进行读取。但是，若使用的是文件，则数据可能被文件系统分散，并存储在磁盘上的多个位置中。当使用一次将文件扩展一页的 SMS 表空间时（这使产生碎片的概率更高），最可能发生此情况。为了供 DMS 表空间使用而预分配的大文件往往在磁盘上是连续的，尤其当该文件分配在一个干净的文件空间中时更是这样。

可以通过在 CREATE TABLESPACE 语句上调整 PREFETCHSIZE 参数，来控制预读取的程度。（该数据库中所有表空间的缺省值由 *dft_prefetch_sz* 数据库配置参数设置。）PREFETCHSIZE 参数告诉 DB2 在触发一个预读取时要读取的页数。通过在 CREATE TABLESPACE 语句上将 PREFETCHSIZE 设置为 EXTENTSIZE 参数的倍数，可以并行读取多个数据块。（该数据库中所有表空间的缺省值由 *dft_extent_sz* 数据库配置参数设置。）EXTENTSIZE 参数指定在跳至下一个容器之前将写入一个容器的 4 KB 大小的页数。

例如，假定有一个表空间使用三个设备。若将 `PREFETCHSIZE` 设置为 `EXTENTSIZE` 的三倍，则 DB2 可以用并行方式从每个设备中执行大块读取，从而显著增加 I/O 吞吐量。此情况假定每个设备是一个单独的物理设备，且控制器具有足够的带宽来处理来自每个设备的数据流。注意，DB2 可能需要根据查询速度、缓冲池的利用情况和其他因素，在运行期动态调整预读取参数。

某些文件系统使用它们自己的预读取方法（如，AIX 上的“日志文件系统”）。在某些情况中，文件系统的预读取设置得比 DB2 的预读取更主动。这可能导致使用文件容器的 SMS 和 DMS 表空间的预读取似乎比使用设备的 DMS 表空间的预读取执行效率更高。但这是误导，因为它可能是在文件系统中发生的附加级别的预读取的结果。DMS 表空间应该能够比任何等效配置的执行效率高。

为提高预读取效率（甚至读取效率），必须存在足够数量的干净缓冲池页。例如，可能有一个并行预读取请求要从一个表空间读取三个数据块，对于正在读取的每一页，从缓冲池写出经过修改的一页。该预读取请求可能被拖慢，导致它跟不上查询的进展。应配置足够数量的页清除程序，以满足预读取请求。对于数据库所用的每个实际磁盘，应定义至少一个页清除程序。有关这些主题的详情，参考管理指南：性能。

将表空间映射至缓冲池

每个表空间都与一个特定的缓冲池相关。缺省缓冲池是 `IBMDEFAULTBP`。若另一个缓冲池要与一个表空间相关，则该缓冲池必须存在（用 `CREATE BUFFERPOOL` 语句定义它），且当创建该表空间（使用 `CREATE TABLESPACE` 语句）时定义关联。表空间与缓冲池之间的关联可以使用 `ALTER TABLESPACE` 语句来更改。

若拥有多个缓冲池，则可以配置数据库使用的内存，以改善整体性能。对于具有一个或多个用户可任意存取的大表的表空间，缓冲池的大小可能受到限制，因为高速缓存该数据页可能没有好处。用于联机事务应用程序的表空间可以与一个较大的缓冲池相关联，以便可以更长地高速缓存应用程序所使用的数据页，导致响应时间更快。配置新缓冲池时必须小心。有关本主题的详情，参考管理指南：性能中的“管理数据库缓冲池”。

注：若确定数据库需要 8 KB、16 KB 或 32 KB 的页大小，必须将使用其中一种页大小的每个表空间映射为具有相同页大小的缓冲池。

当启动数据库时，所有缓冲池必需的存储器对于数据库管理程序必须是可用的。若 DB2 无法获取必需的存储器，则数据库管理程序将使用缺省缓冲池（具有 4 KB、8 KB、16 KB 和 32 KB 页大小的缓冲池各一个）启动，并发出警告。

在分区数据库环境中，可以为该数据库中的所有分区创建大小相同的一个缓冲池。也可以在不同的分区上创建不同大小的缓冲池。有关 CREATE BUFFERPOOL 语句的详情，参考 *SQL Reference*。

将表空间映射至节点组

在分区数据库环境中，每个表空间都与一个特定的节点组相关。这允许将表空间的特征应用于该节点组中的每个节点。该节点组必须存在（用 CREATE NODEGROUP 语句定义它），且当使用 CREATE TABLESPACE 语句创建表空间时定义表空间与该节点组之间的关联。

不能使用 ALTER TABLESPACE 语句来更改表空间与节点组之间的关联。只能为节点组内的个别分区更改表空间规范。在单分区环境中，每个表空间都与一个缺省节点组相关。除非定义了一个系统临时表空间（这将导致使用 IBMTEMPGROUP），否则，当定义一个表空间时，缺省节点组是 IBMDEFAULTGROUP。有关 CREATE NODEGROUP 语句的详情，参考 *SQL Reference*。有关节点组和物理数据库设计的详情，参见第113页的『设计节点组』。

将表映射至表空间

当确定如何将表映射至表空间时，应考虑：

- 表的分区。

至少，应该确保选择的表空间位于含有您所要的分区的节点组中。

- 表中的数据量。

若计划在一个表空间中存储许多小表，则考虑使用 SMS 充当该表空间。对于小表，DMS 表现在 I/O 和空间管理效率方面的优点就没有那么重要。一次分配一页空间的 SMS 的优点（且仅在需要时使用）却对小表更具吸引力。若一个表较大或者您需要更快地存取表中的数据，应考虑具有较小数据块大小的 DMS 表空间。

您可能希望对每个非常大的表都使用单独的表空间，而将所有的小表组合在单个表空间中。这种分隔还允许您根据表空间的使用选择适当的数据块大小。（有关其他信息，参见第131页的『选择数据块大小』。

- 表数据的类型。

例如，有的表可能包含不经常使用的历史数据；最终用户可能愿意接受较长的响应时间，来等待对此数据执行的查询。在这种情况下，您可能会为历史表使用另一个表空间，并将此表空间分配给存取速率较低的较便宜的物理设备。

或者，您也许能够标识某些需要高可用性和快速响应时间的基表。可能要将这些表置于分配给一个快速物理设备的表空间中，这样将有助于支持这些重要的数据需要。

通过使用 DMS 表空间，还可以将表数据分布在三个不同的表空间中：一个存储索引数据；一个存储 LOB 和长整数字段数据；一个存储规则表数据。这允许您选择表空间特征和支持最适合该数据的那些表空间的物理设备。例如，可能会将索引数据置于可找到的最快的设备上，这样性能可显著提高。若将一个表分布在各 DMS 表空间中，如果启用前滚恢复，应考虑一起备份和复原该表的所有部分。SMS 表空间不支持以此方式将表分布在所有表空间中。

- 管理问题。

某些管理功能可以在表空间级执行，但不能在数据库或表级执行。例如，备份表空间（而不是数据库）可以帮助您更好地利用时间和资源。它允许频繁地备份带有大量更改的表空间，同时仅偶尔地备份带有少量更改的表空间。

可以复原数据库或表空间。若不相关的表不共享表空间，就可以选择复原数据库一个较小的部分，以降低成本。

一种好方法是将相关的表编组在一组表空间中。这些表可以通过参考约束相关，也可以通过定义的其他商业约束相关。

若经常需要卸下并重新定义特定表，则应在它自己的表空间中定义该表，因为卸下一个 DMS 表空间比卸下一个表更有效率。

选择数据块大小

表空间的数据块大小表示在将数据写入下一个容器之前，将写入当前容器的表数据的页数。当选择数据块大小时，应考虑：

- 表空间中表的大小和类型。

将 DMS 表空间中的空间分配给一个表，一次一个数据块。当填充该表而一个数据块变满时，会分配新的数据块。

一个表由下列单独的表对象组成：

- 数据对象。它是存储规则列数据的地方。
- 索引对象。在表上定义的所有索引都存储在这里。
- 长整数字段对象。若表有一个或多个 LONG 列，则长整数字段数据存储在此处。
- 两个 LOB 对象。若表有一个或多个 LOB 列，则它们都存储在这两个表对象中：
 - 一个表对象用于存储 LOB 数据
 - 第二个表对象用于存储描述 LOB 数据的元数据。

每个表对象都是单独存储的，每个对象按需要分配新的数据块。每个表对象还与称为数据块映象的一个元数据对象配成一对，该元数据对象描述该表空间中属于该表对象的所有数据块。用于数据块映象的空间也是以一次一个数据块的方式分配。

因此，一个表的空间的初始分配是每个表对象两个数据块。若您在一个表空间中有多个小表，您可能要分配相对大的空间来存储相对少量的数据。在这种情况下，应该指定小的数据块大小或使用一次分配一页的 SMS 表空间。

另一方面，若您有一个增长速率高的非常大的表，且您使用具有较小数据块大小的 DMS 表空间，您可能会有与附加数据块的频繁分配相关的不需要的额外开销。

- 对这些表存取的类型。

若对表的存取包括许多查询或处理大量数据的事务，则从表中预读取数据可以显著改善性能。（有关数据预读取以及它与数据块大小的关系的详情，参考管理指南：性能。）

- 必需的最小数据块数。

若容器中没有足够的空间，以供表空间的五个数据块使用，则将无法创建表空间。

对临时表空间的建议

建议定义一个 SMS 临时表空间，使它的页大小等于大多数规则表空间所使用的页大小。这应该适用于典型的环境和工作负荷。但最好用不同的临时表空间配置和工作负荷进行实验。应该考虑下列几点：

- 在大多数情况下，临时表空间是按顺序成批存取的。也就是说，插入一组行或顺序读取一组行。因此，当需要更少的逻辑或物理页 I/O 请求来读取给定的数据量时，页大小越大，通常所获得的性能越优。当临时表的平均行大小小于页大小除以 255 的值时，情况就不会总是这样。无论是那种页大小，任何一页上最多可有 255 行。例如，若一个查询需要 15 字节一行的临时表，则用 4 KB 页大小的临时表空间更合适，因为 255 行可以全部包含在一个 4 KB 页中。8 KB（或更大的）页大小将在临时表的每一页上浪费至少 4 KB（或更多）字节的空间，最好不要减少需要的 I/O 请求数。
- 若在数据库中有超过一半的规则表空间使用了相同的页大小，建议您定义具有相同页大小的临时表空间。这样做的原因是这种安排可以使临时表空间与大多数或全部的规则表空间共享同一个缓冲池空间。这样可简化缓冲池的调整。
- 当使用临时表空间重组表时，该临时表空间的页大小必须与该表的页大小匹配。由于这个原因，您应确保为现存表使用的每种不同的页大小定义了临时表空间，这样才可使用临时表空间重组这些表。

也可不用临时表空间来执行重组，即“在应有的位置”重组该表；或者说，直接在目标表空间中重组。当然，这种“在应有的位置”重组要求在目标表空间中有额外的空间来完成重组过程。有关表重组的其他信息，参考管理指南：性能。

- 一般而言，当存在具有不同页大小的临时表空间时，优化器常常会选择具有最大缓冲池的临时表空间。在这种情况下，比较聪明的做法是给一个临时表空间分配一个足够大的缓冲池，而给其余临时表空间分配较小的缓冲池。这种缓冲池分配将有助于保证有效利用主存储器。例如，若目录表空间使用 4 KB 页，而其余表空间使用 8 KB 页，则最佳临时表空间配置可能是：具有一个大缓冲池的一个 8 KB 临时表空间；和一个具有小一些的缓冲池的一个 4 KB 表空间。

注：目录表空间只能使用 4 KB 页大小。因此，数据库管理程序始终强制保证 4 KB 临时表空间的存在，以允许目录表重组。

- 一般情况下，定义具有相同页大小的多个临时表空间没有什么好处。
- 对于临时表空间而言，SMS 几乎总是比 DMS 更合适，因为：
 - 在 SMS 中，磁盘空间是按需要分配的；而在 DMS 中必须预先分配它。预先分配可能比较困难：临时表空间保存着瞬时数据，这些数据在某个时候可能需要很大的存储器，但在正常情况下存储器需求却低得多。使用 DMS 时，必须预先分配存储器的峰值需求量；而使用 SMS 时，在高峰期间可以使用额外的磁盘空间来完成其他任务。
 - 数据库管理程序尝试将临时表页保存在内存中，而不是将它们写出至磁盘。因此，DMS 的性能优点就没有那么突出。
 - SMS 容器可以利用文件系统缓冲；DMS 容器则不能。

对目录表空间的建议

建议使用 SMS 表空间来存储数据库目录，原因如下：

- 该数据库目录由许多大小不同的表组成。当使用 DMS 表空间时，对于每个表对象，分配最少两个数据块。根据选择的数据块大小，可能产生大量已分配而未使用的空间。当使用 SMS 表空间时，应选择小的数据块大小（二至四页）；否则，应使用 SMS 表空间。
- 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。从磁盘读取 LOB 降低了性能。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，使得避免 I/O 成为可能（若先前引用了该 LOB 的话）。

将这些因素考虑在内，SMS 表空间更适合用作目录。

另一个要考虑的因素是，将来是否需要扩大目录表空间。虽然某些平台支持扩大 SMS 容器的基本存储器，虽然可以使用重定向复原来扩大 SMS 表空间，但使用 DMS 表空间能简化添加新容器的工作。

工作负荷考虑事项

在您的环境中由 DB2 管理的主要工作负荷类型，会影响您选择要使用的表空间类型以及要指定的页大小。联机事务处理 (OLTP) 工作负荷的特征是，事务需要对数据进行随机存取，并常常返回小的数据集。因为存取是随机的，并且是存取一页或几页，所以预读取是不可能的。

使用设备容器的 DMS 表空间在这种情况下表现得最好。若不需要最大性能，使用文件容器的 DMS 表空间或 SMS 表空间也适用于 OLTP 工作负荷。若期望很少的顺序 I/O 或不期望它，则 CREATE TABLESPACE 语句上的 EXTENTSIZE 和 PREFETCHSIZE 参数的设置对于 I/O 的效率就显得不重要。

查询工作负荷的特征是，事务需要对数据进行顺序存取或部分顺序存取，并常常返回大的数据集。使用多个设备容器（每个容器都在单独的磁盘上）的 DMS 表空间，最有可能提供有效的并行预读取。应该将 CREATE TABLESPACE 语句上的 PREFETCHSIZE 参数的值设置为 EXTENTSIZE 参数的值乘以设备容器数之积。这允许 DB2 以并行方式从所有的容器中预读取。

查询工作负荷的一个合理的替代方法是使用文件（若文件系统有自己的预读取的话）。这些文件可以是使用文件容器的 DMS 类型或 SMS 类型。注意，若使用 SMS，则需要将目录容器映射至单独的物理磁盘，以实现 I/O 并行性。

混合工作负荷的目标是：对于 OLTP 工作负荷，使单个 I/O 请求尽可能有效；而对于查询工作负荷，最大程度地提高并行 I/O 的效率。

确定表空间页大小的考虑事项如下所示：

- 对于执行随机行读写操作的 OLTP 应用程序，通常优先考虑小一点的页大小，因为它为存储不需要的行而浪费的缓冲池空间较少。
- 对于一次存取大量连续行的 DSS 应用程序，大一点的页大小会更好，因为它将减少读取特定行数所需的输入 / 输出请求数。但是，也有例外。若行大小小于：

页大小 / 255

则每页上都会有浪费的空间（每页最多有 255 行）。在这种情况下，更小一点的页大小可能更合适。

- 更大一点的页大小可允许减少索引中的级别数。
- 越大的页，支持的行越长。

- 在缺省的 4 KB 页上，一个表只能有 500 列，而更大的页大小（8 KB、16 KB 和 32 KB）支持 1012 列。
- 表空间的最大大小与表空间的页大小成正比。在 *SQL Reference* 中讲述了这些限制。

选择 SMS 或 DMS 表空间

当确定应使用哪种类型的表空间存储数据时，有一些问题需要权衡一下。

SMS 表空间的优点:

- 直到需要时，系统才分配空间。
- 由于不必预定义容器，所以创建数据库需要的初始工作较少。

DMS 表空间的优点:

- 通过使用 ALTER TABLESPACE 语句，可添加容器来增加表空间的大小。现存数据自动在新的容器集合中重新平衡，以保持最优 I/O 效率。
- 根据存储的数据的类型，一个表可以分布在多个表空间中：
 - 长整数字段和 LOB 数据
 - 索引
 - 规则表数据

可能出于性能方面的考虑而想要分隔表数据，或想要增加存储的表数据量。例如，您可能拥有具有 64GB 规则表数据、64GB 索引数据和 2TB 长整数数据的一个表。若使用的是 8 KB 页，则表数据和索引数据可达 128 GB。若使用的是 16 KB 页，则表数据和索引数据可达 256 GB。若使用的是 32 KB 页，则表数据和索引数据可达 512 GB。

- 可控制数据在磁盘上的位置（若操作系统允许的话）。
- 若所有的表数据都位于单个表空间中，可以使用比卸下和重新定义一个表更少的额外开销来卸下和重新定义一个表空间。
- 通常，精心调整的一组 DMS 表空间的性能将优于 SMS 表空间。

注：在 Solaris 和 PTX (IBM NUMA-Q) 上，强烈建议对性能非常关键的工作负荷使用具有原始设备的 DMS 表空间。

通常，小型个人数据库用 SMS 表空间管理最容易。另一方面，对于大的、增长中的数据库，您可能只希望使用 SMS 表空间用作临时表空间，而使用具有多个容器的单独的 DMS 表空间用于每个表。另外，您可能想将长整数字段数据和索引存储在它们自己的表空间中。

若选择使用带设备容器的 DMS 表空间，您必须乐意调整和管理您的环境。有关详情，参考管理指南：性能中的“DMS 设备的性能考虑事项”。

当将数据存放在 RAID 设备中时优化性能

本节描述当将数据存放在“磁盘冗余阵列”(RAID)设备中时如何优化性能。一般而言，应对使用 RAID 设备的每个表空间执行下列操作：

- 为使用 RAID 设备的表空间定义一个容器。
- 使该表空间的 EXTENTSIZE 等于或数倍于 RAID 条大小。
- 确保该表空间的 PREFETCHSIZE 为：
 - RAID 条大小与 RAID 并行设备数之积（或此积的整数倍），并且是
 - EXTENTSIZE 的一个倍数。
- 使用 DB2_PARALLEL_IO 注册表变量（在下面讨论）对表空间启用并行 I/O。
- 使用 DB2_STRIPED_CONTAINERS 注册表变量（在下面讨论）保证该表空间中的数据块边界是对齐的。

DB2_PARALLEL_IO

当对表空间容器读写数据时，若数据库中的容器数大于 1，DB2 可以使用并行 I/O。但对于某些情况，对单个容器表空间启用并行 I/O 也有好处。例如，若该容器是在由多个物理磁盘组成的单个 RAID 设备上创建的，可发出并行读写调用。

要强制对只有一个容器的表空间执行并行 I/O，可使用 DB2_PARALLEL_IO 注册表变量。可将此变量设置为 "*"（星号），表示每个表空间，或者可将它设置为由逗号分开的表空间 ID 的列表。例如：

```
db2set DB2_PARALLEL_IO=*      {对所有表空间打开并行 I/O}
db2set DB2_PARALLEL_IO=1,2,4,8 {对表空间 1、2、4 和 8 打开并行 I/O}
```

在设置了注册表变量后，必须停止 DB2 (**db2stop**)，然后重新启动它 (**db2start**)，以使更改生效。

DB2_STRIPED_CONTAINERS

目前，当创建 DMS 表空间容器（设备或文件）时，会在该容器开始处存储一个单页标记。其余页可由 DB2 存储数据，并分组成按数据块大小确定的数据块。

当使用 RAID 设备作为表空间容器时，建议用等于或数倍于 RAID 条大小的数据块大小创建表空间。但是，由于存在单页容器标记，因此数据块不会按 RAID 条排列，并且在执行 I/O 请求时可能需要存取比最优情况更多的物理磁盘。

现在创建 DMS 表空间容器时，可将该标记存放于容器自己的（整个）数据块中。这可避免上述问题，但它需要在该容器中占用额外的数据块。要以这种方式创建容器，必须将 DB2 注册表变量 DB2_STRIPED_CONTAINERS 设置为 "ON"，然后停止并重新启动实例：

```
db2set DB2_STRIPED_CONTAINERS=ON
db2stop
db2start
```

（使用 CREATE TABLESPACE 或 ALTER TABLESPACE 语句）创建的任何 DMS 容器的标记都占用一个完整的数据块。现有容器保持不变。

要停止创建具有此属性的容器，重设该变量，然后停止并重新启动实例：

```
db2set DB2_STRIPED_CONTAINERS=
db2stop
db2start
```

“控制中心”和 LIST TABLESPACE CONTAINERS 命令不显示一个容器是否是作为条纹化容器创建的。它们使用标号“文件”或“设备”，这取决于该容器是如何创建的。要验证一个容器是否是作为条纹化容器创建的，可以使用 DB2DART 的 /DTSF 选项来转储表空间和容器信息，然后查看要了解的容器的类型字段。也可使用查询容器 API（**sqlbftcq** 和 **sqlbtcq**）来创建一个显示该类型的简单应用程序。

联合体数据库设计考虑事项

当设计联合体数据库时，考虑下列设计主题：

- 空间要求
- 确定网络优先级。

通常，可从一个联合体数据库存取的数据不存储在该数据库中。对数据源表和视图的引用存储在系统目录中，但实际数据位于数据源中。这样，联合体数据库所需的存储空间可能比常规数据库少。若由于整理系统差别或数据源中缺少某个函数，而必须在本地执行查询，则此一般规则就不适用。在这种情况下，应在 DB2 实现表以进行处理。

由于大多数联合体系统数据通常位于分布在一个网络中的一个或多个数据源中，因此可考虑改变分配给 DB2 和该网络系统的资源。在将多一点的资源分配给 DB2 系统所在的网络，而将少一点的资源分配给数据库管理程序本身之后，性能可能会提高。

第9章 设计分布式数据库

在 DB2 中，一个事务通常称为一个工作单元。工作单元是应用程序进程内可恢复的操作序列。数据库管理程序使用它来确保数据库处于一致状态。对数据库的任何读取或写入都在一个工作单元内执行。

例如，银行事务可能涉及将存款从储蓄帐户转至支票帐户。当应用程序从储蓄帐户中减去一定金额之后，这种不一致状态将一致保持，直到将该金额加入支票帐户为止。在两个步骤都完成后，便到达一致点。可以落实更改，并可使其可用于其他应用程序。

当对数据库发出第一条 SQL 语句时，就启动了一个工作单元。应用程序必须发出 COMMIT 或 ROLLBACK 语句来结束该工作单元。COMMIT 语句使在工作单元内所做的所有更改成为永久性的。ROLLBACK 语句则从该数据库中消除这些更改。若应用程序正常结束，而未显式发出这两条语句中的任何一个，则自动落实该工作单元。若它在执行一个工作单元时异常结束，则自动回滚该工作单元。一旦发出 COMMIT 或 ROLLBACK，就不能停止它们。对于某些多线程应用程序或某些操作系统（如 Windows），若应用程序正常结束，而未显式发出这两条语句中的任何一个，则自动回滚工作单元。建议应用程序始终显式地落实或回滚，以完成工作单元。若部分工作单元未成功完成，则回滚更新，使参与的表保持该事务开始前的状态。这确保了请求既不会丢失，也不会重复。

下列主题提供其他信息：

- 『在一个事务中使用单个数据库』
- 第140页的『在单个事务中使用多个数据库』
- 第145页的『其他配置考虑事项』
- 第147页的『了解两阶段落实过程』
- 第150页的『在两阶段落实期间校正问题』。

有关创建使用分布式数据库的应用程序的详情，参考 *Application Development Guide* 和 *CLI Guide and Reference*。

在一个事务中使用单个数据库

事务的最简单形式是在单个工作单元中只对一个数据库读取和写入。此类数据库存取称为远程工作单元。

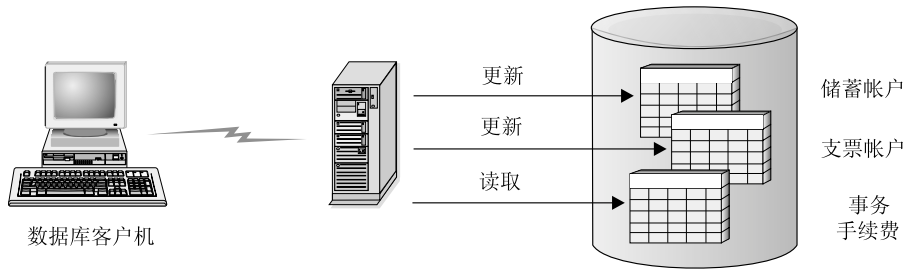


图 38. 在一个事务中使用单个数据库

图38显示了一个运行转帐应用程序的数据库客户机，该应用程序存取包含支票帐户和储蓄帐户表以及银行手续费表的数据库。该应用程序必须：

- 接受从用户界面指定的转帐金额
- 从储蓄帐户中减去该金额，然后确定新的余额
- 读取手续费表，以确定含有给定余额的储蓄帐户的事务手续费
- 从储蓄帐户中减去事务手续费
- 将转帐的金额添加至支票帐户
- 落实该事务（工作单元）。

要设置这样的应用程序，必须：

1. 在同一个数据库中为储蓄帐户、支票帐户和银行业务手续费表创建表（参见管理指南：实现中的“实现设计”）
2. 若实际位置相距较远，则设置数据库服务器以使用适当的通信协议，如安装和配置补遗所述
3. 若实际位置相距较远，则编目节点和数据库，以便在数据库服务器上标识该数据库，如快速入门书籍所述
4. 预编译您的应用程序以指定类型 1 连接；即在 `PRECOMPILE PROGRAM` 命令上指定 `CONNECT 1`（缺省值），如 *Application Development Guide* 所述。

在单个事务中使用多个数据库

当在单个事务中使用多个数据库时，对设置和管理环境的要求是不同的，这取决于在该事务中要更新的数据库的数量。

更新单个数据库

若您的数据分布在多个数据库上，您可能希望在从一个或多个数据库读取的同时更新一个数据库。此类存取称为多站点更新或两阶段落实，它可以在单个工作单

元（事务）中执行。有关多站点更新的另一个示例，参见第142页的『更新多个数据库』。

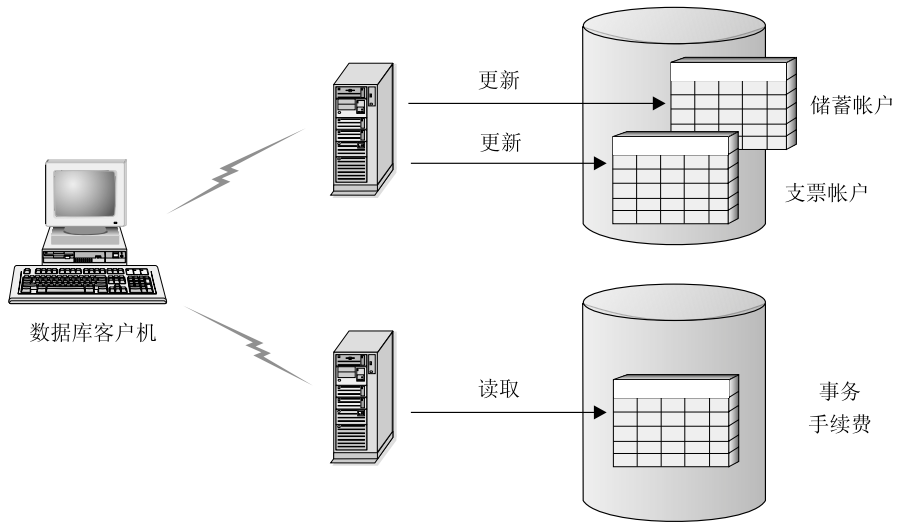


图 39. 在单个事务中使用多个数据库

图39显示了一个运行转帐应用程序的数据库客户机，该应用程序存取两个数据库服务器：一个包含支票和储蓄帐户，另一个包含银行业务手续费表。除了数据库数量和表的位置不同之外，此示例与第140页的图38中显示的示例相似。

要为此环境设置转帐应用程序，必须：

1. 在适当的数据库中创建必需的表（参见管理指南：实现中的“实现设计”
2. 若实际位置相距较远，则设置数据库服务器以使用适当的通信协议，如安装和配置补遗所述
3. 若实际位置相距较远，则编目节点和数据库，以便在数据库服务器上标识该数据库，如快速入门书籍所述
4. 预编译您的应用程序以指定类型 2 连接（即在 PRECOMPILE PROGRAM 命令上指定 CONNECT 2）和一阶段落实（即在 PRECOMPILE PROGRAM 命令上指定 SYNCPOINT ONEPHASE），如 *Application Development Guide* 所述。

若数据库位于主机或 AS/400 数据库服务器上，则需要 DB2 Connect 来与这些服务器连接。有关设置的详情，参考其中一本 DB2 Connect 快速入门书籍。有关使用 DB2 Connect 的详情，参考 *DB2 Connect 用户指南*。

更新多个数据库

若您的数据分布在多个数据库上，则您可能想在单个事务中读取和更新多个数据库。此类数据库存取称为多站点更新。

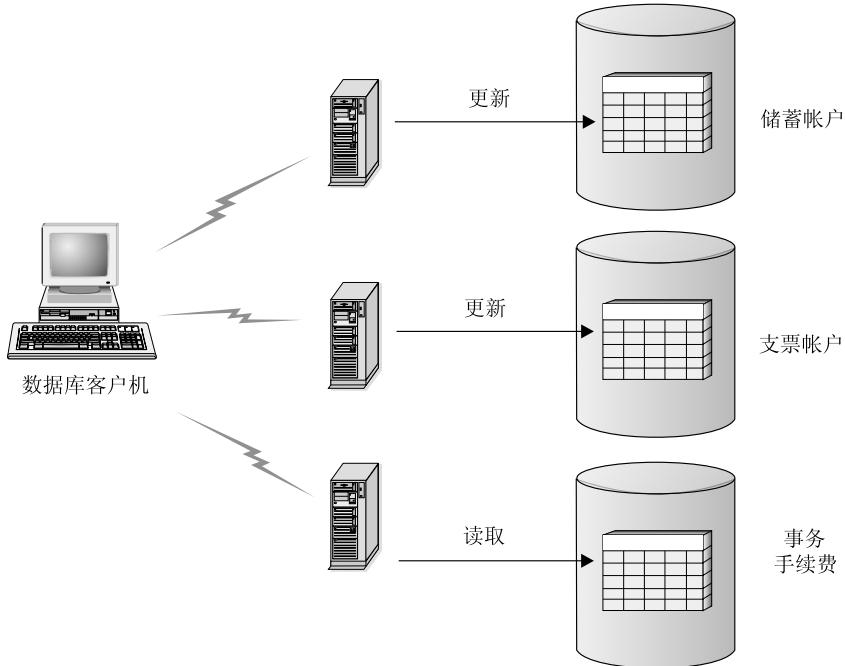


图 40. 在单个事务中更新多个数据库

图40显示了一个运行转帐应用程序的数据库客户机，该应用程序存取三个数据库服务器：一个包含支票帐户，另一个包含储蓄帐户，第三个包含银行业务手续费表。

要为此环境设置转帐应用程序，必须：

1. 在适当的数据库中创建必需的表（参见管理指南：实现中的“实现设计”
2. 若实际位置相距较远，则设置数据库服务器以使用适当的通信协议，如安装和配置补遗所述
3. 若实际位置相距较远，则编目节点和数据库，以便在数据库服务器上标识该数据库，如快速入门书籍所述

4. 预编译您的应用程序以指定类型 2 连接（即在 PRECOMPILE PROGRAM 命令上指定 CONNECT 2）和一阶段落实（即在 PRECOMPILE PROGRAM 命令上指定 SYNCPOINT ONEPHASE），如 *Application Development Guide* 所述。
5. 配置 DB2 事务管理器 (TM)，如『使用 DB2 事务管理器』中所述

使用 DB2 事务管理器

数据库管理程序提供事务管理器功能，这些功能可用于协调在单个工作单元内几个数据库的更新。数据库客户机自动协调工作单元，并使用事务管理器数据库来注册每个事务并跟踪其完成状态。

若您使用的是 XA 兼容的事务管理器，如 IBM TXSeries、BEA Tuxedo 或“Microsoft 事务管理器”，参见第153页的『第10章 针对事务管理器的设计』以获取完整的说明。

当使用用于基于 UNIX 的操作系统、Windows 操作系统或 OS/2 的 DB2 UDB 来协调事务时，必须满足特定配置需求。若您只使用 TCP/IP 来进行通信，且 DB2 UDB 和 DB2 OS/390 版是参与该事务的唯一数据库服务器，则配置就会简化。

DB2 UDB 和 DB2 OS/390 版使用 TCP/IP 连通性： 若以下所有陈述都适合于您的环境，则多站点更新的配置步骤就会简化。

- 与远程数据库服务器（包括 DB2 UDB OS/390 版）的所有通信都只使用 TCP/IP。
- 用于基于 UNIX 的系统、Windows 操作系统、OS/2 或 OS/390 的 DB2 UDB 是参与该事务的唯一数据库服务器。
- 未配置 DB2 Connect 同步点管理程序 (SPM)。

DB2 Connect 同步点管理程序是在创建 DB2 实例时自动配置的，它在下列情况下是必需的：

- 主机或 AS/400 数据库服务器使用 SNA 连通性，来进行多站点更新。
- XA 兼容的事务管理器（如 IBM TXSeries CICS）正在协调两阶段落实。

这适用于主机或 AS/400 数据库服务器使用 SNA 和 TCP/IP 连通性的情况。有关详情，参见第153页的『第10章 针对事务管理器的设计』。若您的环境不需要 DB2 Connect 同步点管理程序，则可以通过在 DB2 Connect 服务器上发出 `db2 update dbm cfg using spm_name NULL` 命令来将其关闭。然后停止并重新启动 DB2。

将用作事务管理器数据库的数据库由数据库客户机的数据库管理程序配置参数 `tm_database` 确定。有关此配置参数的详情，参见管理指南：性能中的“配置 DB2”。当设置此配置参数时，考虑下列因素：

- 事务管理器数据库可以是:
 - 用于基于 UNIX 的系统、Windows 操作系统或 OS/2 的 DB2 UDB 的数据库
 - DB2 OS/390 版的版本 5 或更高版本的数据库。

这是建议用作事务管理器数据库的数据库服务器。OS/390 系统一般比工作站服务器安全，降低了意外断电、重新引导等的可能性。因此，在再同步时使用的恢复日志更安全。

- 若对 *tm_database* 配置参数指定了值 1ST_CONN，则与应用程序连接的第一个数据库被用作事务管理器数据库。

当使用 1ST_CONN 时，必须小心。仅当能够比较容易地确保所有参与的数据库都正确编目时，才应使用此配置，即，如果：

- 启动该事务的数据库客户机位于包含参与数据库（包括事务管理器数据库）的同一个实例中。
- 您正在使用 DCE 目录服务来编目和管理对数据库的存取。

注意若您的应用程序试图与正用作事务管理器数据库的数据库断开，将接收到警告信息，且该连接将被挂起，直到该工作单元落实为止。

其他环境： 在您的环境中，如果：

- 与远程数据库服务器的通信不只使用 TCP/IP（例如，还使用 NETBIOS）
- 存取 DB2 MVS 版的版本 3 或版本 4、DB2 AS/400 版或 DB2 VM 版和 VSE 版
- 使用 SNA 存取 DB2 OS/390 版
- 使用 DB2 Connect 同步点管理程序存取主机或 AS/400 数据库服务器

多站点更新的配置步骤更为复杂。

将用作事务管理器数据库的数据库由数据库客户机的数据库管理程序配置参数 *tm_database* 确定。有关此配置参数的详情，参见管理指南：性能中的“配置 DB2”。当设置此配置参数时，考虑下列因素：

- 事务管理器数据库可以是用于基于 UNIX 的系统、Windows 操作系统或 OS/2 的 DB2 UDB 的数据库。
- 若对 *tm_database* 配置参数指定了值 1ST_CONN，则与应用程序连接的第一个数据库被用作事务管理器数据库。

当使用 1ST_CONN 时，必须小心。仅当能够比较容易地确保所有参与的数据库都正确编目时，才应使用此配置，即，如果：

- 启动该事务的数据库客户机位于包含参与数据库（包括事务管理器数据库）的同一个实例中。

- 您正在使用 DCE 目录服务来编目和管理对数据库的存取。

注意，若您的应用程序试图与正用作事务管理器数据库的数据库断开，则将接收到警告信息，且该连接将被挂起，直到该工作单元落实为止。

其他配置考虑事项

当您设置环境时，应该考虑下列配置参数。有关设置这些参数的其他信息，参考 *DB2 Connect 用户指南*。

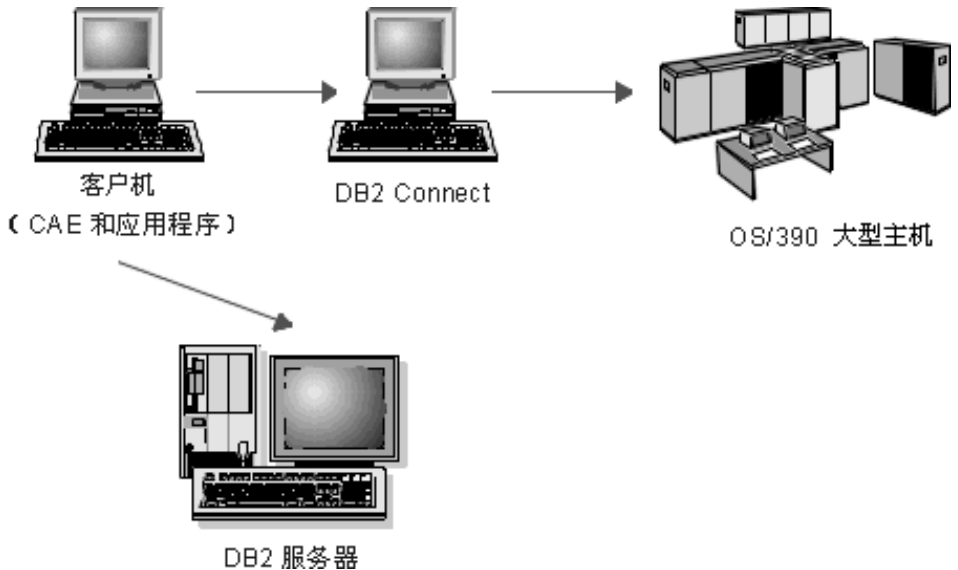


图 41. 配置考虑事项

数据库管理程序配置参数

- *tm_database*
此参数标识每个 DB2 实例的“事务管理器”(TM)数据库的名称。
- *spm_name*
此参数向数据库管理程序标识 DB2 Connect 同步点管理程序实例的名称。为了使再同步成功，该名称必须在您的网络中是唯一的。
- *resync_interval*
此参数标识一个时间间隔（以秒计），在该时间间隔之后，“DB2 事务管理器”、DB2 服务器数据库管理程序以及 DB2 Connect 同步点管理程序或 DB2 UDB 同步点管理程序应该重新尝试恢复任何未完成的不确定事务。
- *spm_log_file_sz*

此参数指定 SPM 日志文件的大小（以 4 KB 页计）。

- *spm_max_resync*

此参数标识可同时执行再同步操作的代理程序数。

- *spm_log_path*

此参数标识 SPM 日志文件的日志路径。

数据库配置参数

- *maxappls*

此参数指定允许的最大活动应用程序数。它的值必须等于或大于连接的应用程序数，加上可能正在同时完成一个两阶段落实或回滚的相同应用程序的数目，加上预期任何时候可能存在的不确定事务的数目之和。有关不确定事务的详情，参见第150页的『在两阶段落实期间校正问题』。

- *autorestart*

此数据库配置参数指定在需要时是否自动调用 RESTART DATABASE 例程。缺省值是 YES（即启用）。

包含不确定事务的数据库需要执行重新启动数据库操作来启动。若未启用 *autorestart*，则当卸下与该数据库的最后一个连接时，下一个连接将失败并需要显式的 RESTART DATABASE 调用。这种情况将一直存在，直到事务管理器的再同步操作或管理员启动的试探性操作除去了不确定事务为止。当发出 RESTART DATABASE 命令时，若该数据库中有任何不确定事务，则将返回一条信息。管理员就可以使用 LIST INDOUBT TRANSACTIONS 命令和其他的命令行处理器命令来获取有关那些不确定事务的信息。

有关这些配置参数的详情，参考管理指南：性能。

在多站点更新中存取基于 LAN 的 DB2 通用数据库服务器的主机或 AS/400 应用程序

“DB2 通用数据库”不支持使用 TCP/IP 连通性从主机或 AS/400 数据库客户机进行多站点更新。在此情况下，只支持 SNA（系统网络体系结构）连通性。多站点更新需要 DB2 同步点管理程序。在此方案中，不使用 DB2 Connect。

从主机或 AS/400 数据库客户机存取的数据服务器不必在带有 DB2 同步点管理程序的本地工作站上。主机或 AS/400 数据库客户机可以将 DB2 同步点管理程序工作站用作一个中间网关来与 DB2 UDB 服务器连接。这允许您将 DB2 同步点管理程序工作站隔离在一个安全的环境中，而实际的 DB2 UDB 服务器位于您公司网络中的远程位置。这也允许 DB2 公共服务器版本 2 数据库参与从主机或 AS/400 数据库客户机发出的多站点更新。

具体步骤如下所示：

- 在主机或 AS/400 应用程序将要直接存取的工作站上:
 1. 安装“DB2 通用数据库企业版”或“DB2 通用数据库扩充企业版”，以便向主机或 AS/400 数据库客户机提供多站点更新支持。
 2. 在同一个系统上创建数据库实例。例如，可以使用缺省实例 DB2，或使用如下命令来创建新实例：

```
db2icrt myinstance
```
 3. 按要求提供特许信息。
 4. 确保注册表的值 DB2COMM 包括值 APPC。
 5. 按要求配置 SNA 通信。当使用受支持的 IBM SNA 产品时，将根据 *spm_name* 数据库管理程序配置参数的值自动创建 DB2 同步点管理程序所需的 SNA 简要表。任何其他受支持的 SNA 堆栈都需要人工配置。有关详情，参考 *安装和配置补遗*。
 6. 确定要对 *spm_name* 数据库管理程序配置参数指定的值。此参数是在创建 DB2 实例时预先配置的，并且是从机器的 TCP/IP 主机名派生的。若它在您的环境中可接受且是唯一的，则不要更改它。
 7. 必要时，使用 UPDATE DATABASE MANAGER CONFIGURATION 命令来在“DB2 通用数据库”服务器上更新 *spm_name*。
 8. 配置此 DB2 工作站连接远程 DB2 UDB 服务器所需的通信（若有的话）。
 9. 配置远程 DB2 UDB 服务器连接此 DB2 服务器所需的通信。
 10. 当首次启动 SPM 时，停止“DB2 通用数据库”服务器上的数据库管理程序，再重新启动它。

应该能够从此 DB2 UDB 工作站与远程 DB2 UDB 服务器连接。
- 在主机或 AS/400 数据库客户机将要存取的每个远程 DB2 UDB 服务器上:
 1. 配置带有 DB2 同步点管理程序的远程 DB2 UDB 工作站与此 DB2 UDB 服务器连接所需的通信。
 2. 停止并重新启动数据库管理程序。

了解两阶段落实过程

第148页的图42举例说明了多站点更新包括的步骤。若两阶段落实过程期间发生错误，则了解一个事务是如何管理的将有助于您解决问题。

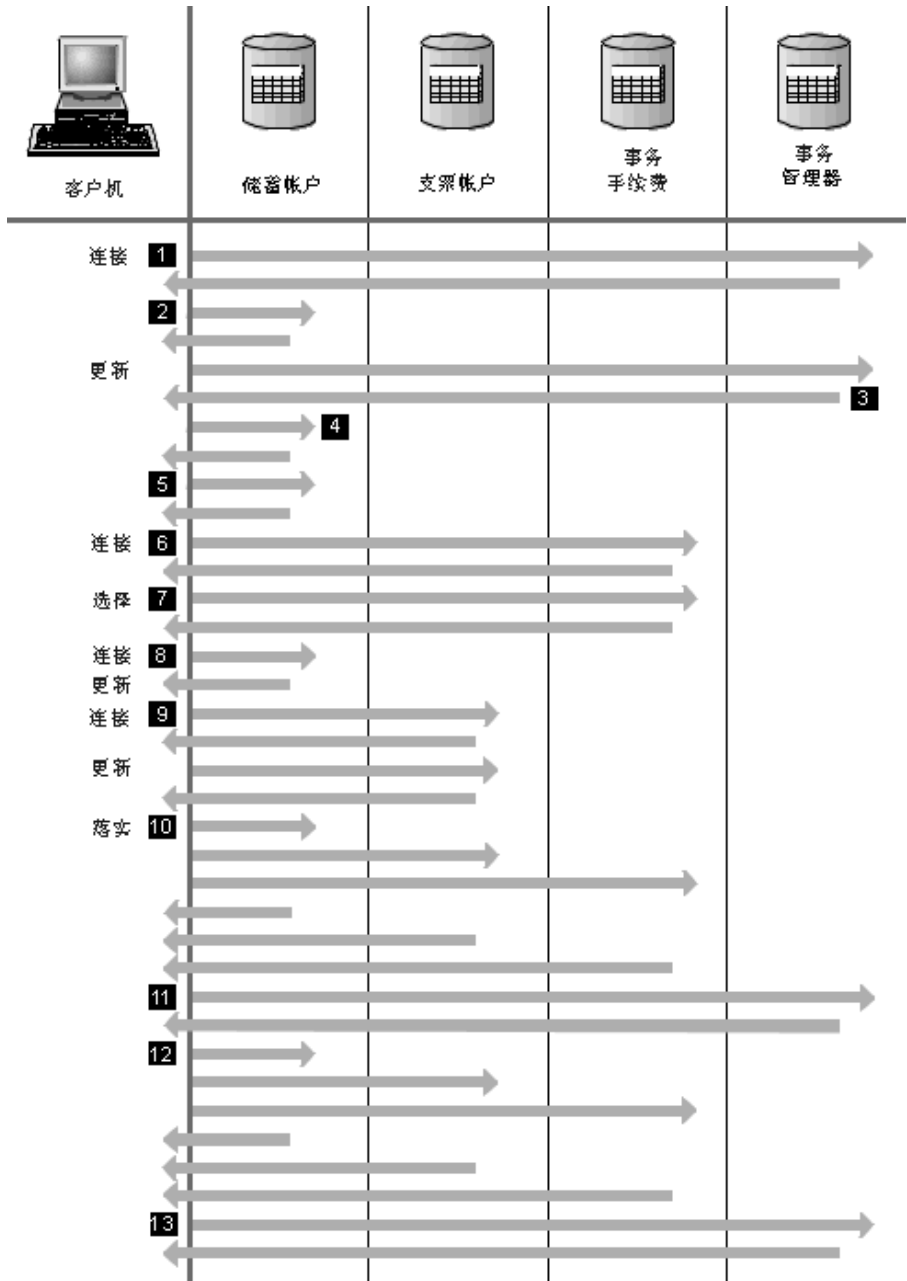


图 42. 更新多个数据库

0 为两阶段落实准备该应用程序。此过程可以通过预编译选项来完成（有关

详情，参考 *Application Development Guide*)。此过程也可以通过 DB2 CLI (调用层接口) 配置来完成 (有关详情，参考 *CLI Guide and Reference*)。

- 1** 当数据库客户机要与 SAVINGS_DB 数据库连接时，它首先在内部与事务管理器 (TM) 数据库连接。该 TM 数据库将确认返回至数据库客户机。若将数据库管理程序配置参数 `tm_database` 设置为 `1ST_CONN`，则 SAVINGS_DB 在此应用程序实例期间成为事务管理器数据库。
- 2** 建立与 SAVINGS_DB 数据库的连接，并确认该连接。
- 3** 数据库客户机开始更新 SAVINGS_ACCOUNT 表。这开始工作单元。TM 数据库提供该工作单元的一个事务 ID，以应答数据库客户机。注意，当运行该工作单元中的第一条 SQL 语句时，而不是在建立连接期间，要对该工作单元进行注册。
- 4** 当接收到该事务 ID 之后，数据库客户机向包含 SAVINGS_ACCOUNT 表的数据库注册该工作单元。将一个应答发送回客户机，以指示该工作单元已成功注册。
- 5** 以正常方式处理对 SAVINGS_DB 数据库发出的 SQL 语句。当使用在一个程序中嵌入的 SQL 语句时，对每条语句的应答都会在 SQLCA 中返回。(在 *Application Development Guide* 和 *SQL Reference* 中描述了 SQLCA。)
- 6** 当在工作单元内第一次存取该数据库期间，在包含 TRANSACTION_FEE 表的该 FEE_DB 数据库中注册该事务 ID。
- 7** 以正常方式处理对 FEE_DB 数据库发出的任何 SQL 语句。
- 8** 可以适当设置该连接，来对 SAVINGS_DB 数据库运行附加的 SQL 语句。由于已向 SAVINGS_DB 数据库注册了该工作单元 **4**，因此数据库客户机不必再次执行注册步骤。
- 9** 连接 CHECKING_DB 数据库，并遵循 **6** 和 **7** 中描述的规则来使用它。
- 10** 当数据库客户机请求落实工作单元时，会将一条准备信息发送至参与该工作单元的所有数据库。每个数据库都将一个 "PREPARED" 记录写至它的日志文件，并答复数据库客户机。
- 11** 在数据库客户机接收到来自所有数据库的肯定应答之后，它会向事务管理器数据库发送一条信息，以通知数据库现在已准备落实工作单元 (PREPARED)。该事务管理器数据库则将一个 "PREPARED" 记录写入它的日志文件，并发送答复，以通知该客户机可以启动落实过程的第二阶段。
- 12** 在落实过程的第二阶段期间，数据库客户机会向所有参与数据库发送一条信息，告知它们要落实。每个数据库则将一个 "COMMITTED" 记录写入它

的日志文件，并释放为此工作单元挂起的锁定。当该数据库完成对更改的落实时，它会将一个答复发送至客户机。

- 13** 在数据库客户机接收到来自所有参与数据库的肯定应答之后，它会向事务管理器数据库发送一条信息，以通知数据库该工作单元已完成。于是，事务管理器数据库将一个“COMMITTED”记录写至其日志文件，指示该工作单元已完成，并答复该客户机，指示它已完成。

在两阶段落实期间校正问题

从错误状态恢复是一个与应用程序设计、系统管理、数据库管理和系统操作相关的正常任务。将数据库分布在几个远程服务器上会增加由于网络或通信故障而导致错误的可能性。为确保数据完整性，数据库管理程序提供了两阶段落实过程，这在第147页的『了解两阶段落实过程』中已有说明（第 **10**、**11** 和 **12** 点）。下面解释数据库管理程序如何在两阶段落实过程期间处理错误：

• 第一阶段的错误

若一个数据库应答“它准备落实工作单元失败”，则数据库客户机将在落实过程的第二阶段期间回滚该工作单元。在这种情况下，不会向事务管理器数据库发送准备信息。

在第二个阶段期间，客户机会向所有在第一阶段成功地准备了落实的参与数据库发送一条回滚信息。于是，每个数据库将一个“ABORT”记录写入它的日志文件，并释放为此工作单元挂起的锁定。

• 第二阶段的错误

此阶段中的错误处理取决于第二阶段将落实还是回滚该事务。若第一阶段遇到错误，则第二阶段将只回滚该事务。

若其中一个参与数据库落实工作单元失败（可能由于通信故障），则事务管理器数据库将对失败的数据库重试落实。然而，将通过 SQLCA 通知应用程序落实成功。DB2 将确保落实数据库服务器中尚未落实的事务。数据库管理程序配置参数 *resync_interval*（参见管理指南：性能中的“配置 DB2”）用于指定事务管理器数据库在试图落实工作单元期间要等待的时间。在数据库服务器上挂起所有锁定，直到落实工作单元为止。

若事务管理器数据库失败，它将在重新启动数据库时使该工作单元再同步。再同步过程将试图完成所有不确定事务；即，已完成第一阶段但未完成第二阶段落实过程的那些事务。与事务管理器数据库相关联的数据库管理程序通过下列各项执行再同步：

1. 与指示在落实过程的第一阶段期间“已准备”（PREPARED）落实的数据库连接。

2. 试图落实那些数据库中的不确定事务。（若找不到不确定事务，则数据库管理程序假定在落实过程的第二阶段期间该数据库已成功落实该事务。）
3. 当在参与数据库中落实了所有不确定事务之后，再落实事务管理器数据库中的不确定事务。

若其中一个参与数据库失败并重新启动，则此数据库的数据库管理程序将查询事务管理器数据库，以获知此事务的状态，来确定是否应该回滚该事务。若在日志中找不到该事务，则数据库管理程序假定该事务已被回滚，且该事务将回滚此数据库中的不确定事务。否则，该数据库等待事务管理器数据库发出落实请求。

若该事务已由事务处理监控程序（XA 兼容的事务管理器）协调，则数据库将始终依靠 TP 监控程序来启动该再同步。

若由于某种原因您不能等待事务管理器自动解决不确定的事务，则您可以执行一些操作，以人工方式解决它们。此人工过程有时称为“作试探性决定”。有关人工恢复不确定事务的详情，参见第163页的『作试探性决定』。

若 **AUTORESTART=OFF**，则再同步不确定事务

有关“DB2 通用数据库”两阶段落实环境中的配置考虑事项，参见第145页的『其他配置考虑事项』。

尤其是，若 *autorestart* 数据库配置参数设置为 OFF，且 TM 或 RM 数据库中存在不确定事务，则需要执行 **RESTART DATABASE** 命令，以启动再同步过程。当从命令行处理器发出 **RESTART DATABASE** 命令时，将使用不同的对话。若您从同一个对话中重新启动另一个数据库，则先前重新启动数据库命令所建立的连接将被卸下，并且必须再次重新启动。在 **LIST INDOUBT TRANSACTIONS** 命令不再返回更多不确定事务之后，发出 **TERMINATE** 命令来卸下该连接。

第10章 针对事务管理器的设计

若有除 DB2 数据库以外的资源，且希望它参与两阶段落实事务，您可能要在 XA 兼容的事务管理器中使用您的数据库。若您的事务只存取 DB2 数据库，则应该使用 DB2 事务管理器，如第142页的『更新多个数据库』所述。

下列主题可以辅助您将数据库管理程序与 XA 兼容的事务管理器（如 IBM TXSeries CICS、IBM TXSeries Encina、BEA Tuxedo 或“Microsoft 事务服务器”）结合使用：

- 第154页的『X/Open 分布式事务处理模型』
- 第157页的『将数据库设置为资源管理程序』
- 第157页的『xa_open 和 xa_close 字符串的用法』
- 第157页的『DB2 版本 7 的新 xa_open 字符串格式』
- 第159页的『TPM 和 TP_MON_NAME 值』
- 第161页的『较早版本的 DB2 的 xa_open 字符串格式』
- 第162页的『更新主机或 AS/400 数据库服务器』
- 第162页的『数据库连接考虑事项』
- 第163页的『作试探性决定』
- 第165页的『安全性考虑事项』
- 第166页的『配置考虑事项』
- 第166页的『受支持的 XA 函数』
- 第168页的『XA 接口的问题确定』
- 第169页的『配置 XA 事务管理器以使用 DB2 UDB』
- 第174页的『配置 Microsoft 事务服务器』。

若您使用的是 XA 兼容的事务管理器，或正在实现该程序，请从我们的技术支持 Web 站点获取更多信息：

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

到达那里之后，选择“DB2 Universal Database”，然后使用关键字“XA”搜索该 web 站点，以获取有关 XA 兼容的事务管理器的最新可用信息。

X/Open 分布式事务处理模型

“X/Open 分布式事务处理” (DTP) 模型包括三个相关的部件:

- 『应用程序 (AP)』
- 第155页的『事务管理器 (TM)』
- 第156页的『资源管理程序 (RM)』。

图43举例说明了此模型，并显示了这些部件之间的关系。

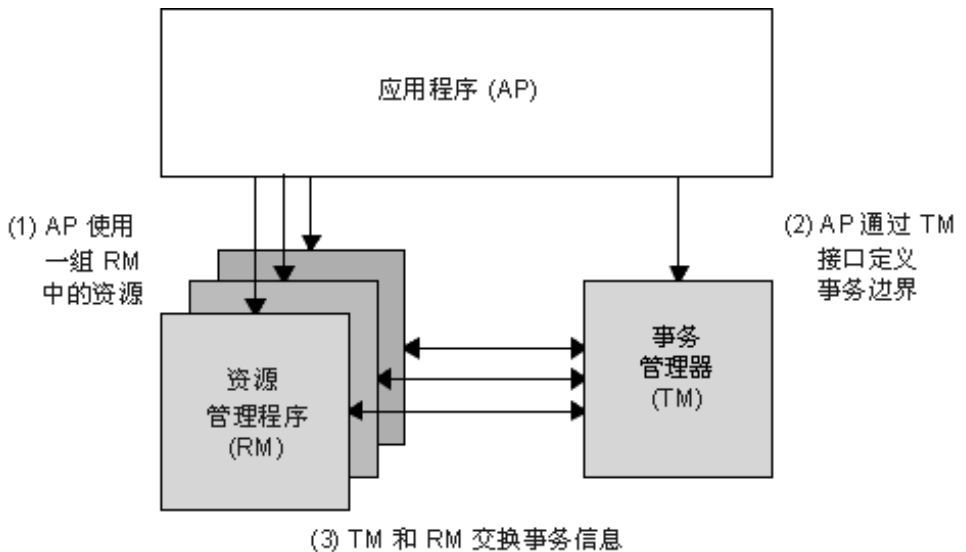


图 43. X/Open 分布式事务处理 (DTP) 模型

应用程序 (AP)

应用程序 (AP) 定义事务边界，并定义构成该事务的应用程序特定的操作。

例如，CICS* 应用程序也许要存取资源管理程序 (RM)，如数据库和“CICS 临时数据队列”，并使用程序设计逻辑来操纵数据。通过针对该资源管理程序的函数调用，将每个存取请求传送至适当的 RM。对于 DB2，这些可能是由 DB2 预编译程序为每条 SQL 语句生成的函数调用，或是直接由程序员使用该 API 来编码的数据库调用。

事务管理器 (TM) 产品通常包括用于运行用户应用程序的事务处理 (TP) 监控程序。TP 监控程序提供 API 以允许应用程序启动和结束事务，在希望运行该应用程序的多个用户之间执行应用程序的调度和负荷平衡。分布式事务处理 (DTP) 环境中的应用程序实际上是用户应用程序和 TP 监控程序的组合。

要实现有效的联机事务处理 (OLTP) 环境, TP 监控程序会在启动时预先分配大量服务器进程, 然后在多个用户事务之间调度和再利用它们。这样通过允许使用更少的服务器进程及其对应的 RM 进程来支持更多的并行用户, 节省了系统资源。再利用这些进程还避免了为每个用户的事务或程序在 TM 和 RM 中启动一个进程的额外开销。(一个程序调用一个或多个事务。) 这还意味着这些服务器进程对于 TM 和 RM 是真正的“用户进程”。它隐含安全性管理和应用程序设计。有关详情, 参见第165页的『安全性考虑事项』。

从 TP 监控程序可进行下列类型的事务:

- 非 XA 事务

这些事务涉及由于未定义给 TM 而未在 TM 的两阶段落实协议下协调的 RM。若该应用程序需要存取不支持 XA 接口的 RM, 这可能是需要的。TP 监控程序只是提供有效的应用程序调度和负荷平衡。由于 TM 不明确“打开”RM 来进行 XA 处理, 所以 RM 将此应用程序作为在非 DTP 环境中运行的任何其他应用程序来处理。

- 全局事务

这些事务涉及定义给 TM 并在 TM 的两阶段落实协议下的 RM。全局事务是可能涉及一个或多个 RM 的工作单元。事务分支是 TM 和 RM 之间支持全局事务的工作的一部分。当通过一个或多个由 TM 协调的应用程序进程来存取多个 RM 时, 一个全局事务可能有多个事务分支。

当多个应用程序进程中的每一个都存取 RM 时, 它们就象在单独的全局事务中一样, 存在松散耦合的全局事务, 但是那些应用程序处于 TM 的协调下。每个应用程序进程在一个 RM 内将有它自己的事务分支。当 AP、TM 或 RM 中任何一个请求落实或回滚时, 会一起完成事务分支。该应用程序负责确保在这些分支之间不发生资源死锁。(注意, DB2 事务管理器为使用 SYNCPOINT(TWOPHASE) 选项准备的应用程序执行的事务协调大致等效于这些松散耦合的全局事务。参见第142页的『更新多个数据库』。

当多个应用程序进程轮流在一个 RM 中的相同事务分支下工作时, 就会存在紧密耦合的全局事务。对于 RM, 这两个应用程序进程是一个单独的实体。RM 必须确保在事务分支内不发生资源死锁。

事务管理器 (TM)

事务管理器 (TM) 向事务指定标识符, 监控它们的进程, 并负责处理事务的完成和失败。事务分支标识符 (称为 XID) 由 TM 指定, 以标识一个 RM 内的全局事务和特定分支。它是一个 TM 中的日志与一个 RM 中的日志之间的关联标记。两阶段落实或回滚需要 XID, 以便在系统启动时执行再同步操作 (也称为再同步 (resync)), 或在需要时允许管理员执行试探操作 (也称为人工干预)。

在 TP 监控程序启动后，它请求 TM 打开一组应用服务器已定义的所有 RM。TM 将 **xa_open** 调用传送至 RM，以便可以将它们初始化以进行 DTP 处理。作为此启动过程的一部分，TM 执行再同步以恢复所有的不确定事务。不确定事务是处于未确定状态的全局事务。当成功完成两阶段落实协议的第一阶段（即，准备阶段）之后 TM（或至少一个 RM）成为不可用时，就会出现此情况。在 TM 可以使自己的日志与再次变成可用的 RM 的日志相符合之前，RM 不会知道要落实还是回滚它的事务分支。要执行再同步操作，TM 会一次或多次地将 **xa_recover** 调用发出至每个 RM，以标识所有不确定事务。TM 将这些回答与它自己的日志中的信息作比较，以确定应该通知 RM **xa_commit** 那些事务，还是 **xa_rollback** 那些事务。若通过管理员执行试探操作，RM 已经落实或回滚了它的不确定事务的分支，则 TM 向该 RM 发出 **xa_forget** 调用来完成再同步操作。

当用户应用程序请求落实或回滚时，它必须使用 TP 监控程序或 TM 提供的 API，以便 TM 可以在所有参与的 RM 之间协调落实和回滚。例如，当 CICS 应用程序发出 CICS SYNCPOINT 请求以落实一个事务时，CICS XA TM（在“Encina 服务器”中实现）将发出 XA 调用，如 **xa_end**、**xa_prepare**、**xa_commit** 或 **xa_rollback**，以请求 RM 落实或回滚该事务。若只涉及一个 RM，或者若 RM 回答它的分支是只读的，则 TM 可选择使用单阶段落实以替代两阶段落实。

资源管理程序 (RM)

资源管理程序 (RM) 提供对共享资源（如数据库）的存取。

作为数据库的资源管理程序的 DB2 可以参与由 XA 兼容的 TM 协调的全局事务。根据 XA 接口的要求，数据库管理程序提供 **xa_switch_t** 类型的 **db2xa_switch** 外部 C 变量，以将 XA 开关结构返回给 TM。此数据结构包含由该 TM 调用的各种 XA 例程的地址和 RM 的操作特征。有关数据库管理程序支持的 XA 函数的详情，参见第166页的『受支持的 XA 函数』。

RM 可使用两种方法来对它参与每个全局事务进行注册：*静态注册*和*动态注册*：

- 静态注册要求 TM（为每个事务）将 **xa_start**、**xa_end** 和 **xa_prepare** 等一系列调用发出至为服务器应用程序定义的所有 RM，而不管给定的 RM 是否正由该事务使用。若并非每个事务都包括每个 RM，则这是低效的，并且低效的程度与已定义的 RM 的数目成比例。
- 动态注册（由 DB2 使用）灵活且高效。仅当 RM 接收到对其资源的请求时，该 RM 才使用 **ax_reg** 调用向 TM 注册。注意，即使只定义了一个 RM 时，或当每个事务都使用每个 RM 时，此方法也不存在性能方面的缺点，因为 **ax_reg** 和 **xa_start** 调用在 TM 中具有相似的路径。

XA 接口提供 TM 和 RM 之间的双向通信。它是两个 DTP 软件部件之间的系统级接口，而不是应用程序开发人员编码的普通应用程序接口。但是，应用程序开发人员应该熟悉该 DTP 软件组件所利用的程序设计限制。有关 X/Open XA 接口程序设计考虑事项的详情，参考 *Application Development Guide*。

虽然 XA 接口是不变的，但是每个 XA 兼容的 TM 可能具有集成 RM 的产品特定方式。有关将 DB2 产品作为资源管理程序与特定的事务管理器集成的详情，参见适当的 TM 产品文档。第169页的『配置 XA 事务管理器以使用 DB2 UDB』中提供了有关最流行的 TP 监控程序的综合信息。

将数据库设置为资源管理程序

将每个数据库定义为事务管理器 (TM) 的一个单独的资源管理程序 (RM)，且必须用 `xa_open` 字符串标识该数据库。有关 DB2 的 `xa_open` 字符串格式的说明，参见『`xa_open` 和 `xa_close` 字符串的用法』。

`xa_open` 和 `xa_close` 字符串的用法

数据库管理程序 `xa_open` 字符串接受两种格式。一种格式是 DB2 版本 7 新增的。第二种格式由较早版本的 DB2 使用，为后面级兼容性而保留。新的实现应使用新格式，在可能的時候，应将旧实现迁移为新格式。DB2 的将来版本可能不支持旧的 `xa_open` 字符串格式。有关原始 `xa_open` 字符串格式的详情，参见第161页的『较早版本的 DB2 的 `xa_open` 字符串格式』。

将一个数据库设置为资源管理程序时，不需要 `xa_close` 字符串。若提供了此字符串，数据库管理程序会忽略它。

DB2 版本 7 的新 `xa_open` 字符串格式

下列 `xa_open` 字符串格式是 DB2 版本 7 新增的：

```
parm_id1 = <parm value>,parm_id2 = <parm value>, ...
```

以何次序指定这些参数并不重要。下表描述了 `parm_id` 的有效值。

表 22. `parm_id` 的有效值

参数名	值	必要的?	区别大小写?	缺省值
DB	数据库别名	是	否	无
应用程序用来存取数据库的数据库别名。				
UID	用户 ID	否	是	无
有权与数据库连接的用户 ID。若指定口令，则是必需的。				

表 22. *parm_id* 的有效值 (续)

参数名	值	必要的?	区别大小写?	缺省值
PWD	口令	否	是	无
与用户 ID 相关联的口令。若指定用户 ID, 则是必需的。				
TPM	事务处理监控程序名	否	否	无
正在使用的 TP 监控程序的名称。有关受支持的值, 参见第159页的「TPM 和 TP_MON_NAME 值」。可指定此参数以允许多个 TP 监控程序使用单一 DB2 实例。指定的值将覆盖 <i>tp_mon_name</i> 数据库管理程序配置参数中指定的值。				
AXLIB	包含 TP 监控程序的 ax_reg 和 ax_unreg 函数的库。	否	是	无
DB2 使用此值来获取必需的 ax_reg 和 ax_unreg 函数的地址。可使用它来覆盖根据 TPM 参数假定的值, 它也可以由未出现在 TPM 列表中的 TP 监控程序使用。				
CHAIN_END	<i>xa_end</i> 链接标志。有效值是 T、F, 或没有值。	否	否	F
XA_END 链接是 DB2 可用于减少网络流的优化。若 TP 监控程序环境可以保证在调用 xa_end 之后, 将立即在同一线程或进程中调用 xa_prepare , 且 CHAIN_END 为开, 则 <i>xa_end</i> 标志将与 xa_prepare 命令链接, 从而消去一个网络流。值 T 表示 CHAIN_END 为开; 值 F 表示 CHAIN_END 为关; 未指定值表示 CHAIN_END 为开。可使用此参数来覆盖从指定的 TPM 值派生的设置。				
SUSPEND_CURSOR	指定当挂起事务控制线程时是否保留游标。有效值是 T、F 或没有值。	否	否	F
挂起事务分支的 TP 监控程序可以将挂起的线程或进程重用于其他事务。在这些情况下, 必须关闭游标, 以使新事务不会继承它们。当继续执行挂起的事务时, 应用程序必须再次获取游标。若 SUSPEND_CURSOR 为开, 则不关闭任何打开的游标, 但不能将线程或进程重用于其他事务。只允许继续执行挂起的事务。值 T 表示 SUSPEND_CURSOR 为开; 值 F 表示 SUSPEND_CURSOR 为关; 未指定值表示 SUSPEND_CURSOR 为开。可使用此参数来覆盖从指定的 TPM 值派生的设置。				
HOLD_CURSOR	指定是否跨事务落实保持游标。有效值是 T、F 或没有值。	否	否	F

表 22. *parm_id* 的有效值 (续)

参数名	值	必要的?	区别大小写?	缺省值
	通常, TP 监控程序将线程或进程重用于多个应用程序。要确保新装入的应用程序不会继承由先前应用程序打开的游标, 游标在落实之后被关闭。若 HOLD_CURSOR 为开, 则跨事务落实保持游标。值 T 表示 HOLD_CURSOR 为开; 值 F 表示 HOLD_CURSOR 为关; 未指定值表示 HOLD_CURSOR 为开。可使用此参数来覆盖从指定的 TPM 值派生的设置。			

TPM 和 TP_MON_NAME 值

xa_open 字符串 TPM 参数和 *tp_mon_name* 数据库管理程序配置参数用来向 DB2 指示正在使用的 TP 监控程序。*tp_mon_name* 值适用于整个 DB2 实例。TPM 参数仅适用于特定 XA 资源管理程序。TPM 值覆盖 *tp_mon_name* 参数。TPM 和 *tp_mon_name* 参数的有效值如下所示:

表 23. TPM 和 *tp_mon_name* 的有效值

TPM 值	TP 监控程序产品	内部设置
CICS	IBM TxSeries CICS	AXLIB=libEncServer (对于 Windows) =/usr/lpp/encina/lib/libEncServer (对于基于 UNIX 的系统) HOLD_CURSOR=T CHAIN_END=T SUSPEND_CURSOR=F
ENCINA	IBM TxSeries Encina 监控程序	AXLIB=libEncServer (对于 Windows) =/usr/lpp/encina/lib/libEncServer (对于基于 UNIX 的系统) HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F
MQ	IBM MQSeries	AXLIB=mqmax (对于 Windows) =/usr/mqm/lib/libmqmax.a (对于 AIX) =/opt/mqm/lib/libmqmax.a (对于 Solaris) HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F
CB	IBM Component Broker	AXLIB=somtrx1i (对于 Windows) =libsomtrx1 (对于基于 UNIX 的系统) HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F

表 23. TPM 和 *tp_mon_name* 的有效值 (续)

TPM 值	TP 监控程序产品	内部设置
SF	IBM San Francisco	AXLIB=ibmsfDB2 HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F
TUXEDO	BEA Tuxedo	AXLIB=libtux HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F
MTS	Microsoft 事务服务器	并不一定要为 MTS 配置 DB2。DB2 的 ODBC 驱动程序自动检测 MTS。
JTA	Java 事务 API	并不一定要为“企业 Java 服务器”(EJS) (例如 IBM WebSphere) 配置 DB2。DB2 的 JDBC 驱动程序自动检测此环境。

示例

- 您正在 Windows NT 上使用 IBM TxSeries CICS。TxSeries 文档指示您需要使用值 `libEncServer:C` 来配置 *tp_mon_name*。这仍是可接受的格式；然而，对于对于 DB2 UDB 或 DB2 Connect 版本 7，您可以选择：
 - 指定值为 CICS 的 *tp_mon_name* (建议用于此方案)：


```
db2 update dbm cfg using tp_mon_name CICS
```

对于向 CICS 定义的每个数据库，在“区域”->“资源”->“产品”->XAD->“资源管理程序初始化字符串”中，指定：

```
db=dbalias,uid=userid,pwd=password
```
 - 对于向 CICS 定义的每个数据库，在“区域”->“资源”->“产品”->XAD->“资源管理程序初始化字符串”中，指定：


```
db=dbalias,uid=userid,pwd=password,tpm=cics
```
- 您正在 Windows NT 上使用 IBM MQSeries。MQSeries 文档指示您需要使用值 `mqmax` 来配置 *tp_mon_name*。这仍是可接受的格式；然而，对于对于 DB2 UDB 或 DB2 Connect 版本 7，您可以选择：
 - 指定值为 MQ 的 *tp_mon_name* (建议用于此方案)：


```
db2 update dbm cfg using tp_mon_name MQ
```

对于向 CICS 定义的每个数据库，在“区域”->“资源”->“产品”->XAD
->“资源管理程序初始化字符串”中，指定：

```
uid=userid,db=dbalias,pwd=password
```

- 对于向 CICS 定义的每个数据库，在“区域”->“资源”->“产品”->XAD
->“资源管理程序初始化字符串”中，指定：

```
uid=userid,db=dbalias,pwd=password,tpm=mq
```

3. 您正在 Windows NT 上同时使用 IBM TxSeries CICS 和 IBM MQSeries。正在使用单个 DB2 实例。在此方案中，应按如下方法进行配置：

- a. 对于向 CICS 定义的每个数据库，在“区域”->“资源”->“产品”->XAD
->“资源管理程序初始化字符串”中，指定：

```
pwd=password,uid=userid,tpm=cics,db=dbalias
```

- b. 对于在队列管理程序特性中定义成资源的每个数据库，将 XaOpenString 指定为：

```
db=dbalias,uid=userid,pwd=password,tpm=mq
```

4. 您正在 Windows NT 上开发您自己的 XA 兼容的事务管理器 (XA TM)，您想要告知 DB2 库 "myaxlib" 有必需的函数 **ax_reg** 和 **ax_unreg**。库 "myaxlib" 在 PATH 语句中指定的目录中。您可以选择：

- 指定值为 myaxlib 的 *tp_mon_name*：

```
db2 update dbm cfg using tp_mon_name myaxlib
```

并且，对于向 XA TM 定义的每个数据库，指定 xa_open 字符串：

```
db=dbalias,uid=userid,pwd=password
```

- 对于向 XA TM 定义的每个数据库，指定 xa_open 字符串：

```
db=dbalias,uid=userid,pwd=password,axlib=myaxlib
```

5. 您正在 Windows NT 上开发您自己的 XA 兼容的事务管理器 (XA TM)，您想要告知 DB2 库 "myaxlib" 有必需的函数 **ax_reg** and **ax_unreg**。库 "myaxlib" 在 PATH 语句中指定的目录中。您还想启用 XA END 链接。您可以选择：

- 对于向 XA TM 定义的每个数据库，指定 xa_open 字符串：

```
db=dbalias,uid=userid,pwd=password,axlib=myaxlib,chain_end=T
```

- 对于向 XA TM 定义的每个数据库，指定 xa_open 字符串：

```
db=dbalias,uid=userid,pwd=password,axlib=myaxlib,chain_end
```

较早版本的 DB2 的 xa_open 字符串格式

较早版本的 DB2 使用这里描述的 xa_open 字符串格式。为了进行兼容，仍支持此格式。在可能的时候，应将应用程序迁移至新格式（参见第157页的『DB2 版本 7 的新 xa_open 字符串格式』）。

将每个数据库定义为事务管理器 (TM) 的一个单独的资源管理程序 (RM)，且必须用 `xa_open` 字符串标识该数据库，该字符串的语法如下：

```
"database_alias<,userid,password>"
```

`database_alias` 用于指定数据库的别名。除非您在创建数据库之后明确地编目了一个别名，否则，别名与数据库名相同。用户名和口令是可选的（这取决于认证方法），它们将用于向数据库提供认证信息。

当将一个数据库设置为资源管理程序时，不需要 `xa_close` 字符串。若提供了此字符串，数据库管理程序会忽略它。

更新主机或 AS/400 数据库服务器

主机和 AS/400 数据库服务器可能是可以更新的，这取决于“XA 事务管理器”的体系结构。要支持来自不同进程的落实序列，必须启用 DB2 Connect 集中器。要启用 DB2 Connect EE 集中器，将数据库管理程序配置参数 `max_logicagents` 设置为大于 `maxagents` 的值。注意，DB2 Connect EE 集中器需要 DB2 版本 7.1 客户机才能支持来自不同进程的 XA 落实序列。有关此环境中允许的 SQL 语句的详情，参考 *Application Development Guide*。有关集中器的详情，参考 *DB2 Connect 用户指南*。

若您要更新主机或 AS/400 数据库服务器，将需要配置了 DB2 同步点管理程序 (SPM) 的 DB2 Connect。参考其中一本快速入门书籍以获取指导。

数据库连接考虑事项

本节包括下列主题：

- 『RELEASE 语句』
- 『存取分区数据库的事务』

RELEASE 语句

若使用 RELEASE 语句来释放与一个数据库的连接，则应使用 CONNECT 语句而不是 SET CONNECTION 来重新连接该数据库。

存取分区数据库的事务

在分区数据库环境中，可将用户数据分布在各数据库分区中。存取该数据库的应用程序与其中一个数据库分区（协调程序节点）连接，并向其发送请求。不同的应用程序可以与不同的数据库分区连接，而同一个应用程序可以为不同的连接选择不同的数据库分区。

对于对分区数据库环境中的某个数据库执行的事务，所有的存取都必须通过同一个数据库分区。即，从启动该事务起，直到落实该事务为止（包括落实时的那一刻），都必须使用同一个数据库分区。

对分区数据库执行的任何事务都必须在断开之前落实。

作试探性决定

XA 兼容的事务管理器（事务处理监控程序）使用一个两阶段落实过程，该过程类似于 DB2 事务管理器所使用的那个过程，如第147页的『了解两阶段落实过程』所述。两个环境之间的主要区别是，TP 监控程序提供记录和控制该事务的功能，而不是 DB2 事务管理器和事务管理器数据库来提供此功能。

当使用 XA 兼容的事务管理器时，可能发生类似于对 DB2 事务管理器讨论的那些错误（参见第150页的『在两阶段落实期间校正问题』）。类似于 DB2 事务管理器，XA 兼容的事务管理器将试图使不确定事务再同步。

若由于某种原因您不能等待事务管理器自动解决不确定事务，则您可以执行一些操作，以人工方式解决它们。此人工过程有时称为“作试探性决定”。

LIST INDOUBT TRANSACTIONS 命令（使用 WITH PROMPTING 选项）或相关的一组 API 允许您查询、落实和回滚不确定事务。另外，它还允许您除去运行记录并释放日志空间，以“忘记”通过试探方式落实或回滚的事务。要从在基于 UNIX 的系统、Windows 操作系统或 OS/2 上的 DB2 UDB 中获取不确定事务信息，与数据库连接，并发出 LIST INDOUBT TRANSACTIONS WITH PROMPTING 命令或等价的 API。有关此命令或相关的管理 API 的详情，参考 *Command Reference* 或 *Administrative API Reference*。

对于与主机或 AS/400 数据库服务器相关的不确定事务信息，您有两个选择：

- 可以直接从主机或 AS/400 服务器获取不确定信息。
要直接从 DB2 OS/390 版获取不确定信息，调用 DISPLAY THREAD TYPE(INDOUBT) 命令。使用 RECOVER 命令来作试探性决定。要直接从 DB2 OS/400 版获取不确定信息，调用 wrkcmdfn 命令。
- 可以从用来存取主机或 AS/400 数据库服务器的 DB2 Connect 服务器获取不确定信息。
要从 DB2 Connect 服务器获取不确定信息，首先通过连接由 spm_name 数据库管理程序配置参数的值所表示的 DB2 实例来与 DB2 同步点管理程序连接。然后发出 LIST DRDA INDOUBT TRANSACTIONS WITH PROMPTING 命令来显示不确定事务并作试探性决定。

务请极为谨慎地使用这些命令（或相关的 API），并且只将其作为最后的手段。最好的策略是等待事务管理器驱动再同步过程。若您在其中一个参与数据库中以人工方式落实或回滚一个事务，并对另一个参与的数据库执行相反的操作，可能会遇到数据完整性问题。校正数据完整性问题要求您了解该应用程序的逻辑、标识已更改或回滚的数据，然后执行数据库的时间点恢复，或以人工方式撤消或重新执行更改。

若您不能等待事务管理器启动再同步过程，且您必须释放不确定事务占用的资源，则必须执行试探性操作。若事务管理器无法在延长期内执行再同步，且该不确定事务占用着急需的资源，则可能发生这种情况。不确定事务占用着事务管理器或资源管理程序成为不可用之前与此事务相关的资源。对于数据库管理程序，这些资源包括对该事务占用的表和索引、日志空间和存储器的锁定。每个不确定事务还会递减（减一）数据库可以处理的并行事务的最大数目。

尽管没有简单明了的方法来执行试探性操作，但下面还是提供了一些一般准则：

1. 与您要完成其所有事务的数据库连接。
2. 使用 `LIST INDOUBT TRANSACTION` 命令显示不确定事务。*xid* 表示全局事务 ID，它与事务管理器和参与事务的其他资源管理程序使用的 *xid* 完全相同。
3. 对于每个不确定事务，使用您掌握的有关该应用程序和操作环境的知识，来确定其他参与的资源管理程序。
4. 确定事务管理器是否可用：
 - 若该事务管理器是可用的，且资源管理程序中的不确定事务是由于在第二个落实阶段期间或更早的再同步过程期间资源管理程序不可用导致的，则您应该检查该事务管理器的日志，以确定对其他资源管理程序执行过什么操作。然后您应该对该数据库执行相同的操作；即使用 `LIST INDOUBT TRANSACTIONS` 命令，试探性落实或试探性回滚该事务。
 - 若事务管理器不可用，您将需要使用其他参与资源管理程序中该事务的状态，以确定您应该执行什么操作：
 - 若其他资源管理程序中至少有一个已经落实该事务，则您应该在所有资源管理程序中试探性落实该事务。
 - 若其他资源管理程序中至少有一个已经回滚该事务，则您应该试探性回滚该事务。
 - 若该事务在所有参与资源管理程序中都处于“已准备好”（未确定）状态，则您应该试探性回滚该事务。
 - 若其他资源管理程序中有一个或多个不可用，则您应该试探性回滚该事务。

除非试探性落实或回滚的事务导致日志满情况（在 `LIST INDOUBT TRANSACTIONS` 命令的输出中指示），否则不要执行试探性忘记功能。试探性忘

记功能释放由不确定事务占用的日志空间。隐含情况为，若一个事务管理器最终对此不确定事务执行了再同步操作，则可能会作出错误的决定来落实或回滚其他资源管理程序，因为在此资源管理程序中没有该事务的运行记录。通常，“丢失”运行记录意味着该资源管理程序已经回滚该事务。

安全性考虑事项

TP 监控程序预分配一组服务器进程，并由在这些服务器进程 ID 下的不同用户运行事务。对于该数据库，每个服务器进程表现为一个有许多工作单元的大应用程序，所有工作单元都在与该服务器进程相关的相同 ID 下运行。

例如，在使用 CICS 的 AIX 环境中，当启动 TXSeries CICS 区域时，该区域与用于定义它的 AIX 用户名相关。所有“CICS 应用服务器”进程也都在 TXSeries CICS 主 ID 下运行，该 ID 通常被定义为“cics”。CICS 用户可以在他们的 DCE 注册 ID 下调用 CICS 事务，且在 CICS 中时，他们还可以使用 CESN 注册事务来更改他们的 ID。在任何一种情况中，RM 都无法使用最终用户的 ID。因此，一个“CICS 应用进程”可能正在代表许多用户运行事务，但是，对于 RM 来说，它就象是一个带有许多工作单元的单一程序，而这些工作单元具有相同的“cics” ID。您可以选择在 `xa_open` 字符串中指定用户 ID 和口令，并将使用该用户 ID 来代替“cics” ID 以与数据库连接。

对于静态 SQL 语句，没有很大影响，因为是使用联编程序的特权而非最终用户的特权来存取数据库。但是，这意味着必须将数据库程序包的 EXECUTE 特权授予服务器 ID，而不是最终用户 ID。

对于动态语句，它们在运行期执行存取认证，必须将数据库对象的存取特权授予服务器 ID，而非那些对象的实际用户。此时必须依靠 TP 监控程序系统来确定允许用户运行的程序，而不是依靠数据库来控制特定用户的存取。必须授予服务器 ID 它的 SQL 用户需要的所有特权。

要确定谁存取了数据库表或视图，您可以执行下列步骤：

1. 从 SYSCAT.PACKAGEDEP 目录视图，获取取决于该表或视图的所有程序包的列表。
2. 通过安装期间使用的命名约定，确定与这些程序包对应的服务器程序（例如，CICS 程序）的名称。
3. 确定可能调用这些程序的客户机程序（例如，CICS 事务 ID），然后使用 TP 监控程序的日志（例如，CICS 日志）来确定谁运行了这些事务或程序以及是何时运行的。

配置考虑事项

当您设置 TP 监控程序环境时，应该考虑下列配置参数：

- *tp_mon_name*

此数据库管理程序配置参数标识正在使用的 TP 监控程序产品的名称（例如，“CICS”或“ENCINA”）。

- *tpname*

此数据库管理程序配置参数标识远程事务程序的名称，当数据库客户机使用 APPC 通信协议向数据库服务器发出分配请求时，必须使用该事务程序。此值是在服务器的配置文件中设置的，且必须与在 SNA 事务程序中配置的事务处理器 (TP) 的名称相同。有关详情，参考快速入门手册。

- *tm_database*

因为 DB2 不协调 XA 环境中的事务，所以不对 XA 协调的事务使用此数据库管理程序配置参数。

- *maxappls*

此数据库配置参数指定允许的活动应用程序的最大数目。此参数的值必须等于或大于连接的应用程序数，加上可能正在同时完成一个两阶段落实或回滚的这些应用程序的数目之和。此和随后应加上在任何时刻可能存在的预期不确定事务数。有关不确定事务的详情，参见第150页的『在两阶段落实期间校正问题』。

对于 TP 监控程序环境（例如，TXSeries CICS），可能需要增加 *maxappls* 参数的值。这有助于确保可以处理所有 TP 监控程序进程。

- *autorestart*

此数据库配置参数指定在需要时是否自动调用 RESTART DATABASE 例程。缺省值是 YES（即，启用）。

需要一个重新启动数据库操作来启动包含不确定事务的数据库。若卸下与该数据库的最后一个连接时未启用 *autorestart*，则下一个连接将失败并需要显式的 RESTART DATABASE 调用。这种情况将一直存在，直到事务管理器的再同步操作或管理员启动的试探性操作除去了不确定事务为止。当发出 RESTART DATABASE 命令时，若该数据库中有任何不确定事务，将返回一条信息。管理员就可以使用 LIST INDOUBT TRANSACTIONS 命令和其他的命令处理程序命令来获取有关那些不确定事务的信息。

受支持的 XA 函数

“DB2 通用数据库”支持在 *X/Open CAE Specification Distributed Transaction Processing: The XA Specification* 中定义的 XA91 规范，但是下列情况除外：

- 异步服务

XA 规范允许该接口使用异步服务，这样可以在稍后的某个时间检查请求的结果。数据库管理程序要求以同步方式调用请求。

- 静态注册

XA 接口允许以两种方式来注册 RM: 静态注册和动态注册。“DB2 通用数据库”只支持动态注册，此方式更高级且更有效率。有关这两种方法的详情，参见第156页的『资源管理程序 (RM)』。

- 关联迁移

DB2 通用数据库不支持控制的线程之间的事务迁移。

有关 `xa_open` 和 `xa_close` 字符串用法的详情，参见第157页的『`xa_open` 和 `xa_close` 字符串的用法』。

XA 开关的用法和位置

根据 XA 接口的要求，数据库管理程序提供 `xa_switch_t` 类型的 `db2xa_switch` 外部 C 变量，以将 XA 开关结构返回至该 TM。不是返回各种 XA 函数的地址，而是返回下列字段：

字段	值
名称	数据库管理程序的产品名。例如，DB2 AIX 版。
标志	TMREGISTER TMNOMIGRATE 明确声明“DB2 通用数据库”使用动态注册，且 TM 不应该使用关联迁移。明确声明不支持异步操作。
版本	必须为零。

使用“DB2 通用数据库”的 XA 开关

XA 体系结构要求“资源管理程序”(RM)提供开关，该开关使“XA 事务管理器”(TM)可以存取 RM 的 `xa_` 例程。RM 开关使用称为 `xa_switch_t` 的结构。该开关包含 RM 的名称、指向 RM 的 XA 入口点的非 NULL 指针、标志和版本号。

基于 UNIX 的系统和 OS/2: DB2 UDB 的开关可以通过下面两种方法之一获得：

- 通过更高一级的间接方法。在 C 程序中，可在使用 `db2xa_switch` 之前定义以下的宏来完成：

```
#define db2xa_switch (*db2xa_switch)
```

- 通过调用 `db2xacic`

DB2 UDB 提供了此 API，它返回 `db2xa_switch` 结构的地址。此函数的原型为：

```
struct xa_switch_t * SQL_API_FN db2xacic ( )
```

无论使用哪一种方法，都必须将应用程序与 `libdb2`（在基于 UNIX 的系统上）或 `db2api.lib`（在 OS/2 上）连接。

Windows NT: 指向 `xa_switch` 结构的指针 `db2xa_switch` 被作为 DLL 数据调出。这意味着使用此结构的 Windows NT 应用程序必须以下列三种方式之一来引用它:

- 通过更高一级的间接方法。在 C 程序中, 可在使用 `db2xa_switch` 之前定义以下的宏来完成:

```
#define db2xa_switch (*db2xa_switch)
```

- 若使用 Microsoft Visual C++ 编译器, 则可以将 `db2xa_switch` 定义为:

```
extern __declspec(dllimport) struct xa_switch_t db2xa_switch
```

- 通过调用 **db2xacic**

DB2 UDB 提供了此 API, 它返回 `db2xa_switch` 结构的地址。此函数的原型为:

```
struct xa_switch_t * SQL_API_FN db2xacic( )
```

使用这些方法中任何一种, 必须将应用程序与 `db2api.lib` 连接。

C 代码示例: 下列代码举例说明在任何 DB2 UDB 平台上通过 C 程序存取 `db2xa_switch` 的不同方法。务必将应用程序与适当的库连接。

```
#include <stdio.h>
#include <xa.h>

struct xa_switch_t * SQL_API_FN db2xacic( );

#ifdef DECLSPEC_DEFN
extern __declspec(dllimport) struct xa_switch_t db2xa_switch;
#else
#define db2xa_switch (*db2xa_switch)
extern struct xa_switch_t db2xa_switch;
#endif

main( )
{
    struct xa_switch_t *foo;
    printf ( "%s \n", db2xa_switch.name );
    foo = db2xacic();
    printf ( "%s \n", foo->name );
    return ;
}
```

XA 接口的问题确定

若从 TM 发出 XA 请求期间检测到错误, 则应用程序可能无法从 TM 获取错误码。若您的程序异常终止, 或从 TP 监控程序或 TM 获得加密的返回码, 则应该检查“首次故障服务日志”, 当诊断级别 3 或更高级别生效时, 它将报告 XA 错误信息。有关“首次故障服务日志”的详情, 参考 *Troubleshooting Guide*。

您还应该参考控制台信息、TM 错误文件或关于所用的外部事务处理软件的其他产品特定信息。

数据库管理程序将所有 XA 特定错误写入“首次故障服务日志”，并使用 SQLCODE -998（事务或试探性错误）和适当的原因码。以下是一些更为常见的错误：

- xa_open 字符串中的语法无效。
- 由于下列原因之一，未能与打开字符串中指定的数据库连接：
 - 数据库未编目。
 - 数据库未启动。
 - 服务器应用程序的用户名或口令无权与该数据库连接。
- 通信错误。

下面是（由于丢失 xa_open 字符串）在 AIX 上生成的 xa_open 错误的错误日志的一个示例：

```
Tue Apr  4 15:59:08 1995
toop pid(83378) process (xatest) XA DTP Support      sqlxa_open Probe:101
DIA4701E Database "" could not be opened for distributed transaction
processing.
String Title : XA Interface SQLCA  pid(83378)
SQLCODE = -998 REASON CODE: 4  SUBCODE: 1
Dump File : /u/toop/diagnostics/83378.dmp Data : SQLCA
```

配置 XA 事务管理器以使用 DB2 UDB

注意，本节中的信息取代管理指南：性能中的类似章节。

下列几节描述如何配置特定的产品，以将 DB2 用作资源管理程序。您可以使用下列任何一种方式：

- 『配置 IBM TXSeries CICS』
- 第170页的『配置 IBM TXSeries Encina』
- 第172页的『配置 BEA Tuxedo』
- 第174页的『配置 Microsoft 事务服务器』。

配置 IBM TXSeries CICS

有关如何配置 IBM TXSeries CICS 以将 DB2 用作资源管理程序的详情，参考 *IBM TXSeries CICS Administration Guide*。TXSeries 文档可以通过访问如下站点联机查看：

http://www.transarc.com/Library/documentation/websphere/WAS-EE/en_US/html/。

主机和 AS/400 数据库服务器可以参与 CICS 协调的事务。

配置 IBM TXSeries Encina

以下是当通过 DB2 Connect 存取时，将“Encina 监控程序”与“DB2 通用数据库”服务器或 DB2 MVS 版、DB2 OS/390 版、DB2 AS/400 版或 DB2 VSE 版和 VM 版集成，所需的各种 API 和配置参数。TXSeries 文档可以通过访问如下站点联机查看：

http://www.transarc.com/Library/documentation/websphere/WAS-EE/en_US/html/。

主机和 AS/400 数据库服务器可以参与 Encina 协调的事务。

配置 DB2

要配置 DB2:

1. 必须在 DB2 数据库目录中定义每个数据库的名称。若该数据库是远程数据库，则还必须定义节点目录项。您可以使用“客户机配置辅助程序”(CCA)或 DB2 命令行处理器 (CLP)来执行该配置。例如:

```
DB2 CATALOG DATABASE inventdb AS inventdb AT NODE host1 AUTH SERVER
DB2 CATALOG TCPIP NODE host1 REMOTE hostname1 SERVER svcname1
```

2. 若 DB2 客户机知道它处理的是 Encina，则它可以针对 Encina 优化它的内部处理。您可以将 *tp_mon_name* 数据库管理程序配置参数设置为 ENCINA 来指定此选项。缺省行为是不进行特殊的优化。若设置了 *tp_mon_name*，则该应用程序必须确保执行该工作单元的线程在结束之后也立即落实该工作。不能启动任何其他工作单元。若您的环境不是这样的，则确保 *tp_mon_name* 值是 NONE (或者，通过 CLP 将此值设置为 NULL)。可以通过“控制中心”或 CLP 更新此参数。CLP 命令是:

```
db2 update dbm cfg using tp_mon_name ENCINA
```

为每个资源管理程序配置 Encina

要为每个资源管理程序 (RM) 配置 Encina，管理员必须为作为资源管理程序的每个 DB2 数据库定义“打开字符串”、“关闭字符串”和“控制线程协议”，然后才能为应用程序中的事务注册该资源管理程序。可以使用 Enconcole 全屏幕界面或 Encina 命令行界面来执行该配置。例如:

```
monadmin create rm inventdb -open "db=inventdb,uid=user1,pwd=password1"
```

每个 DB2 数据库都有一个资源管理程序配置，且每个资源管理程序配置必须具有 rm 名称 (“逻辑 RM 名称”)。为了简化情况，您应该使它与数据库名相同。

xa_open 字符串包含建立与数据库的连接所需的信息。该字符串的内容是 RM 所独有的。DB2 UDB 的 *xa_open* 字符串包含要打开的数据库的别名，以及与该连接相关的可选的用户 ID 和口令。注意，此处定义的数据库名也必须所有数据库存取

都必需的规则数据库目录中编目。有关 DB2 的 xa_open 字符串的详情，参见第157页的『将数据库设置为资源管理程序』。

DB2 不使用 xa_close 字符串。

“控制线程协议”确定一个应用程序的代理程序线程是否一次可以处理多个事务。DB2 UDB 支持 TMXA_SERIALIZE_ALL_OPERATIONS 的缺省值，并且只有当一个事务完成之后才能再次使用一个线程。

若您正存取 DB2 OS/390 版、DB2 MVS 版、DB2 AS/400 版或 DB2 VSE 版和 VM 版，则必须使用 DB2 同步点管理程序。有关配置指导，参考 *DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings* 手册。

从 Encina 应用程序中引用 DB2 数据库

要从 Encina 应用程序中引用 DB2 数据库：

1. 使用“Encina 调度策略 API”，来指定可以从单个 TP 监控程序的应用程序进程中运行多少个应用程序的代理程序。例如：

```
rc = mon_SetSchedulingPolicy (MON_EXCLUSIVE)
```

对于 DB2 (DB2 通用数据库、主机或 AS/400 数据库服务器)，您应该使用 MON_EXCLUSIVE 的缺省设置。这确保：

- 应用程序进程在事务的寿命期间被锁定。
- 应用程序以单线程方式执行。

注：若您使用的是 ODBC 或 DB2 调用层接口，则必须禁用多线程支持。为此，您可以设置 CLI 配置参数 DISABLEMULTITHREAD = 1 (禁用多线程)。DB2 通用数据库的缺省值为 DISABLEMULTITHREAD = 0 (启用多线程)。有关详情，参考 *CLI Guide and Reference*。

2. 使用“Encina RM 注册 API”来提供 XA 开关和逻辑 RM 名称，以供 Encina 在应用程序进程中引用 RM 时使用。例如：

```
rc = mon_RegisterRmi ( &db2xa_switch, /* xa 开关 */  
                      "inventdb", /* 逻辑 RM 名 */  
                      &rmiId ); /* 内部 RM ID */
```

XA 开关包含 RM 中 TM 可以调用的 XA 例程的地址，它还指定由 RM 提供的功能。DB2 通用数据库的 XA 开关是 db2xa_switch，它驻留在 DB2 客户机库 (在 Windows 操作系统和 OS/2 上，是 db2app.dll，在基于 UNIX 的系统上，是 libdb2) 中。

逻辑 RM 名是 Encina 使用的那个名称，而不是在 Encina 下运行的 SQL 应用程序使用的实际数据库名。实际数据库名是在“Encina RM 注册表”API 的 `xa_open` 字符串中指定的。逻辑 RM 名设置为与本示例中的数据库名相同。

第三个参数返回 TM 引用此连接所用的内部标识符或句柄。

注：当通过 TM-XA 接口使用 Encina 来进行 DB2 的事务处理时，注意 Encina 嵌套的事务当前不受 DB2 XA 接口的支持。如果有可能的话，请避免使用这些事务。若无法避免，则确保仅在 Encina 事务系列的一个成员中执行 SQL 工作。

配置 BEA Tuxedo

要配置 Tuxedo 以将 DB2 用作资源管理程序，须执行下列步骤：

1. 按该产品的文档中指定的步骤，安装 Tuxedo。确保执行所有基本的 Tuxedo 配置，包括日志文件和环境变量。

您还需要一个编译程序和 DB2 应用程序开发客户机。需要时安装它们。

2. 在 Tuxedo 服务器 ID 中，设置 `DB2INSTANCE` 环境变量，以引用包含您希望 Tuxedo 使用的数据库的实例。将 `PATH` 变量设置为包括 DB2 程序目录。确认 Tuxedo 服务器 ID 可以与 DB2 数据库连接。
3. 使用值 `TUXEDO` 来更新 `tp_mon_name` 数据库管理程序配置参数。
4. 将 DB2 的定义添加至 Tuxedo 资源管理程序的定义文件。在下面的示例中，`UDB_XA` 是为 DB2 在本地定义的 Tuxedo 资源管理程序名，而 `db2xa_switch` 是类型为 `xa_switch_t` 的一个结构的 DB2 定义的名称：

- 对于 AIX。在文件 `#{TUXDIR}/udataobj/RM` 中，添加定义：

```
# DB2 UDB
UDB_XA:db2xa_switch:-L#{DB2DIR} /lib -ldb2
```

其中，`{TUXDIR}` 是安装了 Tuxedo 的目录，而 `{DB2DIR}` 是 DB2 实例目录。

- 对于 Windows NT。在文件 `%TUXDIR%\udataobj\rm` 中，添加定义：

```
# DB2 UDB
UDB_XA;db2xa_switch;%DB2DIR%\lib\db2api.lib
```

其中，`%TUXDIR%` 是安装了 Tuxedo 的目录，而 `%DB2DIR%` 是 DB2 实例目录。

5. 为 DB2 构建 Tuxedo 事务监控程序服务器程序：

- 对于 AIX:

```
#{TUXDIR}/bin/buildtms -r UDB_XA -o #{TUXDIR}/bin/TMS_UDB
```

其中，`{TUXDIR}` 是安装了 Tuxedo 的目录。

- 对于 Windows NT:

```
%TUXDIR%\bin\buildtms -r UDB_XA -o %TUXDIR%\bin\TMS_UDB
```

6. 构建应用服务器。在下面的示例中，`-r` 选项指定资源管理程序名，`-f` 选项（使用了一次或多次）指定包含应用程序服务的文件，`-s` 选项指定此服务器的应用程序服务名，而 `-o` 选项指定输出服务器文件名:

- 对于 AIX:

```
${TUXDIR}/bin/buildserver -r UDB_XA -f svcfile.o -s SVC1,SVC2  
-o UDBserver
```

其中，{TUXDIR} 是安装了 Tuxedo 的目录。

- 对于 Windows NT:

```
%TUXDIR%\bin\buildserver -r UDB_XA -f svcfile.o -s SVC1,SVC2  
-o UDBserver
```

其中，%TUXDIR% 是安装了 Tuxedo 的目录。

7. 设置 Tuxedo 配置文件以引用 DB2 服务器。在 UDBCONFIG 文件的 *GROUPS 小节中，添加类似如下的项:

```
UDB_GRP LMID=simp GRPNO=3  
TMSNAME=TMS_UDB TMSCOUNT=2  
OPENINFO="UDB_XA:db=sample,uid=db2_user,pwd=db2_user_pwd"
```

其中，TMSNAME 参数指定您先前构建的事务监控程序的服务器程序，而 OPENINFO 参数指定资源管理程序名。其后是数据库名和 DB2 用户和口令（用于认证）。

您先前构建的应用服务器在 Tuxedo 配置文件的 *SERVERS 节中被引用。

8. 若应用程序正在存取驻留在 DB2 OS/390 版、DB2 OS/400 版或 DB2 VM 版和 DB2 VSE 版上的数据，则将需要 DB2 Connect XA 集中器。有关配置细节和限制，参考 *DB2 Connect 用户指南*。

9. 启动 Tuxedo:

```
tmboot -y
```

在该命令完成之后，Tuxedo 信息应指示服务器已启动。另外，若您发出 DB2 命令 `LIST APPLICATIONS ALL`，您应该看到两个连接（在这种情况下，由 Tuxedo 配置文件 UDBCONFIG 中的 UDB 组的 TMSCOUNT 参数指定）。

配置 Microsoft 事务服务器

DB2 UDB V5.2 和更新版本可以完全与 Microsoft 事务服务器 (MTS) 版本 2.0 集成在一起。在 Windows 32 位操作系统上的 MTS 下运行的应用程序，可以使用 MTS 来协调多个 DB2 UDB、主机和 AS/400 数据库服务器以及其他 MTS 兼容的资源管理程序的两阶段落实。

在 DB2 中启用 MTS 支持

自动启用“Microsoft 事务服务器”支持。可将 `tp_mon_name` 数据库管理程序配置参数设置为 MTS，但这没有必要，而且将忽略它。

注：IBM web 站点上可能提供了其他技术信息，以帮助您安装和配置 DB2 MTS 支持。将 URL 设置为 <http://www.ibm.com/software/data/db2/library/>，并使用关键字“MTS”来搜索 DB2 通用数据库“技术注释”。

MTS 软件的先决条件

MTS 支持需要 DB2 客户机应用程序使能器 (CAE) 版本 5.2 或更新版本，且 MTS 必须是带有最新修订 0772 的版本 2.0 或更高版本。

在 Windows 32 位操作系统上安装 DB2 ODBC 驱动程序将自动把一个新关键字添加至该注册表：

```
HKEY_LOCAL_MACHINE\software\ODBC\odbcinit.ini\IBM DB2 ODBC Driver:  
Keyword Value Name: CTimeout  
Data Type: REG_SZ  
Value: 60
```

安装和配置

以下是 MTS 的安装和配置考虑事项的摘要。要使用 DB2 的 MTS 支持，必须：

1. 在运行 MTS 应用程序的机器上安装 MTS 和 DB2 客户机。
2. 若主机或 AS/400 数据库服务器要参与多站点更新：
 - a. 在本地机器或远程机器上安装“DB2 Connect 企业版”(EE)。DB2 Connect EE 允许主机或 AS/400 数据库服务器参与多站点更新事务。
 - b. 确保允许 DB2 Connect EE 服务器参与多站点更新。有关允许 DB2 Connect 参与多站点更新的详情，参考该平台的“DB2 Connect 企业版快速入门”手册。

当运行 DB2 CLI/ODBC 应用程序时，下列配置关键字（与 `db2cli.ini` 文件中设置的相同）必须为缺省值，不得更改：

- CONNECTTYPE 关键字（缺省值为 1）

- MULTICONNECT 关键字（缺省值为 1）
- DISABLEMULTITHREAD 关键字（缺省值为 1）
- CONNECTIONPOOLING 关键字（缺省值为 0）
- KEEPCONNECTION 关键字（缺省值为 0）

为利用 MTS 支持而编写的 DB2 CLI 应用程序不能更改与上述关键字对应的属性值。而且，应用程序不能更改下列属性的缺省值：

- SQL_ATTR_CONNECT_TYPE 属性（缺省值 SQL_CONCURRENT_TRANS）
- SQL_ATTR_CONNECTON_POOLING 属性（缺省值 SQL_CP_OFF）

注：IBM web 站点上可能提供了其他技术信息，以帮助您安装和配置 DB2 MTS 支持。将 URL 设置为 <http://www.ibm.com/software/data/db2/library/>，并使用关键字 "MTS" 来搜索 DB2 通用数据库“技术注释”。

验证安装

1. 配置 DB2 客户机和 DB2 Connect EE 以存取您的 DB2 UDB、主机或 AS/400 服务器。
2. 验证从 DB2 CAE 机器至 DB2 UDB 数据库服务器的连接。
3. 使用 DB2 CLP 来验证从 DB2 Connect 机器至您的主机或 AS/400 数据库服务器的连接，并发出一些查询。
4. 验证从 DB2 CAE 机器经由 DB2 Connect 网关至您的主机或 AS/400 数据库服务器的连接，并发出一些查询。

受支持的 DB2 数据库服务器

支持下列服务器使用 MTS 协调的事务来进行多站点更新：

- DB2 通用数据库企业版的版本 5.2
- DB2 扩充企业版版本 5.2
- DB2 OS/390 版
- DB2 MVS 版
- DB2 AS/400 版
- DB2 VM 版和 VSE 版
- DB2 公共服务器 SCO 版的版本 2
- 带 PTF U453782 的 DB2 通用数据库 AIX 版
- 带 PTF U453784 的 DB2 通用数据库 HP-UX 版
- 带 PTF WR09033 的 DB2 通用数据库企业版 OS/2 版
- 带 PTF U453783 的 DB2 通用数据库 SOLARIS 版

- 带 PTF WR09034 的 DB2 通用数据库企业版 Windows NT 版
- DB2 通用数据库扩充企业版 UNIX 版或 Windows NT 版。

MTS 事务超时和 DB2 连接特性

您可以在 MTS Explorer 工具中设置事务超时值。有关详情，请参考联机 *MTS 管理员指南*。

若一个事务占用的时间超过该事务超时值（缺省值为 60 秒），则 MTS 将以异步方式对所有参与的“资源管理程序”发出异常终止，然后将整个事务异常终止。

对于与 DB2 服务器的连接，该异常终止会被转换成 DB2 回滚请求。象任何其他数据库请求一样，将在该连接上连续执行回滚请求，以保证数据库服务器上的数据的完整性。

其结果为：

- 若该连接是空闲的，则立即执行回滚。
- 如果正在处理长时间运行的 SQL 语句，则回滚请求等待，直到该 SQL 语句完成为止。

连接缓冲

连接缓冲允许一个应用程序使用一个连接缓冲池中的连接，而不必在每次使用时重新建立连接。一旦创建了一个连接并将其置于缓冲池中，则应用程序可以再次使用该连接，而不用执行完整的连接过程。当应用程序与 ODBC 数据源断开时将缓冲该连接，并将它给予一个属性相同的新连接。

连接缓冲一直是“ODBC 驱动程序管理器 2.x”的一个功能部件。在与 MTS 一起交付的最新 ODBC 驱动程序管理器（版本 3.5）中，对于事务性 MTS COM 对象的 ODBC 连接，连接缓冲有一些配置变化和新的特性（参见第177页的『在参与同一个事务的 COM 对象之间再次使用 ODBC 连接』）。

“ODBC 驱动程序管理器 3.5”要求 ODBC 驱动程序在注册表中注册一个新关键字，才允许将连接缓冲激活。该关键字为：

```
Key Name: SOFTWARE\ODBC\ODBCINST.INI\IBM DB2 ODBC DRIVER
Name: CTimeout
Type: REG_SZ
Data: 60
```

32 位操作系统的 DB2 ODBC 驱动程序版本 6 和更高版本完全支持连接缓冲；因此注册了此关键字。版本 5.2 客户机必须安装修订包 3 (WR09024) 或更高版本。

缺省值 60 意味着在连接被断开之前将缓冲 60 秒。

在一个繁忙的环境中，最好是将 CTimeout 的值该为一个较大的数（Microsoft 有时建议对于某些环境使用 10 秒），以阻止过多的物理连接和断开，因为这些操作会占用大量系统资源，包括系统内存和通信堆栈资源。

另外，要确保在多台处理器机器中的同一事务中的对象之间使用同一连接，必须关闭“每个处理器有多个缓冲池”支持。为此，将下列注册表设置复制到名为 `odbcpool.reg` 的文件中，将其作为无格式文本文件保存，并发出命令 **odbcpool.reg**。Windows 操作系统将调入这些注册表设置。

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\ODBC Connection Pooling]
"NumberOfPools"="1"
```

如果不将此关键字设置为 1，则 MTS 可能在不同的缓冲中缓冲连接，导致不重新使用同一连接。

使用 ADO 2.1 和更新版本的 MTS 连接缓冲

如果 MTS COM 对象使用 ADO 来存取数据库，则必须关闭 OLEDB 资源缓冲，以便 ODBC 的 Microsoft OLEDB 提供者 (MSDASQL) 不会干扰 ODBC 连接缓冲。在 ADO 2.0 中，此功能被初始为 OFF，但在 ADO 2.1 中，被初始化为 ON。要关闭 OLEDB 资源缓冲，将下列各行复制到名为 `oledb.reg` 的文件中，并将其作为无格式文本文件保存，然后发出命令 **oledb.reg**。Windows 操作系统将调入此注册表设置。

```
REGEDIT4

[HKEY_CLASSES_ROOT\CLSID\{c8b522cb-5cf3-11ce-ade5-00aa0044773d}]
@="MSDASQL"
"OLEDB_SERVICES"=dword:ffffffff
```

在参与同一个事务的 COM 对象之间再次使用 ODBC 连接

MTS COM 对象中的 ODBC 连接会自动打开连接缓冲（不管该 COM 对象是否是事务性的）。

对于参与同一个事务的多个 MTS COM 对象，可以使用如下方式在两个或多个 COM 对象之间再次使用该连接。

假定有两个 COM 对象 COM1 和 COM2，它们与同一个 ODBC 数据源连接并参与同一个事务。

在 COM1 连接并完成它的工作之后，它就会断开，并缓冲该连接。但是，此连接将被保留，以备同一个事务的其他 COM 对象使用。只有在当前事务结束之后，它才可供其他事务使用。

当在同一个事务中调用 COM2 时，会给予它缓冲的连接。MTS 将确保只将该连接分配给参与同一个事务的 COM 对象。

另一方面，若 COM1 未显式断开，则在该事务结束之后它仍维持着该连接。当在同一个事务中调用 COM2 时，会请求另一个连接。这样，此事务同时维持着两个连接，而不是一个。

由于下列原因，最好对参与同一个事务的 COM 对象使用再次使用连接的功能：

- 它在客户机和服务器中都使用较少的资源。只需要一个连接。
- 它消除了参与同一个事务的两个连接（存取相同的数据库服务器和存取相同的数据）相互锁定的可能性，因为 DB2 服务器将来自 MTS COM 对象的不同连接当作单独的事务来处理。

调整 TCP/IP 通信

若在同时进行太多的物理连接和断开的具有高工作负荷环境中，使用很小的 CTimeout 值，则 TCP/IP 堆栈可能会遇到资源约束。

要消除此问题，使用“TCP/IP 注册表项”。有关信息在 *Windows NT Resource Guide* 的第一卷中有描述。该注册表关键字值位于“HKEY_LOCAL_MACHINE->SYSTEM->CurrentControlSet->服务->TCPIP->参数”中。

其缺省值和建议的设置如下所示：

名称	缺省值	建议的值
KeepAlive 时间	7200000 (2 小时)	相同
KeepAlive 时间间隔	1000 (1 秒)	10000 (10 秒)
TcpKeepCnt	120 (2 分钟)	240 (4 分钟)
TcpKeepTries	20 (20 次重试)	相同
TcpMaxConnectAttempts	3	6
TcpMaxConnectRetransmission	3	6
TcpMaxDataRetransmission	5	8
TcpMaxRetransmissionAttempts	7	10

若未定义注册表的值，则创建它。

使用 MTS “BANK” 样本应用程序测试 DB2

可以使用 MTS 所附带的“BANK”样本程序来测试客户机产品和 MTS 的设置。

执行下列步骤：

1. 更改文件 \Program Files\Common Files\ODBC\Data Sources\ MTSSamples.dsn, 以使它类似于如下所示:

```
[ODBC]
DRIVER=IBM DB2 ODBC DRIVER
UID=your_user_id
PWD=your_password
DSN=your_database_alias
Description=MTS Samples
```

其中:

- *your_user_id* 和 *your_password* 是用于连接主机的用户 ID 和口令。
 - *your_database_alias* 是用于连接数据库服务器的数据库别名。
2. 转至“控制面板”中的“ODBC 管理”，选择 **系统 DSN** 标签并添加数据源:
 - a. 选择“IBM ODBC 驱动程序”，然后选择**结束**。
 - b. 当显示数据库别名的列表时，选择先前指定的那一个。
 - c. 选择**确认**。
 3. 以 ID *your_user_id* 的名义，使用 DB2 CLP 与 DB2 数据库连接，如上所述。
 - a. 联编 db2cli.lst 文件:

```
db2 bind @C:\sql11ib\bnd\db2cli.lst blocking all grant public
```

- b. 联编这些实用程序。

若服务器是 **DRDA** 主机服务器，则根据您连接的主机 (OS/390、AS/400、VSE&VM)，联编 ddcsmvs.lst、ddcs400.lst 或 ddcsvm.lst。例如:

```
db2 bind @C:\sql11ib\bnd\@ddcsmvs.lst blocking all grant public
```

否则，联编 db2ubind.lst 文件:

```
db2 bind @C:\sql11ib\bnd\@db2ubind.lst blocking all grant public
```

- c. 为 MTS 样本应用程序创建样本表和数据，如下所示:

```
db2 create table account (accountno int, balance int)
db2 insert into account values(1, 1)
```
4. 在 DB2 客户机上，确保数据库管理程序配置参数 *tp_mon_name* 设置为 MTS。
 5. 运行“BANK”应用程序。选择**帐户**按钮和 **Visual C++** 选项，然后提交该请求。其他选项可使用专用于 SQL 服务器的 SQL，但可能不起作用。

第11章 针对高可用性的设计

“DB2 通用数据库”在多个平台上提供了高可用性故障恢复支持。故障恢复能力允许在发生硬件故障时，自动将工作负荷从一个处理器转移至另一个处理器。例如，在 AIX 上，DB2 UDB 通过 “IBM 高可用性群集多重处理” (HACMP) 的能力来支持故障恢复。在本节中，使用 AIX 上的示例来介绍与高可用性相关的概念。

HACMP 通过处理器的群集提供更高的可用性，这些处理器共享诸如磁盘或网络存取权这样的资源。若一个处理器发生故障，则群集中的另一个处理器可以替换它。

共有三种方式的故障恢复支持：

热备份 在这种方式中，使用一个处理器来运行 DB2 实例，第二个处理器处于备份方式，准备在第一个处理器运行的环境中发生操作系统或硬件故障时接管该实例。

相互替换

在此方式中：

- 使用两个处理器来运行不同的 DB2 实例。
- 使用一个处理器来运行 DB2 实例，使用另一个来运行 DB2 应用程序。

若在一个处理器上发生操作系统或硬件故障，则另一个处理器接管该失效处理器的任务，最终完成两个处理器的工作。

并行存取

在此方式中，使用多个处理器来构成使用 “DB2 通用数据库扩充企业版” (EEE) 产品的单个数据库实例。这是通过使用无共享模型并将数据分区来实现的，这样一个或多个分区在群集中的每一个处理器上运行。若在一个处理器上发生操作系统或硬件故障，则另一个处理器接管该失效处理器的分区。DB2 UDB EEE 不需要 “并行资源管理程序” 来提供冗余度。通过使用热备份或相互替换方式管理冗余度。只有具有共享体系结构的数据库管理程序才需要并行存取方式的功能。

可使用这些配置中的任何一个来从分区数据库的一个或多个分区的故障中恢复。此外，每种配置都可从一个完整的单分区安装实例的故障中恢复。

热备份

热备份能力可用于从整个单分区数据库实例的故障中恢复，或从一个分区数据库配置的一个分区的故障中恢复。若一个处理器失效，则该群集中的另一个处理器可通过自动转移该实例来替换失效的那个处理器。主处理器和故障恢复处理器必须可以存取数据库实例和实际数据库。

- DB2 安装路径可以在两个系统共享的一条路径上或在非共享文件系统中。若使用非共享文件系统，则安装级别必须相同。
- DB2 实例路径可以在共享文件系统中，或在人工镜像的文件系统上。
- 数据库及相关的容器必须位于两个系统都可存取的文件系统（或设备）上。
- 分区数据库配置中的分区的故障恢复期间，在第二个处理器上重新启动该分区：故障恢复脚本更改 `db2nodes.cfg` 文件，使之在新处理器上指向此分区，并在该处理器上启动此分区。
- 当进行故障恢复时，作为故障恢复过程的一部分，将透明地传送受支持通信协议的外部通信地址。

有关实际安装需求和实例创建的详细信息，参考 *HACMP for AIX, Version 4.2: Installation Guide, SC23-1940*。

示例

（在 DB2 AIX 版安装中，）下列每个示例都在 `sqllib/samples/hacmp` 中存储了一个样本脚本。

实例故障恢复

下列热备份故障恢复方案为：一个双处理器 HACMP 群集在单分区数据库实例上运行（第183页的图44）。有关配置 HACMP 群集的详情，参见第188页的『资源』。

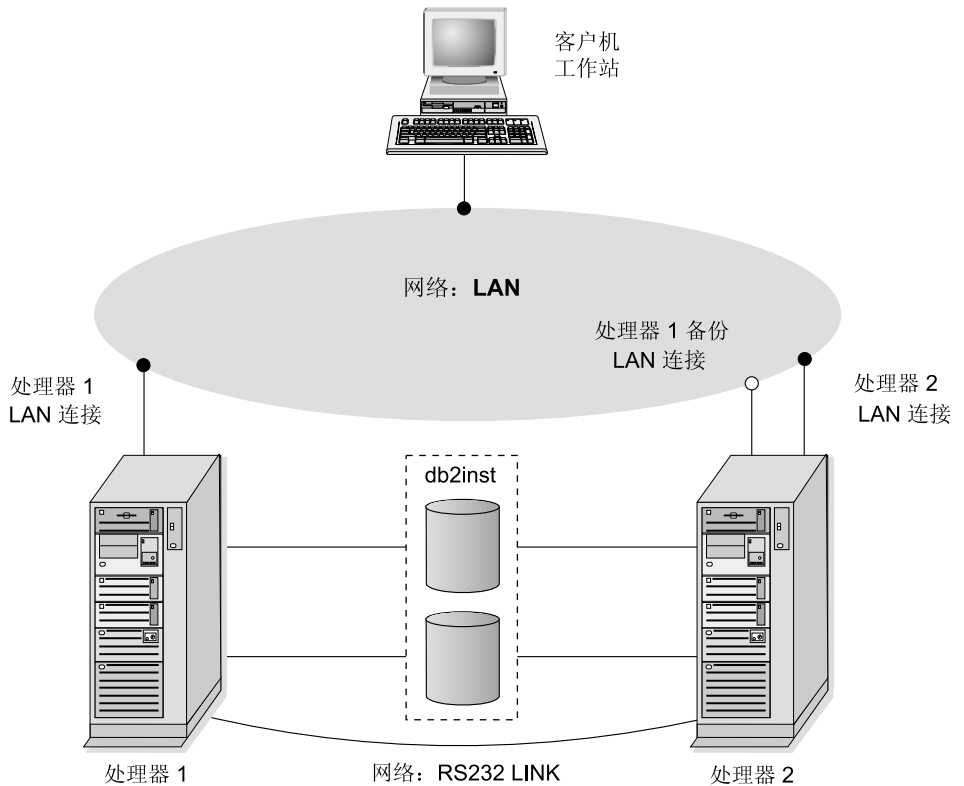


图 44. 热备份故障恢复配置示例

两个处理器对安装目录、实例目录和数据库目录都有存取权。数据库实例“db2inst”当前是活动的，在处理器 1 上执行。处理器 2 不是活动的，被用作一个热备份。处理器 1 上发生故障，处理器 2 接管了该实例。一旦故障恢复完成，远程和本地应用程序就可以存取实例“db2inst”内的数据库。该数据库将必须以人工方式重新启动，或者，若 AUTORESTART 为开，则与该数据库的第一个连接将启动重新启动操作。在提供的样本脚本中，假定 AUTORESTART 为关，故障恢复脚本对该数据库执行重新启动。有关 AUTORESTART 的详情，参见管理指南：实现中的“恢复概述”。

样本脚本：

```
hacmp-s1.sh
```

分区故障恢复

在下面的热备份故障恢复方案中，我们使用一个实例分区，而不是使用整个实例。与上一个示例一样，此方案包括一个双处理器的 HACMP 群集，但是，该机

器表示一个分区数据库服务器的其中一个分区。处理器 1 运行整个配置中的单个分区，处理器 2 用作故障恢复处理器。当处理器 1 失效时，在第二个处理器上重新启动该分区。故障恢复更新 db2nodes.cfg 文件，使之指向处理器 2 的主机名和网络名，然后在新处理器上重新启动该分区。

以下是故障恢复前后的 db2nodes.cfg 文件的一部分。在本示例中，节点号 2 在 HACMP 机器的处理器 1 上运行，该机器的主机名和网络名均为 "node201"。在故障恢复之后，节点号 2 在 HACMP 机器的处理器 2 上运行，该机器的主机名和网络名均为 "node202"。

之前:

```
1 node101 0 node101
2 node201 0 node201    <= HACMP
3 node301 0 node301
```

```
db2start nodenum 2 restart hostname node202 port 0 netname node202
```

之后:

```
1 node101 0 node101
2 node202 0 node202    <= HACMP
3 node301 0 node301
```

样本脚本:

```
hacmp-s2.sh
```

多逻辑节点故障恢复

比上一个示例更复杂的情况是，多逻辑节点从一个处理器到另一个处理器的故障恢复。我们再次使用与前面两例相同的双处理器的 HACMP 群集配置。但是，在本方案中，处理器 1 运行 3 个逻辑分区。设置与简单的故障恢复方案的设置相同，但是在本例中，当处理器 1 失效时，必须在处理器 2 上启动每个逻辑分区。每个逻辑分区必须按在 db2nodes.cfg 文件中定义的次序来启动：始终必须首先启动端口号为 0 的逻辑分区。

以下是故障恢复前后的 db2nodes.cfg 文件的一部分。在本示例中，在一个双处理器的 HACMP 群集的处理器 1 上定义了 3 个逻辑分区。

之前:

```
1 node101 0 node101
2 node201 0 node201    <= HACMP
3 node201 1 node201    <= HACMP
4 node201 2 node201    <= HACMP
5 node301 0 node301
```

```
db2start nodenum 2 restart hostname node202 port 0 netname node202
db2start nodenum 3 restart hostname node202 port 1 netname node202
db2start nodenum 4 restart hostname node202 port 2 netname node202
```

之后:

```
1 node101 0 node101
2 node202 0 node202   <= HACMP
3 node202 1 node202   <= HACMP
4 node202 2 node202   <= HACMP
5 node301 0 node301
```

样本脚本:

```
hacmp-s3.sh
```

相互替换

在相互替换方式中，一个处理器可以从单分区数据库实例的故障或一个分区数据库的分区故障恢复，同时运行另一个实例或一个分区数据库配置的其他分区。使用热备份配置时，安装路径、实例目录和数据库必须可以由可能参与故障恢复处理的每个处理器存取。安装路径和实例路径可在共享文件系统上或在镜象的独立文件系统上。

当使用相互替换策略进行实例故障恢复时，必须将实例定义成两个实例可以同时运行在同一处理器上。有关实际安装需求和实例创建的详细信息，参考 *HACMP for AIX, Version 4.2: Installation Guide, SC23-1940*。

示例

(在 DB2 AIX 版安装中，) 下列每个示例都在 `sqllib/samples/hacmp` 中存储了一个样本脚本。

DB2 实例故障相互恢复

以下实例故障相互恢复方案包含一个带有两个处理器（称为 "node10" 和 "node20"）的 HACMP 系统。

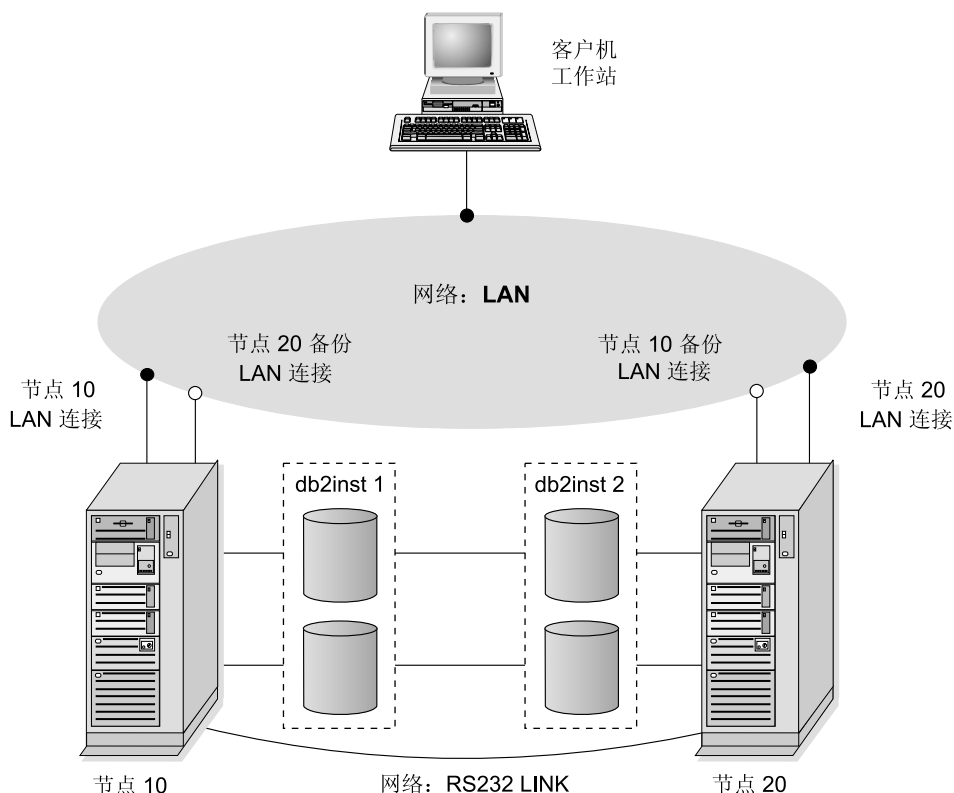


图 45. 实例故障相互恢复配置示例

在一个共享文件系统上的单一安装路径中创建两个实例，即“db2inst1”和“db2inst2”。在 /u/db2inst1 上创建实例“db2inst1”，在 /u/db2inst2 上创建实例“db2inst2”。这两个路径都在两个处理器均可存取的一个共享文件系统上。每个实例有一个具有唯一路径的单一数据库，它也位于两个处理器都可存取的一个共享资源上。

两个实例通过 TCP/IP 协议由远程客户机存取：“db2inst1”使用服务名“db2inst1_port”（端口号 5500），而“db2inst2”使用服务名“db2inst2_port”（端口号 5550）。存取“db2inst1”实例的远程客户机在它们的节点目录中使用“node10”作为主机名来编目此实例。存取“db2inst2”实例的远程客户机在它们的节点目录中使用“node20”作为主机名来编目此实例。在正常的操作条件下，“db2inst1”在“node10”上运行，而“db2inst2”在“node20”上运行。若“node10”失效，则故障恢复脚本将在“node20”上启动“db2inst1”，而与“node10”相关的外部 IP 地址将切换到“node20”。一旦故障恢复脚本启动了该实例并重新启动了该数据库，远程客户机就可以与此实例内的数据库连接，就好象它在“node10”上执行一样。

样本脚本:

```
hacmp-s4.sh
```

DB2 分区故障互相恢复

对一个分区数据库服务器环境中的分区故障进行互相恢复时，要求将该分区作为故障恢复处理器上的逻辑节点进行故障恢复。例如，若一个分区数据库服务器的两个分区分别在配置为相互替换方式的双处理器 HACMP 群集的各处理器上运行，则这两个分区必须作为逻辑节点来进行故障恢复。必须将每个节点上的缺省分区定义为逻辑节点 0，这表示当一个分区从一个处理器转换到另一个处理器来进行故障恢复时，它将作为一个没有直接的远程通信协议监听程序的逻辑节点来启动。这样的分区不能用作协调程序节点。

当配置一个系统进行分区相互替换时，要考虑的另一个重要问题是本地分区数据库路径。当在分区数据库环境中创建数据库时，要在分区数据库服务器不共享的根路径上创建它。例如，考虑如下命令：

```
CREATE DATABASE db_a1 ON /dbpath
```

此命令在实例 "db2inst" 上运行，并在 /dbpath 上创建数据库 db_a1。每个数据库分区都在其本地文件系统上的 /dbpath/db2inst/nodexxxx 下面创建，其中，xxxx 表示节点号。HACMP 故障恢复将尝试安装 /dbpath 文件系统，该文件系统已由另一处理器使用。因此，故障恢复脚本必须在另一个逻辑节点下安装该文件系统，并设置从该文件系统至适当的 /dpath/db2inst/nodexxxx 路径的符号链路。

以下是故障恢复前后的 db2nodes.cfg 文件的一部分。在本示例中，节点号 2 在 HACMP 机器的处理器 1 上运行，该机器的 hostname 和网络名均为 "node201"。节点号 3 在 HACMP 机器的处理器 2 上运行，该机器的 hostname 和网络名均为 "node202"。

之前:

```
1 node101 0 node101
2 node201 0 node201   <= HACMP
3 node202 0 node202   <= HACMP
4 node301 0 node301
```

```
db2start nodenum 2 restart hostname node202 port 1 netname node202
```

之后:

```
1 node101 0 node101
2 node202 1 node202   <= HACMP
3 node202 0 node202   <= HACMP
4 node301 0 node301
```

在执行故障恢复后，尝试直接将节点号 2 作为协调程序来存取的任何远程客户机将不得不重新编目该数据库的节点项，以指向故障恢复节点。不建议将故障相互恢复方案用于协调程序节点。若要求协调程序节点具有冗余度，则使用热备份配置。

样本脚本:

```
hacmp-s5.sh
```

故障恢复后的重新连接

若客户机使用 `SET CLIENT` 语句来与特定节点连接，且该节点在故障恢复期间移至另一主机，则从该客户机发出的下一连接请求将失败。在运行 `SET CLIENT` 语句的节点上发出 `db2stop`，接着发出 `db2start nodenum`，然后重新发出该语句，以便客户机和服务器都能检测到目标节点的新物理位置。

资源

有关 HACMP 的概念、安装和配置的详情，参考下列书籍：

- *HACMP for AIX, Version 4.2: Concepts and Facilities*, SC23-1938
- *HACMP for AIX, Version 4.2: Installation Guide*, SC23-1940
- *HACMP for AIX, Version 4.2: Planning Guide*, SC23-1939.

第4部分 高可用性

第12章 高可用性群集多重处理增强可缩放性 (HACMP ES) AIX 版

“增强可缩放性” (ES) 是 “高可用性群集多重处理 (HACMP) AIX 版” 版本 4.2.2 的一个功能部件，该软件当前只能在 RS/6000 SP 节点上运行。

此功能部件提供的故障恢复与 HACMP 提供的相同，并具有与先前的 HACMP 版本相同的事件结构（参见 *HACMP for AIX, V4.2.2, Enhanced Scalability Installation and Administration Guide*）。增强可缩放性还提供了：

- 更大的 HACMP 群集，每个群集可扩大到 16 个节点。
- 通过用户定义事件扩大错误检测范围。受监控的区域可触发用户定义事件，这些事件各不相同，如一个进程停止或分页空间接近其容量限制。若有必要，这类事件包括可添加至故障恢复进程的前事件和后事件。可将专用于其他实现的额外功能放在 HACMP 的前事件和后事件流中。

规则文件 (`/usr/sbin/cluster/events/rules.hacmprd`) 包含 HACMP 事件。用户定义事件被添加至此文件。事件发生时要运行的脚本文件是此定义的一部分。

有关用户定义事件和规则文件的详情，参见第210页的『HACMP ES 事件监控和用户定义事件』。

- HACMP 客户机实用程序，它们用于从 HACMP 群集外的 AIX 物理节点监控和检测（一个或多个群集中的）状态更改。

HACMP ES 群集中的节点交换的信息称为心跳或保持活动的信息包，每个节点通过此信息通知其他节点有关它的可用性。已停止应答的节点导致该群集中的其余节点调用恢复。该恢复过程称为 `node_down` 事件，也可称为故障恢复。恢复过程完成后，将节点重新集成到该群集中。这称为 `node_up` 事件。

有两类事件：一类是在 HACMP ES 的操作内预计的标准事件，另一类是与硬件和软件部件中的参数监控相关的用户定义事件。

其中一个标准事件是 `node_down` 事件。当计划应执行什么操作来作为恢复过程的一部分时，HACMP 允许两个故障恢复选项：热（或空闲）备份和相互替换。

在热备份配置中，作为替换节点的 AIX 处理器节点不运行任何其他工作负荷和“相互替换”。在相互替换配置中，作为替换节点的 AIX 处理器节点要运行其他工作负荷。

通常，“DB2 通用数据库扩充企业版” (UDB EEE) 以相互替换方式运行，且每个节点上都有分区。目录节点作为热备份配置一部分的方案是一个例外。

当在使用 HACMP ES 的一个 RS/6000 SP 上计划大型 DB2 安装时，需要考虑如何在 RS/6000 SP 计算机内部或之间划分群集的节点。将节点和它的备份置于不同的 SP 计算机中，这样可以在一个计算机停机（即计算机电源 / 配电板发生故障）时进行替换。但是，发生此类故障的可能性非常小，因为每个 SP 计算机中有 $N+1$ 个电源，每个 SP 交换都有备用电路以及 $N+1$ 个风扇和电源。在计算机发生故障的情况下，可能需要人工干预以恢复其余的计算机。在 SP Administration Guide 中记载了此恢复过程。HACMP ES 确保了 SP 节点故障的恢复；计算机故障的恢复取决于一个或多个 SP 计算机内部群集的正确布局。

另一个计划的考虑事项是如何管理大型群集。管理一个小的群集比管理一个大的群集容易得多；然而，管理一个大的群集又比管理许多小的群集容易得多。在计划过程中，要考虑在群集环境中如何使用应用程序。例如，若有一单个的、大型的、同类应用程序在 16 个节点上运行，则将配置作为单个群集管理比管理 8 个两节点群集可能要容易些。若相同的 16 个节点包含与不同网络、磁盘和节点相关的许多不同的应用程序，则可能最好将这些节点分组到更小的群集中。记住，一次只能将一个节点集成到 HACMP 群集中；启动多个群集的一个配置比启动一个大群集快得多。如果某个节点及其备份在同一群集中，那么 HACMP ES 可同时支持单个和多个群集。

HACMP ES 故障恢复允许资源组对物理节点的预定义（也称作级联）分配。该故障恢复过程还允许资源组对物理节点的浮动（也称作旋转）分配。IP 地址和外部磁盘卷组，或文件系统，或 NFS 文件系统，以及每个资源组内的应用服务器指定应用程序或应用程序组件，它们可由 HACMP ES 通过故障恢复和重新集成在物理节点间进行操纵。故障恢复和重新集成行为由所创建的资源组类型和该资源组中的节点数来指定。

例如，考虑一个 DB2 数据库分区（逻辑节点）。若其日志和表空间容器位于外部磁盘上，并且其他节点与那些磁盘链接，则其他那些节点可能可以存取这些磁盘并重新启动该数据库分区（在替换节点上）。这是 HACMP 自动执行的操作类型。HACMP ES 也可用来恢复 DB2 实例主用户目录所使用的 NFS 文件系统。

作为使用 DB2 UDB EEE 来进行恢复的计划的一部分，要仔细阅读 HACMP ES 文档。应阅读 Concepts, Planning, Installation, and Administration Guides, 然后为您的环境构建恢复体系结构。对于根据已知故障点来标识的要恢复的每个子系统，标识您需要的 HACMP 群集，以及恢复节点（热备份或相互替换）。这是填写该文档中包括的 HACMP 工作表的起点。

强烈建议在外部磁盘配置中建立磁盘和适配器的镜象。对于为 HACMP 配置的 DB2 物理节点，需要小心确保卷组上的节点可从共享外部磁盘联机。在相互替换配置中，这种方案需要某些附加计划，以便配对的节点可相互存取对方的卷组而不造成冲突。对于 DB2 UDB EEE，这表示在所有数据库中的所有容器名必须是唯一的。

实现唯一性的一种方式包括分区号作为名称的一部分。当创建 SMS 或 DMS 容器时，可指定容器字符串语法的节点表达式。当指定表达式时，节点号可以是容器名的一部分，或者，若指定了附加自变量，则那些自变量的结果可以是容器名的一部分。使用自变量 "\$N" ([blank]\$N) 来指示节点表达式。自变量必须出现在容器字符串的末尾，且只可使用下列形式之一：

表 24. 用于创建容器的自变量。假定该节点号为 5。

语法	示例	值
[blank]\$N	" \$N"	5
[blank]\$N+[number]	" \$N+1011"	1016
[blank]\$N%[number]	" \$N%3"	2
[blank]\$N+[number]%[number]	" \$N+12%13"	4
[blank]\$N%[number]+[number]	" \$N%3+20"	22
注:		
1. % 是系数。		
2. 在所有情况下，从左往右对运算符求值。		

下面是如何使用此特殊自变量来创建容器的一些示例:

- 创建在两节点系统上使用的容器。

```
CREATE TABLESPACE TS1 MANAGED BY DATABASE USING
(device '/dev/rcont $N' 20000)
```

将使用下列容器:

```
/dev/rcont0 - 在节点 0 上
/dev/rcont1 - 在节点 1 上
```

- 创建在四节点系统上使用的容器。

```
CREATE TABLESPACE TS2 MANAGED BY DATABASE USING
(file '/DB2/containers/TS2/container $N+100' 10000)
```

将使用下列容器:

```
/DB2/containers/TS2/container100 - 在节点 0 上  
/DB2/containers/TS2/container101 - 在节点 1 上  
/DB2/containers/TS2/container102 - 在节点 2 上  
/DB2/containers/TS2/container103 - 在节点 3 上
```

- 创建在两节点系统上使用的容器。

```
CREATE TABLESPACE TS3 MANAGED BY SYSTEM USING  
  ('/TS3/cont $N%2, '/TS3/cont $N%2+2')
```

将使用下列容器:

```
/TS3/cont0 - 在节点 0 上  
/TS3/cont2 - 在节点 0 上  
/TS3/cont1 - 在节点 1 上  
/TS3/cont3 - 在节点 1 上
```

第195页的图46和第196页的图47显示了 DB2 SSA I/O 子系统配置的一个示例，并显示了一个方案的一部分，该方案是确保高度可用的外部磁盘配置并能够存取所有卷组而不发生冲突所必需的。

DB2 SSA I/O 子系统配置 - 无单个故障点

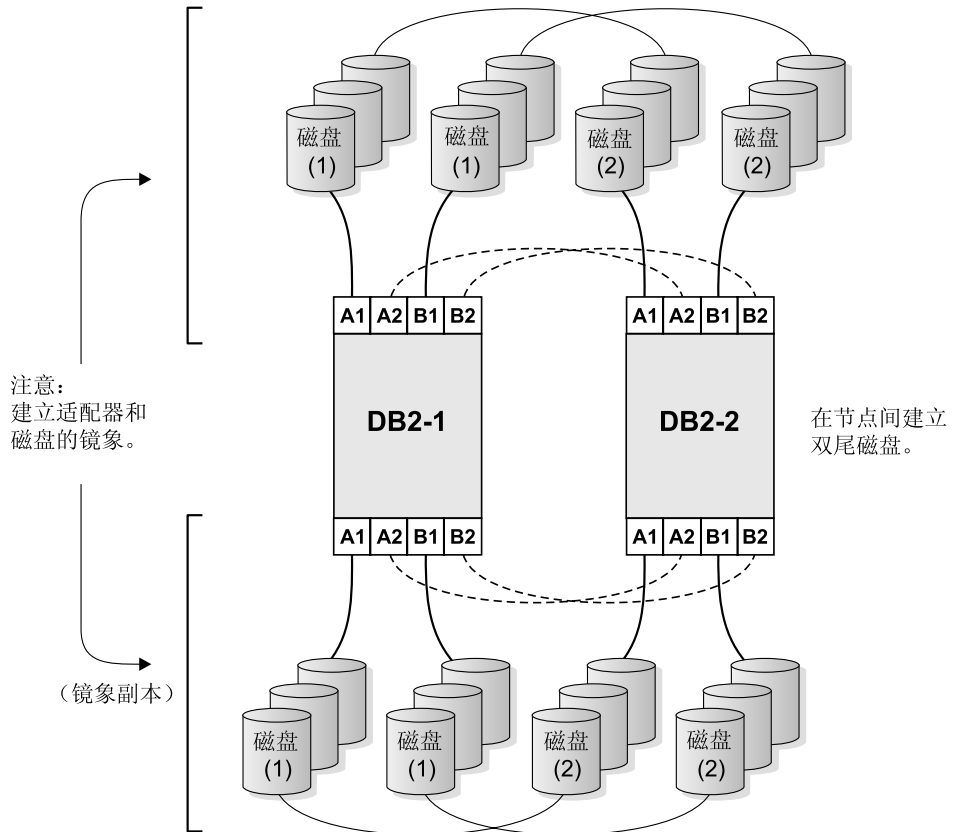


图 46. 没有单个故障点

DB2 SSA I/O 子系统配置 - 卷组和逻辑组设置

在文件系统 /database 实例名 powertp 上的 db2 数据库 testdata

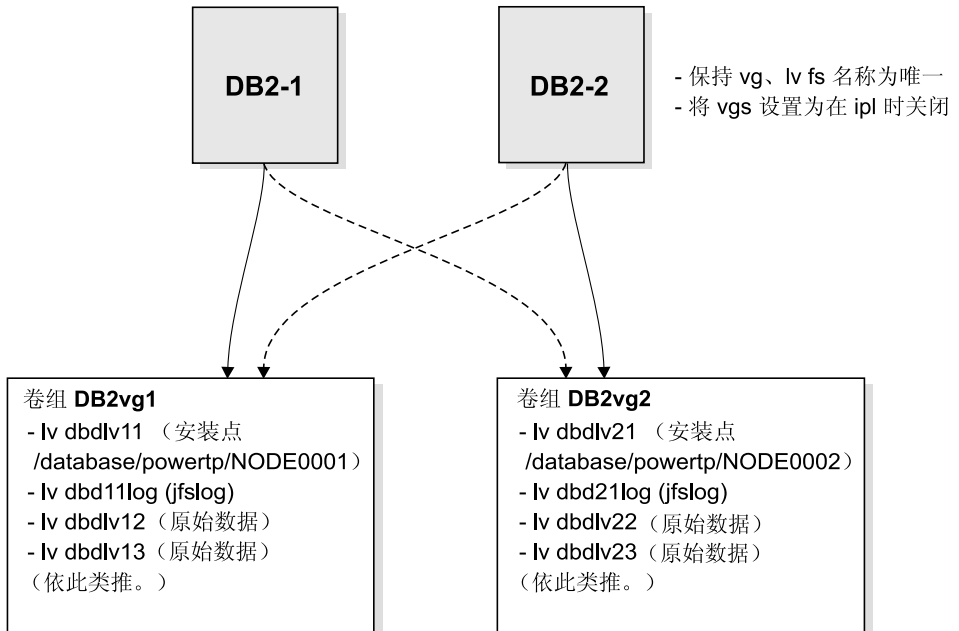


图 47. 卷组和逻辑卷设置

配置 DB2 数据库分区

一旦完成配置，HACMP ES 就启动实例中的每个数据库分区，一次启动一个物理节点。建议使用多个群集来启动多于四个节点的并行 DB2 配置。注意，在 64 节点的并行 DB2 配置中，并行启动 32 个两节点 HACMP 群集比启动四个 16 节点群集要快。

脚本文件 rc.db2pe 与 DB2 UDB EEE 封装在一起（并安装在每个节点的 /usr/bin）中），以对热备份或相互替换节点中的 HACMP ES 故障恢复或恢复进行辅助配置。另外，可在相互替换配置中的故障恢复阶段从 rc.db2pe 内定制 DB2 缓冲池大小。（当两个数据库分区在一个物理节点上运行时，需要配置缓冲池大小以确保合适的性能。）

当在 DB2 数据库分区的 HACMP 配置中创建应用服务器时，将 rc.db2pe 指定为启动和停止脚本，如下所示：

```
/usr/bin/rc.db2pe <instance> <dpn> <secondary dpn> start <use switch>  
/usr/bin/rc.db2pe <instance> <dpn> <secondary dpn> stop <use switch>
```

其中:

<instance> 是实例名。
<dpn> 是数据库分区号。
<secondary dpn> 仅是相互替换配置中的“伙伴”数据库分区号；
在热备份配置中，它与 <dpn> 相同。
<use switch> 通常是空白；当它为空白时，指示
SP 交换网络用于 db2nodes.cfg 文件中的主机名字段
(DB2 的所有通信都通过 SP 交换来按路径发送)；
若不是空白，则使用的名称是要使用的 SP 节点的主机名。

从 rc.db2pe 内使用 DB2 命令 LIST DATABASE DIRECTORY，来查找为此数据库分区配置的所有数据库。然后，该脚本文件查找 /usr/bin/reg.parms.DATABASE 文件和 /usr/bin/failover.parms.DATABASE 文件，其中，DATABASE 是为此数据库分区配置的每个数据库。在相互交换配置中，建议您创建参数文件 reg.parms.xxx 和 failover.parms.xxx。在 failover.parms.xxx 文件中，应调整 BUFFPAGE、DBHEAP 和影响缓冲池的任何其他参数的设置，以确保多个缓冲池的可能性。提供了样本文件 reg.parms.SAMPLE 和 failover.parms.SAMPLE 供您使用。

此环境中的其中一个重要参数是 *start_stop_time* 数据库管理程序配置缺省值，它的缺省值为 10 分钟。然而，rc.db2pe 将此参数设置为两分钟。您应通过 rc.db2pe 将此参数的值设置为 10 分钟，或稍微多些。在此上下文中，指定的持续时间是分区的故障与恢复之间的时间间隔。若在分区上运行的应用程序频繁地发出 COMMIT，则在数据库分区失效后，10 分钟时间应足够回滚未落实的事务，并达到该分区上数据库的一致性点。若工作负荷很重，或者您拥有许多分区，则可能需要增加持续时间，以降低在回滚操作完成之前发生超时的可能性。

下面是热备份配置和相互交换配置的一个示例。在这两个示例中，资源组都包含一个“服务 IP”交换别名地址。此交换别名地址用于:

- 对 DB2 实例所有者文件系统的文件服务器的 NFS 存取
- 在故障恢复、TSM (Tivoli 存储管理器，以前称为 ADMS) 连接或其他类似的操作情况下需要维护的其他客户机存取。

若您的实现不需要这些别名，则可除去它们。若除去了它们，则要确保在 rc.db2pe 脚本文件中将 *MOUNT_NFS* 参数设置为 NO。

热备份配置示例

此示例假设物理节点 1 与 2 之间存在热备份配置，且 DB2 实例名是 POWERTP。数据库分区是 1，数据库是 TESTDATA，该数据库驻留在文件系统 /database 上。

```

Resource group name: db2_dp_1
Node Relationship: cascading
Participating nodenames: node1_eth, node2_eth
Service_IP_label: nfs_switch_1 ( <<< 这是交换别名地址 )
Filesystems: /database/powertp/NODE0001
Volume Groups: DB2vg1
Application Servers: db2_dp1_app
Application Server Start Script: /usr/bin/rc.db2pe powertp 1 1 start
Application Server Stop Script: /usr/bin/rc.db2pe powertp 1 1 stop

```

相互替换配置示例

此示例假设物理节点 1 与 2 之间存在相互替换配置，且 DB2 实例名是 POWERTP。数据库分区是 1 和 2，数据库是 TESTDATA，该数据库驻留在文件系统 /database 上。

```

Resource group name: db2_dp_1
Node Relationship: cascading
Participating nodenames: node1_eth, node2_eth
Service_IP_label: nfs_switch_1 ( <<< 这是交换别名地址 )
Filesystems: /database/powertp/NODE0001
Volume Groups: DB2vg1
Application Servers: db2_dp1_app
Application Server Start Script: /usr/bin/rc.db2pe powertp 1 2 start
Application Server Stop Script: /usr/bin/rc.db2pe powertp 1 2 stop

```

```

Resource group name: db2_pd_2
Node Relationship: cascading
Participating nodenames: node2_eth, node1_eth
Service_IP_label: nfs_switch_2 ( <<< 这是交换别名地址 )
Filesystems: /database/powertp/NODE0002
Volume Groups: DB2vg2
Application Servers: db2_dp2_app
Application Server Start Script: /usr/bin/rc.db2pe powertp 2 1 start
Application Server Stop Script: /usr/bin/rc.db2pe powertp 2 1 stop

```

NFS 服务器节点的配置

还可使用 rc.db2pe 脚本使 DB2 并行实例用户目录的 NFS 安装的目录可用。这可通过在 rc.db2pe 脚本文件中将 *MOUNT_NFS* 参数设置为 YES，并配置 NFS 故障恢复服务器对来完成，如下所示：

- 配置该主目录，并使用 /etc/exports 和 **exportfs** 命令将它作为“根”调出至一个 IP 地址，该地址是与 NFS 服务器的 IP 地址相同的子网中的节点上使用的 IP 地址。包括 HACMP 引导地址和服务地址。NFS 服务器的 IP 地址与可用备份替换的 HACMP 中的服务地址相同。DB2 实例拥有者的主目录应是 NFS 直接安装的，而不是自动安装的。（作为 DB2 实例拥有者主目录的脚本不支持使用自动安装程序。）

- 通过使用 **SMIT** 或一个命令行配置，为此文件系统创建一个单独的 `/etc/filesystems` 项，以便该 **DB2** 并行分组中的所有节点（包括文件服务器）可使用 **NFS** 文件系统命令进行安装。

例如，可将 `/nfshome` **JFS** 文件系统作为 `/dbhome` 调出至所有节点。每个节点都创建一个 **NFS** 文件系统 `/dbname`，它是 `nfs_server:/nfshome`。因此，当实例名为 `"powertp"` 时，**DB2** 实例拥有者的主目录将是 `/dbhome/powertp`。

确保 `/etc/filesystems` 中用于安装的 **NFS** 参数是 `"hard"`、`"bg"`、`"intr"` 和 `"rw"`。

- 确保与 `/etc/passwd` 中的主目录 `/dbhome/powertp` 相关的 **DB2** 实例拥有者定义在所有节点上相同。

在 **SP** 环境中的用户定义通常是在控制工作站上创建的，并且 `"supper"` 或 `"pcp"` 用来将 `/etc/passwd`、`/etc/security/passwd`、`/etc/security/user` 和 `/etc/security/group` 分发至所有节点。

- 不要在已调出的卷组和文件系统的 **HACMP** 资源组中配置“要调出的 `nfs_filesystems`”。而是以正常方式将它配置给 **NFS**。**NFS** 服务器的脚本将控制文件系统的调出。
- 确保文件系统所驻留的卷组的主要编号在主节点和替换节点上是相同的。这通过使用带 **-V** 选项的 `importvg` 来完成。
- 验证 `rc.db2pe` 中的 `MOUNT_NFS` 参数是否被设置为 **YES**，并且每个节点在 `/etc/filesystems` 中是否安装有 **NFS** 文件系统。若不是这样，则 `rc.db2pe` 将不能安装该文件系统并启动 **DB2**。
- 若已创建了 **DB2** 实例拥有者，并且正将用户的目录结构复制到正创建的文件系统，则确保对该目录使用 `tar (-cvf)` 命令。这确保预留符号链路。
- 不要忘记对所创建的文件系统的逻辑卷和文件系统日志的适配器和磁盘进行镜像。

NFS 服务器替换配置的示例

此示例假定在 **IP** 地址 `"nfs_server"` 上的卷组 `nfsvg` 中，存在一个 **NFS** 服务器文件系统 `/nfshome`。该 **DB2** 实例名是 `POWERTP`，主目录是 `/dbhome/powertp`。

```
Resource group name: nfs_server
Node Relationship: cascading
Participating nodenames: node1_eth, node2_eth
Service_IP_label: nfs_server      (<<< 这是交换别名地址)
Filesystems: /nfshome
Volume Groups: nfsvg
Application Servers: nfs_server_app
Application Server Start Script: /usr/bin/rc.db2pe powertp NFS SERVER start
Application Server Stop Script: /usr/bin/rc.db2pe powertp NFS SERVER stop
```

在此示例中:

- 在安装 `nfs_server:/nfshome` 时, 所有节点上的 `/etc/filesystems` 都将包含表示 `/dbhome` 的一项。 `nfs_server` 是一个“服务 IP”交换别名地址。
- `nfs_server` 节点和备份节点上的 `/etc/exports` 将包括引导地址和服务地址, 并包含表示 `/nfsfs -root=nfs_switch_1, nfs_switch_2, ...` 的一项。

配置 SP 交换时的考虑事项

当用 SP 交换实现 HACMP ES 时, 考虑下列情况:

- SP 交换上有“基地址”和“别名地址”。基地址是那些在“SP 系统数据仓库”(SDR) 中定义的地址, 并当“引导”系统时由 `rc.switch` 配置。别名地址是除基地址外、通过带别名属性的 `ifconfig` 命令配置的、至 **css0** 接口中的 IP 地址。例如:

```
ifconfig css0 inet alias sw_alias_1 up
```

- 当配置 `DB2 db2nodes.cfg` 文件时, 应对“主机名”和“网络名”字段使用 SP 交换“基本”IP 地址名。交换 IP 地址别名只用来维护 NFS 连接性。使用 **db2start** (RESTART) 命令重新启动 DB2 (这会更新 `db2nodes.cfg`), 来完成 DB2 故障恢复。
- 不要将交换地址与 `etc/hosts` 别名混淆。SP 交换地址和 SP 交换别名地址实际存在于 `etc/hosts` 或 DNS 中。交换别名地址不是 SP 交换基地址的另一个名称; 它们具有各自的独立地址。
- 当一个节点启动时, SP 交换基地址始终存在于该节点上。HACMP ES 不配置或在节点间移动这些地址。
- 若计划使用 SP 交换别名地址, 则将这些地址作为“心跳”和 IP 地址替换的引导地址和服务地址配置给 HACMP。若不想使用 SP 交换别名地址, 则将基本 SP 交换地址仅 (不进行 IP 地址替换) 作为“心跳”的服务地址配置给 HACMP。不要在任何配置中配置别名地址和交换基地址; HACMP ES 不支持此配置。
- 对于 IP 替换配置, 仅在节点之间移动 SP 交换别名地址 (而不移动 SP 交换基地址)。
- 因为每个节点只能有一个 SP 交换适配器, 因此需要 SP 交换别名。使用别名地址允许一个节点替换另一个节点的交换别名 IP 地址, 而不用添加另一个交换适配器。这在“插槽约束”节点中很有用。有关校正 SP 交换适配器故障的详情, 参见第213页的『HACMP ES 脚本文件』下的网络故障一节。
- 若为 IP 地址替换配置 SP 交换, 则需要为每个节点创建 2 个额外的别名 IP 地址: 一个作为引导地址, 另一个作为服务地址。
- 不要忘记在 SP 交换基本 IP 地址或 SP 交换别名 IP 地址的 HACMP ES 网络名定义中使用“HPS”。

- 当启动 HACMP 时，HACMP 中的 `rc.cluster` 自动在 SP 交换引导地址中执行 `ifconfig`。除创建 IP 地址和名称并将它们定义给 HACMP 外，不需要附加的配置。
- SP 交换的 Eprimary 节点是实现 Estart、Efence 和 Eunfence 命令的服务器。HACMP 脚本尝试在启动 HACMP 时对一个节点执行 Eunfence 或 Estart，并使该交换可用（若将该交换定义为其网络中的一员）。因此，确保 Eprimary 节点在启动 HACMP 时可用。在 HACMP 节点由于出错而退出之前，它最多等待 12 分钟以便 Eprimary 故障恢复完成。
- 由“SP 并行系统支持程序”（PSSP）而不是 HACMP 在节点间移动 SP 交换的 Eprimary 节点。若一个 Eprimary 节点脱机，则 PSSP 自动让一个备份节点执行该 Eprimary 节点的功能。该交换网络不受此更改的影响，且保持运行。

DB2 HACMP 配置示例

下列示例举例说明其他的故障恢复支持配置，并显示当发生故障时出现的情况。

如果是 DB2 HACMP 相互替换配置（第202页的图48、第203页的图49和第204页的图50）：

- HACMP 适配器以及 SP 交换别名的引导别名和服务别名是为以太网定义的 - 不影响基地址。记住在 HACMP 网络名中使用“HPS”字符串。
- 通过交换别名，将 `NFS_server/nfshome` 作为 `/dbhome` 安装在所有节点上。
- `db2nodes.cfg` 文件包含 SP 交换基地址。在 DB2 数据库分区（逻辑节点）故障恢复后，`db2start` (RESTART) 命令会更改 `db2nodes.cfg` 文件。
- 不显示 SP 交换别名引导地址。
- 节点可存在于不同的 SP 计算机中。

使用 NFS 故障恢复的 DB2 HACMP 相互替换 - 正常

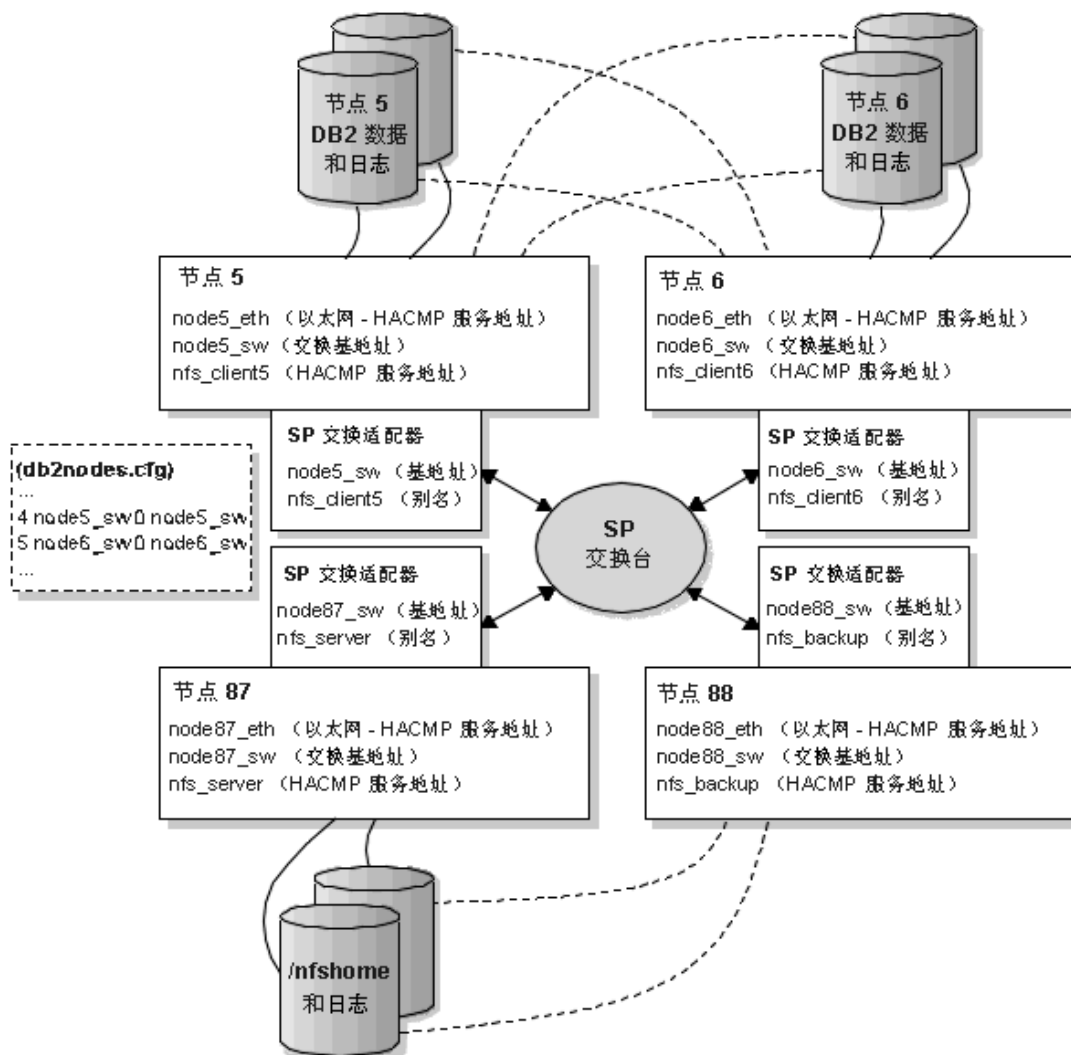


图 48. 使用 NFS 故障恢复的相互替换 - 正常

使用 NFS 故障恢复的 DB2 HACMP 的相互替换 - NFS 故障恢复

- nfs_server SP 开关别名的 IP 地址和 nfs 安装的 /nfshome 从节点 87 移动到节点 88。
- SP 开关 arp 代码具有通过此移动更新所有开关 arp 高速缓存的功能。

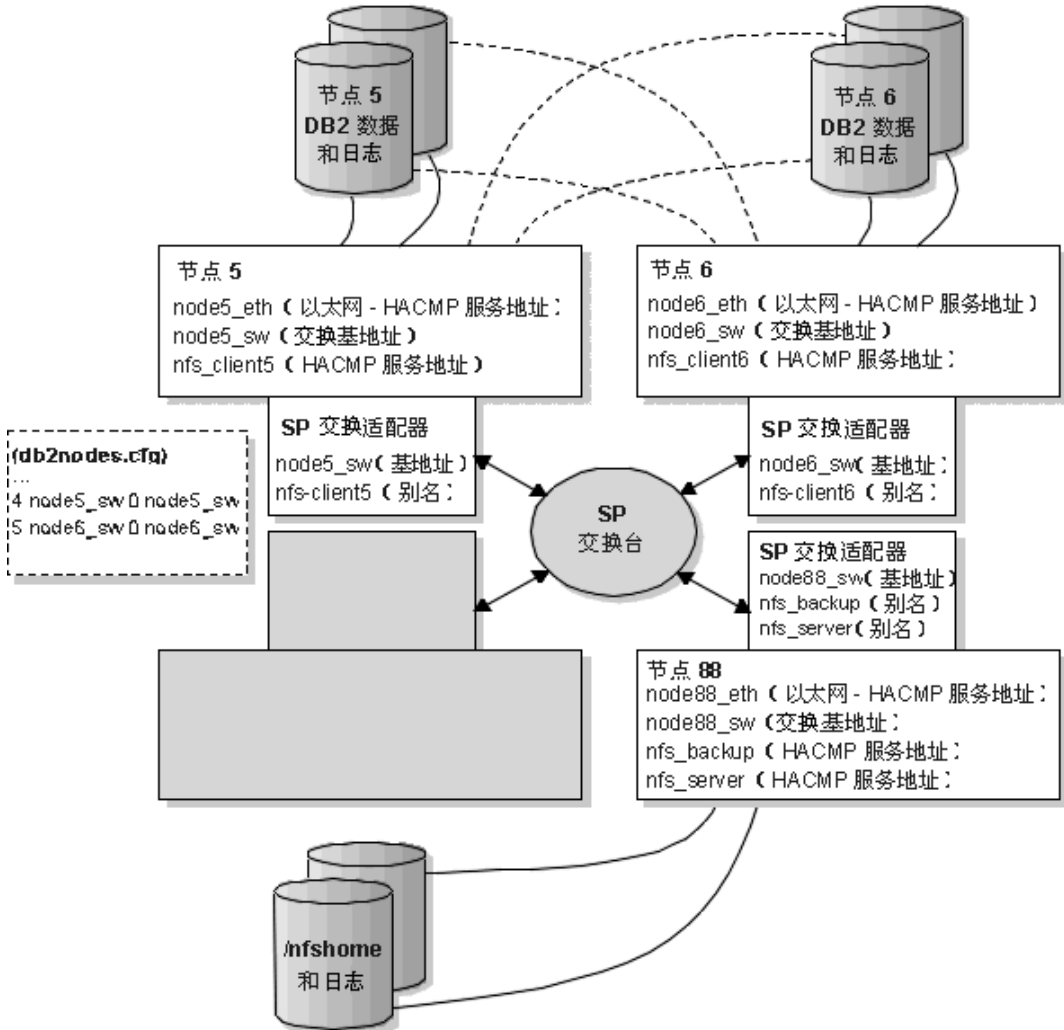


图 49. 使用 NFS 故障恢复的相互替换 - NFS 故障恢复

使用 NFS 故障恢复的 DB2 HACMP 相互替换 - DB2 故障恢复

- 开关 IP 地址替换允许其他服务器（如 ADSM）保持联网
- 节点 5 运行 DB2 的 2 个逻辑节点。

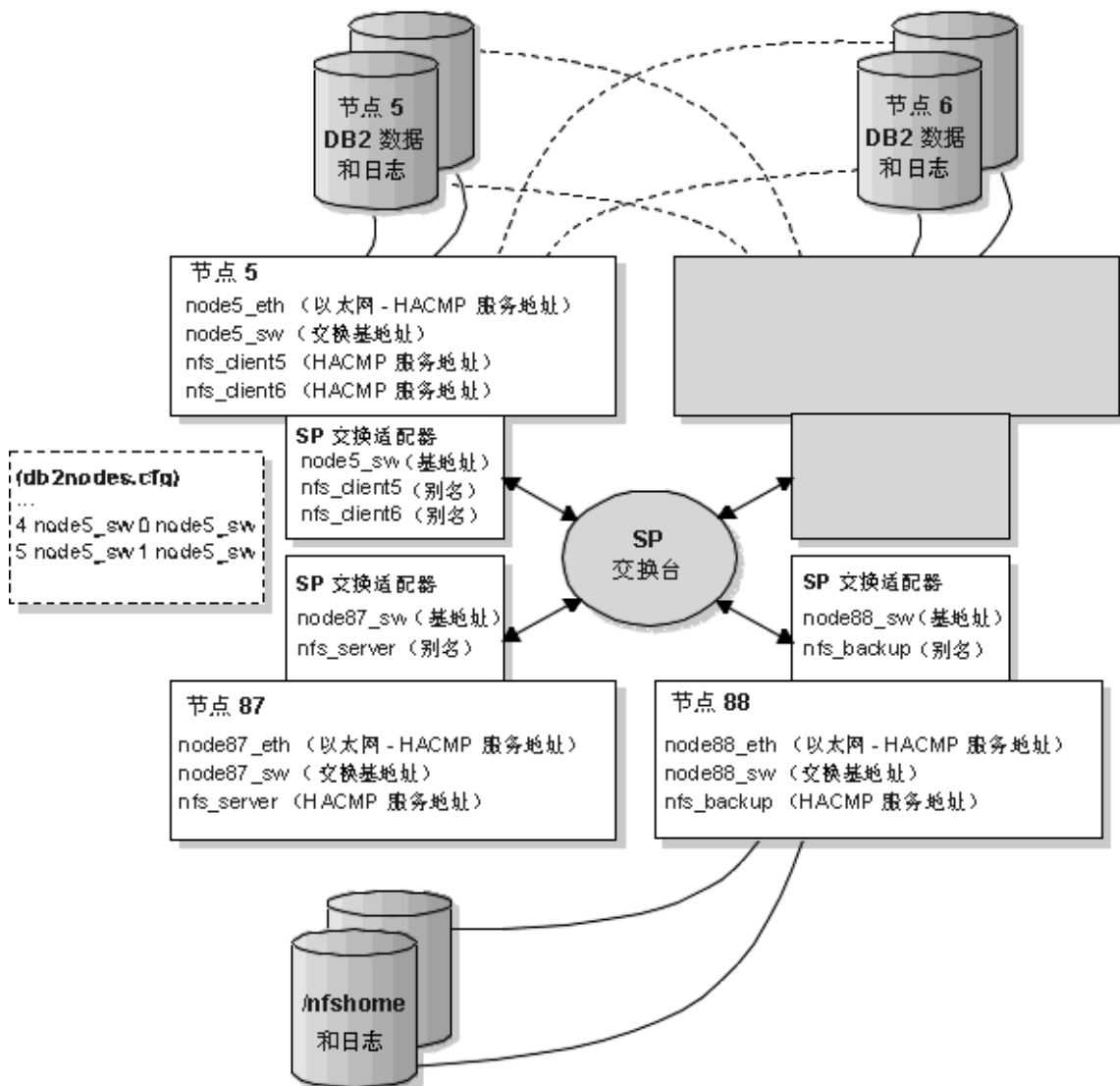


图 50. 使用 NFS 故障恢复的相互替换 - DB2 故障恢复

如果是 DB2 HACMP 热备份配置（第206页的图51和第207页的图52）：

- HACMP 适配器以及 SP 交换别名的引导别名和服务别名是为以太网定义的 - 不影响基地址。记住在 HACMP 网络名中使用 "HPS" 字符串。
- 通过交换别名, 将 NFS_server/nfshome 作为 /dbhome 安装在所有节点上。
- db2nodes.cfg 文件包含 SP 交换基地址。在 DB2 数据库分区 (逻辑节点) 故障恢复后, **db2start** (RESTART) 命令会更改 db2nodes.cfg 文件。
- 不显示 SP 交换别名引导地址。

使用 NFS 故障恢复的 DB2 HACMP 热备份 - 正常

注意：热备份节点可以备份多个节点，取决于磁盘布线。

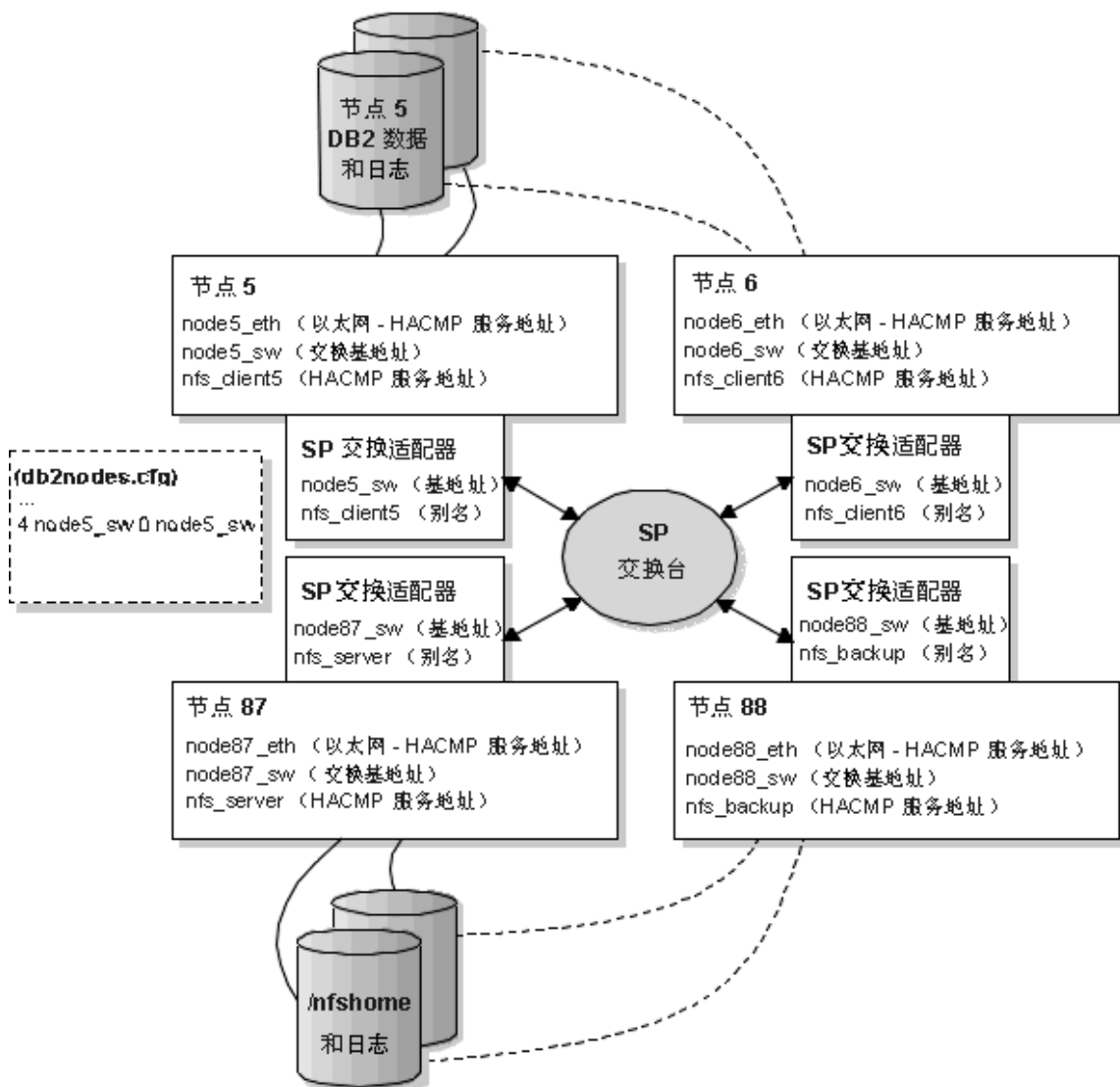


图 51. 使用 NFS 故障恢复的热备份 - 正常

使用 NFS 故障恢复的 DB2 HACMP 热备份 - DB2 故障恢复

注意 热备份节点可备份多个节点，取决于磁盘布线。

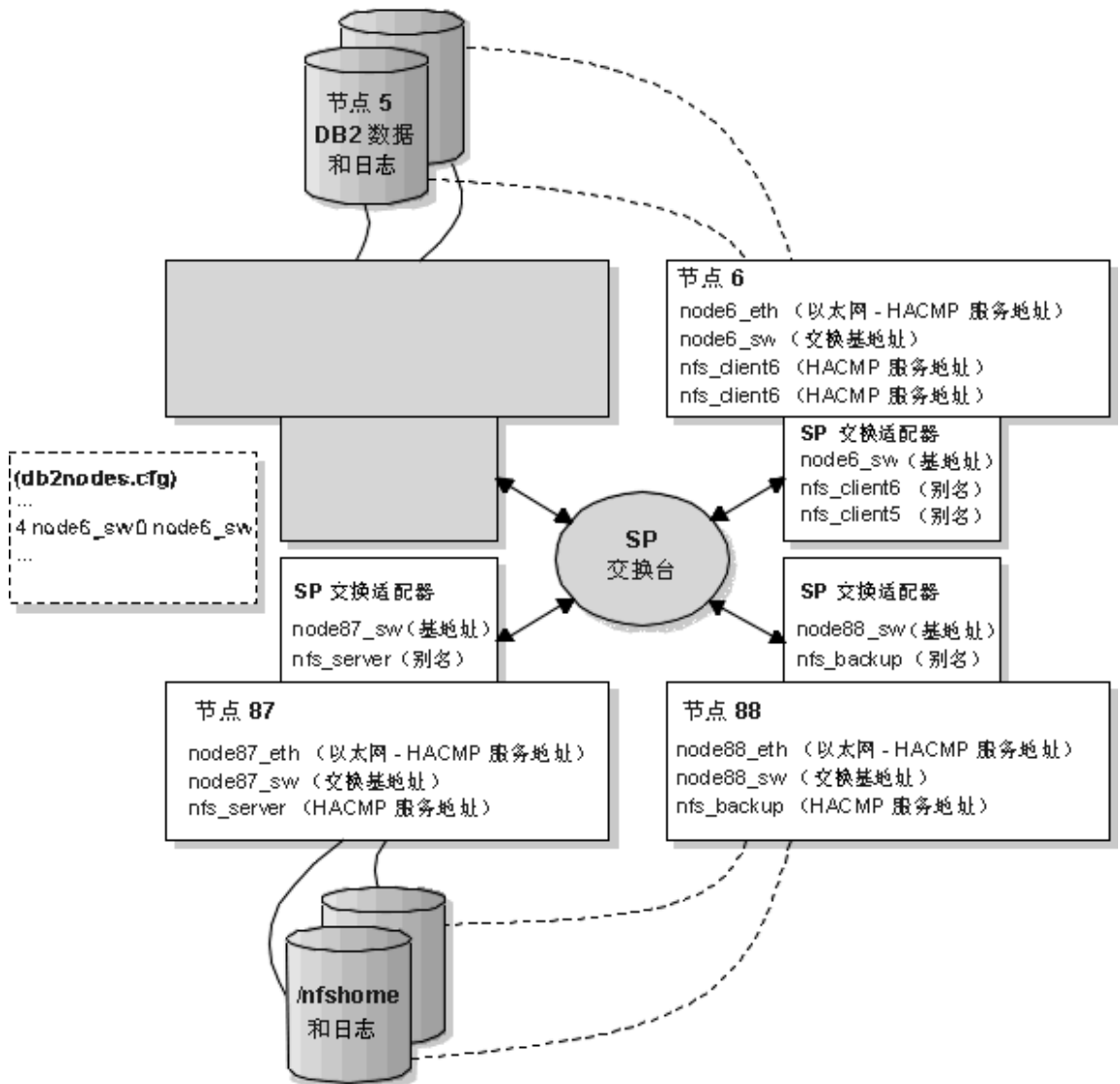


图 52. 使用 NFS 故障恢复的热备份 - DB2 故障恢复

如果是不带 NFS 故障恢复的 DB2 HACMP 相互替换配置（第208页的图53和第209页的图54）：

- HACMP 适配器和 SP 交换基地址是为以太网定义的。记住，当将基地址作为服务地址配置给 HACMP 时，不存在引导地址（只有“心跳”）。不要忘记在 SP 交换的 HACMP 网络名中使用“HPS”字符串。
- db2nodes.cfg 文件包含 SP 交换基地址。在 DB2 数据库分区（逻辑节点）故障恢复后，**db2start** (RESTART) 命令会更改 db2nodes.cfg 文件。
- 不显示 NFS 故障恢复功能。
- 节点可存在于不同的 SP 计算机中。

不使用 NFS 故障恢复的 DB2 HACMP 相互替换 - 正常

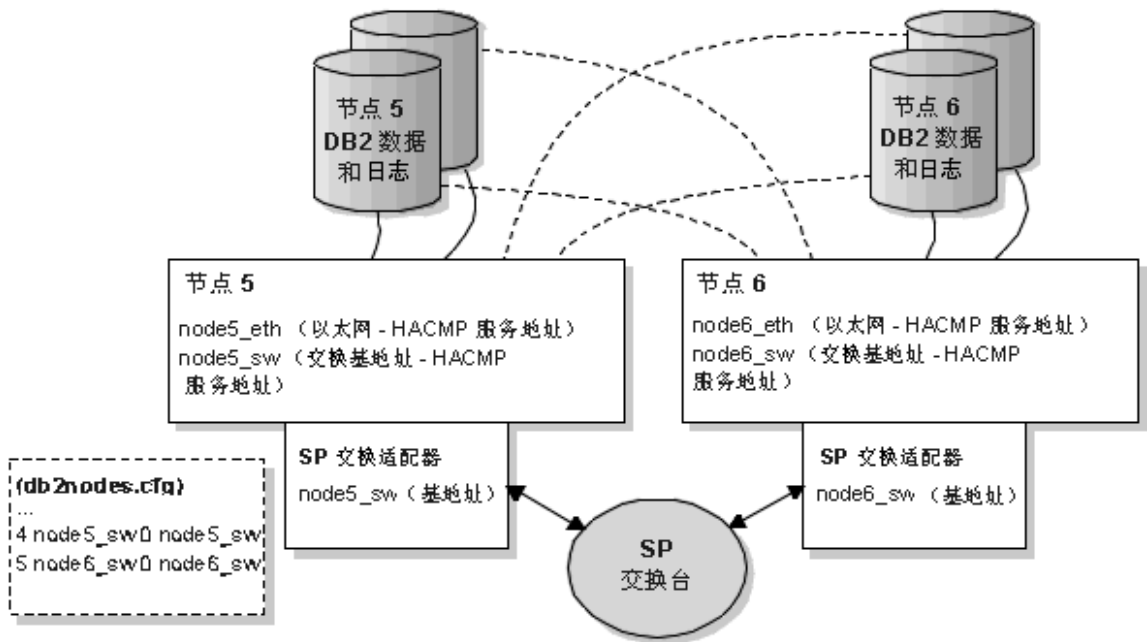


图 53. 不用 NFS 故障恢复的相互替换 - 正常

不使用 NFS 故障恢复的 DB2 HACMP 相互替换 - DB2 故障恢复

- 节点5 运行 DB2 的 2 个逻辑节点。

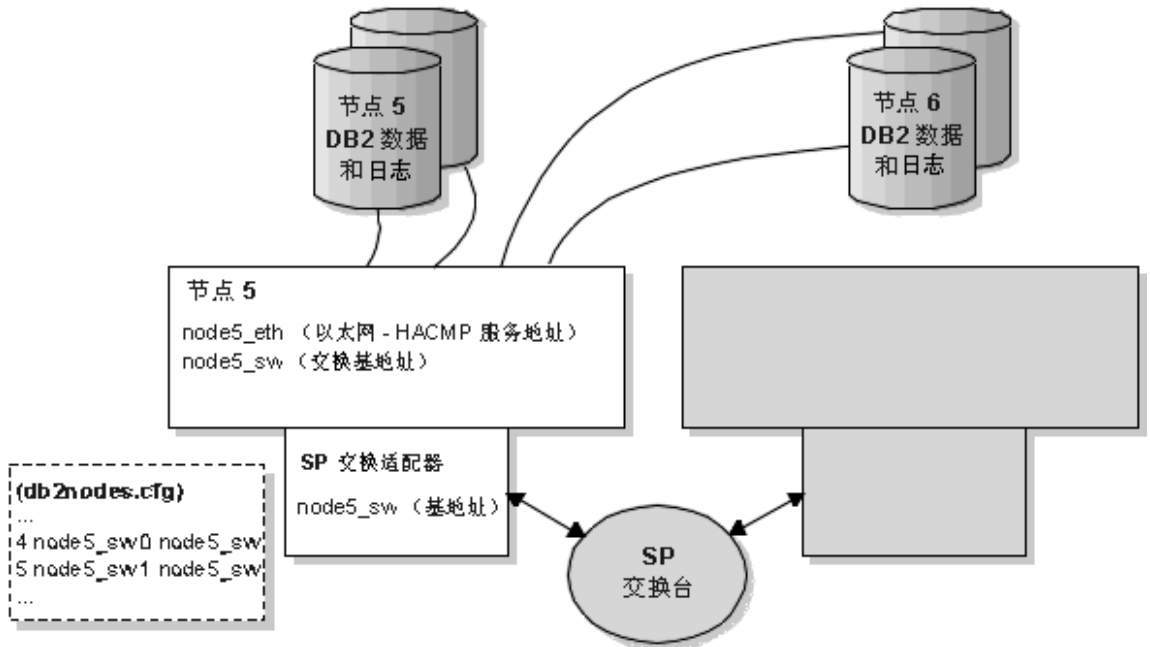


图 54. 不用 NFS 故障恢复的相互替换 - DB2 故障恢复

DB2 HACMP 启动建议

建议不要在 `/etc/inittab` 中指定在引导时启动 HACMP。应在引导这些节点后人工启动 HACMP。这允许对失效的节点进行非中断维护。

以“中断维护”为例，假定一个节点有硬件故障并已崩溃。HACMP 自动启动故障恢复，恢复成功完成。然而，需要修正失效的节点。若 `/etc/inittab` 中将 HACMP 配置为在重新引导时启动，则引导完成后，此节点将尝试重新集成，在此情况下，这并不是所期望的。

对于“非中断维护”，假定在每个节点上人工启动 HACMP。借此，可修正和重新集成失效节点，而不会影响其他节点。提供 `ha_cmd` 脚本，以便从控制工作站控制 HACMP 启动和停止命令。

注：第一次创建 DB2 实例时，下一项被追加至 `/etc/inittab` 文件：

```
rcdb2:2:once:/etc/rc.db2 > /dev/console 2>&1 # Autostart DB2 Services
```

若启用 HACMP 或 HACMP ES，则通过在 HACMP 项前面加上上面这一行来更新 /etc/inittab 文件。下面是 /etc/inittab 文件中的一个样本 HACMP 项：

```
clinit:a:wait:touch /usr/sbin/cluster/.telinit # HACMP for AIX
```

此项必须是 /etc/inittab 文件中的最后一项。

HACMP ES 事件监控和用户定义事件

下面是用户定义事件的两个示例：在调页空间达到某一填充百分比时关闭 AIX 物理节点上的一个 DB2 数据库分区，并重新启动 DB2 数据库分区，或在某进程在给定节点上结束时启动故障恢复操作。可以在 samples 子目录中找到说明用户定义事件的示例，如关闭数据库分区并强制事务异常终止来释放调页空间。

规则文件 /user/sbin/cluster/events/rules.hacmprd 包含 HACMP 事件。此文件中的每个事件说明都有以下 9 个部件：

- 事件名，它必须是唯一的。
- 状态，或事件的限定符。事件名和状态是规则触发器。仅当“HACMP ES 群集管理器”找到具有与该事件名和状态对应的触发器的规则时，它才启动恢复。
- 资源程序路径，这是包含恢复程序的 xxx.rp 文件的完整路径说明。
- 恢复类型。它被保留以备后用。
- 恢复级别。它被保留以备后用。
- 资源变量名，它用于“事件管理器”事件。
- 实例向量，它用于“事件管理器”事件。这是一组格式为“名称=值”的元素。这些值唯一地标识系统中该资源的副本，并用扩展名唯一标识该资源变量的副本。
- 谓词，它用于“事件管理器”事件。这是资源变量与其他元素之间的关系表达式。当此表达式为真时，“事件管理”子系统生成一个事件以通知“群集管理器”和适当的应用程序。
- Rearm 谓词，它用于“事件管理器”事件。这是用来生成改变主谓词状态的事件的谓词。此谓词一般与主要谓词相反。它也可与该事件谓词一起使用来为您感兴趣的条件建立上限和下限。

每个对象在事件定义中都需要一行，即使不使用该行。若除去了这些行，则“HACMP ES 群集管理器”不能正确地对该事件定义进行语法分析，这可能会导致系统挂起。将任何以“#”开始的行视为注释行。

注：该规则文件要求每个事件定义只能有九行（未计算任何注释行）。当在该规则文件的底部添加一个用户定义事件时，除去该文件末尾不需要的空行很重要，否则该节点将挂起。

下面是 `node_up` 的事件定义的一个示例：

```
##### Beginning of the Event Definition: node_up
#
TE_JOIN_NODE
0
/usr/sbin/cluster/events/node_up.rp
2
0
# 6) Resource variable - only used for event management events

# 7) Instance vector - only used for event management events

# 8) Predicate - only used for event management events

# 9) Rearm predicate - only used for event management events

##### End of the Event Definition: node_up
```

此示例仅是在可以在 `rules.hacmprd` 文件中找到的其中一个事件定义。在此示例中，当发生 `node_up` 事件时，将调用恢复程序 `/usr/sbin/cluster/events/node_up.rp`。对状态、恢复类型和恢复级别指定了值。有 4 个空行表示：资源变量、实例变量、谓词和 `rearm` 谓词。

您可定义其他事件，以便对非标准 HACMP ES 事件作出反应。例如，要定义 `/tmp` 文件系统超过 90% 负载的事件，必须修改 `rules.hacmprd` 文件

在“IBM 并行系统支持程序” (PSSP) 中预定义了许多事件。这些事件可被利用（当在用户定义事件内使用时），如下所示：

1. 停止该群集。
2. 编辑 `rules.hacmprd` 文件。在修改该文件之前备份它。人工添加预定义的 PSSP 事件。若需要在该群集中的所有节点将点同步化，则在恢复程序中使用 **barrier** 命令。（在 *HACMP Concepts, Installation, and Administration Guides* 中有关于 **barrier** 命令和恢复程序的同步化的更多信息。）
3. 重新启动该群集。当启动“群集管理器”时，将 `rules.hacmprd` 文件存储在内存中。要准确地实现这些更改，重新启动所有的群集。群集中不应存在任何不兼容的规则。
4. “群集管理器”使用 `rules.hacmprd` 文件中的所有事件。

HACMP ES 使用 PSSP 事件检测以处理用户定义事件。“PSSP 事件管理”子系统通过监控不同硬件和软件资源来提供综合的事件检测。

资源状态由资源变量表示。资源条件表示为称为谓词的表达式。

“事件管理”从“资源监控程序”接收资源变量，该监控程序观察特定系统资源的状态，并将此状态转换为几个资源变量。定期将这些变量传送到“事件管理”。

“事件管理”应用“HACMP ES 群集管理器”在 `rules.hacmprd` 中对每个资源变量指定的谓词。当评估该谓词为真时，会生成一个事件并将该事件发送至“群集管理器”。“群集管理器”启动该表决协议，并在由恢复程序中的“节点集”指定的一组节点上（根据事件优先级）运行该恢复程序文件 (`xxx.rp`)。

该恢复程序文件 (`xxx.rp`) 由一个或多个恢复程序行组成。每一行以下列格式说明：

```
relationship      command_to_run    expected_status   NULL
```

在该行中，每个值之间必须至少有一个空格。“Relationship”是一个值，用来决定哪个程序应在哪种节点上运行。支持三种关系：

- 全部。在当前 HACMP 群集的所有节点上运行指定的命令或程序。
- 事件。仅在发生该事件的节点上运行指定的命令或程序。
- 其他。在所有未发生该事件的节点上运行指定的命令或程序。

“Command_to_run”是一个使用引号定界的字符串，带或不带可执行程序的完整路径说明。仅有 HACMP 传递的事件脚本才可使用相对路径定义。其他脚本或程序必须使用完整路径说明，即使它们与 HACMP 事件脚本位于相同的目录中。

“Expected_states”是指定的命令或程序的返回码。它或者是整数值，或者是“x”。若使用“x”，则表示“群集管理器”不关心该返回码。所有其他代码都必须与期望的返回码相等，否则“群集管理器”将检测到事件故障。此事件的处理“挂起”该进程，直到发生恢复（通过人工干预）为止。若不进行人工干预，该节点就不能与其他节点同步。所有节点同步是“群集管理器”控制所有节点的必要条件。

“NULL”是被保留以备后用的字段。单词“NULL”必须出现在每行（障碍行除外）的末尾。若在两个障碍命令之间或在第一个障碍命令之前指定多个恢复命令，则在该节点自身上和节点之间以并行方式运行这些恢复命令。

该障碍命令用来将所有群集节点中的所有命令同步。当一个节点命中该恢复程序中的障碍语句时，“群集管理器”启动此节点上的障碍协议。因为障碍协议是一个两阶段协议，当所有节点都已遇到该恢复程序中的障碍，并且已“表决”赞成该协议时，则通知所有节点两阶段已完成。

此过程概述如下：

1. “组服务 / ES”（对于预定义事件）或“事件管理”（对于用户定义事件）将该事件通知“HACMP ES 群集管理器”。
2. “群集管理器”读取 `rules.hacmprd` 文件并确定映射至该事件的恢复程序。

3. “群集管理器”运行由一系列恢复命令组成的恢复程序。
4. 恢复程序执行可能是 shell 脚本或二进制命令的恢复命令。（在 HACMP AIX 版中，恢复命令与 HACMP 事件脚本相同。）
5. “群集管理器”从恢复命令接收返回状态。一个意外的状态会“挂起”该群集，直到进行人工干预（使用 `smit cm_rec_aids` 或 `/usr/sbin/cluster/utilities/clruncmd` 命令）为止。

HACMP ES 脚本文件

DB2 UDB EEE 附带包括了以下用于故障恢复和用户定义事件的样本脚本。这些脚本文件位于 `$INSTNAME/sqlllib/samples/hacmp/es` 目录中。这些脚本将“按原样”运行，您也可定制恢复操作。

- DB2 数据库分区恢复脚本 `rc.db2pe`。这是用于启动和停止数据库分区中的 HACMP 配置的脚本文件。它还充当 DB2 实例拥有者的 NFS 服务器的 HACMP 启动和停止脚本。
- HACMP ES 的 DB2 专用的用户定义事件。包括六个缺省事件：一个用于进程恢复，两个用于调页空间，三个用于 NFS 和自动安装程序恢复。
- DB2 实例 NFS 文件服务器故障恢复。此脚本提供 DB2 实例的文件系统服务器至备份的故障恢复。
- 网络故障恢复。脚本 `network_up_complete`、`network_back`、`network_down_complete` 和 `network_down` 允许 SP DB2 数据库分区在其 SP 交换适配器失效时执行故障恢复。
- 定义“SP GUI 透视”的监控事件的脚本。通过“事件和硬件透视”，可以监控故障恢复和用户定义恢复。阅读“PSSP 管理”的文档，以了解有关“透视”的更多信息。
- 安装和除去 HACMP ES 节点上的核心脚本和事件的安装脚本。
- 脚本文件，用来创建和除去用于监控 HACMP 和 DB2 配置的“SP 透视”问题管理 (pman) 资源。

必须在将运行恢复操作的每个节点上安装该恢复脚本。脚本文件可从 SP 控制工作站或其他指定的 SP 节点集中安装：

1. 将这些脚本从 `$INSTNAME/sqlllib/samples/hacmp/es` 目录复制到可运行 `pcp` 和 `pexec` 命令的 SP 控制工作站或另一个 SP 节点。这些命令是安装操作所必需的。
2. 通过为故障恢复配置设置关键字参数（如 `BUFFPAGE`）来为环境定制 `reg.parms.SAMPLE` 和 `failover.parms.SAMPLE` 文件。通常情况下，对于相互替换配置，会将故障设置调低至常规设置大小的一半或更小。而且，将使用自己的名称重命名的这些文件的副本（取代“SAMPLE”）。

3. 定制 `rc.db2pe` 文件中的 5 个参数：`NFS_RETRIES`、`START_RETRIES`、`MOUNT_NFS`、`STOP_RETRIES` 和 `FAILOVER`（如有必要的话）。重试和故障恢复设置应能满足大多数实现的需要。应根据您是否将使用 NFS 服务器可用性软件包来配置 `MOUNT_NFS` 设置。若希望 `rc.db2pe` 为您安装并验证 DB2 实例拥有者的 NFS 主目录，应指定此设置。将 `FAILOVER` 参数设置为 "YES" 将调用 `db2_proc_restart`，并尝试重新启动 DB2 数据库分区。若重新启动操作不成功，则 HACMP 将关闭，并进行故障恢复。

4. 定制事件文件中的 `db2_paging_action`、`db2_proc_recovery` 和 `nfs_auto_recovery`。编辑 `pwq`，将它更改为 DB2 实例所有者。定制 `db2_paging_action`，以指定当调页空间填充程度超过 90% 时要执行的操作。（若真的发生此情况，则停止 DB2 数据库分区。）若需要附加的恢复操作，修改该脚本。

5. 使用 `db2_inst_ha` 在指定的节点上安装这些脚本和事件。（在开始之前，必须在这些节点上预安装 HACMP ES。）`db2_inst_ha` 的语法是：

```
db2_inst_ha $INSTNAME/sqllib/samples/hacmp/es <nodelist> <DATABASENAME>
```

其中，

```
$INSTNAME/sqllib/samples/hacmp/es 是脚本和事件所在的目录  
<nodelist> 是节点的 pcp 或 pexec 式样；例如，1-16  
或 1、2、3、4  
<DATABASENAME> 是规则参数文件和故障恢复参数文件的数据库名。
```

将 `reg.parms.SAMPLE` 和 `failover.parms.SAMPLE` 文件复制到每个节点并重命名为 `reg.parms.DATABASENAME`。`db2_inst_ha` 将文件复制至 `/usr/bin` 中的每个节点，并更新 HACMP 事件文件：

```
/usr/sbin/cluster/events/rules.hacmprd  
/usr/sbin/cluster/events/network_up_complete  
/usr/sbin/cluster/events/network_down_complete
```

6. 用 HACMP 配置系统和脚本。

7. 使用 `create_db2_events` 命令为问题管理资源 (pman) 和“SP GUI 透视”安装监控事件。“透视”中需要附加的配置和定制。有关“透视”的详情，阅读 `PSSP Administration Guide`。

8. 使用 `ha_db2stop` 命令，可关闭数据库分区，而无需进行 HACMP ES 故障恢复。要使用此命令，将该文件复制到数据库用户的主目录中，并确保为该用户设置了许可权和拥有权。要停止数据库而不进行故障恢复，则可作为该用户来输入：

```
ha_db2stop
```

注: 必须等待该命令返回。使用 `ctrl-C` 中断或通过停止进程来退出, 可能会导致过早地再次启用故障恢复, 并且, 某些数据库分区可能不停止。

使用 HACMP ES 执行 DB2 恢复脚本操作

HACMP ES 以下列方式调用 DB2 恢复脚本:

- `node_up_local` (启动节点)

HACMP 运行 `node_up` 序列, 以获取在此节点拥有的 (通过级联) 或赋予此节点的 (通过旋转) 资源组中指定的卷组、逻辑卷、文件系统和 IP 地址。

当运行 `node_up_local_complete` 时, 启动包含 `rc.db2pe` 的应用服务器定义, 以启动在此物理节点上的应用服务器定义中指定的数据库分区。

注: 当 `rc.db2pe` 以启动方式运行时, 它调整在 `reg.parms.DATABASE` 中对与参数 (`parms`) 文件相匹配的数据库目录中的每个 `DATABASE` 指定的 DB2 参数。

每个节点在启动时都遵循此序列。若有多个 HACMP 群集且并行启动它们, 则一次启动多个节点。

- `node_down_remote` (故障恢复)

HACMP 获取在指定的替换节点上的资源组中指定的卷组、逻辑卷、文件系统和 IP 地址。

当运行 `node_down_remote_complete` 时, HACMP 将把 `rc.db2pe` 作为在此数据库分区的资源组中指定的应用服务器启动脚本来运行。

注: 当 `rc.db2pe` 以相互替换方式运行时, 它停止在其上运行的 DB2 数据库分区, 调整 `failover.parms.DATABASE` 中对与参数 (`parms`) 文件相匹配的数据库目录中的每个数据库指定的 DB2 参数, 然后启动该物理替换节点上的两个数据库分区。

- `node_up_remote` (重新集成失效节点 - 级联相互替换资源组)

当在旧的替换节点上运行 `node_up_remote` 时, 该应用服务器定义导致 `rc.db2pe` 以停止方式运行。

注: 当 `rc.db2pe` 以重新集成方式 (相互替换) 运行时, 它停止在其上运行的两个数据库分区, 调整在 `failover.parms.DATABASE` 中对与参数 (`parms`) 文件相匹配的数据库目录中的每个数据库指定的 DB2 参数, 然后只启动此物理替换节点上要保留的数据库分区。

旧的替换节点释放在重新集成节点要拥有的资源组中指定的卷组、逻辑卷、文件系统和 IP 地址。

HACMP 重新获取在重新集成节点现在拥有的资源组中指定的卷组、逻辑卷、文件系统和 IP 地址。

当运行 `node_up_local_complete` 时，启动包含 `rc.db2pe` 的应用服务器定义，以启动在此重新集成的物理节点上的应用服务器定义中指定的 DB2 数据库分区。

注：当 `rc.db2pe` 以启动方式运行时，它调整在 `reg.parms.DATABASE` 中对与参数 (`parms`) 文件相匹配的数据库目录中的每个 `DATABASE` 指定的 DB2 参数。

- `node_down_local` (节点停止或由于替换停止)

当在正在停止的节点上运行 `node_down_local` 时，该应用服务器定义导致 `rc.db2pe` 以停止方式运行。

注：当 `rc.db2pe` 以停止方式运行时，它调整 `failover.parms.DATABASE` 中对与参数 (`parms`) 文件相匹配的数据库目录中的每个 `DATABASE` 指定的 DB2 参数，然后停止该数据库分区 (这用于替换)。

HACMP 释放在该节点现在拥有的资源组中指定的卷组、逻辑卷、文件系统和 IP 地址。

- `db2_proc_recovery` (db2 进程停止)

所有节点都运行 `db2_proc_restart` 脚本。发生故障的节点重新启动正确的 DB2 数据库分区。

- `db2_paging_recovery` (调页空间恢复)

所有节点都运行 `db2_paging_action` 脚本。若一个节点的调页空间被填充了 70% 以上，则发出 `wall` 命令。若一个节点的调页空间被填充了 90% 以上，则停止并重新启动此物理节点上的 DB2 数据库分区。

- `nfs_auto_recovery` (nfs 或自动安装进程失败)

所有节点都以 NFS 方式运行 `rc.db2pe` 脚本。若一个 NFS 进程停止运行，则重新启动它。类似地，若自动安装过程停止运行，则重新启动它。

- `network_down_complete` (网络故障 - SP 交换)

调用 `net_down` 脚本。这验证该网络是否为 SP 交换网络，并验证它是否已关闭。若是这样，则它等待用户定义的时间间隔。缺省时间间隔是 100 秒。

若该 SP 交换网络重新启动 (由 `network_up_complete` 事件指示)，则没有恢复生效。

若达到时间限制，则用故障恢复停止 HACMP。

注：可通过 SP 问题管理和“SP 透视 GUI”监控所有事件。

其他脚本实用程序

您还可使用其他脚本实用程序，包括：

- `ha_cmd`，这是为了从控制工作站启动 SP 节点上的 HACMP 而提供的命令。语法是：

```
ha_cmd <noderange> <START|STOP|TAKE|FORCE>
```

其中，

<noderange> 是 SP 节点范围的 `pcp` 或 `pexec` 式样。

例如，"`ha_cmd 3-6 START`" 将在节点 3、4、5 和 6 上启动 HACMP。

"`ha_cmd 5 TAKE`" 将在节点 5 上为相互替换关闭 HACMP。

- `ha_mon`，这是用于从 SP 控制工作站监控 HACMP `hacmp_out` 文件的命令。语法是：

```
ha_mon <node>
```

其中，

<node> 是要监控的 SP 节点。

`ha_mon` 将对您指定的节点上的 `/tmp/hacmp.out` 文件执行 "`tail -f`" 操作。

- `db2_turnoff_recov`，这是用于临时禁用所有 HACMP 恢复（非故障恢复）的命令，它是专门为极为罕见的情况设计的。不启动 DB2 进程、调页、NFS 和自动安装程序恢复。此功能从 HACMP 规则文件中除去该恢复的事件节。必须停止并重新启动 HACMP。语法是：

```
db2_turnoff_recov <nodelist>
```

- `db2_turnon_recov`，这是用于重新启用 HACMP（非故障恢复）恢复的命令。此命令在 `db2_turnoff_recov` 后使用，以复原 HACMP 规则文件，这样可恢复用户定义事件。必须停止并重新启动 HACMP。语法是：

```
db2_turnon_recov <nodelist>
```

监控 HACMP 群集

除了在 HACMP ES 中已存在的那些监控实用程序外，还提供了一些脚本，用于创建 SP 问题管理 (pman) 事件以监控 DB2 HACMP ES 配置。要从 SP 控制工作站监控 HACMP 状态：

- 在该控制工作站上安装 HACMP 客户机代码。
- 编辑 `/usr/sbin/cluster/etc/clhosts` 文件，并包括您想要监控的这些节点的 SP 以太网 IP 地址。
- 调用命令 `startsrc -s clinfo` 来启动监控群集。

HACMP 提供了一个用于监控群集的接口 (`/usr/sbin/cluster/clstat`)。

要使用“用于 HACMP RS 的 SP 透视 GUI”和用户定义事件来执行问题管理监控:

1. 调用 `create_db2_events <nodelist>`, 其中, *nodelist* 包含 `pcp` 或 `pexec` 样式节点。此脚本创建由“透视”监控的 5 个 `pman` 事件。

注: 在创建这些事件时使用资源变量 `PSSP.pm.User_state12-16`。若这些资源变量已有其他用途, 则必须更新 `create_db2_events` 和 `update_db2_events`, 以使用不同的资源变量。

2. 在控制工作站上启动“透视”。从快速启动板, 选择事件透视。您应该能看到 5 个事件: `db2_hacmp_recovery`、`db2_process_recovery`、`db2_paging_err`、`db2_nfs_err` 和 `Errlog_PERM_entry`。
3. 双击每个事件。在出现的屏幕上, 注册(在“定义表”内)该事件的条件。单击名称: “未命名”旁边的向下箭头, 并选择与您指定为条件的事件相同的名称。选择“应答选项”标签。单击屏幕顶部的按钮(“将信息发送至透视事件对话”)。可指定命令、错误日志项以及 `SNMP` 软中断来触发这些事件。仅通过“透视”对话维护事件日志屏幕; 因此, 可能要为每个对话创建 `AIX` 错误日志项。选择**确认**, 并关闭该窗口。
4. 从“透视”从快速启动板, 选择硬件“透视”。
5. 当硬件框架 GUI 出现时, 选择“视图”, 然后选择“监控”。将提供对于 `SP` 可监控的事件列表。滚动至该列表的底部, 将看到两个附加的事件: 一个用于 `HACMP DB2` 恢复 (`db2_ha_ind`), 另一个用于 `SP` 节点 `PERM` 错误 (`Errlog_PERM_mon`)。选择您想要监控的那些事件。(当一个事件发生时, 节点显示一个红色的“X”。若所有被监控条件都良好, 则节点屏幕是绿色的。)通常情况下, 使用 `host_responds`、`switch_responds` 和 `node_power_LED`。也可监控该节点上的 `DB2 HACMP` 恢复以及 `PERM` 错误。

注: 用于 `pman` 和“透视”的 `db2_hacmp_mon` 和 `db2_hacmp_recovery` 变量不反映 `HACMP` 群集状态。相反, 这些变量反映启动或停止 `DB2` 的 `rc.db2pe` 操作的状态。“真实的”`HACMP` 状态显示在 `HACMP clstat` 监控程序中, 该状态反映 `HACMP` 群集状态。若想要让 `db2_hacmp_ind` 反映类似于 `HACMP` 状态的监控, 则将下面这一行添加至 `/etc/inittab` 文件:

```
haind:2:wait:/usr/bin/db2_update_events HAIND OFF 2>&1 >/dev/null
```

若正计划将 `NetView` 用于您的实现, 考虑使用 `HAVIEW` (它是 `HACMP` 的一部分) 来监控配置。有关配置该产品的详情, 参考 `NetView` 文档。

DB2 SP HACMP ES 安装

为了帮助您规划配合“DB2 通用数据库”安装 HACMP ES，下面提供了安装和迁移过程的逐步概述。

首次安装 DB2 SP HACMP ES

要安装 HACMP ES:

1. 在每个 SP 节点上安装 AIX 操作系统（参考 SP Installation and Administration Guides）。确保在控制工作站和每个 SP 节点上都有适当的调页空间可用。确保已考虑并实现了交换配置，以及任何其他可修改的配置参数。适当的设置您想要使用的 SP 监控（透视）。确保 SP dsh、pcp 和 pexec 命令运行。
2. 设计数据库布局。这至少应包括要使用的节点数、DB2 数据库分区至物理节点的映射、每个节点或分区的磁盘需求和表空间考虑事项。还应考虑主 DB2 实例拥有者将是谁，以及此用户和其他用户将需要什么存取授权。
3. 计划外部 SSA 磁盘配置，包括冗余适配器、镜象磁盘和双尾磁盘。
4. 使用数据库布局和 SSA 配置，填写 HACMP Planning, Installation, and Administration Guides 中的 HACMP 工作表。
5. 实现外部 SSA 磁盘配置。确保微代码级别在所有驱动器上是一致的，并使用 Maymap 实用程序验证并填充工作表中的任何间隙。
6. 在每个 SP 节点上安装 DB2 UDB EEE。
7. 在每个 SP 节点上安装 HACMP ES。
8. 使用 **db2_inst_ha** 命令安装“DB2 UDB EEE HACMP ES on SP 程序包”。
9. 创建 DB2 主实例用户，并确保它可以存取所有节点。在这方面，这是一个可用性不高的用户。这可能是 SP 控制工作站上的一个临时 SP 用户。
10. 创建 DB2 实例和数据库。在转至下一步之前，通过调用 **db2start**，然后调用 **db2stop** 来确保该实例和数据库可操作。
11. 若想要在添加 HACMP 之前装入数据库，现在就应执行此操作。
12. 根据 HACMP 工作表和此文档中的信息，在 SP 节点拓扑和资源组上配置 HACMP ES。
13. 以 DB2 主实例用户的 NFS 服务器节点为起点，根据在此文档中的指定，在所有节点上更改此用户（通过修改 `/etc/security/user` 和 `/etc/passwd`）。此用户将成为可用性很高的 NFS 用户；并且此节点和其备份将更新 `/etc/exports`。通过交换别名 IP 地址，所有节点都将可以使用 NFS 来安装此目录（在每个节点上的 `/etc/filesystems` 中具有一项）。
14. “打包”该主实例用户的主目录，并在新位置“解包”该主目录。
15. 在每个 SP 节点上创建一个 NFS 文件系统，以安装新的主实例主目录。

16. 在 NFS 服务器节点上启动 HACMP。通过查看 /tmp/hacmp.out，验证它已成功启动。当写入此文件时，可使用 **ha_mon** 命令来监控它。
17. 一次启动一个其他节点，通过查看 /tmp/hacmp.out 来验证每个节点是否已成功启动。当写入此文件时，可使用 **ha_mon** 命令来监控它。
18. 通过“透视和问题管理”设置可选的监控。
19. 通过在每个节点上模拟并行维护操作，来验证每个节点上的故障恢复功能。可使用 **ha_cmd** 命令（指定 TAKE 选项）来通过替换平缓停止 HACMP。通过查看 /tmp/hacmp.out 和使用监控工具来验证替换和重新集成是否成功。

DB2 SP HACMP ES 迁移

若要从一个非 HACMP 安装迁移至带 HACMP 的安装，则考虑下列概述：

1. 将现存的外部磁盘转换为高度可用的、双尾的、镜象配置。添加任何额外的硬件和磁盘以完成此配置，记住，当不同节点上的不同逻辑卷为双尾时，它们的名称必须是唯一的。这适用于卷组、逻辑卷和文件系统。
2. 完成该 HACMP 计划和相关的工作表，包括本文档中的工作表。
3. 实现外部 SSA 磁盘配置更改。确保微代码级别在所有驱动器上是一致的，并使用 Maymap 实用程序验证并消去工作表中的任何间隙。

注：支持 RAID5 配置中的 SSA 磁盘。在同一个 RAID 回路中存在两个 SSA 适配器是唯一允许的配置。对于具有 RAID 磁盘双尾端的 HACMP 配置，只支持每个节点一个适配器。在这种配置中，该适配器是存取磁盘的单个故障点，建议用额外的配置检测适配器的运转中断问题，并借助于它来完成 HACMP 故障恢复事件。AIX 出错通知是在 SSA 适配器失效时配置一个节点用于故障恢复的最简单方法。有关 AIX 错误通知的详情，参考 *HACMP for AIX, V4.2.2, Enhanced Scalability Installation and Administration Guide*。

4. 在每个 SP 节点上安装 HACMP ES。
5. 使用 **db2_inst_ha** 命令安装 DB2 UDB EEE HACMP ES on SP 程序包。
6. 根据 HACMP 工作表和此文档中的信息，在 SP 节点拓扑和资源组上配置 HACMP ES。
7. 以 DB2 主实例用户的 NFS 服务器节点为起点，根据在此文档中的指定，在所有节点上更改此用户（通过修改 /etc/security/user 和 /etc/passwd）。此用户将成为可用性很高的 NFS 用户；并且此节点和其备份将更新 /etc/exports。通过交换别名 IP 地址，所有节点都将可以使用 NFS 来安装此目录（在每个节点上的 /etc/filesystems 中具有一项）。
8. “打包”该主实例用户的主目录，并在新位置“解包”该主目录。

9. 在每个 SP 节点上创建一个 NFS 文件系统，以安装新的主实例主目录。
10. 在 NFS 服务器节点上启动 HACMP。通过查看 /tmp/hacmp.out，验证它已成功启动。当写入此文件时，可使用 **ha_mon** 命令来监控它。
11. 一次启动一个其他节点，通过查看 /tmp/hacmp.out 来验证每个节点是否已成功启动。当写入此文件时，可使用 **ha_mon** 命令来监控它。
12. 通过“透视和问题管理”设置可选的监控。
13. 通过在每个节点上模拟并行维护操作，来验证每个节点上的故障恢复功能。可使用 **ha_cmd** 命令（指定 TAKE 选项）来通过替换平缓停止 HACMP。通过查看 /tmp/hacmp.out 和使用监控工具来验证替换和重新集成是否成功。

DB2 SP HACMP ES 工作表

下列工作表设计成配合您在准备外部 SSA 磁盘配置时应该完成的 HACMP 工作表使用（那些工作表在 HACMP Planning, Installation, and Administration Guides 中）。在每种情况下，都提供了一个完成的示例和一个空白的工作表。

下图显示第一个样本工作表中所记载的外部磁盘上的数据库配置。用来创建该数据库的语句是：

```
db2 create database pwq on /newdata
```

为 SSA 外部适配器和外部 SSA 磁盘建立镜象和双尾，以使逻辑卷没有单点故障。此框图描述了一个类似于 **maymap** 命令的输出的配置。Maymap 是一个显示外部 SSA 磁盘配置的实用程序（通过 AIXTOOLS 使用），应在计划设置时使用。

DB2 四节点数据库外部磁盘设置样本

- 显示双尾以获取高可用性。

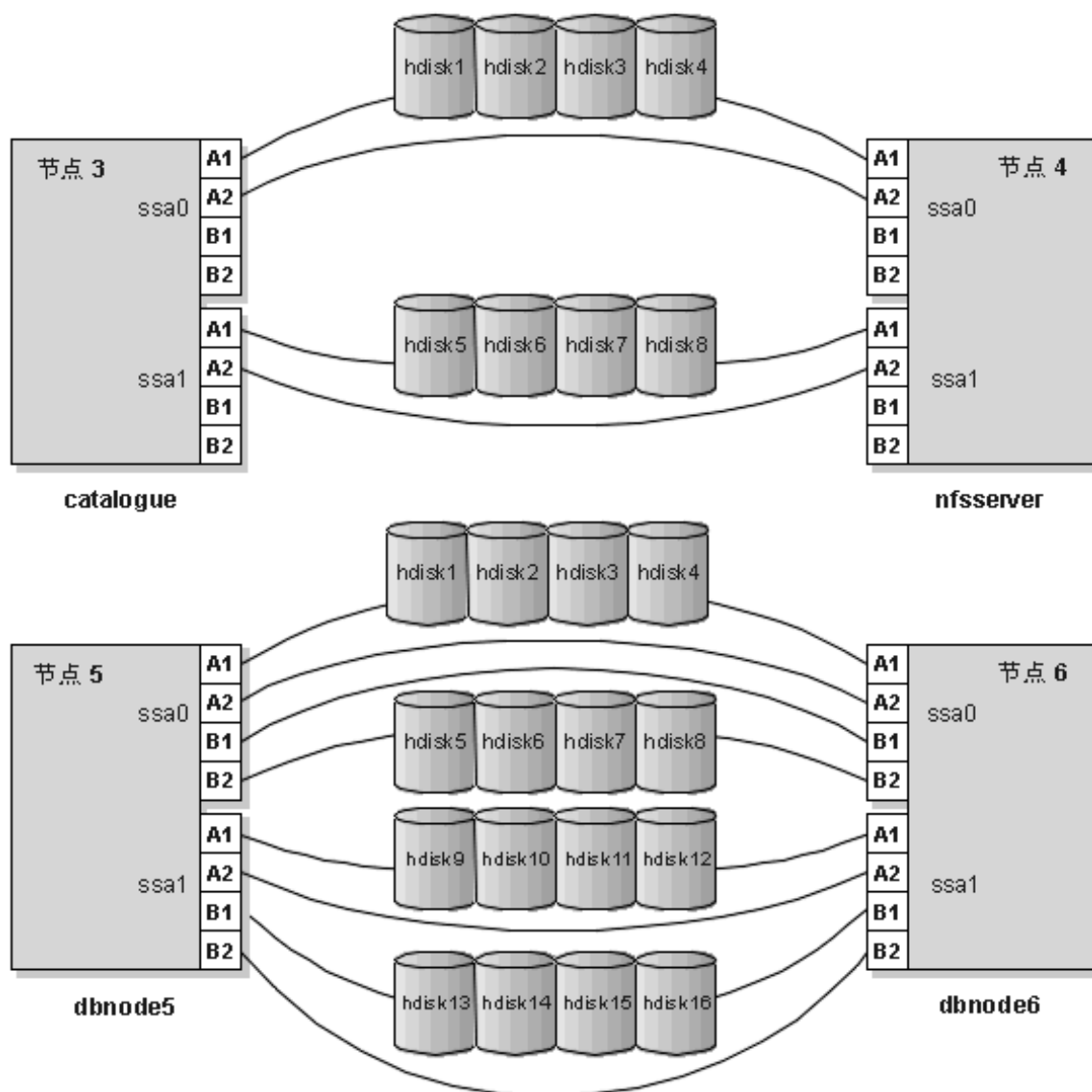


图 55. DB2 四节点数据库外部磁盘设置样本

在复查下表之前，您应仔细阅读 HACMP 文档中有关卷组上的定额设置及逻辑卷上的镜象化写一致性设置的内容。以上两个设置将直接影响可用性和性能。务必复查这些设置并理解其内在含义。“定额”和“镜象化写一致性”的典型设置都是“关”。

表 25. HACMP 卷组、逻辑卷和文件系统

SP 节点	卷组名	PP 大小 (MB)	逻辑卷名	PP 数	份数	硬盘列表	文件系统安装点 (MB)	文件系统日志逻辑卷	节点说明和备份	/dev 逻辑设备的用户拥有者
3	havg3	8	hlv300	10	2	hdisk1 hdisk5	/newdata /pwq /NODE0003	hlog301	目录节点安装点; 节点 4	root *
3	havg3	8	hlog301	1	2	hdisk1 hdisk5	N/A	N/A	目录节点 jfslog; 节点 4	root *
3	havg3	8	hlv301	10	2	hdisk2 hdisk6	N/A	N/A	目录节点原始临时空间; 节点 4	pwq **
4	havg4	8	hlv400	10	2	hdisk3 hdisk7	/dbmnt	hlog401	nfsserver pwq home; 节点 3	root *
4	havg4	8	hlog401	1	2	hdisk3 hdisk7	N/A	N/A	nfsserver jfslog; 节点 3	root *
5	havg5	8	hlv500	10	2	hdisk1 hdisk9	/newdata/ pwq/ NODE0005	HLOG501	Dbnode5 安装点; 节点 6	root *
5	havg5	8	hlog501	1	2	hdisk1 hdisk9	N/A	N/A	Dbnode5 jfslog; 节点 6	root *
5	havg5	8	hlv501	10	2	hdisk2 hdisk10	N/A	N/A	Dbnode5 原始临时空间; 节点 6	pwq **
5	havg5	8	hlv502	100	2	hdisk2 hdisk10	N/A	N/A	Dbnode5 原始表空间; 节点 6	pwq **

表 25. HACMP 卷组、逻辑卷和文件系统 (续)

SP 节点	卷组名	PP 大小 (MB)	逻辑卷名	PP 数	份数	硬盘列表	文件系统安装点 (MB)	文件系统日志逻辑卷	节点说明和备份	/dev 逻辑设备的用户拥有者
5	havg5	8	halv503	100	2	hdisk3 hdisk11	N/A	N/A	Dbnode5 原始表空间; 节点 6	pwq **
5	havg5	8	halv504	100	2	hdisk3 hdisk11	N/A	N/A	Dbnode5 原始表空间; 节点 6	pwq **
5	havg5	8	halv505	100	2	hdisk4 hdisk12	/dbdata5	hlog501	Dbnode6 系统表空间; 节点 6	root *
6	havg6	8	hlv600	10	2	hdisk5 hdisk13	/newdata/ pwq/ NODE0006	hlog601	Dbnode6 安装点; 节点 5	root *
6	havg6	8	hlog601	1	2	hdisk5 hdisk13	N/A	N/A	Dbnode6 jfslog; 节点 5	root *
6	havg6	8	hlv601	10	2	hdisk6 hdisk14	N/A	N/A	Dbnode6 原始临时空间; 节点 5	pwq **
6	havg6	8	hlv602	100	2	hdisk6 hdisk14	N/A	N/A	Dbnode6 原始表空间; 节点 5	pwq **
6	havg6	8	hlv603	100	2	hdisk7 hdisk15	N/A	N/A	Dbnode6 原始表空间; 节点 5	pwq **
6	havg6	8	hlv604	100	2	hdisk7 hdisk15	N/A	N/A	Dbnode6 原始表空间; 节点 5	pwq **
6	havg6	8	hlv605	100	2	hdisk8 hdisk16	/dbdata6	hlog601	Dbnode6 系统表空间; 节点 5	root *

注:

- * jfs 文件系统逻辑卷和日志保留超级用户许可权。
- ** 原始数据库空间获取 /dev 原始文件项 (/dev/rxxxx) 的数据库用户许可权。

表 26. HACMP 卷组、逻辑卷和文件系统 - 空白

SP 节点	卷组名	PP 大小 (MB)	逻辑卷 名	PP 数	份数	硬盘列 表	文件系统安 装点 (MB)	文件系统日 志逻辑卷	节点说明和 备份	/dev 逻 辑设备 的用户 所有者

表 26. HACMP 卷组、逻辑卷和文件系统 - 空白 (续)

SP 节点	卷组名	PP 大小 (MB)	逻辑卷名	PP 数	份数	硬盘列表	文件系统安装点 (MB)	文件系统日志逻辑卷	节点说明和备份	/dev 逻辑设备的用户拥有者

表 26. HACMP 卷组、逻辑卷和文件系统 - 空白 (续)

SP 节点	卷组名	PP 大小 (MB)	逻辑卷名	PP 数	份数	硬盘列表	文件系统安装点 (MB)	文件系统日志逻辑卷	节点说明和备份	/dev 逻辑设备的用户拥有者

表 27. 计划 HACMP NFS 服务器

SP 节点	外部文件系统	备份节点	SP 交换引导和服务 IP 别名对 (/etc/filesystems)	要安装的文件系统	要指定成数据库主目录的文件系统	作为文件系统调出目的地的地址 (/etc/exports)
3	/dbmnt	4	nfs_boot_3 nfs_client_3	nfs_server:/ dbmnt as /dbi	/dbi/pwq	nfs_boot_3 nfs_client_3 nfs_server_boot nfs_server nfs_boot_5 nfs_client_5 nfs_boot_6 nfs_client_6

表 27. 计划 HACMP NFS 服务器 (续)

SP 节点	外部文件系统	备份节点	SP 交换引导和服务 IP 别名对 (/etc/filesystems)	要安装的文件系统	要指定成数据库主目录的文件系统	作为文件系统调出目的地的地址 (/etc/exports)
4	/dbmnt	3	nfs_server_boot nfs_server	nfs_server:/ dbmnt as /dbi	/dbi/pwq	nfs_boot_3 nfs_client_3 nfs_server_boot nfs_server nfs_boot_5 nfs_client_5 nfs_boot_6 nfs_client_6
5	N/A	N/A	nfs_boot_5 nfs_client_5	nfs_server:/ dbmnt as /dbi	/dbi/pwq	N/A
6	N/A	N/A	nfs_boot_6 nfs_client_6	nfs_server:/ dbmnt as /dbi	/dbi/pwq	N/A

注:

1. /etc/passwd 必须在所有节点上都相同可从控制工作站将它同步。
2. 确保外部文件系统具有数据库实例拥有者的许可权。
3. /etc/filesystems 必须具有安装参数: hard、bg、intr 和 rw。
4. /etc/exports 将仅在服务器和其备份上具有
-root=ip1:ip2:ip3

表 28. 计划 HACMP NFS 服务器 - 空白

SP 节点	外部文件系统	备份节点	SP 交换引导和服务 IP 别名对 (/etc/filesystems)	要安装的文件系统	要指定成数据库主目录的文件系统	作为文件系统调出目的地的地址 (/etc/exports)

表 28. 计划 HACMP NFS 服务器 - 空白 (续)

SP 节点	外部文件系统	备份节点	SP 交换引导和服务 IP 别名对 (/etc/filesystems)	要安装的文件系统	要指定成数据仓库主目录的文件系统	作为文件系统调出目的地的地址 (/etc/exports)

表 28. 计划 HACMP NFS 服务器 - 空白 (续)

SP 节点	外部文件系统	备份节点	SP 交换引导和服务 IP 别名对 (/etc/filesystems)	要安装的文件系统	要指定成数据库主目录的文件系统	作为文件系统调出目的地的地址 (/etc/exports)

第13章 Windows NT 环境中的高可用性

您可设置数据库系统，以便当一台机器失效时，该失效机器上的数据库服务器可在另一台机器上运行。在 Windows NT 上，故障恢复支持可通过“Microsoft 群集服务器”(MSCS)实现。要使用 MSCS，您需要“Windows NT 企业版”的版本 4.0，并安装 MSCS 功能部件。

在一个群集环境中，MSCS 可同时执行故障检测和重新启动资源，如物理磁盘和 IP 地址的故障恢复支持。(当失效的机器再次联机时，除非您先前配置资源这样做，否则资源不会自动恢复到原来的状态。有关详情，参见第242页的『回退考虑事项』。)

在为故障恢复支持启用 DB2 实例之前，执行下列计划步骤：

1. 确定要使用哪些磁盘来存储数据。应给每个数据库服务器至少分配一个磁盘来供它自己使用。用于存储数据的磁盘必须与一个共享磁盘子系统连接，且必须配置为 MSCS 磁盘资源。
2. 确保要用于支持远程请求的每个数据库服务器都具有一个 IP 地址。

当设置故障恢复支持时，它可用于一个现存的实例，或您可在实现故障恢复支持时创建一个新的实例。

要启用故障恢复支持，执行下列步骤：

1. 为 DB2MSCS 实用程序创建一个输入文件。
2. 调用 **db2mscs** 命令。
3. 若正在使用一个分区数据库系统，则注册数据库驱动器映射以启用相互替换。参见第242页的『在分区数据库环境中为相互替换配置注册数据库驱动器映射』。

当完成允许实例使用故障恢复支持的工作后，配置将类似于第232页的图56。

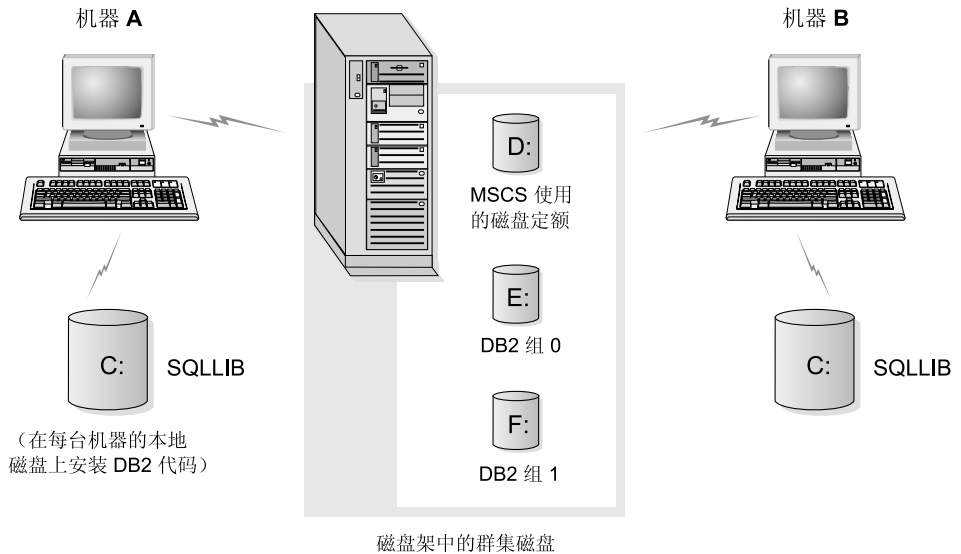


图 56. 样本 MSCS 配置

下列几节描述不同类型的故障恢复支持，以及如何实现它们。在执行以下描述的任何步骤前，您必须已经在 MSCS 群集中要使用的每台机器上安装了该 MSCS 软件。另外，还必须在每台机器上安装了 DB2。

故障恢复配置

可使用两种类型的配置：

- 热备份
- 相互替换

目前，MSCS 支持两台机器的群集。

在一个分区数据库环境中，群集不必全部具有相同类型的配置。您可将一些群集设置为使用热备份，而将其他群集设置为使用相互替换。例如，若您的 DB2 实例由五个工作站组成，则您可将两台机器设置为使用相互替换配置，将另外两个设置为使用热备份配置，而不将一台机器配置为用于故障恢复支持。

热备份配置

在一个热备份配置中，MSCS 群集中的一台机器提供专用的故障恢复支持，而另一台机器参与该数据库系统。若参与该数据库系统的机器失效，则该机器上的数据库服务器将在故障恢复机器上启动。若在一个分区数据库系统中，您正在一台机

器上运行多个逻辑节点，而该机器失效，则这些逻辑节点将在故障恢复机器上启动。图57显示了一个热备份配置的示例。

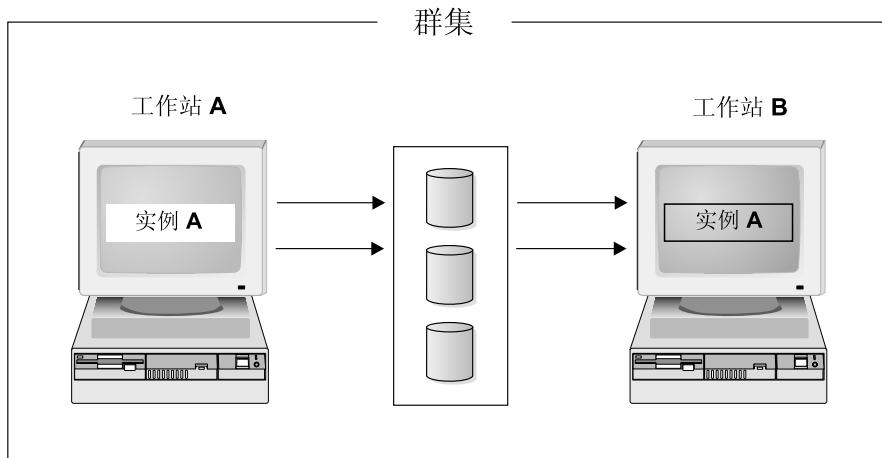


图 57. 热备份配置

相互替换配置

在一个相互替换配置中，两个工作站都参与该数据库系统（即，每台机器上至少有一个数据库服务器在运行）。若 MSCS 群集中的一个工作站失效，则该失效机器上的数据库服务器将在另一台机器上启动以运行。在一个相互替换配置中，一台机器上的数据库服务器失效时不会影响另一台机器上的数据库服务器。在任何给定的时间点，任何数据库服务器可在任何机器上是活动的。第234页的图58显示了一个相互替换配置的示例。

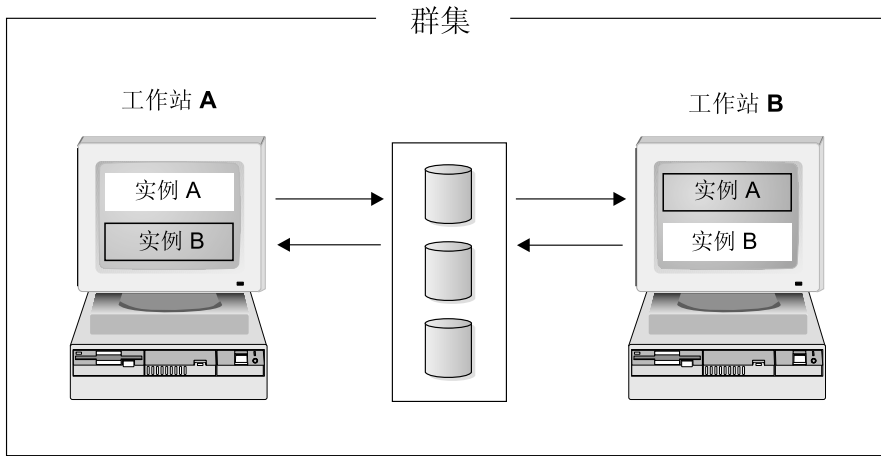


图 58. 相互替换配置

使用 DB2MSCS 实用程序

使用 DB2MSCS 实用程序创建基础结构，以便 DB2 在 Windows NT 环境中使用 MSCS 支持来支持故障恢复。您可使用此实用程序以便在单分区数据库和分区数据库环境中启用故障恢复。

对于每个实例，在它们拥有实例的机器上调用一次 **db2mscs** 命令。若在 MSCS 群集中的一台机器上只有一个 DB2 实例在运行，该操作将设置热备份配置。若在 MSCS 群集中的每台机器上都有一个实例在运行，则应在每个拥有实例的机器上运行 DB2MSCS 一次，以设置相互替换配置。

DB2MSCS 实用程序：

1. 从名为 DB2MSCS.CFG 的输入文件中读取必需的 MSCS 和 DB2 参数。有关全套输入参数的详情，参见第235页的『指定 DB2MSCS.CFG 文件』。
2. 验证输入文件中的参数。
3. 注册 DB2 资源类型。
4. 创建一个或多个 MSCS 组，以包含 MSCS 和 DB2 资源。
5. 创建 IP 资源。
6. 创建“网络名”资源。
7. 将 MSCS 磁盘移至组中。
8. 创建一个或多个 DB2 资源。
9. 添加 DB2 资源的所有必需的从属部件。
10. 将非群集 DB2 实例转换为一个群集实例。

11. 使所有资源联机。

该命令语法如下所示:

```
db2mcs -f:input_file
```

其中:

-f:input_file

指定 MSCS 实用程序要使用的 DB2MSCS.CFG 输入文件。若未指定此参数, 则 DB2MSCS 实用程序读取当前目录中的 DB2MSCS.CFG 文件。

指定 DB2MSCS.CFG 文件

DB2MSCS.CFG 文件是 ASCII 文本文件, 它包含由 DB2MSCS 实用程序读取的参数。使用下列格式在单独的行上指定每个输入参数:

PARAMETER_KEYWORD=*parameter_value*。例如:

```
CLUSTER_NAME=WOLFPACK  
GROUP_NAME=DB2 Group  
IP_ADDRESS=9.21.22.89
```

在 /SQLLIB 目录的 /CFG 子目录中有两个示例配置文件。第一个文件 DB2MSCS.EE 是单分区数据库环境的示例。第二个文件 DB2MSCS.EEE 是分区数据库环境的示例。

DB2MSCS.CFG 文件的参数如下所示:

DB2_INSTANCE

DB2 实例的名称。若未指定实例名, 则使用缺省的实例 (DB2INSTANCE 环境变量的值)。

此参数具有全局作用域, 且只在 DB2MSCS.CFG 文件中指定一次。

此参数是可选的。

示例:

```
DB2_INSTANCE=DB2
```

该实例必须已存在。有关创建实例的详情, 参考 *DB2 Enterprise - Extended Edition for Windows Quick Beginnings* 一书。

DB2_LOGON_USERNAME

DB2 服务的注册帐户名。

此参数具有全局作用域, 且只在 DB2MSCS.CFG 文件中指定一次。

只有 DB2 扩充企业版实例需要此参数。

示例:

```
DB2_LOGON_USERNAME=db2user
```

DB2_LOGON_PASSWORD

DB2 服务的注册帐户口令。若提供 DB2_LOGON_USERNAME 参数, 但没有提供 DB2_LOGON_PASSWORD 参数, 则 DB2MCS 实用程序会提示您输入该口令。在命令行输入口令时不会显示它。

此参数具有全局作用域, 且只在 DB2MCS.CFG 文件中指定一次。

只有 DB2 扩充企业版实例需要此参数。

示例:

```
DB2_LOGON_PASSWORD=xxxxxx
```

CLUSTER_NAME

MCS 群集名。只有指定了另一个 CLUSTER_NAME 标记后, 才在此群集中创建在此行后指定的所有资源。

对每个群集指定此参数一次。

此参数是可选的。若未指定, 则使用本地机器上的 MCS 群集的名称。

示例:

```
CLUSTER_NAME=WOLFPACK
```

GROUP_NAME

MCS 组名。若指定了此参数, 则当 MCS 组不存在时, 会创建一个新的 MCS 组。若该组已存在, 则将它用作目标组。只有在指定了另一个 GROUP_NAME 关键字之后, 才在此组中创建在此行后创建的任何 MCS 资源。

对每个组, 指定此参数一次。

此参数是必需的。

示例:

```
GROUP_NAME=DB2 Group
```

DB2_NODE

当前 MCS 组将包括的数据库分区服务器(节点)的节点号。若同一台机器上存在多个逻辑节点, 则每个节点需要一个单独的 DB2_NODE 关键字。

在 GROUP_NAME 参数后指定此参数, 以便在正确的 MCS 组中创建 DB2 资源。

只有 DB2 扩充企业版实例需要此参数。

示例:

```
DB2_NODE=0
```

IP_NAME

“IP 地址”资源名。IP_NAME 的值是任意的，但必须是唯一的。当指定此参数时，会创建一个类型为“IP 地址”的 MSCS 资源。

远程 TCP/IP 连接需要此参数。您必须为在一个分区数据库环境中的拥有实例的机器指定此参数。在单分区数据库环境中，此参数是可选的。

示例:

```
IP_NAME=IP Address for DB2
```

注: DB2 客户机应使用此 IP 资源的 TCP/IP 地址来编目 TCP/IP 节点项。通过使用 MSCS IP 地址，当数据库服务器故障恢复到其他机器时，DB2 客户机仍可与数据库服务器连接，因为在故障恢复机器上 IP 地址是可用的。

IP 资源的属性如下所示:

IP_ADDRESS

IP 资源的 TCP/IP 地址。指定此关键字，以设置上述 IP 资源的 TCP/IP 地址。

若指定了 IP_NAME 参数，则此参数是必需的。

示例:

```
IP_ADDRESS=9.21.22.34
```

IP_SUBNET

上述 IP 资源的子网掩码。

若指定了 IP_NAME 参数，则此参数是必需的。

示例:

```
IP_SUBNET=255.255.255.0
```

IP_NETWORK

上述 IP 资源所属的 MSCS 网络的名称。若未指定此参数，则使用系统检测到的第一个 MSCS 网络。

此参数是可选的。

示例:

```
IP_NETWORK=Token Ring
```

NETNAME_NAME

“网络名”资源名。指定此参数以创建“网络名”资源。

对于单分区数据库环境，此参数是可选的。对于分区数据库环境，此参数是必需的。

示例:

```
NETNAME_NAME=Network name for DB2
```

“网络名”资源的属性如下所示:

NETNAME_VALUE

“网络名”的值。

若指定了 NETNAME_NAME 参数，则此参数是必需的。

示例:

```
NETNAME_VALUE=DB2SRV
```

NETNAME_DEPENDENCY

“网络名”资源的从属性列表。每个“网络名”资源必须依赖于一个“IP 地址”资源。若未指定此参数，则“网络名”资源依赖于该组中的第一个 IP 资源。

此参数是可选的。

示例:

```
NETNAME_DEPENDENCY=IP Address for DB2
```

DISK_NAME

要移动到当前组中的物理磁盘资源的名称。指定所需的磁盘资源。

注:

1. 磁盘资源必须已存在。
2. 当 DB2MSCS 实用程序为 MSCS 支持配置 DB2 实例时，将该实例目录复制到该组中的第一个 MSCS 磁盘。要为该实例目录指定不同的 MSCS 磁盘，使用 INSTPROF_DISK 参数。

示例:

```
DISK_NAME=Disk E:  
DISK_NAME=Disk F:
```

INSTPROF_DISK

一个可选的参数，指定包含 DB2 实例目录的 MSCS 磁盘。若未指定此参数，则 DB2MSCS 实用程序使用该实例目录所在的同一个组中的第一个 MSCS 磁盘。

该 DB2 实例目录创建在 MSCS 磁盘上的 X:\DB2PROFS 目录下（其中，X 是 MSCS 磁盘驱动器盘符）。

示例:

```
INSTPROF_DISK=Disk E:
```

TARGET_DRVMAP_DISK

用来指定数据库驱动器映射的目标 MSCS 磁盘的可选参数。如果在与节点不属于同一组的 MSCS 磁盘上创建数据库, 则使用目标驱动器映射磁盘来包含数据库分区。如果未指定此参数, 则必须使用 DB2DRVMP 实用程序人工注册数据库驱动器映射。

示例:

```
TARGET_DRVMAP_DISK = Disk E:
```

为单分区数据库系统设置故障恢复

当您为单分区数据库系统运行 DB2MSCS 实用程序时, 一个 MSCS 组包含 DB2 和所有从属的 MSCS 资源 (IP 地址、网络名和磁盘)。例如, 一个单分区数据库系统的 DB2MSCS.CFG 文件的内容将类似于如下内容:

```
#
# DB2MSCS.CFG for a single-partition database system
#
DB2_INSTANCE=DB2
CLUSTER_NAME=MSCS
GROUP_NAME=DB2 Group
IP_NAME=...
IP_ADDRESS=...
IP_SUBNET=...
IP_NETWORK=...
NETNAME_NAME=...
NETNAME_VALUE=...
DISK_NAME=Disk E:
```

为两个单分区数据库系统设置相互替换配置

您可设置两个单分区数据库系统, 每台机器上一个, 这样若任何一台机器上的数据库系统失效, 则在另一个 MSCS 节点上启动它。

要为此配置设置故障恢复支持, 您需要在每个拥有实例的机器上将 DB2MSCS 实用程序运行一次。您必须为每个数据库系统定制配置文件。

假设 DB2 实例称为 DB2A 和 DB2B。DB2A 实例的 DB2MSCS.CFG 文件如下所示:

```
#
# DB2MSCS.CFG for first single-partition database system
#
DB2_INSTANCE=DB2A
CLUSTER_NAME=MSCS
GROUP_NAME=DB2A Group
```

```
IP_NAME=...
IP_ADDRESS=...
IP_SUBNET=...
IP_NETWORK=...
NETNAME_NAME=...
NETNAME_VALUE=...
DISK_NAME=Disk E:
```

DB2A 实例的 DB2MSCS.CFG 文件如下所示:

```
#
# DB2MSCS.CFG for second single-partition database system
#
DB2_INSTANCE=DB2B
CLUSTER_NAME=MSCS
GROUP_NAME=DB2B Group
IP_NAME=...
IP_ADDRESS=...
IP_SUBNET=...
IP_NETWORK=...
NETNAME_NAME=...
NETNAME_VALUE=...
DISK_NAME=Disk F:
```

有关完整的示例, 参见第245页的『示例 - 为相互替换设置两个单分区实例』。

为分区数据库系统设置多个 MSCS 群集

当对一个多分区数据库系统运行 DB2MSCS 实用程序时, 要为参与该系统的每个物理机器创建一个 MSCS 组。DB2MSCS.CFG 文件必须包含多节, 且对于 GROUP_NAME 参数以及该组的所有必需的从属资源, 每一节必须具有不同的值。

而且, 必须对每个 MSCS 组中的每个数据库分区服务器指定 DB2_NODE 参数。若您有多个逻辑节点, 则每个逻辑节点都需要一个单独的 DB2_NODE 关键字。

例如, 假设您有一个多分区数据库系统, 它由四台机器上的四个数据库分区服务器组成, 而您想配置两个 MSCS 群集使用相互替换配置。应按如下所示设置 DB2MSCS.CFG 配置文件:

```
#
# DB2MSCS.CFG for one partitioned database system with
# multiple clusters
DB2_INSTANCE=DB2MPP
DB2_LOGON_USERNAME=db2user
DB2_LOGON_PASSWORD=xxxxxx
CLUSTER_NAME=MSCS1
# Group 1
GROUP_NAME=DB2 Group 1
DB2_NODE=0
IP_NAME=...
```

```

...
# Group 2
GROUP_NAME=DB2 Group 2
DB2_NÓDE=1
IP_NAME=...

...

CLUSTER_NAME=MSCS2
# Group 3
GROUP_NAME=DB2 Group 3
DB2_NÓDE=2
IP_NAME=...

...
# Group 4
GROUP_NAME=DB2 Group 4
DB2_NÓDE=3
IP_NAME=...

...

```

有关完整的示例，参见第247页的『示例 - 为相互替换设置四节点分区数据库系统』。

维护 MSCS 系统

当运行 DB2MSCS 实用程序时，它将为 MSCS 群集中的所有机器创建故障恢复支持的基础结构。要从机器上除去该支持，使用带 "drop" 选项的 **db2iclus** 命令。要对机器重新启用该支持，使用 "add" 选项。

该命令语法如下所示：

```

▶—db2iclus—[add]—[drop]—[/i:—instance_name]—/u:—account_name,password—▶
▶—[/m:—machine_name]—[/c:—cluster_name]—▶

```

其中：

add

将故障恢复支持添加至一个 MSCS 群集，在机器上启用它。这样 DB2 资源（数据库服务器）就可故障恢复到此机器上。

drop

从 MSCS 群集中卸下故障恢复支持，从机器中除去它。

/i: <i>instance_name</i>	实例名称。(此参数替换 DB2INSTANCE 环境变量的设置。)
/u: <i>account_name, password</i>	用作“DB2 服务”的注册帐户名的域帐户。例如: /u:domainA\db2nt,password 只在使用“add”参数时,此参数才是必需的。
/m: <i>machine_name</i>	要添加至 MSCS 群集的或要从 MSCS 群集中卸下的机器的计算机名。若您是从另一台机器而不是正在对其修改故障恢复支持的机器运行该命令,则必须指定此选项。
/c: <i>cluster_name</i>	LAN 上所知道的 MSCS 群集名。此名称是在第一次创建 MSCS 群集时指定的。

回退考虑事项

缺省情况下,将组设置为不回退到原始的(失效的)机器。除非在故障恢复后以人工方式将 DB2 组配置为回退,否则,在解除造成故障恢复的起因后,它将继续在替代的 MSCS 节点上运行。

若将 DB2 组配置为自动回退到原始机器,则一旦原始机器可用,DB2 组中的所有资源包括 DB2 资源将会尽快回退。若在回退期间存在一个数据库连接,则 DB2 资源不能脱机,而回退处理将失败。

若要在回退处理期间强行中断所有数据库连接,则将 DB2_FALLBACK 注册表变量设置为 ON。此变量必须按如下所示的格式设置:

```
db2set DB2_FALLBACK=ON
```

您不必在设置此注册表变量后重新引导或重新启动该群集服务。

在分区数据库环境中为相互替换配置注册数据库驱动器映射

当在分区数据库环境中创建数据库时,可指定一个驱动器盘符,以指示要在何处创建数据库。

注: 不要对单分区数据库环境设置数据库驱动器映射。

当 `CREATE DATABASE` 命令运行时，它希望您指定的驱动器将同时对所有参与该实例的机器可用。因为这是不可能的，所以 DB2 使用数据库驱动器映射，为每台机器的相同驱动器指定不同的名称。

例如，假设名为 DB2 的一个 DB2 实例包含两个数据库分区服务器：

```
NODE0 在机器 WOLF_NODE_0 上是活动的
NODE1 在机器 WOLF_NODE_1 上是活动的
```

还假设共享磁盘 E: 与 NODE0 属于同一个组，而共享磁盘 F: 与 NODE1 属于同一个组。

要在共享磁盘 E 上创建数据库：

```
db2 create database mppdb on e:
```

为了使该命令成功，驱动器 E: 必须同时对两台机器可用。在一个相互替换配置中，每个数据库分区服务器可在不同的机器上是活动的，而群集磁盘 E: 只对一台机器是可用的。在这种情况下，`CREATE DATABASE` 命令将总是失败。

要解决这个问题，数据库驱动器应当按如下所示进行映射：

```
对于 NODE0，该映射是从驱动器 F: 至驱动器 E:
对于 NODE1，该映射是从驱动器 E: 至驱动器 F:
```

然后，将 NODE0 对驱动器 F: 的任何数据库存取映射至驱动器 E:，将 NODE1 对驱动器 E: 的任何数据库存取映射至驱动器 F:。使用驱动器映射，`CREATE DATABASE` 命令将在 NODE0 的驱动器 E: 和 NODE1 的驱动器 F: 上创建数据库文件。

使用 `db2drvmp` 命令来设置驱动器映射。该命令语法如下所示：

```
▶▶—db2drvmp—add—node_number—from_drive—to_drive—▶▶
      |
      |—drop—
      |—query—
      |—reconcile—
```

这些参数如下所示：

- add** 指定新的数据库驱动器映射。
 - drop** 除去现存的数据库驱动器映射。
 - query** 查询数据库映射。
 - reconcile** 当注册表内容损坏时，修复数据库驱动器映射。有关详情，参见第244页的『协调数据库驱动器映射』。
- node_number* 节点号。添加和卸下操作需要此参数。

from_drive 映射的源盘符。添加和卸下操作需要此参数。
to_drive 映射的目标盘符。添加操作需要此参数。该参数对其他操作不适用。

若要设置从 F: 至 NODE0 的 E: 的数据库驱动器映射, 应使用如下命令:

```
db2drvmp add 0 F E
```

注: 数据库驱动器映射不适用于表空间、容器或任何其他数据库存储器对象。

类似地, 要设置从 E: 至 NODE1 的 F: 的数据库驱动器映射, 应发出如下命令:

```
db2drvmp add 1 E F
```

注: 任何对数据库驱动器映射的设置或更改都不会立即生效。要激活数据库驱动器映射, 使用“群集管理员”工具将 DB2 资源脱机, 然后联机。

使用 DB2MISC.CFG 文件中的 TARGET_DRVMAP_DISK 关键字将使驱动器映射自动完成。

协调数据库驱动器映射

当在具有生效的数据库驱动器映射的机器上创建数据库时, 该映射保存在驱动器上的隐含文件中。这防止创建数据库后数据库驱动器被除去。举个例子, 当您意外卸下数据库驱动器映射时, 您将希望协调数据库驱动器映射。要协调该映射, 对包含数据库的每个数据库分区服务器运行 **db2drvmp reconcile** 命令。该命令语法如下所示:

► db2drvmp reconcile —————►
 └── node_number—drive ──┘

这些参数如下所示:

node_number 要修复的节点的节点号。若未指定 *node_number*, 则该命令协调所有节点的映射。

drive 要协调的驱动器。若未指定驱动器, 则该命令协调所有驱动器的映射。

db2drvmp 命令扫描机器上的所有驱动器以查找数据库分区服务器管理的数据库分区, 然后按需要对注册表重新应用数据库驱动器映射。

示例 - 为相互替换设置两个单分区实例

此示例的目的是在一个相互替换配置中设置两个具有故障恢复支持的单分区数据库实例。在此示例中，将四个服务器配置为两个 MSCS 群集。通过使用相互替换配置，当任何一台机器发生故障时，为该机器配置的数据库服务器将故障恢复到替代的机器上（通过使用 MSCS 软件配置），然后在替代机器上运行。

在结果配置中有两个 MSCS 群集。每个群集具有：

- 两个服务器，分别具有 64 MB 内存和一个 2 GB 的本地 SCSI 磁盘
- 一个 SCSI 磁盘组，具有三个共享的 SCSI 磁盘，每个有 2 GB。

另外，每台机器都安装了一个 100X 以太网适配卡。

每台机器都安装了下列软件：

- 安装了 MSCS 功能部件的“Windows NT 企业版的版本 4.0”
- DB2 通用数据库企业版的版本 7。

结果网络配置如下所示：

服务器 1:	服务器 2:
<ul style="list-style-type: none">• 机器名: db2test1• TCP/IP 主机名: db2test1• IP 地址: 9.9.9.1 子网掩码: 255.255.255.0• MSCS 群集名: ClusterA	<ul style="list-style-type: none">• 机器名: db2test2• TCP/IP 主机名: db2test2• IP 地址: 9.9.9.2 子网掩码: 255.255.255.0• MSCS 群集名: ClusterA

网络中的两台机器都配置为使用 TCP/IP，且使用以太网 100 T-base 集线器与专用 LAN 连接。若缺少“域名服务器” (DNS)，则所有机器都有一个本地 TCP/IP hosts 文件，该文件包含下列各项：

```
9.9.9.1 db2test1 # for Server 1
9.9.9.2 db2test2 # for Server 2
9.9.9.3 ClusterA # for MSCS ClusterA
9.9.9.4 db2tcp1 # for DB2 remote client connection to Server 1
9.9.9.5 db2tcp2 # for DB2 remote client connection to Server 2
```

初步任务

在执行下列任务前，假设两台机器属于同一个域 DB2NTD:

1. 在 DB2 将运行的那些机器上，为 DB2 创建一个作为“本地管理员”组的域帐户。使用该帐户来执行所有任务：
 - 将用户名设置为 db2nt。

- 将口令设置为 db2nt。
2. 在机器 db2test1 和 db2test2 上安装 MSCS 功能部件:
 - 将 MSCS 群集命名为 ClusterA。
 - 群集 IP 地址为 9.9.9.3。
 - MSCS 软件将使用共享磁盘 D:。
 - DB2 将使用共享磁盘 E: 和 F:。
 3. 在机器 db2test1 上安装 “DB2 通用数据库企业版版本 7”。在本地驱动器 C:\SQLLIB 上安装该软件。
 4. 在机器 db2test2 上安装 “DB2 通用数据库企业版版本 7”。在本地驱动器 C:\SQLLIB 上安装该软件。

下一步是为每个实例设置 DB2MSCS.CFG 文件，然后对每个实例运行 DB2MSCS 实用程序。

运行 DB2MSCS 实用程序

要设置 db2test1 机器，执行下列任务:

1. 在机器 db2test1 上，作为用户 db2nt 注册。口令是 db2nt。
2. 若 DB2 实例 DB2A 尚不存在，则创建它。创建该实例的命令是:

```
db2icrt DB2A
```

3. 在机器 db2test1 上为 DB2 实例设置 DB2MSCS.CFG 文件:

```
#
# DB2MSCS.CFG for database system
# on machine db2test1
DB2_INSTANCE=DB2A
CLUSTER_NAME=ClusterA
#
# Group 1
GROUP_NAME=DB2A Group
IP_NAME=IP Address for DB2A
IP_ADDRESS=9.9.9.4
IP_SUBNET=255.255.255.0
IP_NETWORK=ClusterA
NETNAME_NAME=Network name for DB2A
NETNAME_VALUE=DB2SRV1
NETNAME_DEPENDENCY=IP Address for DB2A
DISK_NAME=Disk E:
INSTPROF_DISK=Disk E:
```

4. 运行 DB2MSCS 实用程序，如下所示:

```
db2mscs -f:DB2MSCS.CFG
```

5. 从 db2nt 帐户注销。

6. 在机器 db2test2 上, 作为用户 db2nt 注册, 它属于“本地管理员”组。口令是 db2nt。
7. 若 DB2 实例 DB2B 尚不存在, 则创建它。创建该实例的命令是:

```
db2icrt DB2B
```
8. 在机器 db2test2 上为 DB2 实例设置 DB2MSCS.CFG 文件:

```
#  
# DB2MSCS.CFG for database system  
# on machine db2test2  
DB2_INSTANCE=DB2B  
CLUSTER_NAME=ClusterA  
#  
# Group 1  
GROUP_NAME=DB2B Group  
IP_NAME=IP Address for DB2B  
IP_ADDRESS=9.9.9.5  
IP_SUBNET=255.255.255.0  
IP_NETWORK=ClusterA  
NETNAME_NAME=Network name for DB2B  
NETNAME_VALUE=DB2SRV2  
NETNAME_DEPENDENCY=IP Address for DB2B  
DISK_NAME=Disk F:  
INSTPROF_DISK=Disk F:
```
9. 运行 DB2MSCS 实用程序, 如下所示:

```
db2mcs -f:DB2MSCS.CFG
```
10. 从 db2nt 帐户注销。

示例 - 为相互替换设置四节点分区数据库系统

此示例的目的是在一个相互替换配置中设置具有故障恢复支持的四节点分区数据库系统。在此示例中, 将四个服务器配置为两个 MSCS 群集。通过使用相互替换配置, 若任何一台机器发生故障时, 则为该机器配置的数据库分区服务器将故障恢复到替代机器(通过使用 MSCS 软件配置), 然后在替代机器上作为一个逻辑节点运行。

在结果配置中有两个 MSCS 群集。每个群集具有:

- 两个服务器, 分别具有 64 MB 内存和一个 2 GB 的本地 SCSI 磁盘
- 一个 SCSI 磁盘组, 具有三个共享的 SCSI 磁盘, 每个有 2 GB。

另外, 每台机器都安装了一个 100X 以太网适配卡。

每台机器都安装了下列软件:

- 安装了 MSCS 功能部件的“Windows NT 企业版的版本 4.0”
- DB2 通用数据库扩充企业版的版本 7。

结果网络配置如下所示:

服务器 1: <ul style="list-style-type: none">• 机器名: db2test1• TCP/IP 主机名: db2test1• IP 地址: 9.9.9.1 子网掩码: 255.255.255.0• MSCS 群集名: ClusterA	服务器 2: <ul style="list-style-type: none">• 机器名: db2test2• TCP/IP 主机名: db2test2• IP 地址: 9.9.9.2 子网掩码: 255.255.255.0• MSCS 群集名: ClusterA
服务器 3: <ul style="list-style-type: none">• 机器名: db2test3• TCP/IP 主机名: db2test3• IP 地址: 9.9.9.3 子网掩码: 255.255.255.0• MSCS 群集名: ClusterB	服务器 4: <ul style="list-style-type: none">• 机器名: db2test4• TCP/IP 主机名: db2test4• IP 地址: 9.9.9.4 子网掩码: 255.255.255.0• MSCS 群集名: ClusterB

网络中的所有机器都配置为使用 TCP/IP, 且使用以太网 100 T-base 集线器与专用 LAN 连接。若缺少“域名服务器”(DNS), 则所有机器都有一个本地 TCP/IP hosts 文件, 该文件包含下列各项:

```
9.9.9.1 db2test1 # for Server 1
9.9.9.2 db2test2 # for Server 2
9.9.9.3 db2test3 # for Server 3
9.9.9.4 db2test4 # for Server 4
9.9.9.5 ClusterA # for MSCS Cluster 1
9.9.9.6 ClusterB # for MSCS Cluster 2
9.9.9.7 db2tcp # for DB2 remote client connection
```

初步任务

在执行下列任务前, 假设所有四台机器都属于同一个域 DB2NTB:

1. 在 DB2 将运行的那些机器上, 为 DB2 创建一个作为“本地管理员”组的域帐户。使用该帐户来执行所有任务:
 - 将用户名设置为 db2nt。
 - 将口令设置为 db2nt。
2. 创建具有“口令永不到期”特征的第二个域帐户。此帐户将与 DB2 服务相关:
 - 将用户名设置为 db2mpp。
 - 将口令设置为 db2mpp。
3. 在机器 db2test1 和 db2test2 上安装 MSCS 功能部件:
 - 将 MSCS 群集命名为 ClusterA。
 - 该群集的“IP 地址”为 9.9.9.5。

- MSCS 软件将使用共享磁盘 D:。
 - DB2 将使用共享磁盘 E: 和 F:。
4. 在机器 db2test3 和 db2test4 上安装 MSCS 功能部件:
 - 将 MSCS 群集命名为 ClusterB。
 - 该群集的“IP 地址”为 9.9.9.6。
 - MSCS 软件将使用共享磁盘 D:。
 - DB2 将使用共享磁盘 E: 和 F:。
 5. 在机器 db2test1 上安装 DB2 扩充企业版:
 - 选择“此机器将是拥有实例的数据库分区服务器”选项。
 - 该 DB2 服务的帐户是 db2mpp。口令是 db2mpp。
 - 在本地驱动器 C:\SQLLIB 上安装该软件。
 6. 在机器 db2test2、db2test3 和 db2test4 上安装 DB2 扩充企业版:
 - 选择“此机器将是现存分区数据库系统上的一个新节点”选项。
 - 选择 db2test1 作为拥有实例的机器。
 - 该 DB2 服务的帐户是 db2mpp。口令是 db2mpp。
 - 在本地驱动器 C:\SQLLIB 上安装该软件。

下一步是设置 DB2MSCS.CFG 文件，然后运行 DB2MSCS 实用程序。

运行 DB2MSCS 实用程序

要设置 db2test1 机器，执行下列任务：

1. 作为用户 db2nt 注册，它属于“本地管理员”组。口令是 db2nt。
2. 设置 DB2MSCS.CFG 文件：

```
#
# DB2MSCS.CFG for one partitioned database system with
# multiple MSCS clusters
DB2_INSTANCE=DB2MPP
CLUSTER_NAME=ClusterA
DB2_LOGON_USERNAME=db2mpp
DB2_LOGON_PASSWORD=db2mpp
# Group 1
# for DB2 node 0
GROUP_NAME=DB2NODE0
DB2_NODE=0
IP_NAME=IP Address for DB2
IP_ADDRESS=9.9.9.7
IP_SUBNET=255.255.255.0
IP_NETWORK=Ethernet
NETNAME_NAME=Network name for DB2
NETNAME_VALUE=DB2WOLF
```

```

NETNAME_DEPENDENCY=IP Address for DB2
DISK_NAME=Disk E:
INSTPROF_DISK=Disk E:
#
# Group 2
# for DB2 node 1
GROUP_NAME=DB2NODE1
DB2_NODE=1
DISK_NAME=Disk F:
#
CLUSTER_NAME=ClusterB
# Group 3
# for DB2 node 2
GROUP_NAME=DB2NODE2
DB2_NODE=2
DISK_NAME=Disk E:
#
# Group 4
# for DB2 node 3
GROUP_NAME=DB2NODE3
DB2_NODE=3
DISK_NAME=Disk F:

```

3. 运行 DB2MSCS 实用程序，如下所示：

```
db2mcs -f:DB2MSCS.CFG
```

4. 从 db2nt 帐户注销。

最后的步骤是注册两个 MSCS 群集的数据库驱动器映射。

注册 ClusterA 的数据库驱动器映射

要注册 MSCS 群集 ClusterA 的数据库驱动器映射，执行下列任务：

1. 在机器 db2test1 上，作为用户 db2mpp 注册，它是与 DB2 服务相关的帐户。口令是 db2mpp。
2. 要注册数据库驱动器映射，输入下列命令：

```
db2drvmp add 0 F E
```

```
db2drvmp add 1 E F
```

3. 将所有 DB2 资源脱机，然后将它们联机。

注册 ClusterB 的数据库驱动器映射

要注册 MSCS 群集 ClusterB 的数据库驱动器映射，执行下列任务：

1. 在机器 db2test3 上，作为用户 db2mpp 注册，它是与 DB2 服务相关的帐户。口令是 db2mpp。

2. 要注册数据库驱动器映射，输入下列命令：

```
db2drvmp add 2 F E
db2drvmp add 3 E F
```

3. 将所有 DB2 资源脱机，然后将它们联机。

在 MSCS 环境中管理 DB2

若正在使用 MSCS 群集，则您的 DB2 实例需要制订关于日常操作、数据库应用和数据库配置的附加计划。为了使 DB2 在任何 MSCS 节点上透明地执行，必须执行附加管理任务。所有与 DB2 相关的操作系统资源必须在所有 MSCS 节点上可用。其中一些操作系统资源不在 MSCS 作用域之内。即，不能将它们定义为 MSCS 资源。必须确保配置每个系统，以便相同的操作系统资源在所有 MSCS 节点上都可用。后面几节描述必须执行的附加工作。

启动和停止 DB2 资源

必须从“群集管理员”工具启动和停止 DB2 资源。有几种方法可用于启动 DB2 实例，如 **db2start** 命令和“控制面板”中的服务选项。但是，若未从“群集管理员”启动 DB2，则 MSCS 软件将不知道 DB2 实例的状态。若使用“群集管理员”启动 DB2 实例，而使用 **db2stop** 命令停止它，则 MSCS 软件将 **db2stop** 命令解释为软件出错并试图重新启动 DB2。（当前 MSCS 界面不支持资源状态的通知。）

类似地，若使用 **db2start** 启动 DB2 实例，则 MSCS 不能检测该资源是否联机；若数据库服务器失效，则 MSCS 将无法在群集中的故障恢复机器上使用 DB2 资源联机。

有三个操作可用于 DB2 实例：

联机 此操作等效于使用 **db2start** 命令。若 DB2 已经是活动的，则此操作可以只用于通知 MSCS：DB2 是活动的。将把此操作执行期间的任何错误写入“Windows NT 事件日志”。

脱机 此操作等效于使用 **db2stop** 命令。若实例有任何活动的连接，此操作将失败。这与 **db2stop** 的特性是一致的。

使资源失效

此操作等效于使用指定了 **force** 选项的 **db2stop** 命令。DB2 将所有应用程序与 DB2 系统断开，并停止所有数据库服务器。

运行脚本

在将 DB2 资源联机前和联机后都可运行脚本。这些脚本必须驻留在为 DB2INSTPROF 环境变量指定的实例简要表目录中。此目录是 **db2icrt** 命令的 "-p" 参数所指定的目录路径。通过发出如下命令可获取此值：

```
db2set -i:instance_name DB2INSTPROF
```

此文件路径必须在一个群集磁盘上，以便该实例目录在所有群集节点上可用。

这些脚本文件不是必需的，且仅当它们在实例目录中存在时才运行。它们由“MSCS 群集服务”在后台启动。这些脚本文件必须重定向标准输出，以便捕捉脚本文件中命令返回的任何信息。该输出不显示到屏幕上。

在一个分区数据库环境中，缺省情况下，实例中的每个数据库分区服务器都使用相同的脚本。若需要区分该实例中的不同数据库分区服务器，则使用 DB2NODE 环境变量的不同赋值以标记特定的节点号（例如，在 db2cpre.bat 和 db2cpost.bat 文件中使用 IF 语句）。

在将 DB2 资源联机前运行脚本

若要在将 DB2 资源联机前运行一个脚本，则必须将该脚本命名为 db2cpre.bat。在将 DB2 资源联机前，DB2 调用函数，以便从 Windows NT 命令行处理器 (CLP) 启动此批处理文件，然后等待 CLP 完成执行。您可使用此批处理文件执行一些任务，如修改 DB2 数据库管理程序配置。若故障恢复系统是受约束的，则可能要更改某些数据库管理程序的参数值，且必须减少 DB2 所占用的系统资源。

位于 db2cpre.bat 脚本中的命令应当同步执行。否则，可能在脚本中的所有任务完成前将 DB2 资源联机，这将引起意外的行为。特别地，**db2cmd** 不应该在 db2cpre.bat 脚本中调用，因为这样的话它会启动另一个命令处理器，该命令处理器将与 **db2cmd** 程序异步运行 DB2 命令。

若要在 db2cpre.bat 脚本中使用 DB2 CLP 命令，应将这些命令放入一个文件，并在为 DB2 命令行处理器初始化 DB2 环境的一个程序内，将该文件作为 CLP 批处理文件来运行，然后等待 DB2 命令行处理器的完成。例如：

```
#include <windows.h>

int WINAPI DB2SetCLPEnv_api(DWORD pid);

void main (int argc, char *argv [ ] )
{
    STARTUPINFO          startInfo  = {0};
    PROCESS_INFORMATION  pidInfo    = {0};
    char title [32]      = "Run Synchronously";
    char runCmd [64]     =
```

```

                                "DB2 -z c:\\run.out -tvf c:\\run.clp";
/* Invoke API to setup a CLP Environment */
    if ( DB2SetCLPEnv_api (GetCurrentProcessId ()) == 0 ) 1
    {
        startInfo.cb           = sizeof(STARTUPINFO);
        startInfo.lpReserved   = NULL;
        startInfo.lpTitle      = title;
        startInfo.lpDesktop    = NULL;
        startInfo.dwX          = 0;
        startInfo.dwY          = 0;
        startInfo.dwXSize      = 0;
        startInfo.dwYSize      = 0;
        startInfo.dwFlags      = 0L;
        startInfo.wShowWindow  = SW_HIDE;
        startInfo.lpReserved2  = NULL;
        startInfo.cbReserved2  = 0;
    if ( CreateProcessA( NULL,
                        runCmd, 2
                        NULL,
                        NULL,
                        FALSE,
                        NORMAL_PRIORITY_CLASS CREATE_NEW_CONSOLE,
                        NULL,
                        NULL,
                        &startInfo,
                        &pidInfo))
        {
            WaitForSingleObject (pidInfo.hProcess, INFINITE);
            CloseHandle (pidInfo.hProcess);
            CloseHandle (pidInfo.hThread);
        }
    }
    return;
}

```

1 通过调入库 DB2API.LIB 解析 API DB2SetCLPEnv_api。此 API 设置一个允许调用 CLP 命令的环境。若从 db2cpre.bat 脚本调用此程序，命令处理器将等待 CLP 命令完成。

2 runCmd 是包含 DB2 CLP 命令的脚本文件的名称。

一个名为 db2clpex.exe 的样本程序可在 DB2 安装路径的 MISC 子目录中找到。此可执行文件与提供的示例相似，但它接受 DB2 CLP 命令作为命令行自变量。若要使用此样本程序，将它复制到 BIN 子目录中。可在 db2cpre.bat 脚本中使用此可执行文件，如下所示（INSTHOME 是实例目录）：

```
db2clpex "DB2 -Z INSTHOME\pre.log -tvf INSTHOME\pre.clp"
```

所有 DB2 ATTACH 命令或 CONNECT 语句应当明确指定用户，否则，将在与群集服务相关的用户帐户下执行它们。CLP 脚本也应当以 TERMINATE 命令完成，以结束 CLP 后台进程。

以下是 db2cpre.bat 文件的一个示例:

```
db2cpre.bat : 1
-----
db2c1pex "db2 -z INSTHOME\pre-%DB2NODE%.log -tvf INSTHOME\pre.clp" 2 - 5
-----

PRE.CLP 6
-----
update dbm cfg using MAXAGENTS 200;
get dbm cfg;
terminate;
-----
```

- 1 db2cpre.bat 脚本在与“群集服务”相关的用户帐户下执行。若必需 DB2 操作，则与“群集服务”相关的用户帐户必须是一个有效的 SQL 标识符，它由 DB2 定义。
- 2 INSTHOME 是实例目录。
- 3 对于每个节点，日志文件的名称必须不同，以避免当两个逻辑节点同时联机时可能产生的文件争用。
- 4 db2c1pex.exe 是一个样本程序，它使用命令行自变量来指定要调用的 CLP 命令。
- 5 必须使 db2c1pex.exe 样本程序在所有 MSCS 群集节点上可用
- 6 此示例中的 CLP 命令设置代理程序数的限制。

将 DB2 资源联机后运行脚本

若要在将 DB2 资源联机后运行脚本，则必须将脚本命名为 db2cpost.bat。当 DB2 资源已成功联机后，将从 MSCS 异步运行该脚本。可在此脚本中使用 **db2cmd** 命令，以执行 DB2 CLP 脚本文件。使用 **db2cmd** 命令的“-c”参数，来指定一旦该任务完成后实用程序应关闭所有窗口。例如:

```
db2cmd -c db2 -tvf mycmds.clp
```

“-c”参数必须是 **db2cmd** 命令的第一个自变量，因为它防止后台存在孤立的命令处理器。

若要在 DB2 资源进行故障恢复且成为活动后立即执行数据库活动，则 db2cpost.bat 脚本是有用的。例如，您可重新启动或激活该实例中的数据库，以便它们准备好用于用户存取。

以下是 db2cpost.bat 脚本的一个示例:

```
db2cpost.bat 1
-----
db2cmd -c db2 -z INSTHOME\post-%DB2NODE%.log -tvf INSTHOME\post.clp 2 - 4
```

POST.CLP **5**

```
restart database SAMPLE;  
connect reset;  
activate database SAMPLE;  
terminate;
```

- 1** db2cpost.bat 脚本在与“群集服务”相关的用户帐户下运行。若必需 DB2 操作，则与“群集服务”相关的用户帐户必须是一个有效的 SQL 标识符，它由 DB2 定义。
- 2** INSTHOME 是实例目录。
- 3** 对于每个节点，日志文件的名称必须不同，以避免当两个逻辑节点同时联机时可能产生的文件争用。
- 4** 可使用 **db2cmd** 命令，因为 db2cpost.bat 脚本可异步运行。必须使用“-c”参数来终止命令处理器。
- 5** 此示例中的 CLP 脚本包含重新启动和激活数据库的命令。此脚本将数据库返回到紧接在数据库管理程序启动后的一个活动状态。在一个分区数据库系统中，您应当除去 **ACTIVATE DATABASE** 命令，因为可将多个 DB2 资源同时联机。**RESTART DATABASE** 命令可能失败，因为另一个节点正在激活该数据库。若发生这种情况，重新运行该脚本，以确保正确地重新启动了数据库。

数据库考虑事项

当创建一个数据库时，确保数据库路径指向一个共享磁盘。这允许在所有 MSCS 节点上查看该数据库。所有日志和其他数据库文件也必须指向群集磁盘，以便 DB2 成功地进行故障恢复。若不执行这些步骤，则 DB2 系统将发生故障，因为在 DB2 看来好象是文件已删除或不可用。

还要确保设置数据库管理程序和数据库配置参数，以便在任何一个 MSCS 节点上支持 DB2 所占用的系统资源量。应将 *autorestart* 数据库配置参数设置为 ON，以便在进行故障恢复时的第一个数据库连接将使数据库处于一致的状态。

(*autorestart* 的缺省设置为 ON。)也可使用 db2cpost.bat 脚本来重新启动并激活数据库，使该数据库处于就绪状态。此方法更好，因为这样不会依赖于 *autorestart*，且使数据库处于就绪状态，而不受用户连接请求影响。

用户和组支持

DB2 依赖于 Windows NT 进行用户认证和组支持。要使一个 DB2 实例以无缝方式从一个 MSCS 节点到另一个进行故障恢复，每个 MSCS 节点都必须对相同的 Windows NT 安全性数据库具有存取权。您可使用“Windows NT 域安全性”来达到此目的。

在“域安全性”数据库中定义所有 DB2 用户和组。MSCS 节点必须是此域的成员，或此域必须是“可信赖域”。然后，DB2 将使用“域安全性”数据库来进行认证和组支持，与 DB2 正在哪个 MSCS 节点上运行无关。

若您使用的是本地帐户，必须在每个 MSCS 节点上复制该帐户。不建议使用此方法，因为它容易出错，且需要双重维护。

“DCE 安全性”也是一种受支持的认证模式，只要所有 MSCS 节点为相同 DCE 单元中的客户机。

您应当使 MSCS 服务与符合 DB2 命名约定的一个用户帐户相关。这允许 MSCS 服务对 DB2 执行在 `db2cpre.bat` 和 `db2cpost.bat` 脚本中可能必需的操作。

有关 Windows NT 用户和组支持的详情，参见 *管理指南：实现中的“使用 DB2 Windows NT 版进行用户认证”*。

通信考虑事项

DB2 在 MSCS 环境中支持两个 LAN 协议：

- TCP/IP
- NetBIOS

支持 TCP/IP，因为它是受支持的群集资源类型。要允许 DB2 使用 TCP/IP 作为一个分区数据库系统的通信协议，创建一个“IP 地址”资源，然后将它置于 DB2 资源所在的同一个组中，该组表示您要用作远程应用程序的协调程序节点的数据库分区服务器。然后，使用“群集管理员”工具创建从属性，以确保在启动 DB2 资源前 IP 资源是联机的。从而，DB2 客户机可编目 TCP/IP 节点目录项以使用此 TCP/IP 地址。

与 `svcname` 数据库管理程序配置参数相关的 TCP/IP 端口必须保留，以供参与该实例的所有机器上的 DB2 实例使用。与该端口号码相关的服务名也必须在所有机器上的 `services` 文件中是相同的。

虽然 NetBIOS 不是受支持的群集资源，但是可使用 NetBIOS 作为 LAN 协议，因为该协议确保 NetBIOS 名在 LAN 上是唯一的。当 DB2 注册 NetBIOS 名时，

NetBIOS 确保该名称没有在 LAN 上使用。在一个故障恢复方案中，当将 DB2 从一个系统移到另一个系统时，将从 MSCS 群集中的一个伙伴机器上注销 DB2 使用的 *nname*，而在另一台机器上注册。

DB2 NetBIOS 支持使用“NetBIOS 帧”(NBF)。此协议堆栈可与不同的逻辑适配卡号 (LANA) 相关。要确保 NetBIOS 对服务器的一致性存取，与 NBF 协议堆栈相关的 LANA 应当在所有群集节点上相同。可使用“控制面板”中的**网络**选项来配置它。您应当将 NBF 与 LANA 0 相关，因为它是 DB2 期望的缺省设置。

系统时间考虑事项

DB2 使用系统时间来记录特定操作的时间戳记。所有参与 DB2 故障恢复的 MSCS 节点必须使系统时区和系统时间同步，以确保 DB2 在所有机器上的操作取得一致。

使用“控制面板”对话框中的**日期/时间**选项，来设置系统时区。MSCS 具有一个时间服务，当 MSCS 节点连接为一个群集时，该服务用来校准日期和时间。但是，该时间服务只是每 12 个小时校准时间一次，若在一个系统上更改了时间，而在将时间同步前 DB2 进行了故障恢复，这可能引起问题。

若在 MSCS 群集的一个节点上更改了时间，则应当使用如下命令在其他群集节点上人工校准时间：

```
net time /set /y \\remote_node
```

其中 *remote_node* 是群集节点的机器名。

分区数据库环境中的管理服务器和控制中心考虑事项

“DB2 管理服务器”是在安装 DB2 通用数据库期间（可选地）创建的。它不是分区数据库系统。控制中心使用“管理服务器”提供的服务，来管理 DB2 实例和数据库。

在一个分区数据库系统中，一个 DB2 实例可驻留在多个 MSCS 节点上。这意味着一个 DB2 实例必须在控制中心下的多个系统上编目，以便无论该 DB2 实例在哪个 MSCS 节点上是活动的，该实例都是可存取的。

“管理服务器”实例目录不是共享的。您必须将“管理服务器”目录中的所有用户定义文件镜像到所有 MSCS 节点上，以对所有 MSCS 节点提供相同级别的管理。特别是，必须使用户脚本和调度的可执行文件在所有节点上可用。还必须确保调度的活动在一个 MSCS 群集中的所有机器上进行调度。

或者，代替在所有机器上复制“管理服务器”的方法是，您可能要使“管理服务器”进行故障恢复。对于以下示例，假设该群集中有两个 MSCS 节点，它们称为 MACH0 和 MACH1。MACH0 具有对“管理服务器”将使用的一个群集磁盘的存取权。还假设 MACH0 和 MACH1 都具有“管理服务器”。您将执行下列步骤，以使“管理服务器”高度可用：

1. 在每台机器上调用 **db2admin stop** 命令，以同时将两台机器上的“管理服务器”停止。
2. 在所有管理客户机上，使用 **UNCATALOG NODE** 命令，以取消编目对 MACH0 和 MACH1 上的“管理服务器”的所有引用。（可在客户机上使用 **LIST NODE DIRECTORY** 命令，以确定是否存在对“管理服务器”的任何引用。）
3. 通过从 MACH1 上调用 **db2admin drop** 命令，将“管理服务器”从 MACH1 上卸下。（若两台机器上都具有“管理服务器”，则只执行此步骤。）
4. 从 MACH0 上发出 **db2admin** 命令，确定“管理服务器”的名称。（缺省名称是 DB2DAS00。）
5. 使用 DB2MSCS 实用程序，为“管理服务器”设置故障恢复支持。这必然会在 MSCS 上创建一个名为 DB2DAS00 的 DB2 资源，它依赖于 IP 和磁盘资源。（若您具有一个相互替换配置，应当将该资源放在为 NODE0 保存 DB2 资源的组中。）此资源将用作支持“管理服务器”的 MSCS 资源。DB2MSCS.ADMIN 文件应是如下所示的格式：

```
#
# db2mscs.admin for Administration Server
# run db2mscs -f:db2mscs.admin
#
DB2_INSTANCE=DB2DAS00
CLUSTER_NAME=CLUSTERA
DB2_LOGON_USERNAME=db2admin
DB2_LOGON_PASSWORD=db2admin
# put Administration server in the same group as DB2 Node 0
GROUP_NAME=DB2NODE0 1
DISK_NAME=DISK E:
INSTPROF_DISK=DISK E:
IP_NAME= IP Address for Administration Server
IP_ADDRESS=9.9.9.8
IP_SUBNET=255.255.255.0
IP_NETWORK=Ethernet
```

1 此组可与现存组相同。这样，就不需要附加磁盘来用于实例简要表目录。

6. 在 MACH1 上调用如下命令，以将 DB2DAS00 设置为“管理服务器”：
`db2set -g db2adminsriver=DB2DAS00`
7. 在 MACH0 上，通过 Services 程序修改 DB2DAS00 的启动特性，以便人工而不是自动地启动它，因为 DB2DAS00 现在由 MSCS 控制。

当使“管理服务器”可进行故障恢复时，所有远程存取都应当使用 MSCS IP 资源来与“管理服务器”通信。现在，“管理服务器”将具有下列特性：

- “管理服务器”实例目录将使用“管理服务器”来进行故障恢复。
- 客户机将只编目单个节点以与“管理服务器”通信，而不管它在哪个 MSCS 节点上是活动的。
- 只需要对“管理服务器”调度作业一次。
- 仅当“管理服务器”在本地实例所在的相同 MSCS 节点上活动时，“管理服务器”才可控制本地实例。
- 若“群集服务”不是活动的，则“管理服务器”不可存取。

限制

当在 MSCS 环境中运行 DB2 时：

- 除非共享磁盘在两个 MSCS 节点上具有相同的物理磁盘号，否则，不能在共享磁盘上使用物理 I/O。可使用逻辑 I/O，因为磁盘是使用分区标识符来存取的。
- 必须为 MSCS 支持配置所有 DB2 资源。若没有这样做，DB2 运行期间将发生系统错误（在缺少系统资源的情况下，DB2 不能正确操作）。例如，若数据库日志没有在 MSCS 共享磁盘上，则 DB2 不能重新启动数据库。
- 必须从“群集管理员”工具管理 DB2 实例。MSCS 将把用于启动和停止数据库管理程序的其他方法看作是软件不一致性。例如，若您使用 MSCS 来启动 DB2，而使用 **db2stop** 命令来停止 DB2，则 MSCS 将把这种情况检测为软件错误，并重新启动实例。这还意味着您不应当使用控制中心来启动和停止 DB2。
- 要卸装 DB2，必须首先停止 MSCS。

第14章 Sun Cluster 2.2 上的 DB2 和高可用性

本章详细描述 DB2 如何使用 Sun Cluster 2.x (SC2.x) 来达到高可用性，包括高可用性代理程序的说明，该代理程序用作两个软件产品之间的中介（参见图59）。



图 59. DB2、Sun Cluster 2.x 和高可用性

高可用性

提供数据服务的计算机系统包含许多独特的部件，每个部件都有一个相关联的“平均无故障时间” (MTBF)。MTBF 是部件保持可用的平均时间。高质量硬盘驱动器的 MTBF 大约是一百万小时（大约 114 年）。虽然这看起来象是很长一段时间，但每 200 个硬盘中，就有一个有可能在 6 个月内失效。

虽然有很多方法可以提高数据服务的可用性，但最常用的方法还是 HA 群集。当用于高可用性时，群集由两台或更多机器、一组专用网络接口、一个或多个公用网络接口以及一些共享磁盘组成。这种特殊的配置允许将数据服务从一台机器移至另一机器。通过将数据服务移至群集中的另一机器，应该能够继续提供对其数据的存取。将数据服务从一台机器移至另一机器称为故障恢复，第262页的图60对其进行了说明。

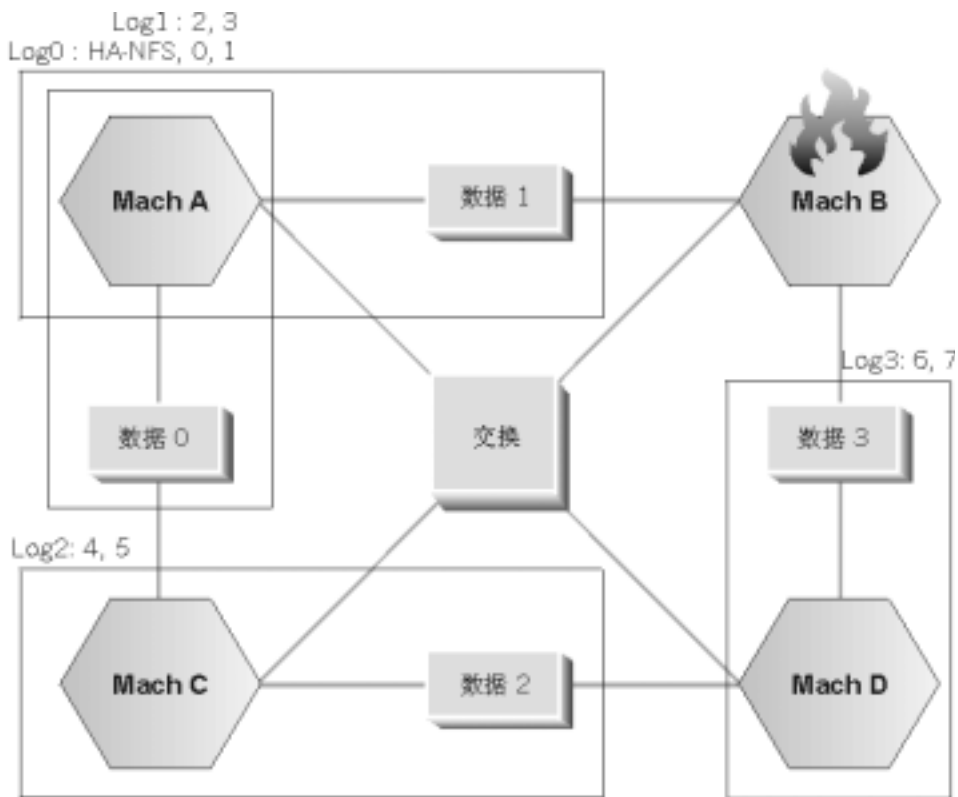


图 60. 故障恢复

专用网络接口用来在群集中的机器之间发送心跳信息和控制信息。公用网络接口用来直接与 HA 群集的客户机通信。HA 群集中的磁盘与群集中的两台或更多机器相连，因此，当一台机器失效时，另一机器可以存取它们。

在 HA 群集上运行的数据服务有一个或多个逻辑公用网络接口和一组磁盘与其相关联。HA 数据服务的客户机只通过 TCP/IP 与该数据服务的逻辑网络接口相连。如果执行故障恢复，则该数据服务及其逻辑网络接口和磁盘组便移至另一机器。

HA 群集的其中一个好处是可以在没有支持人员辅助的情况下恢复数据服务，并且，可以随时这样做。另一个好处是冗余。群集中的所有部件都应该是冗余的，包括机器本身。群集应当能够经受住任何单一故障点。

虽然高度可用的数据服务在本质上可以有非常大的区别，但它们还是有一些公共的需求。高度可用数据服务的客户机期望该数据服务的网络地址和主机名保持不变，并期望能够以同一方式发出请求，而不考虑该数据服务具体是在哪一台机器上。

假定一个 Web 浏览器正在存取高度可用 Web 服务器。请求是使用 URL（统一资源定位器）发出的，该 URL 包含主机名和 Web 服务器上文件的路径。浏览器期望该主机名和路径在 Web 服务器进行故障恢复后保持不变。如果浏览器从 Web 服务器下载文件，而该服务器正在进行故障恢复，则该浏览器将需要重新发出该请求。

数据服务的可用性是通过数据服务可供其用户使用的时间量测量的。最常用的可用性计量单位是“运行时间”的百分比；此百分比通常用数字“9”来表示：

99.99% => 服务（最多）停止 52.6 分钟/年
99.999% => 服务（最多）停止 5.26 分钟/年
99.9999% => 服务（最多）停止 31.5 秒/年

当设计和测试 HA 群集时：

1. 确保群集管理员熟悉系统并了解发生故障恢复时应发生的情况。
2. 确保群集的每一部分都确实是冗余的，当它失效时，可以被快速替换。
3. 强制一个测试系统在受控环境中失效，并确保它每次都能正确地进行故障恢复。
4. 跟踪每次故障恢复的原因。虽然这应该不会经常发生，但找出任何导致群集不稳定的问题非常重要。例如，如果群集的一个部分一个月导致了 5 次故障恢复，则找出原因并修正它。
5. 当发生故障恢复时，务必通知该群集的支持人员。
6. 不能使群集过载。确保其余的系统在发生故障恢复之后，仍能够处理适量的工作负荷。
7. 经常检查容易引起故障的部件（如磁盘），以便可以在发生问题之前进行替换。

故障容错和连续可用性

另一种提高数据服务可用性的方法是故障容错。故障容错机器内置了其所有冗余，应该能够经受住任何部件（包括 CPU 和内存）的单一故障。故障容错机器最常用在小型市场中，其实现成本比较高。机器位于不同地理位置的 HA 群集还有一个优点，即可以从仅影响那些位置的一部分的灾难中恢复。

连续可用性是一个基于高可用性的步骤。它保护其客户机不受已计划的和未曾计划的停机时间的影响。有了连续可用性配置，即使其中一台提供数据服务的机器失效或为了进行维护而被关闭，客户机也完全不受影响。连续可用性配置非常复杂，实现成本更高。

由于 HA 群集可缩放、易于使用以及实现成本相对较低，所以它是最常见的提高可用性的解决方案。

Sun Cluster 2.2

Sun Cluster 2.2 (SC2.2) 是 Sun Microsystems 的分群和高可用性产品。SC2.2 在单一群集中最多支持 4 台机器。通过使用 4 个 Ultra Enterprise 10000, 一个群集最多可以带有 256 个 CPU 和 256 GB 内存。

受支持的系统

系统	UltraSPARC	内存容量	I/O
Ultra Enterprise 1	1	64MB-1GB	3 SBus
Ultra Enterprise 2	1-2	64MB-2GB	4 SBus
Ultra Enterprise 450	1-4	32MB-4GB	10 PCI
Ultra Enterprise 3000	1-6	64MB-6GB	9 SBus
Ultra Enterprise 4000	1-14	64MB-14GB	21 SBus
Ultra Enterprise 5000	1-14	64MB-14GB	21 SBus
Ultra Enterprise 6000	1-30	64MB-30GB	45 SBus
Ultra Enterprise 10000	1-64	512MB-64GB	64 SBus

代理程序

Sun Cluster 软件包括许多受 SC2.2 产品支持并随其一起交付的高可用性代理程序。其他 HA 代理程序, 如用于 DB2 的代理程序, 不是由 Sun 开发的, 它们未随 Sun Cluster 软件一起交付。DB2 附带交付了用于 DB2 的 HA 代理程序, 它受 IBM 支持。

Sun Cluster 软件通过提供注册特定方法 (脚本或程序) 的机会来使用高度可用数据服务, 这些方法与 Sun Cluster 软件的各种部件相对应。利用这些方法, SC2.2 软件便可以控制数据服务, 而无需对其进行深入了解。这些方法包括:

START

用来在逻辑网络接口联机之前启动一部分数据服务。

START_NET

用来在逻辑网络接口联机之后启动一部分数据服务。

STOP 用来在逻辑网络接口脱机之后停止一部分数据服务。

STOP_NET

用来在逻辑网络接口脱机之前停止一部分数据服务。

ABORT

与 STOP 方法类似, 不同的是它刚好在群集软件关闭机器之前运行。在此

情况下，该机器的“健康”是有疑问的，在该机器关闭之前，数据服务可能想要执行“临终愿望”请求。在逻辑网络接口脱机后运行。

ABORT_NET

与 ABORT 方法类似，不同的是它在逻辑网络接口脱机之前运行。

FM_INIT

用来初始化故障监控程序。

FM_START

用来启动故障监控程序。

FM_STOP

用来停止故障监控程序。

FM_CHECK

由 **hactl** 命令调用。返回对应的数据服务的当前状态。

DB2 代理程序包含下列脚本：START_NET、STOP_NET、FM_START 和 FM_STOP。群集重新配置期间，不运行下列脚本：ABORT、ABORT_NET 和 FM_CHECK。

高可用性代理程序由这些方法中的一个或多个组成。这些方法通过 **hareg** 命令向 SC2.2 注册。一旦注册，Sun Cluster 软件就将调用相对应的方法来控制数据服务。

重要的是记住不能调用服务的 ABORT 和 STOP 方法。这些方法用于数据服务的受控关闭，如果机器失效，而未调用这些方法，则该数据服务必须能够恢复。

有关详情，参考 Sun Cluster 文档。

逻辑主机

SC2.2 软件使用了逻辑主机概念。逻辑主机由一组磁盘和一个或多个逻辑公用网络接口组成。高度可用数据服务与一个逻辑主机相关联，并需要该逻辑主机的磁盘组中的磁盘。逻辑主机可以驻留在群集中不同的机器上，并可以“借用”运行它们的机器的 CPU 和内存。

逻辑网络接口

与其他基于 UNIX 的操作系统一样，除用于网络接口的主 IP 地址外，Solaris 还有能力具有附加的 IP 地址。附加的 IP 地址驻留在一个逻辑接口上，其驻留方式与驻留在物理网络接口上的主 IP 地址的方式相同。下面是群集中两台机器上的逻辑接口的一个示例。共有两个逻辑主机，它们当前都在机器“thrash”上。

```

scadmin@crackle(202)# netstat -in
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 127.0.0.0 127.0.0.1 289966 0 289966 0 0 0
hme0 1500 9.21.55.0 9.21.55.98 121657 6098 764122 0 0 0
scid0 16321 204.152.65.0 204.152.65.1 489307 0 476479 0 0 0
scid0:1 16321 204.152.65.32 204.152.65.33 0 0 0 0 0 0
scid1 16321 204.152.65.16 204.152.65.17 347317 0 348073 0 0 0

```

1. lo0 是反馈接口
2. hme0 是公用网络接口（以太网）
3. scid0 是第一个专用网络接口（SCI 或可缩放 Coherent 接口）
4. scid0:1 是 Sun Cluster 软件内部使用的逻辑网络接口
5. scid1 是第二个专用网络接口

```

scadmin@thrash(203)# netstat -in
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis Queue
lo0 8232 127.0.0.0 127.0.0.1 1128780 0 118780 0 0 0
hme0 1500 9.21.55.0 9.21.55.92 1741422 5692 757127 0 0 0
hme0:1 1500 9.21.55.0 9.21.55.109 0 0 0 0 0 0
hme0:2 1500 9.21.55.0 9.21.55.110 0 0 0 0 0 0
scid0 16321 204.152.65.0 204.152.65.2 476641 0 489476 0 0 0
scid0:1 16321 204.152.65.32 204.152.65.34 0 0 0 0 0 0
scid1 16321 204.152.65.16 204.152.65.18 348199 0 347444 0 0 0

```

1. hme0:1 是逻辑主机的逻辑网络接口
2. hme0:2 是另一逻辑主机的逻辑网络接口

逻辑主机可以有一个或多个逻辑接口与其相关联。这些逻辑接口随逻辑主机一起在机器间移动，它们用来存取与该逻辑主机相关联的数据服务。因为这些逻辑接口随逻辑主机一起移动，所以客户机可以独立于数据服务所在的机器来存取数据服务。

高度可用数据服务应联编至 TCP/IP 地址 INADDR_ANY。这确保系统上的每个 IP 地址都可以接受数据服务连接。相反，若数据服务联编至某一特定 IP 地址，则它必须联编至与提供数据服务的逻辑主机相关联的逻辑接口。联编至 INADDR_ANY 还有一个好处，即当系统上新增了数据服务所需的新 IP 地址时，无需重新联编至该地址。

注：HA 实例的客户机应当使用逻辑主机的逻辑 IP 地址的主机名来编目数据库。它们决不应使用一台机器的主主机名，因为不能保证 DB2 将在该机器上运行。

磁盘组和文件系统

数据服务的磁盘与组中的一个逻辑主机相关联。如果群集正在运行 Sun StorEdge Volume Manager (Veritas)，则 Sun Cluster 软件使用 Veritas "vxvg" 实用程序来

为每个逻辑主机调入和调出磁盘组。下面是两个逻辑主机 "log0" 和 "log1" 的磁盘组的示例，这两个逻辑主机驻留在名为 "thrash" 的机器上。名为 "crackle" 的机器当前并未存放任何逻辑主机。

```
scadmin@crackle(206)# vxdg list
NAME STATE ID
rootdg enabled 899825206.1025.crackle

scadmin@thrash(205)# vxdg list
NAME STATE ID
rootdg enabled 924176206.1025.thrash
data0 enabled 925142028.1157.crackle=
data1 enabled 899826248.1108.crackle
```

磁盘组 "data0" 和 "data1" 分别与逻辑主机 "log0" 和 "log1" 相对应。磁盘组 "data0" 可以从 "thrash" 中调出，方法是运行

```
vxdg deport data0
```

并可调入 "crackle"，方法是运行

```
vxdg import data1
```

这是由 Sun Cluster 软件自动完成的，不应在活动群集上人工执行。

每个磁盘组都包含许多可以由群集中的两台或更多机器共享的磁盘。逻辑主机只能移至另一能够对其拥有的磁盘组中的磁盘进行物理存取的机器。

有两个文件控制着每个逻辑主机的文件系统：

```
/etc/opt/SUNWcluster/conf/hanfs/vfstab.<logical_host>
/etc/opt/SUNWcluster/conf/hanfs/dfstab.<logical_host>
```

其中，*logical_host* 是相关联的逻辑主机的名称。

vfstab 文件与 */etc/vfstab* 文件类似，不同的是它包含在为逻辑主机调入磁盘组之后要安装的文件系统的项。*dfstab* 文件与 */etc/dfs/dfstab* 文件类似，不同的是它包含要通过 HA-NFS 为逻辑主机调出的文件系统的项。每台机器都有这些文件的它们自己的副本，您应仔细确保这些文件在群集中每台机器上的内容都相同。

注：逻辑主机的 *vfstab* 和 *dfstab* 文件的路径容易令人误解，因为它们包含目录 *hanfs*。只有逻辑主机的 *dfstab* 文件才用于 HA-NFS。即使未配置 HA-NFS，也会使用 *vfstab* 文件。

下面是来自在相互替换配置中运行“DB2 通用数据库扩充企业版” (EEE) 的群集的示例：

```

scadmin@thrash(217)# ls -l /etc/opt/SUNWcluster/conf/hanfs
total 8
-rw-r--r-- 1 root build 173 Apr 14 15:01 dfstab.log0
-rw-r--r-- 1 root build 316 Apr 26 12:07 vfstab.log0
-rw-r--r-- 1 root build 389 Apr 13 21:04 vfstab.log1

scadmin@thrash(218)# cat dfstab.log0
share -F nfs -o root=crackle:thrash:\
jolt:bump:crackle.torolab.ibm.com:thrash.torolab.ibm.com:\
jolt.torolab.ibm.com:bump.torolab.ibm.com /log0/home

```

主机（被授予安装文件系统 /log0/home 的许可权）来自群集中每台机器上的所有网络接口（逻辑接口和物理接口）。文件系统使用超级用户许可权调出。

```

scadmin@thrash(220)# cat vfstab.log0
#device to mount          device to fsck          mount      FS   fsck mount  options
#                          #                          point      type pass at boot

/dev/vx/dsk/data0/data1-stat /dev/vx/rdisk/data0/data1-stat /log0      ufs  2   no   -
/dev/vx/dsk/data0/vo101     /dev/vx/rdisk/data0/vo101     /log0/home ufs  2   no   -
/dev/vx/dsk/data0/vo102     /dev/vx/rdisk/data0/vo102     /log0/data ufs  2   no   -

scadmin@thrash(221)# cat vfstab.log1
#device to mount          device to fsck          mount      FS   fsck mount  options
#                          #                          point      type pass at boot

/dev/vx/dsk/data1/data1-stat /dev/vx/rdisk/data1/data1-stat /log1      ufs  2   no   -
/dev/vx/dsk/data1/vo101     /dev/vx/rdisk/data1/vo101     /log1/home ufs  2   no   -
/dev/vx/dsk/data1/vo102     /dev/vx/rdisk/data1/vo102     /log1/data ufs  2   no   -
/dev/vx/dsk/data1/vo103     /dev/vx/rdisk/data1/vo103     /log1/data1 ufs  2   no   -

```

vfstab.log0 文件包含 /log0 目录下的文件系统的三个有效项。注意，逻辑主机 log0 的文件系统使用逻辑卷设备，这些逻辑设备是与该逻辑主机相关联的磁盘组 data0 的一部分。

vfstab 文件中的文件系统按从上到下的次序安装，因此重要的是确保以正确的次序列示文件系统。安装在特定文件系统下面的那些文件系统应列示在该文件系统下面。逻辑主机所需的实际文件系统取决于数据服务的需要，与这些示例会有很大的不同。

故障恢复期间，SC2.2 软件负责确保与一个逻辑主机相关联的磁盘组和逻辑接口跟着该逻辑主机在群集中的机器间移动。在故障恢复之后，高度可用数据服务预期这些资源中至少有一个在新系统上可用。事实上，许多数据服务甚至不知道它们是高度可用的，在故障恢复之后，必须使这些资源“看起来”丝毫未变。

控制方法

注册控制方法应使用

```
hareg(1m)
```

在注册 HA 服务之后，SC2.2 就负责在群集重新配置或故障恢复期间的适当时候调用已对 HA 服务注册的方法。

在群集重新配置（受控故障恢复）期间，（以给出的次序）执行下列操作：如果机器崩溃，将不执行步骤 5c 之前的操作。（有关群集重新配置的详情，参考 SC2.2 文档。）

1. 运行 FM_STOP 方法。
 2. 运行 STOP_NET 方法。
 3. 使逻辑主机的逻辑接口脱机。
 - ifconfig hme0:1 0.0.0.0 down
 4. 运行 STOP 方法。
 5. 移动磁盘组和文件系统。
 - a. 卸下逻辑主机文件系统。
 - b. 在一台机器上对磁盘组执行 vxvg deport 操作。
- - 如果机器崩溃，则只运行下面的步骤 - -
- c. 在另一机器上对磁盘组执行 vxvg import 操作。
 - d. 对逻辑主机文件系统执行 fsck。
 - e. 安装逻辑主机文件系统。
 6. 运行 START 方法。
 7. 使逻辑主机的逻辑接口联机。
 - ifconfig hme0:1 <ip address> up
 8. 运行 START_NET 方法。
 9. 运行 FM_INIT 方法。
 10. 运行 FM_START 方法。

控制方法是使用下列命令行自变量运行的：

```
METHOD <logical hosts being hosted> <logical hosts not being hosted> <time-out>
```

第一个自变量是当前存放的逻辑主机的逗号定界列表，第二个自变量是未存放的逻辑主机的逗号定界列表。最后一个自变量是该方法的超时，即 SC2.2 软件在异常终止该方法之前允许它运行的时间量。

磁盘和文件系统配置

SC2.2 支持两个卷管理器：Sun StorEdge Volume Manager (Veritas) 和 Solstice Disk Suite。虽然这两个软件的表现都很优良，但 StorEdge Volume Manager 在群集环境中更有优势。在某些群集配置中，群集中每台机器的磁盘罩的控制器编号可以不同。若控制器编号不同，则该控制器的磁盘设备的路径也将不同。因为 Disk Suite 直接使用磁盘设备路径，所以它在此环境中不能很好地工作。StorEdge Volume Manager 使用磁盘本身，而不考虑控制器编号，因而即使控制器编号不同也不受影响。

因为 HA 的目标是提高数据服务的可用性，所以重要的是确保所有文件系统和磁盘设备都被镜像，或采用 RAID 配置。这能预防因磁盘失效而导致的故障恢复，并能提高群集的稳定性的。

HA-NFS

当跨多台机器配置一个实例时，DB2 UDB EEE 需要一个共享文件系统。典型的 DB2 UDB EEE 配置是通过 NFS 从一台机器调出主目录，并将其安装在所有参与 EEE 实例的机器上。对于相互替换配置，DB2 UDB EEE 依靠 HA-NFS 来提供共享的、高度可用的文件系统。其中一个逻辑主机通过 HA-NFS 调出一个文件系统，然后，群集中的每台机器安装该文件系统作为 EEE 实例的主目录。有关 HA-NFS 的详情，参考 Sun Cluster 文档。

cconsole 和 ctelnet 实用程序

SC2.2 附带提供了两个很有用的实用程序：cconsole 和 ctelnet。这些实用程序可以用来同时向群集中的数台机器发出单一命令。使用这些实用程序编辑配置文件可以确保该文件在群集中的所有机器上保持完全相同。这些实用程序还可用来以完全相同的方式在每台机器上安装软件。有关这些实用程序的详情，参考 Sun Cluster 文档。

校园分群和大陆分群

当一个群集的机器不在同一建筑物中时，该群集被称为校园群集。校园群集对于作为单一故障点除去建筑本身很有用。例如，如果群集中的机器全都在同一建筑物中，而该建筑物烧毁了，则整个群集都受到影响。然而，如果机器在不同的建筑物中，则即使有一座建筑物烧毁了，群集也可以幸免于难。

大陆群集的机器分布在不同的城市中。在这种情况下，目标是作为单一故障点除去某个地理区域。这种类型的群集提供了针对灾难事件（如地震和海啸）的保护。

目前，Sun Cluster 支持机器最多相距 10 公里（即大约 6 英里）。这使校园分群对于那些在两个不同站点之间需要高速连接的人而言成为可行的选项。群集需要两个专用互连，并需要许多光纤电缆用于共享磁盘。两个站点之间进行高速连接的成本可能会抵消掉它的优点。

常见问题

SC2.2 软件使用“群集配置数据库”（即 CCD(4)）来为群集配置提供单群集范围的储存库。CCD 有一个专用 API，它存储在 /etc/opt/SUNWcluster/conf 目录下。在极为罕见的情况下，CCD 会变得不同步，可能需要修复。在这种情况下，最好的 CCD 修复方法是从副本复原它。

要备份 CCD，需在群集中的所有机器上关闭群集软件，“压缩”（tar）/etc/opt/SUNWcluster/conf 目录，并将 tar 文件存储在安全的地方。如果在进行

备份时未关闭群集软件，则复原 CCD 时可能会发生问题。在每次更改群集配置时都执行新备份，以确保副本总是最新的。要复原 CCD，需在群集中的所有机器上关闭群集软件，将 conf 目录移至 conf.old，并“解压”(untar)副本。之后，便可以使用新的 CCD 启动群集。

DB2 考虑事项

本节包括下列主题:

- 『连接 HA 实例的应用程序』
- 第272页的『EE 和 EEE 实例的磁盘布局』
- 第273页的『EE 和 EEE 实例的主目录布局』
- 第274页的『逻辑主机和 DB2 UDB EEE』
- 第275页的『DB2 安装位置和选项』
- 第275页的『数据库和数据库管理程序配置参数』
- 第276页的『应急恢复』
- 第276页的『通过数据复制的高可用性』。

连接 HA 实例的应用程序

依赖高度可用 DB2 实例的应用程序必须能够在发生故障恢复时进行重新连接。因为逻辑主机的主机名和 IP 地址保持不变，所以无需连接另一主机名或重新编目数据库。

假设有一个群集带有两台机器和一个“DB2 通用数据库企业版”(EE)实例。EE 实例通常驻留在群集中的其中一台机器上。HA 实例的客户机将连接与该 HA 实例相关联的逻辑主机的逻辑 IP 地址(或主机名)。

依照 HA 客户机，有两种类型的故障恢复。其中一种类型的故障恢复在存放 HA 实例的机器崩溃时发生。另一种类型的故障恢复在 HA 有机会平缓关闭时发生。

若一台机器崩溃并关闭 HA 实例，则与数据库的现存连接和新连接都将挂起。连接挂起的原因是网络上没有任何机器具有客户机用于该数据库的 IP 地址。若数据库平缓关闭，则 db2stop force 断开与数据库的现存连接，并返回错误信息。

故障恢复期间，与数据库相关联的逻辑 IP 地址脱机，或者是因为 SC2.2 软件使其脱机，或者是因为存放逻辑主机的机器崩溃。此时，与该数据库的任何新连接都将被短暂挂起。

在 DB2 启动之前，与该数据库相关联的逻辑 IP 地址最终在另一机器上联机。在此阶段，与该数据库的连接不被挂起，但将接收到通信错误，因为 DB2 尚未在系

统上启动。仍与该数据库相连的 DB2 客户机也将开始接收通信错误。虽然那些客户机仍相信它们是连接着的，但开始存放该逻辑 IP 地址的机器却不了解任何现存的连接。连接仅仅是进行复位，DB2 客户机接收到通信错误。片刻之后，DB2 将在该机器上启动，现在可以成功地连接该数据库了。此时，数据库可能不一致，客户机可能必须要等待它恢复。

当针对 HA 环境设计应用程序时，不必为数据库连接的挂起阶段编写特殊的代码。连接只是在 Sun Cluster 软件移动逻辑 IP 地址时被短暂挂起。在此阶段，在 Sun Cluster 上运行的任何数据服务都将遇到同一连接挂起问题。无论数据库是如何关闭的，客户机都将接收到错误信息，必须尝试重新连接，直到成功为止。从客户机的角度而言，就象是 HA 实例先是关闭，然后在同一机器上重新打开。在受控故障恢复中，在客户机看来，就象是 HA 实例被强制关闭，而客户机稍后可以重新连接同一机器上的数据库。在不受控故障恢复中，在客户机看来，就象是数据库服务器崩溃，然后很快在同一机器上重新启动。

EE 和 EEE 实例的磁盘布局

DB2 期望它所需的磁盘设备或文件系统在群集中的每台机器上看起来都是相同的。要确保发生这种情况，应将必需的磁盘或文件系统配置为：它们跟随与 HA 实例相关联的逻辑主机，并在群集中的每台机器上都有相同的路径名。

DMS 和 SMS 表空间在 HA 环境中都受支持。DMS 表空间的设备容器必须使用由卷管理器创建的原始设备，它们或者是镜像的，或者采用 RAID 配置。不应使用规则磁盘设备，如 `/dev/rdisk/c20t0d0s0`，原因是：

- 它增加了同时从多台机器写该设备的可能性。
- 控制器编号在另一机器上可能不同。

如果 DB2 在此情况下进行故障恢复，则它所需的磁盘设备看起来与它们在另一机器上时不同，并且 DB2 也不启动。DMS 表空间的文件容器以及 SMS 表空间的容器必须驻留在已安装的文件系统上。当一个逻辑主机的 `vfstab` 文件中包括了该逻辑主机的文件系统时，便会自动安装那些文件系统。

一个逻辑主机的 `vfstab` 文件在以下路径中：

```
/etc/opt/SUNWcluster/conf/hanfs/vfstab.<logical_host>
```

其中，`logical_host` 是与该 `vfstab` 文件相关联的逻辑主机的名称。

每个逻辑主机都有它自己的 `vfstab` 文件，该文件包含将该逻辑主机的磁盘组传送至当前机器之后（但在启动 HA 服务之前）要安装的文件系统。Sun Cluster 软件在运行 `fsck`（文件系统检查）以确保文件系统健康之后，它将尝试安装任何正确定义的文件系统。若 `fsck` 失败，则不安装文件系统，并记录错误信息。

注：若一个进程打开了文件，或其当前工作目录在安装点之下，则安装将失败。
为了预防这种情况，确保逻辑主机的 `vfstab` 文件中包含的安装点下面没有留下任何进程。

当使用 SMS 表空间时，可以对 EEE 实例的文件系统布局使用任何约定。下面是 `hadb2_setup` 实用程序所使用的约定：

```
scadmin@crackle(190)# pwd
/export/ha_home/db2eee/db2eee
scadmin@crackle(191)# ls -l
total 18
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0000 -> /log0/disks/db2eee/NODE0000
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0001 -> /log0/disks/db2eee/NODE0001
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0002 -> /log0/disks/db2eee/NODE0002
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0003 -> /log0/disks/db2eee/NODE0003
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0004 -> /log0/disks/db2eee/NODE0004
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0005 -> /log1/disks/db2eee/NODE0005
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0006 -> /log1/disks/db2eee/NODE0006
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0007 -> /log1/disks/db2eee/NODE0007
lrwxrwxrwx 1 root build 28 Aug 12 19:08 NODE0008 -> /log1/disks/db2eee/NODE0008
scadmin@crackle(192)#
```

实例拥有者是 `db2eee`，`db2eee` 实例的缺省数据库目录是 `/export/ha_home/db2eee`。逻辑主机 `log0` 存放数据库分区 0、1、2 和 3，而逻辑主机 `log1` 存放数据库分区 4、5、6、7 和 8。

对于每个数据库分区，都有一个对应的 `NODExxxx` 目录。数据库分区的节点目录指向相关联的逻辑主机文件系统下面的一个目录。

当选择路径约定时，确保：

1. 文件系统的磁盘在特定逻辑主机的磁盘组中，该逻辑主机对需要这些磁盘的数据库分区负责。
2. 通过逻辑主机的 `vfstab` 文件安装存放容器的文件系统。

EE 和 EEE 实例的主目录布局

对于 EE 实例，主目录应该是逻辑主机的 `vfstab` 文件中定义的文件系统。此目录在 DB2 启动之前便可用，并随 DB2 一起被传送到群集中逻辑主机所移到的位置。每台机器都有它自己的 `vfstab` 文件副本，您应仔细确保该文件在每台机器上的内容都相同。下面是一个 EE 实例的主目录的示例：

```
/log0/home/db2ee
```

其中，`/log0` 是逻辑主机 `log0` 的逻辑主机文件系统，而 `db2ee` 是 DB2 实例的名称。应将这个主目录路径放在群集中可以存放“`db2ee`”实例的每台机器的 `/etc/passwd` 文件中。

对于 EEE 实例，有两种设置主目录的方法。对于热备份配置，设置主目录的方法可以与用于 EE 实例的方法相同。对于相互替换配置，必须将 HA-NFS 用于主目录，且必须在设置 EEE 实例之前正确配置它。

群集中的其中一台机器必须使用所选逻辑主机的 `dfstab` 文件来为 EEE 实例调出文件系统。`dfstab` 文件包含当一台机器存放逻辑主机时应通过 NFS 调出的文件系统。每台机器都有它自己的 `dfstab` 文件副本，您应仔细确保该文件在每台机器上的内容都相同。

有关 HA-NFS 文件系统的信息（通过 `hadb2_setup` 程序）放在 `hadb2tab` 文件中。当 HA 代理程序读取实例的信息时，它自动为该实例安装 HA-NFS 文件系统（参见第277页的『`hadb2tab` 文件』）。

HA-NFS 文件系统的安装点通常是 `/export/ha_home`。在群集中的每台机器上，这将是调出 HA-NFS 目录的逻辑主机安装的 NFS。EEE 实例拥有者的主目录放在此目录下面，名为：

```
/export/ha_home/<instance>
```

其中，*instance* 是实例拥有者的名称。

实例在每台机器上都可以有一个主目录，这样可以避免反复安装和卸下。但这样将进行额外的管理，才能确保主目录在每台机器上都是完全相同的。未能这样做会导致 DB2 不能正确启动，或导致它使用另一配置启动。这不是受支持的配置。

逻辑主机和 DB2 UDB EEE

通常选择逻辑主机来存放一个或多个数据库分区，以及调出 HA-NFS 文件系统。例如，若群集中有四个数据库分区和两台机器，则每台机器都应该有一个逻辑主机（第275页的图61）。其中一个逻辑主机可以存放两个数据库分区并调出 HA-NFS 文件系统，而另一个逻辑主机可以存放其余两个数据库分区。

在缺省情况下，DB2 UDB EEE 实例分配足够的资源，以成功地向已带有一个或多个用于该实例的活动数据库分区的机器添加最多两个数据库分区。例如，若群集中的单个实例有四个数据库分区，则仅当每个逻辑主机有一个数据库分区，或一个逻辑主机存放了三个数据库分区时，才会有所影响。在任何一种情况下，都有可能将三个数据库分区执行故障恢复至已存放有一个用于同一实例的数据库分区的机器。

可使用 `DB2_NUM_FAILOVER_NODES` 注册表变量来增加为进行故障恢复的数据库分区保留的资源量。

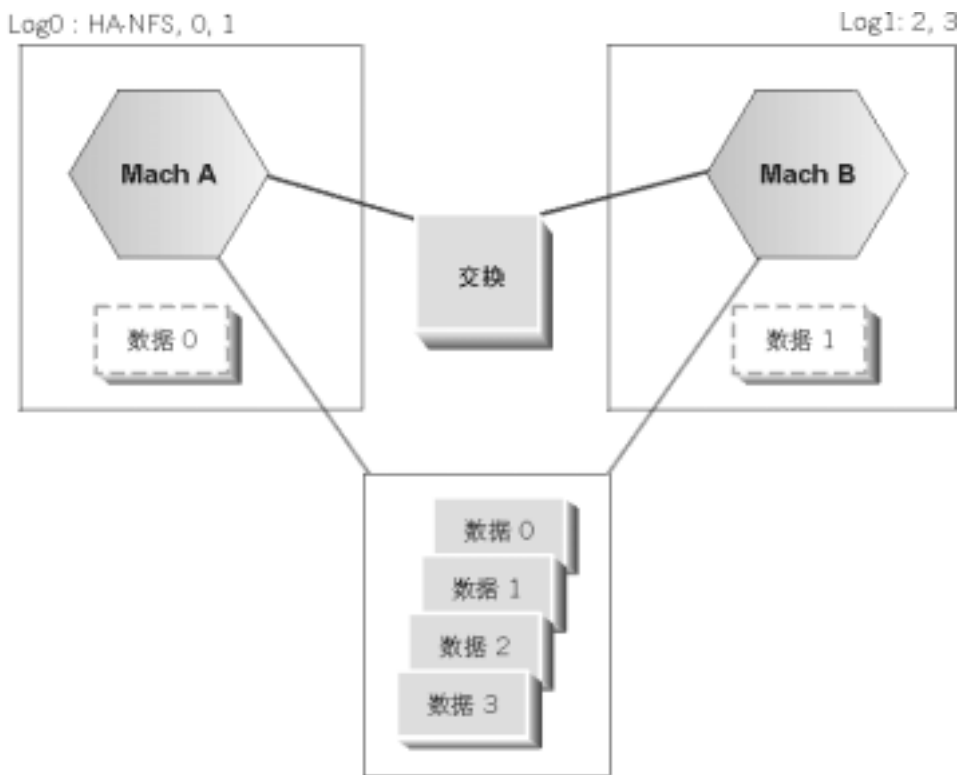


图 61. 每台机器一个逻辑主机

DB2 安装位置和选项

应对安装有 DB2 的文件系统执行镜像，或者，它至少应采用 RAID 配置。若 DB2 安装在规则磁盘上，则更有可能发生磁盘故障；所导致的故障恢复被认为是可以预防的，这降低了群集的稳定性的。

因为 HA 代理程序始终需要对 DB2 库具有存取权，所以不能将 DB2 安装在用于逻辑主机的磁盘组中的磁盘上。若 HA 代理程序对 DB2 库不具有存取权，则它们将失败。必须将 DB2 正常地安装在群集中的每台机器上。

数据库和数据库管理程序配置参数

可以使用 `pre_db2start` 脚本来在故障恢复之后但在 DB2 启动之前更改数据库管理程序配置参数（参见第279页的『用户脚本』）。这个可执行脚本在实例拥有者的主目录的 `sql1lib/ha` 目录下运行（如果它存在的话）。正如其名称所暗示的那样，它刚好在 `db2start` 之前运行。除非该实例是 EEE 实例，否则将传送给控制

方法的相同自变量传送给 `pre_db2start` 脚本。对于 `EEE` 实例，还将 `db2start` 命令的节点号传送给 `pre_db2start` 脚本。

应急恢复

`HA` 环境中的应急恢复与规则环境中的应急恢复相同。即使启动 `HA` 实例的机器与其发生崩溃时所在的机器不同，该实例的文件和磁盘设备看起来也是一样的，而恢复数据库所需的操作也不会有什么不同。有关应急恢复和其他形式的数据库恢复的详情，参见 *管理指南：实现中的“恢复数据库”*。

虽然可以人工地（或通过其中一个用户脚本）重新启动数据库，但还是建议您将 `autorestart` 数据库配置参数设置为 `ON`，特别是对于 `EEE` 实例而言更是如此。这将把数据库处于不一致状态的时间长度缩至最短。

通过数据复制的高可用性

也可以通过复制增强数据可用性。通过在两台服务器之间复制数据，可以达到某种形式的高可用性。如果其中一台服务器关闭，则另一服务器应能够替换它并继续提供数据服务。

但是因为复制是异步执行的，因此当另一服务器关闭时，可能尚未将某些更改传播至该服务器。

DB2 高可用性代理程序

`DB2` 高可用性代理程序在 `DB2` 与 `SC2.x` 之间起中介作用。它为 `Sun Cluster 2.2` 软件提供了一种控制分群环境中的 `DB2` 的方法，而无需深入了解 `DB2`。`EE` 和 `EEE` 实例都有一个代理程序。代理程序既支持管理实例，又支持数据库实例。

注册 `hadb2` 服务

要使用 `SC2.2`，必须注册 `DB2 HA` 代理程序。注册数据服务告知 `SC2.2` 有哪些控制方法可用，以及它们驻留在哪一个目录中。`HA` 代理程序附带交付了一个名为 `hadb2_reg` 的特殊脚本，它可以为 `EE` 和 `EEE` 实例注册 `hadb2` 服务。只需要对整个群集运行一次 `hadb2_reg` 脚本。

虽然只有一组用于 `DB2 HA` 代理程序的控制方法，但它们的注册方式取决于是否将在相互替换配置中使用 `EEE` 实例。对于热备份配置中的 `EE` 实例或 `EEE` 实例，不使用 `HA-NFS`；因此，不需要 `-d nfs` 开关（它告知 `SC2.2` 软件 `hadb2` 服务依赖于 `HA-NFS`）。

hadb2_reg 用来为 EEE 实例注册 DB2 V7.1 控制方法的实际命令是:

```
hareg -r hadb2 -b /opt/IBMDB2/V7.1/ha -m
START=hadb2_start,START_NET=hadb2_startnet,STOP_NET=hadb2_stopnet,
FM_START=hadb2_fmstart,FM_STOP=hadb2_fmstop
-t START_NET=$TIMEOUT,STOP_NET=$TIMEOUT -d nfs
```

-b 开关告知 SC2.x 在 opt/IBMDB2/V7.1/ha 目录中查找所有控制方法。-m 开关定义 hadb2 服务的实际控制方法。-t 开关定义 START_NET 和 STOP_NET 控制方法的超时。有关每个控制方法的详细说明,参考 Sun Cluster 文档。

可使用 hadb2_unreg 脚本来注销 hadb2 服务,并且,与 hadb2_reg 一样,它只需要对群集运行一次。

hadb2tab 文件

hadb2tab 文件是 DB2 HA 代理程序的主配置文件。每个控制方法都查阅此文件,以了解哪些实例是高度可用的。hadb2tab 文件位于 DB2 UDB 版本 7.1 的 /var/db2/v71/ 目录下面。此文件支持多个实例,每一个非注释行都表示一个不同的 HA 实例。下面是 hadb2tab 文件的一个示例:

```
<scadmin@thrash(203)# cat hadb2tab
EEE DATA db2eee jolt ON /export/ha_home /log0/home #Added by DB2 HA software
EE ADMIN db2ee log1 ON - - #Added by DB2 HA software
```

第一个字段向 DB2 HA 代理程序指示该实例是 EE 实例还是 EEE 实例。第二个字段指示该实例是数据实例还是管理实例。第三个字段包含该 HA 实例的用户名。第四个字段是该实例的逻辑主机或 HA-NFS 主机,这取决于它是 EE 实例还是 EEE 实例。第五个字段指示该实例的故障监控是打开的还是关闭的。最后两个字段分别是本地安装点和远程 HA-NFS 目录。如果不使用这些字段,则应将它们设置为 - (连字符),并且,这些字段只应配合 EEE 相互替换配置使用。hadb2tab 文件中允许注释,如果一行中“#”标记前面的信息的长度为零,或该信息不是一个实例的有效定义,则该行被视为注释。

控制方法

SC2.2 代理程序的控制方法可以是一组脚本或程序。Solaris 上 DB2 的代理程序是一组包括下列方法的程序:

START_NET

hadb2_startnet, 用来启动 DB2

STOP_NET

hadb2_stopnet, 用来停止 DB2

FM_START

hadb2_fmstart, 用来启动 DB2 的故障监控程序

FM_STOP

hadb2_fmstop, 用来停止 DB2 的故障监控程序

有关这些控制方法的详情, 参考 Sun Cluster 文档。

对于 EE 实例, 与该实例相关联的逻辑主机就是在 hadb2tab 文件中定义的。然而, 对于 EEE 实例, 还必须在以下文件中查找控制方法:

```
~<instance>/sql1lib/ha/hadb2-eee.cfg
```

其中, ~<instance> 是实例拥有者的主目录。此文件对每个数据库分区都包含一行, 它用来将数据库分区与逻辑主机相关联。有效的 hadb2-eee.cfg 文件的一个示例是:

```
crackle % cat hadb2-eee.cfg
NODE:log0 0
NODE:log0 1
NODE:log1 2
NODE:log1 3
```

实例或数据库分区跟随着相对应的逻辑主机围绕在群集周围。逻辑主机可以移至群集中任何受下层硬件和 SC2.2 支持的机器。若配置设置正确, 则 DB2 将支持任何受 SC2.2 软件支持的拓扑结构。

在读取了一个实例的所有信息之后, 控制方法便知道有哪些逻辑主机与该实例相关联。在对命令行自变量进行了语法分析之后, 控制方法还知道当前机器存放了哪些逻辑主机, 以及未存放哪些逻辑主机。

下表显示了各控制方法在运行时所执行的操作, 以及当前机器上是否存放了与数据库分区或实例相关联的逻辑主机。

控制方法	存放相关联的逻辑主机	不存放相关联的逻辑主机
START_NET	启动 DB2 实例或数据库分区	无操作
STOP_NET	无操作	停止 DB2 实例或数据库分区
FM_START	启动实例的故障监控程序	无操作
FM_STOP	无操作	停止实例的故障监控

执行启动操作的控制方法只关心当前存放的逻辑主机, 执行停止操作的控制方法只关心当前未存放的逻辑主机。

若是在使用 HA-NFS, 则控制方法还需要以一种特殊的方法安装 HA-NFS 目录。若未将 HA-NFS 的本地安装点和目录定义为 - (连字符), 则控制方法在本地安

装点上运行 `statvfs(2)`。若本地安装点的文件系统类型不是 `nfs`，则代理程序尝试使用 `hadb2tab` 行中的信息来安装文件系统。若已将 HA-NFS 的安装点和目录定义为 `-`（连字符），则需要对应的逻辑主机的 `vfstab` 文件来安装包含该实例的主目录的文件系统。对于 `EE` 和 `EEE` 热备份配置，只应将 HA-NFS 的本地安装点和远程目录定义为 `-`（连字符）。

用户脚本

这些脚本从控制方法中运行，以添加附加的功能，它们与控制方法传送相同的命令行自变量，此外，它们是由系统管理员或数据库管理员编写的。

若必须从不在后台运行的脚本中运行一个程序，则考虑使用 `nohup(1)` 使该程序在后台运行。`nohup` 程序保护执行的程序不受 `SIGHUP`（或挂断）信号干扰。若没有 `nohup`，则脚本中以后台方式运行的程序可能会被该脚本完成时发出的 `SIGHUP` 信号结束。

控制方法运行下列脚本：

- `/var/db2/v61/failover`
- `~<instance>/sqllib/ha/pre_db2start`
- `~<instance>/sqllib/ha/post_db2start`
- `~<instance>%s/sqllib/ha/post_failover`
- `~<instance>/sqllib/ha/pre_db2stop`
- `~<instance>/sqllib/ha/fm_warning`

其中，`~instance` 是 HA 实例的主目录。

除 `fm_warning` 脚本之外，每个用户脚本都是使用调用它的控制方法的自变量运行的。使用 `EEE` 实例时，还将数据库分区编号（作为最后一个自变量）传送给用户脚本。

`/var/db2/v71/failover` 脚本是在 `START_NET` 方法开始时调用的，它在后台运行。举个例子，这样的脚本可用来在发生故障恢复时向支持人员发送电子邮件。下面是故障恢复脚本的一个示例：

```
#!/bin/ksh
# E-mail or page support staff to notify them that a failover has occurred.
echo "Failover occurred on machine 'hostname':Running $0!" |/bin/mail admin@sphere.torolab.ibm.com
```

要成功地从脚本中发送电子邮件，必须在系统上正确地配置 `sendmail(1m)`。

正如其名称所暗示的那样，`pre_db2start` 脚本刚好在调用 `db2start` 之前运行。此脚本可用于诸如更改数据库管理程序配置参数之类的任务。它必须在 20 秒之内完

成。对于 **EEE** 实例，在每个数据库分区上调用 **db2start** 之前运行此脚本。此脚本仅适用于数据实例，而不适用于管理实例。

类似地，刚好在调用 **db2start** 之后运行 **post_db2start** 脚本。此脚本可用于诸如重新启动数据库之类的任务。它在后台运行，以确保其执行时间不会干扰其他实例。此脚本仅适用于数据实例，而不适用于管理实例。

在处理实例之后，运行实例所有者主目录下面的 **post_failover** 脚本。可使用此脚本来通知客户机应用程序：DB2 现在已开始工作、来激活数据库或将状态文件发送给管理员。它在后台运行，以防止其执行时间延迟其他 HA 实例的操作。下面是后故障恢复脚本的一个示例：

```
#!/bin/ksh
#

# Send the status file to the administrator.
mail admin@sphere.torolab.ibm.com </tmp/HA.info.db2eee
```

DB2 HA 代理程序的 **START_NET** 和 **STOP_NET** 方法都在处理每个实例之后创建一个状态文件。状态文件的名称是：

```
/tmp/HA.info.<instance>
```

其中，*instance* 是实例拥有者的用户名。此状态文件包含该实例的启动和停止报告，以及运行控制方法所花费的时间。下面是状态文件的一个示例：

```
scadmin@crackle(173)# cat /tmp/HA.info.db2eee
----- Elapsed Time: 00:00:18 -----
----- Elapsed Time: 00:00:00 (HA-NFS) -----

NODE      ACTION      RESULT      TRIES      RC
-----
4         stop       success     3          1064
5         stop       success     1          1064
6         stop       success     2          1064
7         stop       success     2          1064
8         stop       success     1          1064
-----
```

pre_db2stop 脚本刚好在调用 **db2stop** 之前运行。可使用此脚本来通知客户机应用程序：DB2 将要停止。它必须在 20 秒之内完成。此脚本仅适用于数据实例，而不适用于管理实例。

当 DB2 因为意外关闭而重新启动时，故障监控程序也将运行一个用户脚本。此脚本名为：

```
~/<instance>/sqlllib/ha/fm_warning
```

可使用 `fm_warning` 脚本来通知系统管理员：故障监控程序重新启动了 DB2。系统管理员应尝试查出 DB2 意外关闭的原因，并采取适当的措施来防止它再次发生。`fm_warning` 脚本在后台运行。

其他考虑事项

若关闭了 HA 数据服务，则故障恢复或群集重新配置期间只运行停止方法；仅当正确注册并打开了 HA 数据服务时，才运行其他方法。

确保群集中的每台机器都有足够的资源来运行它所负责的所有数据服务。在将群集投入生产环境之前，必须要考虑资源，如 CPU 负荷、内存、交换和核心参数。例如，若群集中的一台机器可能需要运行两个 DB2 实例，则该机器的核心参数需求将是这两个实例的需求之和。

故障监控程序

若打开故障监控，则群集重新配置或故障恢复期间，将启动故障监控程序。若 `START_NET` 脚本未启动 DB2，则故障监控程序将自行启动 DB2。故障监控程序可以检测出 DB2 是尚未启动，还是因为未知的原因而关闭了。正因为如此，重要的是不要在打开故障监控程序时人工关闭 DB2。故障监控程序将把这视为意外关闭，并重新启动 DB2。如果发生的次数太多，则它将对适当的逻辑主机执行故障恢复。

当对实例启用了故障监控时，正确的人工启动或停止实例的方法是首先关闭故障监控或 `hadb2` 服务。这两个操作都可以通过 `hadb2_setup` 命令使用 `-f` 和 `-s` 开关启动（参见第286页的『`hadb2_setup` 命令』）。

注：不要对同一逻辑主机使用多个实例。如果有多个实例与一个逻辑主机相关联，则健康的实例可能会与不健康的实例一起进行故障恢复。

EEE 考虑事项

当决定将哪些数据库分区与一个逻辑主机相关联时，重要的是考虑它们将如何进行故障恢复。假设一个由两台机器组成的群集要配合这两台机器之间的四个数据库分区使用，如第282页的图62所示。

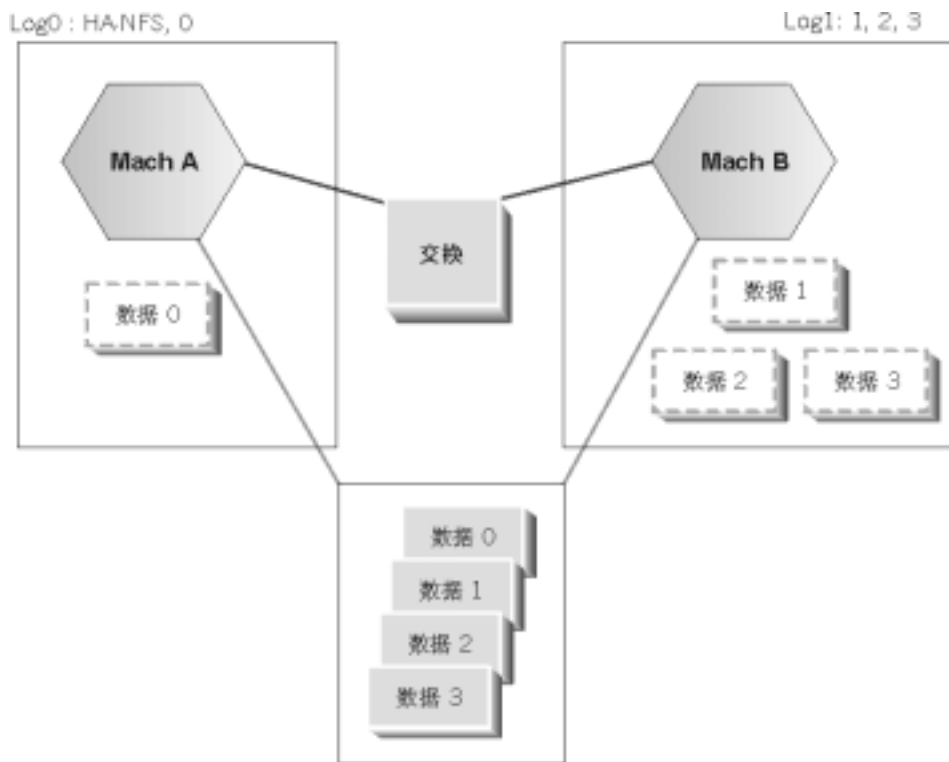


图 62. 由两台机器组成的群集与四个数据库分区

可将一个逻辑主机与每个数据库分区相关联，并将一个逻辑主机用于 HA-NFS。在这种情况下，若所有逻辑主机都驻留在一个系统上，则会有问题。若该系统失效，则必须同时将所有逻辑主机移出该系统。不幸的是，Sun Cluster 软件并非按任何可预测的次序移动逻辑主机，在移动与 HA-NFS 相关联的逻辑主机之前，可能会移动带有相关联的数据库分区的逻辑主机。通常，最好是根据单一系统上将要存放的数据库分区来将那些数据库分区分组到一起。这意味着正常情况下存放在一台机器上的两个数据库分区应与单一逻辑主机相关联。

将更新 EEE 实例使用的 db2nodes.cfg 文件，以指示数据库分区所在的机器。例如，若所有数据库分区都在一台名为 "crackle" 的机器上，则 db2nodes.cfg 文件类似于：

```
scadmin@crackle(193)# cat db2nodes.cfg
0 crackle 0 204.152.65.33
1 crackle 1 204.152.65.33
2 crackle 2 204.152.65.33
3 crackle 3 204.152.65.33
4 crackle 4 204.152.65.33
```



```
5 crackle 5 204.152.65.33
6 crackle 6 204.152.65.33
7 crackle 7 204.152.65.33
8 crackle 8 204.152.65.33
```

若这些数据库分区中的某一些被移至一台名为 "thrash" 的机器, 则 db2nodes.cfg 文件被更新为:

```
scadmin@crackle(193)# cat db2nodes.cfg
0 crackle 0 204.152.65.33
1 crackle 1 204.152.65.33
2 crackle 2 204.152.65.33
3 crackle 3 204.152.65.33
4 thrash 0 204.152.65.34
5 thrash 1 204.152.65.34
6 thrash 2 204.152.65.34
7 thrash 3 204.152.65.34
8 thrash 4 204.152.65.34
```

注意, 主机名和开关名都更改为反映机器名 "thrash", 端口编号也不同了。

HA.config 文件

若 /etc/HA.config 文件存在的话, 它可以包含许多配置选项, 包括下列各项:

```
scadmin@thrash(204)# cat /etc/HA.config
SYSLOG_FACILITY=LOG_LOCAL3
SYSLOG_LPRIORITY=LOG_INFO
SYSLOG_EPRIORITY=LOG_ERR
USE_INTERCONNECT=auto
SWITCH_NAME=204.152.65.18
DEBUG_LEVEL=2
FAILS_PER_HOUR=2
FAILS_PER_DAY=4
FAILS_PER_WEEK=10
FM_FAIL_SEV=soft
DB2START_TIMEOUT=60
DB2STOP_TIMEOUT=500
SCRIPT_USER=bin
```

注: 若 HA.config 文件不存在, 则使用缺省值。

`SYSLOG_FACILITY` 变量设置用于记录信息和错误的 `SYSLOG` 设施。`SYSLOG_LPRIORITY` 和 `SYSLOG_EPRIORITY` 变量分别设置用于记录资料式信息和错误信息的 `SYSLOG` 优先级。

可能需要进行一些更改, 以使 `SYSLOG` 精灵程序能够记录来自 `DB2 HA` 代理程序的信息。例如, 将下列两行之一添加至 `/etc/syslog.conf` 文件将告知 `SYSLOG` 精灵程序把信息写至一个日志文件。

```
*.notice  
local3.info
```

```
/var/adm/SC.x  
/var/adm/SC.LOG_LOCAL3
```

Sun Cluster 通常使用高速互连。要对 DB2 使用高速互连，请将 USE_INTERCONNECT 设置为 auto 或 override。auto 设置（缺省值）使用 Sun 内部逻辑网络接口。若初始接口失效，则此接口将被传送到另一物理接口。若将 USE_INTERCONNECT 设置为 override，则开关名从 SWITCH_NAME 变量中获取。另一个选项是将 USE_INTERCONNECT 设置为 no，它指定不使用高速互连。

DEBUG_LEVEL 指定故障恢复期间要记录多少信息。它是一个介于 0 与 10 之间的数字，其中，10 为最高调试级别。使用指定的 SYSLOG 优先级和设施记录信息。若遇到任何问题，则将调试级别设置为最高级别，将 SYSLOG 配置为记录 HA 代理程序的输出，并将 SYSLOG 输出发送至 IBM 服务。

有三个变量可以帮助 DB2 故障监控程序确定何时对逻辑主机执行故障恢复：FAILS_PER_HOUR、FAILS_PER_DAY 和 FAILS_PER_WEEK。每一个 HA 环境都是不同的；您必须确定可以接受的 DB2 故障数量。在每个“可接受的”故障之后，在同一机器上重新启动 DB2。当超过这三个故障阈值之一时，与实例或数据库分区相关联的逻辑主机执行故障恢复。

FM_FAIL_SEV 变量指定故障恢复是“软的”还是“硬的”。有关详情，参考 Sun Cluster 文档中有关 hact1(1m) 的部分。

DB2START_TIMEOUT 和 DB2STOP_TIMEOUT 变量指定允许 db2start 和 db2stop 运行的最大秒数。在指定的时间间隔过去之后，HA 代理程序认为该操作已失败，并尝试重新启动实例。

有些用户脚本不与任何特定实例相关联。正常情况下，这些脚本作为超级用户运行；这一点可以被 SCRIPT_USER 变量覆盖，可设置此变量以指定可以运行这些脚本的用户 ID。

控制方法是如何运行 DB2 命令的

DB2 HA 代理程序使用 su 命令来作为实例所有者运行命令。实际的命令将类似于：

```
su - <instance> -c "db2stop"
```

其中，instance 是实例的用户名。

重要的是确保实例拥有者的 .profile 文件支持 su。若不支持，则 su 命令不能正确工作。请人工调用 su 命令，或从脚本调用，以验证该命令是否能够成功运行。

设置

在阅读本节之前，您务必要熟悉 SC2.2 软件。本节假定您了解如何设置 SC2.2 和 HA-NFS，并且了解如何使用卷管理器。除 DB2 所必需的其他补丁程序之外，HA 代理程序还需要下列补丁程序：

```
Solaris 2.6:  
105210-17 (或更新版本)  
105786-05 (或更新版本)
```

注：Solaris 7 (Solaris 2.7) 没有必需的补丁程序。

公共安装步骤

1. 在群集中的所有机器上安装 SC2.2。安装期间，SC2.2 将询问要安装哪些代理程序。因为 DB2 不是由 SC2.2 附带交付的，所以它不在代理程序列表中。用于 DB2 的代理程序将随 DB2 一起安装，并通过 **hadb2_reg** 命令注册。
2. 使用磁盘组和逻辑 IP 地址配置逻辑主机。

DB2 UDB 企业版上的设置

1. 在逻辑主机的逻辑主机文件系统下面为实例创建主目录。
2. 在群集中的所有机器上安装 DB2。
3. 在群集中当前带有用于该实例的主目录的机器上创建该实例。
4. 向群集中的其他机器添加用于该实例的用户，确保数字用户 ID 相同。
5. 使用 **hadb2_reg** 命令注册 hadb2 服务。
6. 运行 **hadb2_setup** 命令来为该实例设置 HA。

DB2 UDB 扩充企业版上的设置

1. 为 HA 实例所有者创建主目录：
 - a. 对于热备份，在逻辑主机的逻辑主机文件系统下面为实例创建主目录。
 - b. 对于相互替换配置，配置 HA-NFS，并从其中一个逻辑主机调出主目录。在其中一台机器上，在选择安装点下面安装 HA-NFS 目录。
2. 在群集中的所有机器上安装 DB2。
3. 在安装有 HA-NFS 文件系统的机器上创建实例。
4. 向群集中的其他机器添加用于该实例的用户，确保数字用户 ID 相同。
5. 使用 **hadb2_reg** 命令注册 hadb2 服务。
6. 运行 **hadb2_setup** 命令来为该实例设置 HA。

注：因为 NIS 会引入单一故障点，所以不建议使用 NIS 来定义 HA 实例的信息。

hadb2_setup 命令

hadb2_setup 命令是 DB2 HA 代理程序附带的程序的中心。可以使用它来设置、修改或删除实例。可以使用它来打开和关闭 hadb2_setup 服务。有了此命令，您无需人工编辑 hadb2tab 文件。

注：hadb2_setup 命令只对运行它的机器执行操作。对一台机器所作的更改也应应用于群集中的其他机器。

支持下列自变量：

添加 EE 实例：

```
-----  
hadb2_setup -a -i <instance> -f [on|off] -h <logical_host> -p [DATA|ADMIN] -t EE
```

例如：

```
hadb2_setup -a -i db2ee -f off -h log1 -p DATA -t EE
```

添加 EEE 实例：

```
-----  
hadb2_setup -a -i <instance> -f [on|off] -h <nfs_host> -l <mount_point> \  
-r <ha-nfs_dir> -p [DATA|ADMIN] -t EEE -n "<node_info>"
```

例如：

```
hadb2_setup -a -i db2eee -f off -h ha-sun1 -l /export/ha_home \  
-r /log0/home -p DATA -t EEE -n "log0[0,10,20],log1[30,40,50]"
```

删除实例：

```
-----  
hadb2_setup -d -i <instance>
```

修改实例：

```
-----  
hadb2_setup -m -i <instance> [-f [on|off] | -l <mount_point> | \  
-h <host> | -p [DATA|ADMIN] | -r <ha-nfs_dir> | -t [EE|EEE] ]
```

其他选项：

```
-----  
-s <on|off>          (对所有 HA 实例) 启动或关闭 hadb2  
-y                  假设 yes, 以进行安全性检查
```

要打开或关闭 hadb2 服务，指定 -s 开关。这与使用带有 -n 和 -y 开关，并指定 hadb2 服务的 **hareg** 命令等效。有关 **hareg(1m)** 命令的详情，参考 Sun Cluster 文档。

可使用 `-f` 开关来关闭实例的故障监控程序。其作用为在本地机器上停止该实例的故障监控程序，以及修改 `hadb2tab` 文件，以反映故障监控已关闭这一事实。

对于 EE 实例，建议在实例进行故障恢复时在所有机器上关闭故障监控。对于 EEE 实例，在人工关闭该实例之前，必须在所有存放有该实例的数据库分区的机器上关闭故障监控。

要删除实例，请使用 `-d` 开关。这只会从 `hadb2tab` 文件中除去该实例，而不除去或修改任何其他文件或目录。因为 `hadb2tab` 文件是 HA-DB2 代理程序的主配置文件，所以从此文件中除去一个实例将使控制方法不知道它的存在。

要修改实例，请使用 `-m` 开关。这只会更改 `hadb2tab` 文件中的信息，而不除去或修改任何其他文件或目录。`-m` 开关可以与任何处理 `hadb2tab` 文件中的信息的开关一起使用。因为 `hadb2_setup` 命令不支持修改 `db2nodes.cfg` 文件和 `hadb2-eee.cfg` 文件，所以在初始设置之后，必须人工更改这些文件。

添加实例稍微有些棘手。

对于 EE 实例，需要下列自变量：

```
hadb2_setup -a -i <instance> -f <fm> -h <logical_host> -t <EEE_or_EE> -p <purpose>
```

其中，*instance* 是要添加的实例的名称，*fm* 指定最初是打开还是关闭故障监控，*logical_host* 是相关联的逻辑主机，*EEE_or_EE* 设置为 EE，而 *purpose* 可以是 DATA 或 ADMIN。

对于 EEE 实例，需要下列自变量：

```
hadb2_setup -a -i <instance> -f <fm> -h <nfs_host> -t <EEE_or_EE> -p  
<purpose> -l <mount_point> -r <HA-NFS_directory> -n <node_info>
```

其中，*instance* 是要添加的实例的名称，*fm* 指定最初是打开还是关闭故障监控，*nfs_host* 是调出 HA-NFS 文件系统的逻辑主机的主机名，*EEE_or_EE* 设置为 EEE，*purpose* 可以是 DATA 或 ADMIN，*mount_point* 是 HA-NFS 目录的本地安装点，*HA-NFS_directory* 是 HA-NFS 目录，而 *node_info* 是将数据库分区与一个逻辑主机相关联的信息。例如：

```
hadb2_setup -a -i db2eee -f on -h jolt -l /export/ha_home -p DATA -t EEE -r  
/log1/home -n "log0[0,1],log1[2,3]"
```

添加 EEE 实例时，必须将节点信息括在引号中。在此示例中，实例 "db2eee" 将与两个逻辑主机 "log0" 和 "log1" 相关联。"db2eee" 实例的数据库分区 "0" 和 "1" 将与逻辑主机 "log0" 相关联，数据库分区 "2" 和 "3" 将与逻辑主机 "log1" 相关联。

使用 `hadb2_setup` 命令来向群集中的所有机器添加实例。然后，可以通过强制群集重新配置，或通过先关闭再打开 `hadb2` 服务来启动实例。这可以通过 `hareg` 命令完成，也可以通过 `hadb2_setup` 命令的 `-s` 开关完成。若该实例未启动，则参见第291页的『故障排除』。

当 `hadb2_setup` 命令添加 `EEE` 实例时，以透明方式执行下列操作：

- 检查指定的信息。这包括确保系统上存在该用户，且 `HA-NFS` 正在运行。
- 创建一个 `db2nodes.cfg` 文件。
- 创建一个 `hadb2-eee.cfg` 文件。
- 为 `EEE` 实例创建 `.rhosts` 文件。
- 创建从缺省数据库路径到相关联的逻辑主机数据目录的符号连接。
- 向 `hadb2tab` 文件添加一行。

为了预防配置错误，并确保在运行 `hadb2_setup` 命令之后能够启动 `HA` 实例，该命令在添加新实例之前要执行相当大量的测试。

创建 `db2nodes.cfg` 文件，并将关于当前群集状态的信息存入其中。例如，若逻辑主机 `"log0"` 正驻留在机器 `"crackle"` 上，则与 `"log0"` 相关联的数据库分区的项将包含机器名 `"crackle"` 和用于 `"crackle"` 的高速互连：

```
scadmin@crackle(193)# cat db2nodes.cfg
0 crackle 0 204.152.65.33
1 crackle 1 204.152.65.33
2 thrash 0 204.152.65.34
3 thrash 1 204.152.65.34
```

仅根据命令上指定的节点信息创建 `hadb2-eee.cfg` 文件。对于每个数据库分区都有一行：

```
sphere % cat hadb2-eee.cfg
NODE:log0 0
NODE:log0 1
NODE:log1 2
NODE:log1 3
```

`DB2 UDB EEE` 需要 `.rhost` 文件，该文件应包含群集中每台机器的所有主机名（或 IP 地址）。例如：

```
crackle db2eee
204.152.65.1 db2eee
204.152.65.17 db2eee
thrash db2eee
204.152.65.2 db2eee
204.152.65.18 db2eee
crackle db2eee
```

```
jolt db2eee
bump db2eee
thrash.torolab.ibm.com db2eee
crackle.torolab.ibm.com db2eee
```

根据 SMS 表空间的文件系统布局，**hadb2_setup** 命令设置许多目录和符号连接。它们包括：

- 每个逻辑主机的逻辑主机文件系统下面的一个名为 "data" 的目录。
- 与该逻辑主机相关联的每个数据库分区的节点目录（在这个 "data" 目录下面）。
- 缺省数据库路径中的符号连接，它们位于 `~<instance>` 下面，其中，`~instance` 是实例的主目录。对于每个数据库分区，都有一个符号连接指向相对应的节点目录。有关详情，参见第272页的『EE 和 EEE 实例的磁盘布局』。

故障恢复时间

故障恢复时间从数据最初不可用时算起，直到它再次可用为止。故障恢复期间发生的许多事件会对故障恢复时间产生显著影响：

- 磁盘调出和调入。

虽然调出和调入磁盘对整体停机时间确实有所影响，但与其他事件相比，它们通常不会花费很长时间。故障恢复期间需要从一台机器移至另一机器的磁盘越多，此过程所花费的时间也越长。若有磁盘损坏，则此过程还要长一些。
- 对为逻辑主机安装的文件系统执行 **fsck**。

在可以安装逻辑主机的文件系统之前，这些文件系统必须通过 **fsck**，以确保它们健康。文件系统越大，此过程就越长。通过使用日志文件系统，可以大幅缩短此时间。因为日志文件系统通常是在 HA 环境中使用的，所以 **fsck** 时间通常不是一个问题。
- 从 HA 代理程序调用的用户脚本。

HA 代理程序将调用用户脚本（如果它们存在且可执行的话）。这些脚本中的一些是同步运行的，它们可能会增加启动 HA 实例所需的时间。确保它们能够尽可能快地运行；请考虑在后台运行由这些脚本调用的任何外部程序。
- HA-NFS。

对于相互替换配置中的单一 EEE 实例，必须将 HA-NFS 用于实例拥有者的主目录。因为 *lockd* 有宽限期（在 HA-NFS 的 HA 代理程序中定义，当运行 HA-NFS 时，为 90 秒），所以 HA-NFS 会增加故障恢复时间。在故障恢复之后，因为任何在 HA-NFS 文件系统上锁定文件的进程都必须等待宽限期结束，所以这会影响到故障恢复时间。在故障恢复之后，DB2 的 HA 代理程序是第一个在实例拥有者的主目录下锁定文件的进程，它记录获取第一个锁定所花费的时间。在故障恢复之后，此时间显示在状态报告中。

- 启动 DB2。

启动 DB2 对故障恢复时间的影响不大。对于 EE 实例，平均大约增加 5-15 秒。对于 EEE 实例，大约增加 10 秒，每个正在执行故障恢复的数据库分区另外还要增加 5 秒。举个例子，如果有三个数据库分区正在进行故障恢复，则启动这三个数据库分区将会使故障恢复时间增加大约 25 秒。这不包括该实例的数据库的应急恢复。

- 数据库应急恢复。

应急恢复通常导致与故障恢复相关联的停机时间中的绝大部分。恢复数据库所需时间的长短取决于许多因素，包括：

- 客户机工作负荷。事务日志中只记录对数据库的更改。若客户机工作负荷主要是只读操作，则应急恢复期间，必须对数据库应用的事务就相对少一些。
- 磁盘和机器速度。磁盘以及存放 HA 实例的机器的速度对恢复数据库所需时间的长短也有影响。系统越快，应急恢复时间也就越短。
- *softmax* 数据库配置参数的值。*softmax* 的值是日志文件大小的百分比，达到此百分比时，设置一个软检查点，并写日志控制文件。应急恢复期间，使用日志控制文件来确定哪些日志记录是将数据库复原为一致状态真正必需的。减小此值将导致数据库管理程序更频繁地触发页清除程序，并设置更多的软检查点；虽然性能会降低，但数据库恢复却更快。
- 该实例是 EE 还是 EEE。若该实例是 EEE 实例，则数据库重新启动操作将并行完成。每个数据库分区都负责重新启动它自己的数据库部分。若数据库有 50 GB 的数据，则带有四个数据库分区的实例将能够以大约四倍于 EE 实例的速度恢复数据库。

故障排除

下表标识了您可能会遇到的问题、其可能的原因以及为了解决它们而需执行的操作。

表 29. 在 Sun Cluster 2.2 上排除高可用性的故障

症状	可能的原因	操作
不能安装逻辑主机文件系统	逻辑主机的故障恢复期间，已正常安装和卸下逻辑主机文件系统。故障恢复期间，逻辑主机文件系统下面不应有活动的进程或打开的文件。在非常罕见的情况下，不能被删掉的进程在逻辑主机文件系统下面有它们的当前工作目录。要了解安装点下面是否有进程，使用 <code>fuser(1m)</code> ，或使用名为 <code>lsdf</code> 的 GNU 实用程序。当不能安装逻辑主机文件系统时，会生成错误信息。 ^a	重新引导系统，或将该逻辑主机文件系统转移为另一名称，并重新创建它。这样做允许冻结的进程停留在该目录下面（因为不能删掉它），并使安装能够发生。 ^b
<code>db2start</code> 或 <code>db2stop</code> 超时不起作用	<code>SIGALRM</code> 信号不能中断阻塞的系统调用。相反，该系统调用将重新启动，就象使用 <code>sigaction()</code> 设置了 <code>SA_RESTART</code> 标志一样。这导致 <code>DB2 HA</code> 代理程序的超时被忽略，代理程序方法将挂起，而不是从挂起的 <code>db2start</code> 或 <code>db2stop</code> 命令恢复。	对 Solaris 2.6 应用必需的补丁程序，即 105210-17（或更新版本）。
对实例的注册挂起	虽然发生此问题的原因相当多，但最常见的原因包括 NFS 问题和 <code>/usr/sbin/quota</code> 程序。	检查 NFS 安装，确保它们是健康的，并寻找由实例所有者拥有的限额进程。根据系统管理员的判断，将限额程序更改为指向 <code>/bin/true</code> 的符号连接可以解决此问题。这不是建议的解决方法，但可能管用。
我刚刚设置了 <code>EEE</code> 实例，但它不启动	<code>hadb2_setup</code> 命令未向 <code>/etc/services</code> 文件添加端口；期望管理员人工添加它们。返回了错误信息。 ^c	确保 <code>/etc/services</code> 文件中命名了适当的端口。

表 29. 在 Sun Cluster 2.2 上排除高可用性的故障 (续)

症状	可能的原因	操作
START_NET 方法不能启动 DB2		<p>关闭故障监控，以确保实例不执行故障恢复。作为实例所有者注册，并尝试人工启动 DB2。</p> <ol style="list-style-type: none"> 1. 确保 <code>hadb2tab</code> 配置文件指定了正确的实例类型。例如，将 <code>db2nodes.cfg</code> 文件用于 EE 管理实例将导致问题，HA 代理程序方法将无法从此问题恢复。 2. 确保 <code>.rhosts</code> 文件存在，且其中带有有效的项。 3. 确保群集中的所有机器使用超级用户许可权来共享 HA-NFS 文件系统。 4. 检查核心参数，并确保它们正确。 5. 确保 <code>/etc/services</code> 文件包含用于该实例的项。
实例只在一台机器上工作	<ul style="list-style-type: none"> • 实例的数字 <code>uid</code> 可能没有在群集中的每台机器上都相同。 • 核心参数可能没有在群集中的每台机器上都有效。 • <code>hadb2tab</code> 文件可能没有在群集中的每台机器上都相同。 • 其他配置文件（如逻辑主机 <code>vfstab</code> 文件）可能没有在群集中的每台机器上都相同。 	<p>若这些原因似乎都不适用，则尝试作为实例所有者注册，并人工启动 DB2。对于 EE 实例，若存放实例的逻辑主机驻留在当前机器上，则此方法应管用。对于 EEE 实例，此方法可以对群集中任何可以存放数据库分区的机器起作用。</p>
<code>su</code> <instance> -c "db2start" 不 工作	<ul style="list-style-type: none"> • 该实例的 <code>.profile</code> 可能不支持 <code>su</code>。 • Bourne shell (<code>/bin/sh</code>) 有一个已知的问题，即 <code>su</code> 命令能以人工方式工作，但不能通过 HA 代理程序工作。 	<ul style="list-style-type: none"> • 尝试作为超级用户人工运行此命令，在通过 HA 代理程序再试之前，确保它能工作。 • 如有必要，切换至 Korn shell (<code>/bin/ksh</code>)。
我的 EEE 实例不能启动， 但主目录已安装	<p>可能未使用“超级用户”许可权来将 HA-NFS 目录调出至群集中的机器。DB2 和 HA 代理程序都需要此目录才能正确运行。</p>	<p>要进行测试，尝试（作为超级用户）在实例拥有者的主目录下面创建一个文件。</p>

表 29. 在 Sun Cluster 2.2 上排除高可用性的故障 (续)

症状	可能的原因	操作
尝试存取 EEE 实例目录时返回“NFS 文件句柄已过时”错误	实例拥有者的主目录下面可能仍有进程。	卸下实例拥有者的主目录，并允许 HA 代理程序重新安装它。若先关闭并再次打开 hadb2 服务（参见第286页的『hadb2_setup 命令』中有关 hadb2_setup 命令上的 -s 开关的说明），则 HA 代理程序将重新安装该目录。
未能通过 SC2.2 成功运行控制方法	可能未向 Sun Cluster 软件注册 hadb2 服务，或者可能是未打开它。	<p>若该控制方法似乎能从命令行正常运行，则检查 SYSLOG 文件中的错误信息可能会有助于您解释该问题。确保向 Sun Cluster 软件注册了 hadb2 服务，且它已打开。</p> <p>人工运行方法对于调试问题而言很有用。^d</p> <p>应作为超级用户来运行方法，并给出适当的命令行自变量。若逻辑主机列表为空，则给出自变量 ""。不带空格分隔符的双引号指示空白自变量。例如：</p> <pre>hadb2_startnet log0,log1 "" 600</pre> <p>第一个自变量 log0,log1 告知 hadb2_startnet 方法：逻辑主机 log0 和 log1 驻留在当前机器上。第二个自变量为空，它告知 hadb2_startnet 方法：没有将其他逻辑主机存放在群集中的其他机器上（它们全都在当前机器上）。第三个自变量告知该方法：SC2.2 将在 600 秒之后超时。</p>
用户脚本不运行	仅当用户脚本存在于适当的目录中且可执行时，它们才能运行。	检查文件所有权和属性。若脚本仍未能运行，则与 IBM 服务联系。寄上未运行的脚本的目录列表，以及应该已运行了该脚本的故障恢复或群集重新配置的 SYSLOG 输出。

表 29. 在 Sun Cluster 2.2 上排除高可用性的故障 (续)

症状	可能的原因	操作
未将信息记录至 /etc/syslog.conf 中指定的文件		使用 <code>touch(1)</code> 来创建 /etc/syslog.conf 文件中指定的文件, 然后重新启动 SYSLOG 精灵程序。
<p>^a在不能安装逻辑主机文件系统时, 生成的错误信息可能类似于:</p> <pre>Aug 17 11:14:01 rash ID[SUNWcluster.loghost.1170]: importing data1 Aug 17 11:14:06 rash ID[SUNWcluster.scnfs.3040]: mount -F ufs -o "" /dev/vx/dsk/data1/data1-stat /log1 failed. Aug 17 11:14:07 rash ID[SUNWcluster.ccd.cccd.5304]: error freeze cmd = /opt/SUNWcluster/bin/loghost_sync CCDSYNC_POST_ADDU LOGHOST_CM:log1:rash /etc/opt/SUNWcluster/conf/ccd.database 2 "0 1" 1 error code = 1</pre> <p>^b 例如:</p> <pre>scadmin@rash(218)# ps -fe egrep db2 db2ee 1984 1 0 0:01 <defunct></pre> <p>解决方法:</p> <pre>scadmin@rash(229)# cd / scadmin@rash(230)# mv /log1 /log1.bkp scadmin@rash(231)# mkdir /log1</pre> <p>^c 错误信息可能类似于:</p> <pre>SQL6030N START or STOP DATABASE MANAGER failed. Reason code "13".</pre> <p>^d 例如, 若 <code>hadb2_startnet</code> 方法找不到 <code>libdb2.so.1</code>, 但它能通过 Sun Cluster 软件正常运行, 则不会报告错误。人工运行该方法将生成下列信息:</p> <pre>scadmin@crackle(213)# hadb2_startnet '''log0,log1' 600 ld.so.1: hadb2_startnet: fatal: libdb2.so.1: open failed: No such file or directory Killed</pre>		

第5部分 附录及附属资料

附录A. 使用 DB2 资料库

DB2 通用数据库由联机帮助、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。

要存取联机产品信息，可以使用“信息中心”。有关详情，参见第310页的『用“信息中心”存取“信息”』。可以查看任务信息、DB2 书籍、疑难解答信息、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；参见第306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 表示本国语言环境）目录下安装多种语言版本的 HTML 文件。有关详情，参考适当的快速入门书籍。

您可以各种方法来获取 DB2 书籍并存取信息:

- 第309页的『查看联机信息』
- 第313页的『搜索联机信息』
- 第306页的『订购打印书籍』
- 第305页的『打印 PDF 书籍』

表 30. DB2 信息

名称	说明	书号	HTML 目录
		PDF 文件名	
DB2 指南和参考信息			
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 db2d1x70	db2d0
	管理指南: 实现提供有关实现问题 (如实现设计、存取数据库、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 应用程序设计接口 (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 db2b0x70	db2b0
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 应用程序的逐步指导。	SB84-0220 db2axx70	db2ax
<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 db2apx70	db2ap
	仅有 HTML 格式的版本。		

表 30. DB2 信息 (续)

名称	说明	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) 来存取 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) 来存取 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 (如调入、调出、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 “信息目录管理程序” 集成的信息。	SC26-9994 db2adx70	db2ad
<i>DB2 Connect 用户指南</i>	提供 DB2 Connect 产品的概念、程序设计以及一般用法信息。	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 30. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
<i>DB2 Query Patroller</i> 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 及其部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	
<i>Image, Audio, and Video Extenders</i> 管理和程序设计	提供有关 DB2 Extender 的一般信息, 有关 Image, Audio and Video (IAV) Extender 的管理和配置的信息, 以及有关使用 IAV Extender 进行程序设计的信息。它包括参考信息、诊断资料 (带有信息) 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“信息目录管理程序”的体系结构接口的定义。	SC26-9997	db2bi
		db2bix70	
信息目录管理程序用户指南	提供有关使用“信息目录管理程序”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、配置分布式请求和存取多机种数据源的信息。	GB84-0127	db2iy
		db2iyx70	
信息参考	列出由 DB2、信息目录管理程序和数据仓库中心发出的信息和代码, 并描述应执行的操作。 在北美, 您可订购两卷英文版的信息参考 (使用书号 SBOF-8932)。	第 1 卷 GB84-0216	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“管理程序”部件。	SC27-0787	n/a
		db2dpx70	

表 30. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
<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”）创建和填充 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
<i>OLAP Setup and User's Guide</i>	提供 OLAP Starter Kit 的配置和设置信息。	SC27-0702 db2ipx70	db2ip
<i>OLAP Spreadsheet Add-in User's Guide for Excel</i>	描述如何使用 Excel 电子表格程序来分析 OLAP 数据。	SA40-1756 db2epx70	db2ep
<i>OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3</i>	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
<i>Spatial Extender 用户指南和参考</i>	提供关于 Spatial Extender 的安装、配置、管理、程序设计和疑难解答的信息。还提供空间数据概念的重要说明，并提供 Spatial Extender 特定的参考资料（信息和 SQL）。	SB84-0249 db2sbx70	db2sb
<i>SQL 入门</i>	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
<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 30. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
<i>Text Extender 管理和程序设计</i>	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 进行程序设计的信息。它包括参考信息、诊断资料(带有信息)和样本。	SB84-0248 desu9x70	desu9
<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 db2p0x70	db2p0
新增内容	描述 DB2 通用数据库(版本 7)中的新特性、函数和增强功能。	SB84-0224 db2q0x70	db2q0
DB2 安装和配置信息			
<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 db2c6x70	db2c6
<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 db2cyx70	db2cy
<i>DB2 Connect 个人版快速入门</i>	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 db2c1x70	db2c1
<i>DB2 Connect Personal Edition Quick Beginnings Linux 版</i>	在进行所有受支持的 Linux 分布式系统时, 提供“DB2 Connect 个人版”的计划、安装、迁移和配置信息。	GC09-2962 db2c4x70	db2c4
<i>DB2 DataLinks Manager 快速入门</i>	提供“DB2 DataLinks 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 db2z6x70	db2z6
<i>DB2 扩充企业版 UNIX 版快速入门</i>	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 db2v3x70	db2v3

表 30. DB2 信息 (续)

名称	说明	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 (OS/2 版) 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
<i>DB2 (UNIX 版) 快速入门</i>	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
<i>DB2 Windows 版快速入门</i>	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
<i>DB2 个人版快速入门</i>	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 Linux 版</i>	在进行所有受支持的 Linux 分布式系统时，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
<i>DB2 Query Patroller 安装指南</i>	提供有关 DB2 Query Patroller 的安装信息。	GB84-0208 db2iwx70	db2iw
<i>DB2 数据仓库管理程序安装指南</i>	提供仓库代理程序、仓库变换器和“信息目录管理程序”的安装信息。	GB84-0122 db2idx70	db2id
HTML 格式的跨平台样本程序			

表 30. DB2 信息 (续)

名称	说明	书号	HTML 目录
		PDF 文件名	
HTML 格式的样本程序	为所有受 DB2 支持的平台上的程序设计语言提供 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程序设计语言的版本。HTML 样本仅当安装了“DB2 应用程序开发客户机”时才可用。 有关这些程序的详情，参考应用程序构建指南。	无书号	db2hs
发行说明			
<i>DB2 Connect</i> 发行说明	提供 DB2 书籍中未能包括的最新信息。	参见注释 2。	db2cr
<i>DB2</i> 安装注释	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，参见 `Release.Notes` 文件。此文件位于 `DB2DIR/Readme/%L` 目录中，其中 `%L` 表示本国语言环境名，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，参见 `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 读入程序，可打印整本书籍或特定范围内的页。有关库中每本书的文件名，参见第298页的表30。

可从 Adobe Web 站点（网址 <http://www.adobe.com>）获取 Adobe Acrobat 读入程序的最新版本。

这些 PDF 文件包括在 DB2 出版物 CD-ROM 上，文件扩展名为 PDF。要存取这些 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。参考快速入门一书以了解安装过程。
2. 启动 Acrobat 读入程序。
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` 表示期望的本国语言环境的名称。

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

可通过使用销售单 (SBOF) 书号单本地或成套地订购打印的 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 (网址为 <http://www.elink.ibm.link.ibm.com/pbl/pbl>) 订购这些书籍。

有两套书籍。SBOF-8935 提供了“DB2 仓库管理程序”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 31. 订购打印书籍

SBOF 号	包括的书籍
SBOF-8931	<ul style="list-style-type: none"> • Administration Guide: Planning • Administration Guide: Implementation • Administration Guide: Performance • Administrative API Reference • Application Building Guide • Application Development Guide • CLI Guide and Reference • Command Reference • Data Movement Utilities Guide and Reference • Data Warehouse Center Administration Guide • Data Warehouse Center Application Integration Guide • DB2 Connect User's Guide • Installation and Configuration Supplement • Image, Audio, and Video Extenders Administration and Programming • Message Reference, Volumes 1 and 2 • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP Setup and User's Guide • OLAP Spreadsheet Add-in User's Guide for Excel • OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3 • Replication Guide and Reference • Spatial Extender Administration and Programming Guide • SQL Getting Started • SQL Reference, Volumes 1 and 2 • System Monitor Guide and Reference • Text Extender Administration and Programming • Troubleshooting Guide • What's New
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager User's Guide • Information Catalog Manager Programming Guide and Reference • Query Patroller Administration Guide • Query Patroller User's Guide

DB2 联机文档

存取联机帮助

随所有 DB2 部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	如何存取...
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前提条件信息，并描述如何使用窗口或笔记本控件。	从窗口或笔记本，单击 帮助 按钮或按 F1 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		
信息目录管理程序帮助		
卫星管理中心帮助		
脚本中心帮助		
信息帮助	描述信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 <i>XXXnnnnn</i> 表示有效的信息标识符。</p> <p>例如，<code>? SQL30081</code> 显示关于 SQL30081 信息的帮助。</p> <p>要每次查看一屏信息帮助，可输入： <code>? XXXnnnnn 尚有</code></p> <p>要在文件中保存信息帮助，可输入： <code>? XXXnnnnn > filename.ext</code></p> <p>其中 <i>filename.ext</i> 表示想要保存信息帮助的文件。</p>

帮助类型	内容	如何存取...
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：</p> <pre>help statement</pre> <p>其中，<i>statement</i> 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。</p> <p>注：在基于 UNIX 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：</p> <pre>? sqlstate 或 ? class code</pre> <p>其中，<i>sqlstate</i> 表示有效的五位 SQL 状态，而 <i>class code</i> 表示该 SQL 状态的头两位。</p> <p>例如，<code>? 08003</code> 显示 08003 SQL 状态的帮助，而 <code>? 08</code> 显示 08 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 表示本国语言环境名称

- 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导，执行：

1. 插入 Netscape CD-ROM。
2. 安装 CD-ROM（仅限于在基于 UNIX 的平台上）。参考快速入门一书以了解安装过程。
3. 有关安装指导，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。

用“信息中心”存取“信息”

“信息中心”提供对 DB2 产品信息的快速存取。在所有装有 DB2 管理工具的平台，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存取“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。

任务	可使用 DB2 执行的关键任务。
参考	DB2 参考信息，如关键字、命令以及 API。
书籍	DB2 书籍。
疑难解答	错误信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。要存取此信息，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...	如何存取...
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、创建并调度应急计划。	从“控制中心”，用鼠标右键单击想要备份的数据库并选择 备份 → 数据库 （使用向导）。
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”，用鼠标右键单击 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”，用鼠标右键单击 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关键字。	从“控制中心”，用鼠标右键单击图符，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”，用鼠标右键单击 表空间 图符，并选择 创建 → 表空间 （使用向导）。

向导	帮助您...	如何存取...
创建索引	建议对于所有查询要创建和卸下哪些索引。	从“控制中心”，用鼠标右键单击索引图符，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。	从“控制中心”，用鼠标右键单击想要调整的数据库并选择 使用向导配置性能 。 对于分区数据库环境，从“数据库分区”视图，用鼠标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 。
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”，用鼠标右键单击想要复原的数据库并选择 复原 → 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。这表示需要存取 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。参见第310页的『用“信息中心”存取“信息”』以获取详情。

附录B. 命名规则

当您为下列数据库和数据库对象提供名称时，使用下面显示的命名规则：

- 数据库名称
- 数据库名和数据库别名
- 用户 ID 和口令
- 模式名
- 组名和用户名
- 对象名。

不要使用 IBM SQL 或 ISO/ANSI SQL92 保留字来命名表、视图、列、索引或授权 ID。 *SQL Reference* 中包括这些保留字的一个列表。

有关授权 ID（包括用户名和组名）和工作站的命名规则，以及其他平台限制，参考快速入门手册。

数据库名称

每当创建一个新数据库时，数据库管理程序就创建一个单独的目录来存储该数据库的控制文件和数据文件。

这些目录的命名方案是 SQL00001 至 SQLnnnnn，其中，SQL00001 包含与创建的第一个数据库相关的控制文件，SQL00002 包含与创建的第二个数据库相关的控制文件，依此类推。

这些目录是自动维护的。要避免潜在的目录命名问题，不要使用数据库管理程序所用的相同命名模式来创建您自己的目录，也不要操纵已经由数据库管理程序创建的目录。

数据库名和数据库别名

数据库名是您或您的用户提供来作为 CREATE DATABASE 命令或 API 的一部分的标识名称。这些名称在它们被编目的位置中必须是唯一的。例如，在基于 UNIX 的 DB2 实现上，此位置是一个目录路径，而在 OS/2 实现上，它是驱动器盘符。

数据库别名是赋予本地或远程数据库的本地同义词。这些名称在“系统数据库目录”中必须是唯一的，在此目录中存储了数据库管理程序的单个实例的所有别

名。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。

当命名一个数据库或数据库别名时，您指定的名称：

- 可包含 1 至 8 个字符
- 必须以下列其中一项开始：
 - A 至 Z（将小写字母转换为大写）
 - @、# 或 \$
- 可包括：
 - A 至 Z（将小写字母转换为大写）
 - 0 至 9
 - @、#、\$ 和 _（下划线字符）

注：为避免潜在的问题，若您打算在通信环境中使用数据库，则不要在该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，因此若您计划在另一个国家使用该数据库，则不要使用这些特殊字符。最后，在 Windows NT 系统上，确保没有实例名与服务名相同。

用户 ID 和口令

当创建用户 ID 或口令时，创建的名称：

- 不能是下列其中任何一项：
 - *SQL Reference* 中列示的 USERS、ADMINS、GUESTS、PUBLIC、LOCAL 或任何 SQL 保留字。
- 不能以下列其中一项开始：
 - SQL、SYS 或 IBM
- 可包括：
 - A 至 Z

注：有些操作系统允许区别大小写的用户 ID 和口令。您应检查操作系统文档，看看是否是这种情况。

- 0 至 9
- @、# 或 \$
- 用户 ID 不能超过 30 个字符。

注：您可能需要执行口令维护任务。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口令的方法。例如，DB2 OS/390 版的版本 5 支持这种更改用户口令的方法。若收到错误信息 SQL1404N “口令到期”，则使用 CONNECT 语句来更改口令，如下所示：

```
CONNECT TO <database> USER <userid> USING <password>
NEW <new_password> VERIFY <new_password>
```

也可使用“DB2 客户机配置辅助程序”(CCA)的“口令更改”对话来更改口令。有关这些口令更改方法的详情，参考 *SQL Reference* 和 CCA 联机帮助。

模式名

下列模式名是保留字，因此不能使用：

- SYSCAT
- SYSFUN
- SYSIBM
- SYSSTAT.

通常，您应当避免以 SYS 开头的模式名，以免将来发生迁移问题。数据库管理程序将不允许您使用以 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。

还建议您不要将 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续表同名的临时表，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。

组名和用户名

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。

在 OS/2 上，组和用户不能有相同的名称。

在 Windows NT 上，“本地组名”、“全局组名”和“用户 ID”不能同名。

组名不能超过 8 个字符。

对象名

数据库对象包括下列各项:

- 模式
- 表
- 视图
- 列
- 索引
- 用户定义函数 (UDF)
- 用户定义类型 (UDT)
- 触发器
- 别名
- 表空间
- 存储过程
- 方法
- 节点组
- 缓冲池
- 事件监控程序

当命名数据库对象时, 指定的名称:

- 可以包含 1 至 18 个字符 (字节)

注: 有以下例外:

- 模式和列允许 1 至 30 个字符
- 表、视图、相关名和别名允许 1 至 128 个字符。
- 必须以下列其中一项开始:
 - A 至 Z (将小写字母转换为大写)
 - 有效的重音字母 (如 ö)
 - 除多字节空格外的多字节字符 (用于多字节环境)
- 可包括:
 - A 至 Z (将小写字母转换为大写)
 - 有效的重音字母 (如 ö)
 - 0 至 9
 - @、#、\$ 和 _ (下划线字符)

- 除多字节空格外的多字节字符（用于多字节环境）

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。有关定界标识符的详情，参考 *SQL Reference*。

为获得最大可移植性，使用 IBM SQL 和 ISO/ANSI SQL92 保留字。有关这些保留字的列表，参考 *SQL Reference*。

注：

1. 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；然而，后续的使用有可能导致错误情况。为避免使用和操作数据库时发生潜在的问题，**不要**违反上述规则。

例如，若创建在名称中带有 + 号或 - 号的一列，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。

联合体数据库对象名

联合体数据库对象包括：

- 索引规范
- 诨名
- 服务器
- 封装器
- 函数映射
- 类型映射
- 用户映射。

命名联合体数据库对象时存在限制。有关对象名的完整列表以及相关标识符限制和要求，参阅 *SQL Reference*。概括地说，对象名：

- 有限制。诨名、映射、索引规范、服务器和封装器名不能超过 128 个字符。
- 必须以下列其中一项开始：
 - A 至 Z（没有引号的名称转换为大写）
 - 有效的重音字母（如 ö）
 - 除多字节空格外的多字节字符（用于多字节环境）
- 必须遵守内部命名约定。非前导字符可以包括：
 - A 至 Z
 - 有效的重音字母（如 ö）
 - 0 至 9

- @、#、\$ 和 _（下划线字符）
- 除多字节空格外的多字节字符（用于多字节环境）

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。有关定界标识符的详情，参考 *SQL Reference*。

为获得最大可移植性，使用 IBM SQL 和 ISO/ANSI SQL92 保留字。有关这些保留字的列表，参考 *SQL Reference*。

选项（服务器、译名）和选项设置限制为 255 字节。

如何在联合体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别大小写的标识符和口令。为了确保将它们传送到数据源时大小写正确，请遵循这些准则：

- 用要求的字体指定它们并用双引号引起来。
- 若指定的是用户 ID，应将数据源的 fold_id 服务器选项设置为 "n"（否，不更改大小写）。若指定的是口令，应将数据源的 fold_pw 服务器选项设置为 "n"。

用户 ID 和口令有替代项。若数据源要求用户 ID 为小写，可以任何字体指定它，并将 fold_id 服务器选项设置为 "l"（以小写字体将此 ID 发送至数据源）。若数据源要求此 ID 为大写，可以任何字体指定它，并将 fold_id 服务器选项设置为 "u"（以大写字体将此 ID 发送至数据源）。同样，若数据源要求口令为小写或大写，可将 fold_pw 服务器选项设置为 "l" 或 "u" 来满足此要求。

有关服务器选项的详情，参见 *管理指南：实现中的“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别大小写的标识符或口令，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 系列数据源中的许多定界标识符要区分大小写。假设您希望为驻留在数据源 NORBASE 中的 DB2 for CS 视图 "my_schema"."wkly_sal" 创建译名 NICK1。

在基于 UNIX 的系统的命令提示处，应输入：

```
db2 'create nickname nick1 for norbase."my_schema"."wkly_sal"'
```

在 Windows NT 命令提示处，应输入：

```
db2 create nickname nick1 for norbase.\\"my_schema\\".\\"wkly_sal\""
```

若从 DB2 交互式命令提示符输入语句，或者若在应用程序中指定它，则不需要单引号或斜杠。例如，在基于 UNIX 的系统或 Windows NT 上的 DB2 命令提示处，应输入：

```
create nickname nick1 for norbase."my_schema"."wkly_sal"
```

附录C. 数据库迁移计划

本节提供迁移过程的概述。注意，您不需要将 DB2 UDB 版本 6 数据库迁移至版本 7。有关迁移 DB2 UDB 版本 5.x 数据库的详细信息可在操作系统的快速入门手册中找到。

在迁移数据库时：

- 迁移下列数据库实体：
 - 数据库配置文件
 - 数据库系统目录表
 - 数据库目录
 - 数据库日志文件头
- 更改系统目录表，如下所示：
 - 添加新列。
 - 创建新表。
 - 在 SYSCAT 模式中，迁移一组目录视图，并创建一组新目录视图。
 - 在 SYSSTAT 模式中创建一组可更新的目录视图。
 - 在 SYSFUN 模式中，保留一组通用的标量函数，并创建一组新的通用标量函数。在数据库迁移期间，只卸下并重建 SYSFUN.DIFFERENCE 标量函数。
- 在数据库目录中创建一个数据库历史文件和它的影子。此文件包含备份信息的摘要，若必须复原一个数据库可使用该信息，而且无论何时对此数据库执行特定操作，都会更新该文件。也保存备份信息的摘要，以用于表空间的备份和复原操作。

迁移考虑事项

要成功迁移用数据库管理程序的先前版本创建的数据库，必须考虑如下事项：

- 『迁移限制』
- 第324页的『安全性与授权』
- 第324页的『存储器需求』
- 第324页的『发行版间的不兼容性』

迁移限制

在尝试将数据库迁移至版本 7 之前，应当注意一些特定的先决条件或限制：

- 仅支持自 V5.x 或 V6 的迁移。不支持自 DB2 V1.2 并行版的迁移。在迁移到 V7 之前，必须将较早版本的 DB2（数据库管理程序）迁移到 V5.x 或 V6。
- 支持从 V7 客户机发出迁移命令以将数据库迁移至 V7 服务器；然而，不支持从较旧版本的 DB2 客户机发出迁移命令来将数据库迁移至 V7 服务器。
- 不支持平台之间的迁移。
- 数据库中的用户对象不能将 V7 保留的模式名用作对象限定符。这些保留的模式名包括：SYSCAT、SYSSTAT 和 SYSFUN。
- 在迁移数据库之前，必须重命名使用名称 BIGINT、REAL、DATALINK 或 REFERENCE 的用户定义单值类型。
- 您不能迁移处于下列状态之一的数据库：
 - 备份暂挂
 - 前滚暂挂
 - 一个或多个表空间未处于正常状态
 - 事务不一致
- 支持低级别（V5.x 或 V6）的数据库备份的复原，但不支持低级别日志的前滚。

安全性与授权

您需要 SYSADM 权限，才可迁移数据库。

存储器需求

迁移期间，旧目录和新目录都需要空间。必需的磁盘空间量将随数据库的复杂性以及数据库对象的数目和大小的不同而不同。这些对象包括所有的表和视图。您应当提供至少相当于该数据库目录当前所占空间两倍的磁盘空间。若无足够的磁盘空间，迁移会失败。

若 SYSCAT 表空间是 SMS 类型的表空间，则还应考虑更新与日志文件相关联的数据库配置参数。应增大 *logfilesiz*、*logprimary* 和 *logsecond* 的值，以防止用于这些日志文件的空间用完（SQL1704N，原因码为 3）。若发生这种情况，则须增加日志空间参数并重新发出 MIGRATE DATABASE 命令。

发行版间的不兼容性

当计划迁移数据库时，请考虑在该产品的两个版本之间的不兼容性的影响。

要利用版本 7 的增强功能，在迁移数据库后，应当调整数据库和数据库管理程序配置。要简化此调整，可记录和比较迁移前后的配置参数值。（有关 GET DATABASE CONFIGURATION 命令和 GET DATABASE MANAGER CONFIGURATION 命令的说明，参考 *Command Reference*。）

迁移数据库

以下是为迁移数据库而必须执行的步骤。必须启动数据库管理程序，才能开始迁移。

在迁移之前:

注: 必须对上一个发行版执行迁移前的步骤（即，在迁移至或安装新发行版前在您的当前发行版上进行）。

1. 验证是否已没有与第323页的『迁移限制』相关的未解决的问题。
2. 断开所有应用程序和最终用户与正在迁移的每个数据库的连接（根据需要使用 LIST APPLICATIONS 命令或 FORCE APPLICATIONS 命令）。
3. 使用 DB2CKMIG 预迁移实用程序来确定是否可迁移该数据库（有关使用此实用程序的详情，参阅您的平台的快速入门一书）。注意，在 Windows NT 或 OS/2 上，安装期间会提示您运行此工具，但在基于 UNIX 的系统上，实例迁移期间会自动调用此工具。
4. 备份数据库。

迁移不是可恢复的过程。若在更改版本 6 保留的模式名前备份数据库，则无法使用 DB2 UDB 版本 7 复原该数据库。要复原该数据库，必须使用先前版本的数据库管理程序。

注意! 若没有数据库的备份，而迁移又失败了，将无法使用 DB2 UDB 版本 7 或数据库管理程序的先前版本复原该数据库。

您还应知道，从执行备份开始到完成迁移至版本 7 为止这段时间内所执行的任何数据库事务都是不能恢复的。即，若在安装和迁移至版本 7 完成后的某个时间，需要复原该数据库（至版本 7 级别），则版本 7 安装之前所写的日志就不可用于前滚恢复。

迁移:

5. 使用以下方法之一迁移数据库:
 - MIGRATE DATABASE 命令
 - 当复原数据库的完整备份时用 RESTORE DATABASE 命令
 - sqllemgdb - 迁移数据库 API。

在 OS/2 上: 在“配置/安装/分发”(CID)体系结构环境中工作的 DB2CIDMG 迁移实用程序只在 DB2 OS/2 版上可用。它允许在基于 LAN 的工作站上进行远程无人照管安装和配置。您必须在 LAN 上装有 NetView DM/2，才可使用 CID 迁移。

在基于 UNIX 的系统上: 您的平台的快速入门一书描述了若您不想迁移一个给定实例中的所有数据库该如何做。

迁移之后:

6. 您可选择使用 **DB2UIDDL** 实用程序按您自己的计划来简化唯一索引的分阶段迁移管理。(版本 5 中创建的 DB2 版本 5 数据库不需要此工具来利用延迟唯一性检查, 因为版本 5 中创建的所有唯一索引已具有这些语义。然而, 对于先前迁移至版本 5 的数据库, 除非使用 **DB2UIDDL** 实用程序来更改唯一索引, 否则这些语义不是自动的。) 此实用程序为用户表上的唯一索引生成 **CREATE UNIQUE INDEX** 语句, 并将它们写至一个文件。将此文件作为 **DB2 CLP** 命令文件运行, 将导致把唯一索引转换为版本 7 的语义。有关使用此实用程序的详情, 参考其中一本快速入门书籍。
7. 可选择对主要影响 **SQL** 查询性能的那些表发出 **RUNSTATS** 命令。旧统计信息保留在迁移后的数据库中, 除非调用 **RUNSTATS** 命令, 否则不更新它们。
8. 您可选择使用 **DB2RBIND** 实用程序来使所有程序包重新有效, 或者允许在第一次使用一个程序包时以隐含方式使程序包再次有效。
9. 若计划在版本 7 中使用解释表, 则可选择迁移它们。有关详情, 参见 *管理指南: 性能* 中的 “**SQL 解释设施**”。
10. 调整数据库和数据库管理程序配置参数以利用版本 7 的增强功能。

附录D. 发行版间的不兼容性

本节标识存在于 DB2 通用数据库和 DB2 的先前发行版之间的不兼容性。

不兼容性是指 DB2 通用数据库中工作方式不同于先前版本 DB2 的那一部分。若在现存应用程序中使用，将生成预期不到的结果，使更改应用程序成为必要，或降低性能。在本上下文中，“应用程序”是指：

- 应用程序代码
- 第三方实用程序
- 交互式 SQL 查询
- 命令或 API 调用。

描述了“DB2 通用数据库”版本 6 和版本 7 引入的不兼容性。它们根据下列类别分组：

- 系统目录视图
- 应用程序设计
- SQL
- 数据库安全性和调整
- 实用程序和工具
- 连通性和共存
- 配置参数。

每个不兼容性小节都包括该不兼容性的说明、该不兼容性的症状或效果，以及可能的解决方案。每个不兼容性说明的开头还有一个指示符，它标识该不兼容性适用的操作系统：

WIN DB2 支持的 Microsoft Windows 平台

UNIX DB2 支持的基于 UNIX 的平台

OS/2 OS/2

注：对于“DB2 通用数据库”版本 6，不再支持版本 1.x 和版本 2.x 客户机，包括“DB2 并行版”版本 1.2 服务器附带封装的客户机。

“DB2 通用数据库” 计划的不兼容性

本节描述“DB2 通用数据库”用户在编码新应用程序或修改现存应用程序时应该记住的将来的不兼容性。这将简化面向将来版本的 DB2 UDB 的迁移。

“DB2 通用数据库” 未来版本中的只读视图

WIN	UNIX	OS/2
-----	------	------

更改

系统目录视图将是只读视图。SYSSTAT 视图将继续是可更新的。

症状

应用于 SYSCAT 视图中的列的 UPDATE 语句将失败。

说明

编写的工具或应用程序通过更新 SYSCAT 视图中定义的列来更改目录中的值。

解决方案

更改该工具或应用程序，以便通过更新 SYSCAT 视图中定义的列来更改目录。

“DB2 通用数据库” 未来版本中的 PK_COLNAMES 和 FK_COLNAMES

WIN	UNIX	OS/2
-----	------	------

更改

SYSCAT.REFERENCES 的列 PK_COLNAMES 和 FK_COLNAMES 不再可用。

症状

列不存在并返回错误。

说明

编写的工具或应用程序使用过时的 PK_COLNAMES 和 FK_COLNAMES 列。

解决方案

更改该工具或应用程序，以改用 SYSCAT.KEYCOLUSE 视图。

在“DB2 通用数据库” 未来版本中 COLNAMES 不再可用

WIN	UNIX	OS/2
-----	------	------

更改

SYSCAT.INDEXES 的列 COLNAMES 不再可用。

症状

列不存在并返回错误。

说明

编写的工具或应用程序使用过时的 COLNAMES 列。

解决方案

更改该工具或应用程序，以改用 SYSCAT.INDEXCOLUSE 视图。

“DB2 通用数据库”版本 7 的不兼容性

本节标识“DB2 通用数据库”版本 7 引入的不兼容性。

应用程序设计

Query Patroller 通用客户机

WIN	UNIX	OS/2
-----	------	------

更改： 因为有新的存储过程，所以这个新版本的客户机应用程序使能器 (CAE) 只能使用“Query Patroller 服务器”版本 7。CAE 是 DB2 的应用程序接口，所有应用程序最终必须通过它来存取数据库。

症状： 若对后面级服务器运行此 CAE，则返回信息 SQL29001。

对象变换函数和结构化类型

WIN	UNIX	OS/2
-----	------	------

更改： 在版本 7 之前的客户机与版本 7 服务器之间有一个次要的且可能性极小的不兼容性，此不兼容性与对 SQLDA 所作的更改相关。如 *Application Development Guide* 所述，除了值 X'00' 和 X'01' 之外，第二个 SQLVAR 的字节 8 的值现在还可以是 X'12'。不期望新值的应用程序可能会受此扩展影响。

解决方案： 因为将来的发行版可能会对此字段进行其他扩展，所以建议开发者仅测试显式定义的值。

JVM 使用的类和 Jar 文件的版本

WIN	UNIX	OS/2
-----	------	------

更改： 以前，在 Java 存储过程或用户定义函数 (UDF) 启动后，“Java 虚拟机” (JVM) 便锁定 CLASSPATH 中给出的所有文件（包括 sqllib/function 中的那些文件）。在数据库管理程序停止之前，JVM 使用这些文件。根据运行存储过程或

UDF 的环境的不同（即，根据 *keepdari* 数据库管理程序配置参数的值，以及该存储过程是否受保护），刷新类允许您替换类和 jar 文件，而不会停止数据库管理程序。这与先前行为不同。

“安装 Jar”、“替换 Jar”和“除去 Jar”命令更改了功能

WIN	UNIX	OS/2
-----	------	------

更改：以前，安装 jar 会导致清除所有 DARI（数据库应用程序远程接口）进程。这样，保证在下一调用时挑选新的存储过程类。目前，jar 命令不清除 DARI 进程。要确保挑选新安装或替换的 jar 中的类，必须显式发出 `SQLJ.REFRESH_CLASSES` 命令。

由于不清除 DARI 进程而引入的另一不兼容性是这样一个事实：对于受保护存储过程，当 *keepdari* 数据库管理程序配置参数的值设置为“YES”时，客户机可能会获得不同版本的 jar 文件。考虑以下方案：

1. 用户 A 替换一个 jar，但未刷新类。
2. 然后，用户 A 从该 jar 调用一个存储过程。假定此调用使用同一 DARI 进程，则用户 A 将得到旧版本的 jar 文件。
3. 用户 B 调用同一存储过程。此调用使用一个新的 DARI，这表示新创建的类装入程序将挑选新版本的 jar 文件。

换而言之，若在 jar 操作后不刷新类，则可能会调用不同版本的 jar 的存储过程，这取决于正在使用的 DARI 进程。这与先前行为不同，先前行为（通过清除 DARI 进程）确保始终使用新类。

32 位应用程序不兼容性

	UNIX	
--	------	--

更改：32 位可执行文件（DB2 应用程序）不能对新的 64 位数据库引擎运行。

症状：应用程序连接失败。当尝试对 64 位 DB2 应用程序库连接 32 位目标时，显示操作系统连接程序错误信息。

解决方案：必须将该应用程序重新编译为 64 位可执行程序，并对新的 64 位 DB2 库重新连接。

更改便签簿的长度字段

WIN	UNIX	OS/2
-----	------	------

更改：现在，任何更改传送给用户定义函数（UDF）的便签簿的长度字段的 UDF 都将接收到 `SQLCODE -450`。

症状: 更改便签簿的长度字段的 UDF 失败。调用语句接收到 SQLCODE -450, 其中填入了模式和特定函数名。

解决方案: 重新编写 UDF 主体, 不更改便签簿的长度字段。

SQL

使用由模式 **SESSION** 限定的规则表的应用程序

WIN	UNIX	OS/2
-----	------	------

更改: 临时表只允许 **SESSION** 模式, 现在, DB2 使用此模式来指示 **SESSION** 限定的表可引用临时表。然而, **SESSION** 不是一个为临时表保留的关键字, 可以用作规则基本表的模式。因此, 应用程序会发现同时存在一个 **SESSION.T1** 实表和一个 **SESSION.T1** 已说明临时表。当联编程序包时, 若遇到包含由 "**SESSION**" (显式或隐式) 限定的表引用的静态语句, 则既不将此语句的一节也不将此语句的从属存储在目录中。而是, 需要在运行期以增量方式联编此节。这将把该节的一个副本放在动态 **SQL** 高速缓存中, 高速缓存的副本供该应用程序的唯一实例专用。在运行期, 若存在与该表名相匹配的已说明临时表, 则即使存在同名的持久基本表, 也使用该已说明临时表。

症状: 在版本 6 (和更早版本) 中, 任何带有包含由 **SESSION** 限定的表的静态语句的程序包都始终引用持久基本表。当联编该程序包时, 将该语句的一节以及相关从属记录保存在目录中。在版本 7 中, 这些语句不在联编期联编, 可以在运行期将它们解析为同名的已说明临时表。因此, 会发生下列情况:

- 从版本 5 迁移。若版本 5 中存在这样的程序包, 则将在版本 6 中再次联编它, 静态语句现在将以增量方式联编。这会降低性能, 因为除了不能在其他应用程序 (甚至是同一应用程序可执行程序的不同实例) 之间共享高速缓存的动态节外, 这些以增量方式联编的节的行为与高速缓存的动态 **SQL** 相同。
- 从版本 6 迁移至版本 7。即使版本 6 中存在这样的程序包, 也无需在版本 7 中再次联编它。而是, 那些语句仍作为常规静态 **SQL** 执行, 使用在原始联编期保存在目录中的节。然而, 若重新联编此程序包 (无论是隐式联编还是显式联编), 则不再存储程序包中具有 **SESSION** 限定的表引用的语句, 它们将需要增量联编。这会降低性能。

总而言之, 在版本 7 中联编的具有引用 **SESSION** 限定表的静态语句的任何程序包, 都不再象静态 **SQL** 那样执行, 因为它们需要增量联编。事实上, 若应用程序进程向与现存 **SESSION** 限定表、视图或别名同名的一个表发出 **DECLARE GLOBAL TEMPORARY TABLE** 语句, 则对那些对象的引用始终被改为引用已说明临时表。

解决方案： 有可能的话，更改持久表的模式名，使它们不是“SESSION”。否则，没有其他方法，性能会受到影响，并可能与已说明临时表发生冲突。

可使用以下查询来标识在应用程序使用临时表时可能会受影响的表、视图和别名：

```
select tabschema, tabname from SYSCAT.TABLES where tabschema = 'SESSION'
```

可使用以下查询来标识在目录中存储了静态节、并且在重新联编程序包时其行为可能会更改的版本 7 联编程序包（仅当从版本 6 移至版本 7 时才相关）：

```
select pkgschema, pkgname, bschema, bname from syscat.packagedep
where bschema = 'SESSION' and btype in ('T', 'V', 'I')
```

实用程序和工具

Solaris 上的 DataLinks 文件管理器和文件系统过滤器

	UNIX	
--	------	--

更改： “DataLinks 文件管理器”和“文件系统过滤器”在 Solaris OS 2.5.1 上不受支持。

AIX 和 Solaris 上的 db2set

	UNIX	
--	------	--

更改： 命令“db2set -ul (user level)”及其相关函数未移植至 AIX 或 Solaris。

连通性和共存

32 位客户机不兼容性

WIN	UNIX	OS/2
-----	------	------

更改： 32 位客户机不能连接实例，也不能与 64 位服务器上的数据库连接。

症状： 若客户机和服务器都在运行版本 7 代码，则返回 SQL1434N；否则，连接失败，并返回 SQLCODE -30081。

解决方案： 使用 64 位客户机。

“DB2 通用数据库版本 6”的不兼容性

本节标识“DB2 通用数据库”版本 6 引入的不兼容性。

系统目录视图

“DB2 通用数据库版本 6”中的系统目录视图

WIN	UNIX	OS/2
-----	------	------

更改: 在系统目录视图中引入了新的代码: “U” 表示类型表, “W” 表示类型视图。

症状: 使用类型码 “T” 表示表, “V” 表示视图, 在系统目录中搜索表和视图的查询再不会查找到类型表和类型视图。

说明: 几个系统目录, 包含称为 TABLES、PACKAGEDEP、TRIGDEP 和 VIEWDEP 的系统目录视图, 都有称为 TYPE 或 BTYPE 的一列, 它们包含单字母类型码。在版本 5.2 中, 类型码 “T” 用来表示所有的表, 类型码 “V” 用来表示所有的视图。在版本 6 中, 继续用类型码 “T” 表示非类型表, 而用新的类型码 “U” 表示类型表。同样, 继续用类型码 “V” 表示非类型视图, 而用新的类型码 “W” 表示类型视图。另外, 在系统目录表中将出现一种新的表, 它称为分级结构表, 用类型码 “H” 表示, 用户不能直接创建这种类型表, 而系统可使用它来实现表的分级结构。

解决方案: 更改该工具或应用程序, 以便可识别类型表和类型视图的代码。若该工具或应用程序需要表的逻辑视图, 应使用类型码 “T”、“U”、“V” 和 “W”。若该工具或应用程序需要表的物理视图, 包括分级结构表, 应使用类型码 “T” 和 “H”。

“DB2 通用数据库版本 6”中的主关键字列名和外部关键字列名

WIN	UNIX	OS/2
-----	------	------

更改: SYSCAT.REFERENCES 的两列 PK_COLNAMES 和 FK_COLNAMES 的数据类型, 由 VARCHAR(320) 更改为 VARCHAR(640)。

症状: 主关键字或外部关键字的列名被截断、不正确或丢失。

说明: 当主关键字或外部关键字的列名长度超过 18 字节时, 则列名的列表不能以相同的格式保存在这两列中。在长度 (n) 超过 18 个字节的列后面出现的、以 20 字节空白定界的列名, 将向右移 $n-18$ 个字节。另外, 若列名的列表超过 640 字节, 该列将包含空字符串。

解决方案: SYSCAT.KEYCOLUSE 视图包含构成一个主关键字、外部关键字和唯一关键字的那些列的列表, 并应该用来替代 SYSCAT.REFERENCES 中的那些列。或者, 用户可以限制列名的长度为 18 字节, 或限制列的列表总长度为 640 字节。

“DB2 通用数据库版本 6”中的 SYSCAT.VIEWS 列 TEXT

WIN	UNIX	OS/2
-----	------	------

更改: SYSCAT.VIEWS 列 TEXT 中的视图文本将不再分成多行。该数据类型已由 VARCHAR(3600) 更改为 CLOB(64K)。

症状: 工具或应用程序没有给出完整的视图文本。

说明: 工具或应用程序被编写为一次从 TEXT 列返回的视图文本不会超过 3600（或 3900）字节，这样的话，则不能处理此字段增加的字节。不再需要使用 SEQNO 字段检索多行并重构视图文本的机制。SEQNO 值将始终是 1。

解决方案: 更改该工具或应用程序，以便能够处理来自 TEXT 列的大于 3600 字节的值。或者，重写视图 TEXT 以便不超过 3600 字节。

“DB2 通用数据库版本 6”中的 SYSCAT.STATEMENTS 列 TEXT

WIN	UNIX	OS/2
-----	------	------

更改: SYSCAT.STATEMENTS 列 TEXT 中的语句文本将不再分成多行。该数据类型已由 VARCHAR(3600) 更改为 CLOB(64K)。

症状: 工具或在应用程序没有给出完整的语句文本。

说明: 工具或应用程序被编写为一次从 TEXT 列返回的视图文本不会超过 3600（或 3900）字节，这样的话，则不能处理此字段增加的大小。不再需要使用 SEQNO 字段检索多行并重构语句文本的机制。SEQNO 值将始终是 1。

解决方案: 更改该工具或应用程序，以便能够处理来自 TEXT 列的大于 3600 字节的值。或者，重写语句 TEXT 以便不超过 3600 字节。

“DB2 通用数据库版本 6”中的 SYSCAT.INDEXES 列 COLNAMES

WIN	UNIX	OS/2
-----	------	------

更改: SYSCAT.INDEXES 列 COLNAMES 的数据类型由 VARCHAR(320) 更改为 VARCHAR(640)。

症状: 索引中的列名丢失。

说明: 编写的工具或应用程序只能从数据类型为 VARCHAR(320) 的列检索数据，不能处理此字段增加的字节。

解决方案： SYSCAT.INDEXCOLUSE 视图包含构成一个索引的那些列的列表，应用来代替 COLNAMES 列。或者，从索引中除去一列，或减小列名大小，以便列名（前缀为 + 或 -）的列表不超过 320 字节。

“DB2 通用数据库版本 6”中的 SYSCAT.CHECKS 列 TEXT

WIN	UNIX	OS/2
-----	------	------

更改： CHECKS 列 TEXT 的数据类型由 CLOB(32K) 更改为 CLOB(64K)。

症状： 检查约束子句不完整。

说明： 编写的工具或应用程序只能从数据类型为 CLOB(32K) 的列检索数据，不能处理此字段增加的字节。

解决方案： 更改该工具或应用程序，以便能够处理来自 TEXT 列的长于 32 KB 的值。另外，重写检查约束子句以便使用更少的字符，使它不超过 32 KB。

“DB2 通用数据库版本 6”中列数据类型更改为 BIGINT

WIN	UNIX	OS/2
-----	------	------

更改： 几个系统目录视图已将其列数据类型由 INTEGER 更改为 BIGINT。

症状： 值比期望的要小得多（或大得多），特别是统计信息。

说明： 编写的工具或应用程序只能从数据类型为 INTEGER 的列检索数据，不能处理此字段增加的字节。

解决方案： 更改该工具或应用程序，以便能够处理比可存储在 INTEGER 字段中的最大值更大或比最小值更小的值。另外，更改导致该值落在可在 INTEGER 字段中表示的值外部的基本结构或 SQL 代码。

“DB2 通用数据库版本 6”中的列不匹配

WIN	UNIX	OS/2
-----	------	------

更改： 在 SYSCAT 视图定义中，没有在视图结尾插入新列。

症状： 由于几个列不匹配或列数据类型不匹配，重新预处理失败。

说明： 将新列插入系统目录视图，并将其放在对特定查询环境有利的位置；确切地说，将较短的列放在很长的列之前，REMARKS 列始终是最后一列。

解决方案： 在 select 列表中显式命名那些列，而不是编码 "SELECT *"。

“DB2 通用数据库版本 6”中的 SYSCAT.COLUMNS 和 SYSCAT.ATTRIBUTES

WIN	UNIX	OS/2
-----	------	------

更改： SYSCAT.COLUMNS 和 SYSCAT.ATTRIBUTES 现在包含继承的列和属性的项。

症状： 若查询的主题是子表、子视图或子类型，则对 SYSCAT.COLUMNS 的查询（检索类型表或类型视图的列）和对 SYSCAT.ATTRIBUTES 的查询（检索结构化类型的属性）在版本 6 中返回的行数比在版本 5.2 中返回的多。

说明： 在版本 5.2 中，对于给定的表、视图或结构化类型，COLUMNS 和 ATTRIBUTES 目录只包含通过该表、视图或类型引入的列和属性的项。从超表或超类型继承的列和属性不出现在目录中。然而，在版本 6 中，COLUMNS 和 ATTRIBUTES 目录将包含继承的列和属性的项。

解决方案： 更改该工具或应用程序，以便可识别 COLUMNS 和 ATTRIBUTES 目录中的新项。

“DB2 通用数据库版本 6”中不再支持 OBJCAT 视图

WIN	UNIX	OS/2
-----	------	------

更改： 版本 5.2 的 OBJCAT 模式中的递归目录视图不再是交付的“DB2 通用数据库”产品的一部分。

症状： 针对 OBJCAT 目录视图编写的查询将不再能够成功运行。

解决方案： OBJCAT 视图中大多数先前的信息已合并到规则的 SYSCAT 目录视图中。在大多数情况下，可以从系统目录视图中获取信息。若从版本 5.2 迁移，且 OBJCAT 目录视图存在，则应将它们卸下。为此，可运行 CLP 脚本 objcatdp.db2，该脚本可在 sqllib 目录的 misc 子目录中找到。

也可创建一组自己的 OBJCAT 视图，它们与版本 5.2 中支持的目录视图等效。

在版本 5.2 中，*SQL Reference* 的附录 E 提醒用户 OBJCAT 目录视图只是临时的，在未来版本中不受支持。

“DB2 通用数据库版本 6”中更改的关联代码

WIN	UNIX	OS/2
-----	------	------

更改： 在系统目录视图中，以前由“H”表示的分层从属性现在由“O”表示。

症状: 在目录视图中搜索由代码 "H" 表示的分层从属性的查询将不再生效。

说明: 几个系统目录, 包含系统目录视图 PACKAGEDEP、TRIGDEP 和 VIEWDEP, 都有 BTYPE 列。在版本 5.2 中, OBJCAT 视图通过代码 "H" 表示分层从属性。在版本 6 中, 这些从属性由代码 "O" 表示。

解决方案: 修订这些查询, 以搜索代码 "O"。

“DB2 通用数据库版本 6”中的 SYSIBM 基本目录表

WIN	UNIX	OS/2
-----	------	------

更改: 以下是对 SYSIBM 基本目录表的更改, 仍可使用它来代替 SYSCAT 视图:

- 删除了字段 (但仍在 SYSCAT 视图中):
 - SYSSTMT.SEQNO
 - SYSVIEWS.SEQNO
- 重命名了目录表: SYSTRIGDEP 更改为 SYSDEPENDENCIES。并且, 将列 BCREATOR 和 DCREATOR 分别重命名为 BSHEMA 和 DSHEMA。视图 SYSCAT.TRIGDEP 不变。
- 删除了字段 (未曾在 SYSCAT 视图中出现过):
 - SYSATTRIBUTES.DEFAULT_VALUE
 - SYSATTRIBUTES.NULLS
 - SYSCOLUMNS.SERVERTYPE
 - SYSDATATYPES.REFREP_TYPENAME
 - SYSDATATYPES.REFREP_TYPESHEMA
 - SYSDATATYPES.REFREP_LENGTH
 - SYSDATATYPES.REFREP_SCALE
 - SYSDATATYPES.REFREP_CODEPAGE
 - SYSINDEXES.TEXT
(以前在视图中, 但保留下来仅供将来使用。)
 - SYSPLANDEP.PUBLICPRIV
 - SYSSECTION.SEQNO
 - SYSTABAUTH.UPDATE_BY_COLS
 - SYSTABAUTH.REF_BY_COLS
 - SYSTABLES.MINPDLENGTH
 - SYSTABLESPACES.READONLY
 - SYSTABLESPACES.REMOVABLEMEDIA

- 数据类型更改:
 - SYSSECTION.SECTION, 由 VARCHAR(3600) 更改为 CLOB(10M)
 - SYSPLANDEP.COLUSAGE, 由 VARCHAR(3000) FOR BIT DATA 更改为 BLOB(5K)

应用程序设计

“DB2 通用数据库版本 6”中的 VARCHAR 数据类型

WIN	UNIX	OS/2
-----	------	------

更改: 在版本 6 中, VARCHAR (VARGRAPHIC) 数据类型的最大可能大小已由 4000 字符 (2000 双字节字符) 增加到 32672 字符 (16336 双字节字符)。

症状: 对 VARCHAR (VARGRAPHIC) 数据类型使用 4000 字节定长缓冲区的应用程序, 若它将一个大于 4000 字节的 VARCHAR 字段读入一个很小的缓冲区中, 可能造成缓冲区被覆盖或截断。CLI 函数 - SQLGetTypeInfo() 现在将 VARCHAR 的大小返回为 32672。由于没有足够页大小的表空间可用, 在表 DLL 中使用此值的 CLI 应用程序可能收到错误信息。有关表空间页大小的详情, 参见第106页的『用户表数据』。

解决方案: 当编码应用程序时, 建议首先描述结果集的列 (使用 DESCRIBE 语句), 然后使用大小基于 DESCRIBE 语句返回的长度的缓冲区。

在“DB2 通用数据库版本 6”中通过 Java 程序设计定位的 UPDATE 和 DELETE

WIN	UNIX	OS/2
-----	------	------

更改: 在版本 6 中使用 Java 进行程序设计时, 定位的 UPDATE 和 DELETE 语句使用联编游标程序包的那个人的缺省授权标识符。这不同于版本 5.2, 在版本 5.2 中使用运行程序包的那个人的授权标识符。

症状: 包含定位 UPDATE 和 DELETE 语句的程序包可能不能运行, 因为联编程序包的那个人的授权标识符没有足够的权限。

解决方案: 必须给联编程序包的那个人的授权标识符授予足够的权限, 以运行程序包中的定位 UPDATE 和 DELETE 语句。授予正确的特权并重新联编程序包。

“DB2 通用数据库版本 6”中 FOR UPDATE 子句的语法更改

WIN	UNIX	OS/2
-----	------	------

更改: 在版本 5.2 中, 可在 SQLJ 程序中使用 SELECT 语句中的 FOR UPDATE 子句来标识可在后续的定位 UPDATE 语句中更新的列。在版本 6 中, 语法已更改。

症状: 若 SELECT 语句包含 FOR UPDATE 子句, 将接收到出错信息 SQJ0204E。

解决方案: 除去 SELECT 语句中的 FOR UPDATE 子句。通过迭代声明子句指定可更新的迭代程序。例如:

```
#sql public iterator DelByName implements sqlj.runtime.ForUpdate(String EmpNo)
with updateColumns = (salary);
```

若想显式标识哪些列可更新, 通过 updateColumns 关键字以及 WITH 子句指定它们。

有关定位迭代声明的详情, 参考 *Application Development Guide*。

“DB2 通用数据库版本 6”中的字符名大小

WIN	UNIX	OS/2
-----	------	------

更改: “DB2 通用数据库版本 6”支持 128 字节的表名、视图名和别名以及 30 字节的列名。先前版本中的上述每个实体名只能为 18 字节。

USER 和 CURRENT SCHEMA 专用寄存器以前是 CHAR(8), 而现在是 VARCHAR(128)。CURRENT EXPLAIN MODE 专用寄存器以前是 CHAR(8), 而现在是 VARCHAR(254)。TYPE_SCHEMA 和 TABLE_SCHEMA 内部函数的输出以前是 CHAR(8), 而现在是 VARCHAR(128)。

症状: 若对没有使用长名限制的版本 6 数据库运行早于版本 6 开发的应用程序, 则该应用程序的行为不会有任何更改。然而, 若对确实使用了长名的版本 6 数据库运行这些应用程序, 可能导致某些副作用, 这取决于这些应用程序是如何编码的。

下面是一些示例:

- 考虑一个现存的应用程序, 它将一个表名或列名 (一般来自目录视图) FETCH 进被定义为 18 字节长的主变量中。由于 18 字节是版本 6 之前的表名或列名的大小限制, 所以此应用程序在检查 SQLCA 的 sqlwarn1 位时可能不会遇到麻烦。将 (不正确地) 认为不会发生截断。
- 考虑一个应用程序, 它将表名或列名 (一般来自目录视图) FETCH 进 SQLDA 中, 其中 sqldata 字段的大小根据 sqlllen 字段 (由该 SELECT 的 DESCRIBE 确定) 分配。即使表名或列名的大小可能已增加, 应用程序也会得到正确 (未

被截断)的结果。若其他应用程序逻辑假定列名限长 18 字节,则可能会以不期望的方式处理更长的名称;例如,更长的列名的显示可能在第 18 个字节后被截断。

- 由于 SQLCA 记号字段 (sqlerrmc) 限长 70 字节,尝试向表中插入一行的现存应用程序可能会受到影响。作为对错误 SQL0204N 的响应,这样的应用程序根据 SQLCA sqlerrmc 字段确定表名,然后根据该对象名执行一些操作。对于较早版本的 DB2,表或模式标识符限制保证整个表名可以包含在 SQLCA 中。在版本 6 中,情况并非如此。
- 使用后面级 API 的应用程序将只得到表名的前 18 个字节。
- 当与一个支持 18 字节名称的服务器连接时,使用模式函数(如 SQLTables()、SQLColumns() 等)的现存的 CLI 和 ODBC 应用程序将会受影响。尽管会出现截断警告,但应用程序可能会忽略此警告,使用截断名称继续运行。

解决方案: 解决这类问题的最好方法是重新编码应用程序,以便可处理更长的表名和列名。否则,确保不要对使用超过 18 字节的名称的版本 6 数据库运行这些应用程序。

“DB2 通用数据库版本 6”中的 PC/IXF 格式更改

WIN	UNIX	OS/2
-----	------	------

更改: “DB2 通用数据库版本 6”支持 128 字节的表名、视图名和别名以及 30 字节的列名。先前版本中的上述每个实体名只能为 18 字节。

症状: “DB2 通用数据库版本 5”客户机不能调入由“DB2 通用数据库版本 6”客户机调出的 PC/IXF 文件(错误 SQL3059N)。(从“DB2 通用数据库版本 6”客户机中调出的)PC/IXF 文件不能装入“DB2 通用数据库版本 5”数据库中(错误 SQL3059N)。

解决方案: 当调入或装入 PC/IXF 数据时,需使用“DB2 通用数据库”的兼容版本。

“DB2 通用数据库版本 6”中非双精度 SQLVAR 中的 SQLNAME

WIN	UNIX	OS/2
-----	------	------

更改: “DB2 通用数据库版本 6”支持 30 字节的列名。先前版本只支持 18 字节的名称。在版本 5 中,所记载的行为是将“0xFF”放在非双精度 SQLVAR 的 SQLNAME 字段的第 30 个字节中;对于系统生成的名称以及用户在“AS”子句中指定的列名,还将“0x00”放在第 30 个字节中。

在版本 6 中,仅当名称是系统生成的时,才在第 30 个字节中返回“0xFF”。

症状: 若用户指定的列名由 30 个字符组成, 根据 SQLNAME 字段的第 30 个字节来确定是用户指定的列名还是系统生成的名称的任何应用程序, 可能接收到意外的逻辑检查结果。这种情况几乎不会发生。

解决方案: 若 SQLNAME 字段的长度小于 30 字节, 应修改这些应用程序应以便只检查该字段的第 30 个字节处的 "0xFF"。在这种情况下, 名称是由用户生成的。

“DB2 通用数据库版本 6”中过时的 DB2 CLI/ODBC 配置关键字

WIN		
-----	--	--

更改: 当迁移至新版本的 DB2 UDB 时, 通过在 db2cli.ini 文件中指定一组可选关键字, 可更改 DB2 CLI/ODBC 驱动程序的行为。

在版本 6 中, TRANSLATEDLL 和 TRANSLATEOPTION 关键字已过时。

症状: 若这些关键字仍存在, 则会被忽略。可能注意到由于除去了这些设置而引起的行为更改。

解决方案: 将需要复查新的有效参数列表以决定适合于您的环境的关键字和设置。有关这些关键字的详情, 参考 *CLI Guide and Reference*。

“DB2 通用数据库版本 6”中的事件监控程序输出流格式

WIN	UNIX	OS/2
-----	------	------

更改: 事件监控程序输出流没有版本控制。因此, 要增加对超过 18 字节的表名的支持, 要求更改为一种输出流格式。

症状: 对事件监控程序输出流进行语法分析的应用程序不再能够正确工作。

解决方案: 您有两个选择:

- 更新应用程序以便使用新的数据流。
- 设置注册表变量

DB20LDEVMON=evmonname1, evmonname2, ...

其中, *evmonname* 是要以旧数据格式编写的事件监控程序的名称。注意, 事件监控程序中的任何新字段都不可以旧数据格式存取。

SQL

DB2 通用数据库版本 6 中的 DATALINK 列

	UNIX	
--	------	--

更改: 在“DB2 通用数据库”版本 6 下面插入的 DATALINK 值要求在列值描述符中多占四字节的空间。

症状: 当更新在版本 5.2 中创建的 DATALINK 列时, 需要在数据页中多占四字节以便存储新的列值。结果是, 数据页没有足够多的空间完成更新, 这样不得不将它移动到新的页中。这可能造成更新操作用完所有的空间。

解决方案: 需要在系统上添加更多空间以允许更新。

“DB2 通用数据库版本 6”中的 SYSFUN 字符串函数特征

WIN	UNIX	OS/2
-----	------	------

更改: SYSFUN 模式中许多字符串函数现已在 SYSIBM 模式 (内部函数) 中定义了改进的版本。这些函数名为 LCASE、LTRIM、RTRIM 和 UCASE。

症状: 当准备语句或创建视图时, 从任何这些函数返回的数据类型都可能与在版本 6 中不同。造成这种情况的原因是, 内部函数 (在 SYSIBM 中定义的) 通常在 SYSFUN 模式中的函数解析之前被解析。

解决方案: 不需要任何操作。内部函数通常优先于 SYSFUN 模式中的函数。可通过转换 SQL 路径复原先前版本的行为 (以便 SYSFUN 在 SYSIBM 之前发生), 但性能会下降。也可用模式名 SYSFUN 限定函数名来调用先前版本的函数。

若没有执行显式操作, 如显式联编程序包、重新创建视图、摘要表、触发器或约束, 则引用这些函数的已迁移的程序包、视图、摘要表、触发器和约束将继续使用 SYSFUN 模式中的版本。

“DB2 通用数据库版本 6”中具有新的完整性状态的 SYSTABLE 列更改

WIN	UNIX	OS/2
-----	------	------

更改: 当运行 SET INTEGRITY ... OFF 语句时, SYSCAT.TABLES 的 CONST_CHECKED 列中的 “U” 状态发生更改。

症状: 在版本 6 之前的版本中, 当运行 SET INTEGRITY ... OFF 语句时, CONST_CHECKED 列中的 “U” 状态更改为 “N” 状态。“U” 状态现在更改为 “W” 状态。

解决方案: 不需要任何操作。CONST_CHECKED 列中的新状态 “W” 用来指示约束类型以前被该用户检查过, 且表中的某些数据可能需要作完整性检查。

“N” 状态未澄清是否存在数据库管理程序尚未验证过的任何旧数据。当后续 SET INTEGRITY ... IMMEDIATE CHECKED INCREMENTAL 语句运行时, 数据库管理程序必须返回错误, 因为如果只检查新更改, 便不能保证数据完整性。另一方

面，可将“W”状态更改回“U”状态（若指定 INCREMENTAL 选项的话），以指示用户仍对表的数据完整性负责。若未指定 INCREMENTAL 选项，数据库管理程序将选择完全处理，将“W”状态更改为“Y”状态，并承担维护数据完整性的责任。

数据库安全性和协调

在“DB2 通用数据库版本 6”中使用客户机创建数据库

WIN	UNIX	OS/2
-----	------	------

更改： 客户机用来创建数据库的方法。

症状： 使用后面级客户机来创建数据库将导致错误。

解决方案： 当使用客户机来创建数据库时，确保客户机和服务器运行同一级别的 DB2 代码。

“DB2 通用数据库版本 6”中需要对分级结构拥有 SELECT 特权

WIN 32 位	UNIX	OS/2
----------	------	------

更改： 对一个表指定 ONLY 关键字时，现在要求用户具有指定的类型表的所有子表的 SELECT 特权。同样，对一个视图指定 ONLY 关键字时，现在要求用户具有指定的类型视图的所有子视图的 SELECT 特权。先前版本的 DB2 只需要指定的表或视图的 SELECT 特权。

症状： 此更改有两个可能的症状：

- 当重新联编包含 SQL 语句（在 FROM 子句中指定 ONLY 关键字）的程序包时，若联编程序包所用的授权 ID 缺少命名的类型表（或视图）的子表的 SELECT 特权，将发生授权错误（SQLCODE -551、SQLSTATE 42501）。
- 若一个视图或触发器的定义包含在 FROM 子句中的 ONLY 关键字，则该视图或触发器将继续正常工作。然而，除非创建者拥有命名的表（或视图）的全部子表的 SELECT 特权，否则不能再使用该视图或触发器的定义来创建新的视图或触发器。

解决方案： 应给需要重新联编程序包或创建新视图或触发器的授权 ID 授予对 ONLY 关键字后指定的表（或视图）的全部子表（和子视图）的 SELECT 特权。

“DB2 通用数据库版本 6”中过时的简要表注册表变量和环境变量

WIN	UNIX	OS/2
-----	------	------

更改： 下列简要表注册表变量或环境变量是过时的：

- DB2_VECTOR

解决方案: 不再需要这些变量。

实用程序和工具

“DB2 通用数据库版本 6”中当前的解释方式

WIN	UNIX	OS/2
-----	------	------

更改: “CURRENT EXPLAIN MODE” 专用寄存器的类型已由 CHAR(8) 更改为 VARCHAR(254)。

症状: 若应用程序认为该类型仍为 CHAR(8), 该值可能由 254 字节截断为 8 字节。

解决方案: 将读取专用寄存器的所有主变量的类型由 CHAR(8) 重新定义为 VARCHAR(254)。

此更改一定要包含 “CURRENT EXPLAIN MODE” 专用寄存器的两个新值。这些新值是 “EVALUATE INDEXES” 和 “RECOMMEND INDEXES”。

“DB2 通用数据库版本 6”中的 USING 和 SORT BUFFER 参数

WIN	UNIX	OS/2
-----	------	------

更改: 在版本 6 中, 不再支持 LOAD 命令的 USING 和 SORT BUFFER 参数。这些参数被忽略。

症状: 返回一警告信息, 指出不再支持 USING 和 SORT BUFFER 参数, 装入实用程序将忽略这两个参数。

解决方案: 忽略该警告信息。有关其他信息, 参考 *Data Movement Utilities Guide and Reference*。

连通性和共存

在“DB2 通用数据库版本 6”中用 PCOMM 替换 RUMBA

WIN		
-----	--	--

更改: 在版本 6 中, RUMBA 在 Windows NT、Windows 98 和 Windows 95 (而不包括 Windows 3.1) 上被 PCOMM 替换。

症状: 无。

解决方案： 无。

配置参数

过时的数据库配置参数

WIN	UNIX	OS/2
-----	------	------

更改： 下列数据库配置参数已过时：

- DL_NUM_BACKUP (被 NUM_DB_BACKUP 数据库配置参数替换)

解决方案： 从应用程序中除去对这些参数的所有引用。

附录E. 国家语言支持 (NLS)

本节包含有关 DB2 提供的国家语言支持 (NLS) 的信息，包括有关支持的国家、语言和代码页（代码集）的信息，并介绍如何在您的数据库和应用程序中配置和使用 DB2 NLS 功能。

国家代码和代码页支持

表32显示受数据库服务器支持的语言和代码集以及如何将这些值映射为数据库管理程序使用的国家代码或代码页值。

以下是该表每一列的说明：

- **代码页**显示 IBM 定义的代码页，如从操作系统代码集所映射的一样。
- **组**显示代码页是单字节 ("S") 还是多字节 ("D")。"-n" 是一个用于创建字母-数字组合的数。匹配组合显示 DB2 在何处允许连接和转换。例如，所有的 "S-1" 个组都可一起工作。
- **代码集**显示与受支持的语言相关的代码集。代码集被映射为 DB2 代码页。
- **Tr.** 显示两字母领域标识符。
- **国家代码**显示数据库管理程序内部用来提供特定国家支持的国家代码。
- **本国语言环境**显示数据库管理程序所支持的本国语言环境值。
- **OS** 显示支持此语言和代码集的操作系统。
- **国家名**显示一个或一些国家的名称。

表 32. 受支持的语言和代码集

代码页	组	代码集	Tr.	国家代码	本国语言环境	OS	国家名
437	S-1	IBM-437	AL	355	-	OS2	阿尔巴尼亚
850	S-1	IBM-850	AL	355	-	OS2	阿尔巴尼亚
819	S-1	ISO8859-1	AL	355	sq_AL	AIX	阿尔巴尼亚
850	S-1	IBM-850	AL	355	Sq_AL	AIX	阿尔巴尼亚
819	S-1	iso88591	AL	355	-	HP	阿尔巴尼亚
1051	S-1	roman8	AL	355	-	HP	阿尔巴尼亚
819	S-1	ISO8859-1	AL	355	-	Sun	阿尔巴尼亚
1252	S-1	1252	AL	355	-	WIN	阿尔巴尼亚
1275	S-1	1275	AL	355	-	Mac	阿尔巴尼亚
37	S-1	IBM-37	AL	355	-	HOST	阿尔巴尼亚
1140	S-1	IBM-1140	AL	355	-	HOST	阿尔巴尼亚

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
864	S-6	IBM-864	AA	785	-	OS2	阿拉伯国家
1046	S-6	IBM-1046	AA	785	Ar_AA	AIX	阿拉伯国家
1089	S-6	ISO8859-6	AA	785	ar_AA	AIX	阿拉伯国家
1089	S-6	iso88596	AA	785	ar_SA.iso88596	HP	阿拉伯国家
1256	S-6	1256	AA	785	-	WIN	阿拉伯国家
420	S-6	IBM-420	AA	785	-	HOST	阿拉伯国家
437	S-1	IBM-437	AU	61	-	OS2	澳大利亚
850	S-1	IBM-850	AU	61	-	OS2	澳大利亚
819	S-1	ISO8859-1	AU	61	en_AU	AIX	澳大利亚
850	S-1	IBM-850	AU	61	En_AU	AIX	澳大利亚
819	S-1	iso88591	AU	61	-	HP	澳大利亚
1051	S-1	roman8	AU	61	-	HP	澳大利亚
819	S-1	ISO8859-1	AU	61	en_AU	Sun	澳大利亚
819	S-1	ISO8859-1	AU	61	en_AU	SCO	澳大利亚
1252	S-1	1252	AU	61	-	WIN	澳大利亚
1275	S-1	1275	AU	61	-	Mac	澳大利亚
37	S-1	IBM-37	AU	61	-	HOST	澳大利亚
1140	S-1	IBM-1140	AU	61	-	HOST	澳大利亚
437	S-1	IBM-437	AT	43	-	OS2	奥地利
850	S-1	IBM-850	AT	43	-	OS2	奥地利
819	S-1	ISO8859-1	AT	43	ge_AT	AIX	奥地利
850	S-1	IBM-850	AT	43	Ge_AT	AIX	奥地利
819	S-1	iso88591	AT	43	-	HP	奥地利
1051	S-1	roman8	AT	43	-	HP	奥地利
819	S-1	ISO8859-1	AT	43	de_AT	SCO	奥地利
819	S-1	ISO-8859-1	AT	43	de_AT	Linux	奥地利
819	S-1	ISO8859-1	AT	43	de_AT	Sun	奥地利
1252	S-1	1252	AT	43	-	WIN	奥地利
1275	S-1	1275	AT	43	-	Mac	奥地利
37	S-1	IBM-37	AT	43	-	HOST	奥地利
1140	S-1	IBM-1140	AT	43	-	HOST	奥地利
915	S-5	ISO8859-5	BY	375	-	OS2	白俄罗斯
915	S-5	ISO8859-5	BY	375	be_BY	AIX	白俄罗斯
1131	S-5	IBM-1131	BY	375	-	OS2	白俄罗斯
1251	S-5	1251	BY	375	-	WIN	白俄罗斯
1283	S-5	1283	BY	375	-	Mac	白俄罗斯
1025	S-5	IBM-1025	BY	375	-	HOST	白俄罗斯

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
437	S-1	IBM-437	BE	32	-	OS2	比利时
850	S-1	IBM-850	BE	32	-	OS2	比利时
819	S-1	ISO8859-1	BE	32	nl_BE	AIX	比利时
850	S-1	IBM-850	BE	32	Nl_BE	AIX	比利时
819	S-1	iso88591	BE	32	-	HP	比利时
819	S-1	ISO8859-1	BE	32	fr_BE	SCO	比利时
819	S-1	ISO8859-1	BE	32	nl_BE	SCO	比利时
819	S-1	ISO-8859-1	BE	32	nl_BE	Linux	比利时
819	S-1	ISO8859-1	BE	32	nl_BE	Sun	比利时
1252	S-1	1252	BE	32	-	WIN	比利时
1275	S-1	1275	BE	32	-	Mac	比利时
500	S-1	IBM-500	BE	32	-	HOST	比利时
1148	S-1	IBM-1148	BE	32	-	HOST	比利时
855	S-5	IBM-855	BG	359	-	OS2	保加利亚
915	S-5	ISO8859-5	BG	359	-	OS2	保加利亚
915	S-5	ISO8859-5	BG	359	bg_BG	AIX	保加利亚
915	S-5	iso88595	BG	359	bg_BG.iso88595	HP	保加利亚
1251	S-5	1251	BG	359	-	WIN	保加利亚
1283	S-5	1283	BG	359	-	Mac	保加利亚
1025	S-5	IBM-1025	BG	359	-	HOST	保加利亚
850	S-1	IBM-850	BR	55	-	OS2	巴西
850	S-1	IBM-850	BR	55	-	AIX	巴西
819	S-1	ISO8859-1	BR	55	pt_BR	AIX	巴西
819	S-1	ISO8859-1	BR	55	-	HP	巴西
819	S-1	ISO8859-1	BR	55	pt_BR	SCO	巴西
819	S-1	ISO8859-1	BR	55	pt_BR	Sun	巴西
819	S-1	ISO-8859-1	BR	55	pt_BR	Linux	巴西
1252	S-1	1252	BR	55	-	WIN	巴西
37	S-1	IBM-37	BR	55	-	HOST	巴西
1140	S-1	IBM-1140	BR	55	-	HOST	巴西
850	S-1	IBM-850	CA	1	-	OS2	加拿大
850	S-1	IBM-850	CA	1	En_CA	AIX	加拿大
819	S-1	ISO8859-1	CA	1	en_CA	AIX	加拿大
819	S-1	iso88591	CA	1	fr_CA.iso88591	HP	加拿大
1051	S-1	roman8	CA	1	fr_CA.roman8	HP	加拿大
819	S-1	ISO8859-1	CA	1	en_CA	SCO	加拿大
819	S-1	ISO8859-1	CA	1	fr_CA	SCO	加拿大
819	S-1	ISO8859-1	CA	1	en_CA	Sun	加拿大
819	S-1	ISO8859-1	CA	1	en_CA	Sun	加拿大
819	S-1	ISO-8859-1	CA	1	en_CA	Linux	加拿大
1252	S-1	1252	CA	1	-	WIN	加拿大
1275	S-1	1275	CA	1	-	Mac	加拿大
37	S-1	IBM-37	CA	1	-	HOST	加拿大
1140	S-1	IBM-1140	CA	1	-	HOST	加拿大
863	S-1	IBM-863	CA	2	-	OS2	加拿大 (法语)

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
1381	D-4	IBM-1381	CN	86	-	OS2	中国 (PRC)
1386	D-4	GBK	CN	86	-	OS2	中国 (PRC)
1383	D-4	IBM-eucCN	CN	86	zh_CN	AIX	中国 (PRC)
1386	D-4	GBK	CN	86	Zh_CN.GBK	AIX	中国 (PRC)
1383	D-4	hp15CN	CN	86	zh_CN.hp15CN	HP	中国 (PRC)
1383	D-4	eucCN	CN	86	zh_CN	SCO	中国 (PRC)
1383	D-4	eucCN	CN	86	zh_CN.eucCN	SCO	中国 (PRC)
1383	D-4	gb2312	CN	86	zh	Sun	中国 (PRC)
1381	D-4	IBM-1381	CN	86	-	WIN	中国 (PRC)
1386	D-4	GBK	CN	86	-	WIN	中国 (PRC)
935	D-4	IBM-935	CN	86	-	HOST	中国 (PRC)
1388	D-4	IBM-1388	CN	86	-	HOST	中国 (PRC)
852	S-2	IBM-852	HR	385	-	OS2	克罗地亚
912	S-2	ISO8859-2	HR	385	hr_HR	AIX	克罗地亚
912	S-2	iso88592	HR	385	hr_HR.iso88592	HP	克罗地亚
912	S-2	ISO8859-2	HR	385	hr_HR.ISO8859-2	SCO	克罗地亚
912	S-2	ISO-8859-2	HR	385	hr_HR	Linux	克罗地亚
1250	S-2	1250	HR	385	-	WIN	克罗地亚
1282	S-2	1282	HR	385	-	Mac	克罗地亚
870	S-2	IBM-870	HR	385	-	HOST	克罗地亚
852	S-2	IBM-852	CZ	421	-	OS2	捷克共和国
912	S-2	ISO8859-2	CZ	421	cs_CZ	AIX	捷克共和国
912	S-2	iso88592	CZ	421	cs_CZ.iso88592	HP	捷克共和国
912	S-2	ISO8859-2	CZ	421	cs_CZ.ISO8859-2	SCO	捷克共和国
912	S-2	ISO-8859-2	CZ	421	cs_CZ	Linux	捷克共和国
1250	S-2	1250	CZ	421	-	WIN	捷克共和国
1282	S-2	1282	CZ	421	-	Mac	捷克共和国
870	S-2	IBM-870	CZ	421	-	HOST	捷克共和国
850	S-1	IBM-850	DK	45	-	OS2	丹麦
819	S-1	ISO8859-1	DK	45	da_DK	AIX	丹麦
850	S-1	IBM-850	DK	45	Da_DK	AIX	丹麦
819	S-1	iso88591	DK	45	da_DK.iso88591	HP	丹麦
1051	S-1	roman8	DK	45	da_DK.roman8	HP	丹麦
819	S-1	ISO8859-1	DK	45	da	SCO	丹麦
819	S-1	ISO8859-1	DK	45	da_DA	SCO	丹麦
819	S-1	ISO8859-1	DK	45	da_DK	SCO	丹麦
819	S-1	ISO8859-1	DK	45	da	Sun	丹麦
819	S-1	ISO8859-1	DK	45	da	Sun	丹麦
819	S-1	ISO-8859-1	DK	45	da_DK	Linux	丹麦
1252	S-1	1252	DK	45	-	WIN	丹麦
1275	S-1	1275	DK	45	-	Mac	丹麦
277	S-1	IBM-277	DK	45	-	HOST	丹麦
1142	S-1	IBM-1142	DK	45	-	HOST	丹麦

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
922	S-10	IBM-922	EE	372	-	OS2	爱沙尼亚
922	S-10	IBM-922	EE	372	Et_EE	AIX	爱沙尼亚
922	S-10	IBM-922	EE	372	-	WIN	爱沙尼亚
1122	S-10	IBM-1122	EE	372	-	HOST	爱沙尼亚
437	S-1	IBM-437	FI	358	-	OS2	芬兰
850	S-1	IBM-850	FI	358	-	OS2	芬兰
819	S-1	ISO8859-1	FI	358	fi_FI	AIX	芬兰
850	S-1	IBM-850	FI	358	Fi_FI	AIX	芬兰
819	S-1	iso88591	FI	358	fi_FI.iso88591	HP	芬兰
819	S-1	ISO8859-1	FI	358	fi	SCO	芬兰
819	S-1	ISO8859-1	FI	358	fi_FI	SCO	芬兰
819	S-1	ISO8859-1	FI	358	sv_FI	SCO	芬兰
819	S-1	ISO8859-1	FI	358	-	Sun	芬兰
819	S-1	ISO-8859-1	FI	358	fi_FI	Linux	芬兰
1051	S-1	roman8	FI	358	-	HP	芬兰
1252	S-1	1252	FI	358	-	WIN	芬兰
1275	S-1	1275	FI	358	-	Mac	芬兰
278	S-1	IBM-278	FI	358	-	HOST	芬兰
1143	S-1	IBM-1143	FI	358	-	HOST	芬兰
855	S-5	IBM-855	MK	389	-	OS2	FYR 马其顿
915	S-5	ISO8859-5	MK	389	-	OS2	FYR 马其顿
915	S-5	ISO8859-5	MK	389	mk_MK	AIX	FYR 马其顿
915	S-5	iso88595	MK	389	-	HP	FYR 马其顿
1251	S-5	1251	MK	389	-	WIN	FYR 马其顿
1283	S-5	1283	MK	389	-	Mac	FYR 马其顿
1025	S-5	IBM-1025	MK	389	-	HOST	FYR 马其顿
437	S-1	IBM-437	FR	33	-	OS2	法国
850	S-1	IBM-850	FR	33	-	OS2	法国
819	S-1	ISO8859-1	FR	33	fr_FR	AIX	法国
850	S-1	IBM-850	FR	33	Fr_FR	AIX	法国
819	S-1	iso88591	FR	33	fr_FR.iso88591	HP	法国
1051	S-1	roman8	FR	33	fr_FR.roman8	HP	法国
819	S-1	ISO8859-1	FR	33	fr	Sun	法国
819	S-1	ISO8859-1	FR	33	fr	SCO	法国
819	S-1	ISO8859-1	FR	33	fr_FR	SCO	法国
819	S-1	ISO-8859-1	FR	33	fr_FR	Linux	法国
1252	S-1	1252	FR	33	-	WIN	法国
1275	S-1	1275	FR	33	-	Mac	法国
297	S-1	IBM-297	FR	33	-	HOST	法国
1147	S-1	IBM-1147	FR	33	-	HOST	法国

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
437	S-1	IBM-437	DE	49	-	OS2	德国
850	S-1	IBM-850	DE	49	-	OS2	德国
819	S-1	ISO8859-1	DE	49	de_DE	AIX	德国
850	S-1	IBM-850	DE	49	De_DE	AIX	德国
819	S-1	iso88591	DE	49	de_DE.iso88591	HP	德国
1051	S-1	roman8	DE	49	de_DE.roman8	HP	德国
819	S-1	ISO8859-1	DE	49	de	SCO	德国
819	S-1	ISO8859-1	DE	49	de_DE	SCO	德国
819	S-1	ISO8859-1	DE	49	de	Sun	德国
819	S-1	ISO-8859-1	DE	49	de_DE	Linux	德国
1252	S-1	1252	DE	49	-	WIN	德国
1275	S-1	1275	DE	49	-	Mac	德国
273	S-1	IBM-273	DE	49	-	HOST	德国
1141	S-1	IBM-1141	DE	49	-	HOST	德国
819	S-1	ISO8859-1	DE	49	De_DE.88591	SINIX	德国
819	S-1	ISO8859-1	DE	49	De_DE.6937	SINIX	德国
813	S-7	ISO8859-7	GR	30	-	OS2	希腊
869	S-7	IBM-869	GR	30	-	OS2	希腊
813	S-7	ISO8859-7	GR	30	e1_GR	AIX	希腊
813	S-7	iso88597	GR	30	e1_GR.iso88597	HP	希腊
813	S-7	ISO8859-7	GR	30	e1_GR.ISO8859-7	SCO	希腊
813	S-7	ISO-8859-7	GR	30	gr_GR	Linux	希腊
737	S-7	737	GR	30	-	WIN	希腊
1253	S-7	1253	GR	30	-	WIN	希腊
1280	S-7	1280	GR	30	-	Mac	希腊
423	S-7	IBM-423	GR	30	-	HOST	希腊
875	S-7	IBM-875	GR	30	-	HOST	希腊
852	S-2	IBM-852	HU	36	-	OS2	匈牙利
912	S-2	ISO8859-2	HU	36	hu_HU	AIX	匈牙利
912	S-2	iso88592	HU	36	hu_HU.iso88592	HP	匈牙利
912	S-2	ISO8859-2	HU	36	hu_HU.ISO8859-2	SCO	匈牙利
912	S-2	ISO-8859-2	HU	36	hu_HU	Linux	匈牙利
1250	S-2	1250	HU	36	-	WIN	匈牙利
1282	S-2	1282	HU	36	-	Mac	匈牙利
870	S-2	IBM-870	HU	36	-	HOST	匈牙利

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
850	S-1	IBM-850	IS	354	-	OS2	冰岛
819	S-1	ISO8859-1	IS	354	is_IS	AIX	冰岛
850	S-1	IBM-850	IS	354	Is_IS	AIX	冰岛
819	S-1	iso88591	IS	354	is_IS.iso88591	HP	冰岛
1051	S-1	roman8	IS	354	is_IS.roman8	HP	冰岛
819	S-1	ISO8859-1	IS	354	is	SCO	冰岛
819	S-1	ISO8859-1	IS	354	is_IS	SCO	冰岛
819	S-1	ISO8859-1	IS	354	-	Sun	冰岛
819	S-1	ISO-8859-1	IS	354	is_IS	Linux	冰岛
1252	S-1	1252	IS	354	-	WIN	冰岛
1275	S-1	1275	IS	354	-	Mac	冰岛
871	S-1	IBM-871	IS	354	-	HOST	冰岛
1149	S-1	IBM-1149	IS	354	-	HOST	冰岛
437	S-1	IBM-437	IE	353	-	OS2	爱尔兰
850	S-1	IBM-850	IE	353	-	OS2	爱尔兰
819	S-1	ISO8859-1	IE	353	en_IE	AIX	爱尔兰
850	S-1	IBM-850	IE	353	En_IE	AIX	爱尔兰
819	S-1	iso88591	IE	353	-	HP	爱尔兰
1051	S-1	roman8	IE	353	-	HP	爱尔兰
819	S-1	ISO8859-1	IE	353	en_IE	Sun	爱尔兰
819	S-1	ISO8859-1	IE	353	en_IE.IS08859-1	SCO	爱尔兰
819	S-1	ISO-8859-1	IE	353	en_IE	Linux	爱尔兰
1252	S-1	1252	IE	353	-	WIN	爱尔兰
1275	S-1	1275	IE	353	-	Mac	爱尔兰
285	S-1	IBM-285	IE	353	-	HOST	爱尔兰
1146	S-1	IBM-1146	IE	353	-	HOST	爱尔兰
806	S-12	IBM-806	IN	91	hi_IN	-	印度
1137	S-12	IBM-1137	IN	91	-	HOST	印度
862	S-8	IBM-862	IL	972	-	OS2	以色列
916	S-8	ISO8859-8	IL	972	iw_IL	AIX	以色列
916	S-8	ISO-8859-8	IL	972	iw_IL	Linux	以色列
1255	S-8	1255	IL	972	-	WIN	以色列
424	S-8	IBM-424	IL	972	-	HOST	以色列

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
437	S-1	IBM-437	IT	39	-	OS2	意大利
850	S-1	IBM-850	IT	39	-	OS2	意大利
819	S-1	ISO8859-1	IT	39	it_IT	AIX	意大利
850	S-1	IBM-850	IT	39	It_IT	AIX	意大利
819	S-1	iso88591	IT	39	it_IT.iso88591	HP	意大利
1051	S-1	roman8	IT	39	it_IT.roman8	HP	意大利
819	S-1	ISO8859-1	IT	39	it	SCO	意大利
819	S-1	ISO8859-1	IT	39	it_IT	SCO	意大利
819	S-1	ISO8859-1	IT	39	it	Sun	意大利
819	S-1	ISO-8859-1	IT	39	it_IT	Linux	意大利
1252	S-1	1252	IT	39	-	WIN	意大利
1275	S-1	1275	IT	39	-	Mac	意大利
280	S-1	IBM-280	IT	39	-	HOST	意大利
1144	S-1	IBM-1144	IT	39	-	HOST	意大利
932	D-1	IBM-932	JP	81	-	OS2	日本
942	D-1	IBM-942	JP	81	-	OS2	日本
943	D-1	IBM-943	JP	81	-	OS2	日本
954	D-1	IBM-eucJP	JP	81	ja_JP	AIX	日本
932	D-1	IBM-932	JP	81	Ja_JP	AIX	日本
954	D-1	eucJP	JP	81	ja_JP.eucJP	HP	日本
5039	D-1	SJIS	JP	81	ja_JP.SJIS	HP	日本
954	D-1	eucJP	JP	81	ja	SCO	日本
954	D-1	eucJP	JP	81	ja_JP	SCO	日本
954	D-1	eucJP	JP	81	ja_JP.EUC	SCO	日本
954	D-1	eucJP	JP	81	ja_JP.eucJP	SCO	日本
954	D-1	eucJP	JP	81	ja	Sun	日本
954	D-1	EUC-JP	JP	81	ja_JP	Linux	日本
943	D-1	IBM-943	JP	81	-	WIN	日本
930	D-1	IBM-930	JP	81	-	HOST	日本
939	D-1	IBM-939	JP	81	-	HOST	日本
5026	D-1	IBM-5026	JP	81	-	HOST	日本
5035	D-1	IBM-5035	JP	81	-	HOST	日本
1390	D-1		JP	81	-	HOST	日本
1399	D-1		JP	81	-	HOST	日本
949	D-3	IBM-949	KR	82	-	OS2	南韩
970	D-3	IBM-eucKR	KR	82	ko_KR	AIX	南韩
970	D-3	eucKR	KR	82	ko_KR.eucKR	HP	南韩
970	D-3	eucKR	KR	82	ko_KR.eucKR	SGI	南韩
970	D-3	5601	KR	82	ko	Sun	南韩
1363	D-3	1363	KR	82	-	WIN	南韩
933	D-3	IBM-933	KR	82	-	HOST	南韩
1364	D-3	IBM-1364	KR	82	-	HOST	南韩

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
437	S-1	IBM-437	Lat	3	-	OS2	拉丁美洲
850	S-1	IBM-850	Lat	3	-	OS2	拉丁美洲
819	S-1	ISO8859-1	Lat	3	-	AIX	拉丁美洲
850	S-1	IBM-850	Lat	3	-	AIX	拉丁美洲
819	S-1	iso88591	Lat	3	-	HP	拉丁美洲
819	S-1	ISO8859-1	Lat	3	-	Sun	拉丁美洲
819	S-1	ISO-8859-1	Lat	3	-	Linux	拉丁美洲
1051	S-1	roman8	Lat	3	-	HP	拉丁美洲
1252	S-1	1252	Lat	3	-	WIN	拉丁美洲
1275	S-1	1275	Lat	3	-	Mac	拉丁美洲
284	S-1	IBM-284	Lat	3	-	HOST	拉丁美洲
1145	S-1	IBM-1145	Lat	3	-	HOST	拉丁美洲
<hr/>							
921	S-10	IBM-921	LV	371	-	OS2	拉脱维亚
921	S-10	IBM-921	LV	371	Lv_LV	AIX	拉脱维亚
921	S-10	IBM-921	LV	371	-	WIN	拉脱维亚
1112	S-10	IBM-1112	LV	371	-	HOST	拉脱维亚
<hr/>							
921	S-10	IBM-921	LT	370	-	OS2	立陶宛
921	S-10	IBM-921	LT	370	Lt_LT	AIX	立陶宛
921	S-10	IBM-921	LV	370	-	WIN	立陶宛
1112	S-10	IBM-1112	LV	370	-	HOST	立陶宛
<hr/>							
437	S-1	IBM-437	NL	31	-	OS2	荷兰
850	S-1	IBM-850	NL	31	-	OS2	荷兰
819	S-1	ISO8859-1	NL	31	n1_NL	AIX	荷兰
850	S-1	IBM-850	NL	31	N1_NL	AIX	荷兰
819	S-1	iso88591	NL	31	n1_NL.iso88591	HP	荷兰
1051	S-1	roman8	NL	31	n1_NL.roman8	HP	荷兰
819	S-1	ISO8859-1	NL	31	n1	SCO	荷兰
819	S-1	ISO8859-1	NL	31	n1_NL	SCO	荷兰
819	S-1	ISO8859-1	NL	31	n1	Sun	荷兰
819	S-1	ISO-8859-1	NL	31	n1_NL	Linux	荷兰
1252	S-1	1252	NL	31	-	WIN	荷兰
1275	S-1	1275	NL	31	-	Mac	荷兰
37	S-1	IBM-37	NL	31	-	HOST	荷兰
1140	S-1	IBM-1140	NL	31	-	HOST	荷兰
<hr/>							
850	S-1	IBM-850	NZ	64	-	OS2	新西兰
850	S-1	IBM-850	NZ	64	En_NZ	AIX	新西兰
819	S-1	ISO8859-1	NZ	64	en_NZ	AIX	新西兰
819	S-1	ISO8859-1	NZ	64	-	HP	新西兰
819	S-1	ISO8859-1	NZ	64	en_NZ	SCO	新西兰
819	S-1	ISO8859-1	NZ	64	en_NZ	Sun	新西兰
1252	S-1	1252	NZ	64	-	WIN	新西兰
37	S-1	IBM-37	NZ	64	-	HOST	新西兰
1140	S-1	IBM-1140	NZ	64	-	HOST	新西兰

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
850	S-1	IBM-850	NO	47	-	OS2	挪威
819	S-1	ISO8859-1	NO	47	no_NO	AIX	挪威
850	S-1	IBM-850	NO	47	No_NO	AIX	挪威
819	S-1	iso88591	NO	47	no_NO.iso88591	HP	挪威
1051	S-1	roman8	NO	47	no_NO.roman8	HP	挪威
819	S-1	ISO8859-1	NO	47	no	SCO	挪威
819	S-1	ISO8859-1	NO	47	no_NO	SCO	挪威
819	S-1	ISO8859-1	NO	47	no	Sun	挪威
819	S-1	ISO-8859-1	NO	47	no_NO	Linux	挪威
1252	S-1	1252	NO	47	-	WIN	挪威
1275	S-1	1275	NO	47	-	Mac	挪威
277	S-1	IBM-277	NO	47	-	HOST	挪威
1142	S-1	IBM-1142	NO	47	-	HOST	挪威
852	S-2	IBM-852	PL	48	-	OS2	波兰
912	S-2	ISO8859-2	PL	48	pl_PL	AIX	波兰
912	S-2	iso88592	PL	48	pl_PL.iso88592	HP	波兰
912	S-2	ISO8859-2	PL	48	pl_PL.ISO8859-2	SCO	波兰
912	S-2	ISO-8859-2	PL	48	pl_PL	Linux	波兰
1250	S-2	1250	PL	48	-	WIN	波兰
1282	S-2	1282	PL	48	-	Mac	波兰
870	S-2	IBM-870	PL	48	-	HOST	波兰
860	S-1	IBM-860	PT	351	-	OS2	葡萄牙
850	S-1	IBM-850	PT	351	-	OS2	葡萄牙
819	S-1	ISO8859-1	PT	351	pt_PT	AIX	葡萄牙
850	S-1	IBM-850	PT	351	Pt_PT	AIX	葡萄牙
819	S-1	iso88591	PT	351	pt_PT.iso88591	HP	葡萄牙
1051	S-1	roman8	PT	351	pt_PT.roman8	HP	葡萄牙
819	S-1	ISO8859-1	PT	351	pt	SCO	葡萄牙
819	S-1	ISO8859-1	PT	351	pt_PT	SCO	葡萄牙
819	S-1	ISO8859-1	PT	351	pt	Sun	葡萄牙
819	S-1	ISO-8859-1	PT	351	pt_PT	Linux	葡萄牙
1252	S-1	1252	PT	351	-	WIN	葡萄牙
1275	S-1	1275	PT	351	-	Mac	葡萄牙
37	S-1	IBM-37	PT	351	-	HOST	葡萄牙
1140	S-1	IBM-1140	PT	351	-	HOST	葡萄牙
852	S-2	IBM-852	RO	40	-	OS2	罗马尼亚
912	S-2	ISO8859-2	RO	40	ro_RO	AIX	罗马尼亚
912	S-2	iso88592	RO	40	ro_RO.iso88592	HP	罗马尼亚
912	S-2	ISO8859-2	RO	40	ro_RO.ISO8859-2	SCO	罗马尼亚
912	S-2	ISO-8859-2	RO	40	ro_RO	Linux	罗马尼亚
1250	S-2	1250	RO	40	-	WIN	罗马尼亚
1282	S-2	1282	RO	40	-	Mac	罗马尼亚
870	S-2	IBM-870	RO	40	-	HOST	罗马尼亚

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
866	S-5	IBM-866	RU	7	-	OS2	俄国
915	S-5	ISO8859-5	RU	7	-	OS2	俄国
915	S-5	ISO8859-5	RU	7	ru_RU	AIX	俄国
915	S-5	iso88595	RU	7	ru_RU.iso88595	HP	俄国
915	S-5	ISO8859-5	RU	7	ru_RU.IS08859-5	SCO	俄国
915	S-5	ISO-8859-5	RU	7	ru_RU	Linux	俄国
1251	S-5	1251	RU	7	-	WIN	俄国
1283	S-5	1283	RU	7	-	Mac	俄国
1025	S-5	IBM-1025	RU	7	-	HOST	俄国
<hr/>							
855	S-5	IBM-855	SP	381	-	OS2	塞尔维亚 / 蒙的内哥罗
915	S-5	ISO8859-5	SP	381	-	OS2	塞尔维亚 / 蒙的内哥罗
915	S-5	ISO8859-5	SP	381	sr_SP	AIX	塞尔维亚 / 蒙的内哥罗
915	S-5	iso88595	SP	381	-	HP	塞尔维亚 / 蒙的内哥罗
1251	S-5	1251	SP	381	-	WIN	塞尔维亚 / 蒙的内哥罗
1283	S-5	1283	SP	381	-	Mac	塞尔维亚 / 蒙的内哥罗
1025	S-5	IBM-1025	SP	381	-	HOST	塞尔维亚 / 蒙的内哥罗
<hr/>							
852	S-2	IBM-852	SK	422	-	OS2	斯洛伐克
912	S-2	ISO8859-2	SK	422	sk_SK	AIX	斯洛伐克
912	S-2	iso88592	SK	422	sk_SK.iso88592	HP	斯洛伐克
912	S-2	ISO8859-2	SK	422	sk_SK.IS08859-2	SCO	斯洛伐克
1250	S-2	1250	SK	422	-	WIN	斯洛伐克
1282	S-2	1282	SK	422	-	Mac	斯洛伐克
870	S-2	IBM-870	SK	422	-	HOST	斯洛伐克
<hr/>							
852	S-2	IBM-852	SI	386	-	OS2	斯洛文尼亚
912	S-2	ISO8859-2	SI	386	sl_SI	AIX	斯洛文尼亚
912	S-2	iso88592	SI	386	sl_SI.iso88592	HP	斯洛文尼亚
912	S-2	ISO8859-2	SI	386	sl_SI.IS08859-2	SCO	斯洛文尼亚
912	S-2	ISO-8859-2	SI	386	sl_SI	Linux	斯洛文尼亚
1250	S-2	1250	SI	386	-	WIN	斯洛文尼亚
1282	S-2	1282	SI	386	-	Mac	斯洛文尼亚
870	S-2	IBM-870	SI	386	-	HOST	斯洛文尼亚
<hr/>							
437	S-1	IBM-437	ZA	27	-	OS2	南非
850	S-1	IBM-850	ZA	27	-	OS2	南非
819	S-1	ISO8859-1	ZA	27	en_ZA	AIX	南非
850	S-1	IBM-850	ZA	27	En_ZA	AIX	南非
819	S-1	iso88591	ZA	27	-	HP	南非
1051	S-1	roman8	ZA	27	-	HP	南非
819	S-1	ISO8859-1	ZA	27	-	Sun	南非
819	S-1	ISO8859-1	ZA	27	en_ZA.IS08859-1	SCO	南非
1252	S-1	1252	ZA	27	-	WIN	南非
1275	S-1	1275	ZA	27	-	Mac	南非
285	S-1	IBM-285	ZA	27	-	HOST	南非
1146	S-1	IBM-1146	ZA	27	-	HOST	南非

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
437	S-1	IBM-437	ES	34	-	OS2	西班牙
850	S-1	IBM-850	ES	34	-	OS2	西班牙
819	S-1	ISO8859-1	ES	34	es_ES	AIX	西班牙
850	S-1	IBM-850	ES	34	Es_ES	AIX	西班牙
819	S-1	iso88591	ES	34	es_ES.iso88591	HP	西班牙
1051	S-1	roman8	ES	34	es_ES.roman8	HP	西班牙
819	S-1	ISO8859-1	ES	34	es	Sun	西班牙
819	S-1	ISO8859-1	ES	34	es	SCO	西班牙
819	S-1	ISO8859-1	ES	34	es_ES	SCO	西班牙
819	S-1	ISO-8859-1	ES	34	es_ES	Linux	西班牙
1252	S-1	1252	ES	34	-	WIN	西班牙
1275	S-1	1275	ES	34	-	Mac	西班牙
284	S-1	IBM-284	ES	34	-	HOST	西班牙
1145	S-1	IBM-1145	ES	34	-	HOST	西班牙
437	S-1	IBM-437	SE	46	-	OS2	瑞典
850	S-1	IBM-850	SE	46	-	OS2	瑞典
819	S-1	ISO8859-1	SE	46	sv_SE	AIX	瑞典
850	S-1	IBM-850	SE	46	Sv_SE	AIX	瑞典
819	S-1	iso88591	SE	46	sv_SE.iso88591	HP	瑞典
1051	S-1	roman8	SE	46	sv_SE.roman8	HP	瑞典
819	S-1	ISO8859-1	SE	46	sv	SCO	瑞典
819	S-1	ISO8859-1	SE	46	sv_SE	SCO	瑞典
819	S-1	ISO8859-1	SE	46	sv	Sun	瑞典
819	S-1	ISO-8859-1	SE	46	sv_SE	Linux	瑞典
1252	S-1	1252	SE	46	-	WIN	瑞典
1275	S-1	1275	SE	46	-	Mac	瑞典
278	S-1	IBM-278	SE	46	-	HOST	瑞典
1143	S-1	IBM-1143	SE	46	-	HOST	瑞典
437	S-1	IBM-437	CH	41	-	OS2	瑞士
850	S-1	IBM-850	CH	41	-	OS2	瑞士
819	S-1	ISO8859-1	CH	41	de_CH	AIX	瑞士
850	S-1	IBM-850	CH	41	De_CH	AIX	瑞士
819	S-1	iso88591	CH	41	-	HP	瑞士
1051	S-1	roman8	CH	41	-	HP	瑞士
819	S-1	ISO8859-1	CH	41	de_CH	SCO	瑞士
819	S-1	ISO8859-1	CH	41	fr_CH	SCO	瑞士
819	S-1	ISO8859-1	CH	41	it_CH	SCO	瑞士
819	S-1	ISO8859-1	CH	41	de_CH	Sun	瑞士
819	S-1	ISO-8859-1	CH	41	de_CH	Linux	瑞士
1252	S-1	1252	CH	41	-	WIN	瑞士
1275	S-1	1275	CH	41	-	Mac	瑞士
500	S-1	IBM-500	CH	41	-	HOST	瑞士
1148	S-1	IBM-1148	CH	41	-	HOST	瑞士

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
938	D-2	IBM-938	TW	88	-	OS2	台湾
948	D-2	IBM-948	TW	88	-	OS2	台湾
950	D-2	big5	TW	88	-	OS2	台湾
950	D-2	big5	TW	88	Zh_TW	AIX	台湾
964	D-2	IBM-eucTW	TW	88	zh_TW	AIX	台湾
950	D-2	big5	TW	88	zh_TW.big5	HP	台湾
964	D-2	eucTW	TW	88	zh_TW.eucTW	HP	台湾
950	D-2	big5	TW	88	big5	Sun	台湾
964	D-2	cns11643	TW	88	zh_TW	Sun	台湾
950	D-2	big5	TW	88	-	WIN	台湾
937	D-2	IBM-937	TW	88	-	HOST	台湾
874	S-20	TIS620-1	TH	66	-	OS2	泰国
874	S-20	TIS620-1	TH	66	Th_TH	AIX	泰国
874	S-20	tis620	TH	66	th_TH.tis620	HP	泰国
874	S-20	TIS620-1	TH	66	-	WIN	泰国
838	S-20	IBM-838	TH	66	-	HOST	泰国
857	S-9	IBM-857	TR	90	-	OS2	土耳其
920	S-9	ISO8859-9	TR	90	tr_TR	AIX	土耳其
920	S-9	iso88599	TR	90	tr_TR.iso88599	HP	土耳其
920	S-9	ISO8859-9	TR	90	tr_TR.ISO8859-9	SCO	土耳其
920	S-9	ISO-8859-9	TR	90	tr_TR	Linux	土耳其
1254	S-9	1254	TR	90	-	WIN	土耳其
1281	S-9	1281	TR	90	-	Mac	土耳其
1026	S-9	IBM-1026	TR	90	-	HOST	土耳其
437	S-1	IBM-437	GB	44	-	OS2	英国
850	S-1	IBM-850	GB	44	-	OS2	英国
819	S-1	ISO8859-1	GB	44	en_GB	AIX	英国
850	S-1	IBM-850	GB	44	En_GB	AIX	英国
819	S-1	iso88591	GB	44	en_GB.iso88591	HP	英国
1051	S-1	roman8	GB	44	en_GB.roman8	HP	英国
819	S-1	ISO8859-1	GB	44	en_UK	Sun	英国
819	S-1	ISO8859-1	GB	44	en_GB	SCO	英国
819	S-1	ISO8859-1	GB	44	en	SCO	英国
819	S-1	ISO-8859-1	GB	44	en_GB	Linux	英国
1252	S-1	1252	GB	44	-	WIN	英国
1275	S-1	1275	GB	44	-	Mac	英国
285	S-1	IBM-285	GB	44	-	HOST	英国
1146	S-1	IBM-1146	GB	44	-	HOST	英国
819	S-1	88591	GB	44	En_GB.88591	SINIX	英国
819	S-1	ISO8859-1	GB	44	En_GB.6937	SINIX	英国
1125	S-5	IBM-1125	UA	380	-	OS2	乌克兰
1124	S-5	IBM-1124	UA	380	uk_UA	AIX	乌克兰
1251	S-5	1251	UA	380	-	WIN	乌克兰
1123	S-5	IBM-1123	UA	380	-	HOST	乌克兰

表 32. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 代码	本国语言环境	OS	国家名
----	-----	-----	--	---	-----	----	-----
437	S-1	IBM-437	US	1	-	OS2	美国
850	S-1	IBM-850	US	1	-	OS2	美国
819	S-1	ISO8859-1	US	1	en_US	AIX	美国
850	S-1	IBM-850	US	1	En_US	AIX	美国
819	S-1	iso88591	US	1	en_US.iso88591	HP	美国
1051	S-1	roman8	US	1	en_US.roman8	HP	美国
819	S-1	ISO8859-1	US	1	en_US	Sun	美国
819	S-1	ISO8859-1	US	1	en_US	SGI	美国
819	S-1	ISO8859-1	US	1	en_US	SCO	美国
819	S-1	ISO-8859-1	US	1	en_US	Linux	美国
1252	S-1	1252	US	1	-	WIN	美国
1275	S-1	1275	US	1	-	Mac	美国
37	S-1	IBM-37	US	1	-	HOST	美国
1140	S-1	IBM-1140	US	1	-	HOST	美国
<hr/>							
1163	S-11	IBM-1163	VN	84	-	OS2	越南
1163	S-11	IBM-1163	VN	84	vi_VN	AIX	越南
1258	S-11	1258	VN	84	-	WIN	越南
1164	S-11	IBM-1164	VN	84	-	HOST	越南

以下映射至阿拉伯国家 (AA):

```

/* 阿拉伯语 (沙特阿拉伯) */
/* 阿拉伯语 (伊拉克) */
/* 阿拉伯语 (埃及) */
/* 阿拉伯语 (利比亚) */
/* 阿拉伯语 (阿尔及利亚) */
/* 阿拉伯语 (摩洛哥) */
/* 阿拉伯语 (突尼斯) */
/* 阿拉伯语 (阿曼) */
/* 阿拉伯语 (也门) */
/* 阿拉伯语 (叙利亚) */
/* 阿拉伯语 (约旦) */
/* 阿拉伯语 (黎巴嫩) */
/* 阿拉伯语 (科威特) */
/* 阿拉伯语 (阿拉伯联合酋长国) */
/* 阿拉伯语 (巴林) */
/* 阿拉伯语 (卡塔尔) */

```

以下映射至英语 (US):

```

/* 英语 (牙买加) */
/* 英语 (加勒比地区) */

```

表 32. 受支持的语言和代码集 (续)

代码页	组	代码集	Tr.	国家代码	本国语言环境	OS	国家名
-----	---	-----	-----	------	--------	----	-----

以下映射至拉丁美洲 (Lat):

```

/* 西班牙语 (墨西哥) */
/* 西班牙语 (危地马拉) */
/* 西班牙语 (哥斯达黎加) */
/* 西班牙语 (巴拿马) */
/* 西班牙语 (多米尼加共和国) */
/* 西班牙语 (委内瑞拉) */
/* 西班牙语 (哥伦比亚) */
/* 西班牙语 (秘鲁) */
/* 西班牙语 (阿根廷) */
/* 西班牙语 (厄瓜多尔) */
/* 西班牙语 (智利) */
/* 西班牙语 (乌拉圭) */
/* 西班牙语 (巴拉圭) */
/* 西班牙语 (玻利维亚) */

```

注: Solaris 代码页 950 不支持 IBM 950 中的下列字符:

代码范围	说明	Sun Big-5	IBM Big-5
C6A1-C8FE	符号	保留区	符号
F9D6-F9FE	ETen 扩充	保留区	ETen 扩充
F286-F9A0	IBM 选择的字符	保留区	IBM 选择的

注: DB2 UDB 的此版本提供欧元符号的支持。根据最新的 Microsoft 定义修改 Microsoft Windows ANSI 代码页以在位置 0x80 包含欧元符号。此位置先前未定义。另外, 更改代码页 850 的定义以用欧元符号来替换字符去点的 i (可在位置 0xD5 找到)。DB2 UDB 使用这些代码页的新定义为缺省值以提供欧元符号支持。此实现是给当前需要欧元符号支持的 DB2 UDB 客户的恰当的缺省值, 且不应影响其他客户。但是, 若您要继续使用这些代码页的先前的定义, 则可在安装结束后拷贝下列文件:

- 12520850.cnv
- 08501252.cnv
- IBM00850.ucs
- IBM01252.ucs

从目录

sql/lib/conv/alt/

到目录

sqllib/conv/

在安装完成之后。在拷贝非欧元版本覆盖之前，您可能希望备份当前的 IBM01252.usc 和 IBM00850.ucs 文件。拷贝文件之后，将没有 DB2 UDB 支持的欧元符号。

派生代码页值

当进行数据库连接时，应用程序代码页由活动的环境派生而来若设置了 DB2CODEPAGE 注册表变量，则将其值用作应用程序代码页。但是，不需要设置 DB2CODEPAGE 注册表变量，因为 DB2 将根据操作系统确定适当的代码页值。将 DB2CODEPAGE 注册表变量设置为不正确的值可能导致不可预测的结果。

数据库代码页则从创建数据库时指定的值（显式或缺省）派生而来例如，下面定义如何在不同的操作环境中确定活动环境:

UNIX	在基于 UNIX 的操作系统上，活动环境是根据本国语言环境设置确定的，它包含有关语言、区域和代码集的信息。
OS/2	在 OS/2 上，在 CONFIG.SYS 文件中指定主代码页和次代码页。可以使用 chcp 命令来显示和动态更改给定对话中的代码页
Macintosh	对于 Macintosh 操作系统，若未设置 DB2CODEPAGE 注册表变量，则 Macintosh 代码页由已安装脚本的“地区”版本代码派生而来。
Windows	对于 Windows 操作系统，若未设置 DB2CODEPAGE 注册表变量，则 Windows 代码页由国家 ID 派生而来，它由 Windows 的 WIN.INI 文件中 [intl] 节的 iCountry 值指定。
Windows 32 位操作系统	对于所有 Windows 32 位操作系统，若未设置 DB2CODEPAGE 注册表变量，则代码页由“注册表”中的 ANSI 代码页设置派生而得。

有关代码页值的环境映射的完整列表，参见第347页的表32。

字符集

一般而言，数据库管理程序不限制应用程序可以使用的字符集。有关受 DB2 支持的多字节字符集 (MBCS) 的详细说明，参考 *Application Development Guide*。

标识符的字符集

可以在数据库名中使用的基本字符集包括单字节大写和小写拉丁字母 (A...Z, a...z)、阿拉伯数字 (0...9) 和下划线字符 (_)。此列表增加了三个特殊字符 (#、@ 和 \$) 以提供与主机数据库产品的兼容性。但是, 在 NLS 环境中使用这些特殊字符应小心, 因为它们不包括在 NLS 主机 (EBCDIC) 不变量字符集中。

当命名数据库对象 (如表和视图)、程序标号、主变量、游标时, 也可使用扩充字符集 (例如, 带有相异标记的字母) 中的元素。到底哪些字符可用取决于所用的代码页。若是在一个多代码页环境中使用该数据库, 您必须确保所有代码页都支持您计划使用的扩充字符集中的任何元素。有关带有不属于扩充字符集的字符, 但可以在 SQL 语句中使用的定界标识符的详情, 参考 *SQL Reference*。

DBCS 标识符的扩充字符集定义

在 DBCS 环境中, 扩充字符集包含基本字符集中的所有字符以及下列各项:

- 除双字节空间外, 每个 DBCS 代码页中的所有双字节字符都是有效的字母。
- 双字节空间是特殊字符。
- 在每个混合的代码页中可用的单字节字符被分配给各种类别, 如下所示:

类别 在每个混合代码页内有效的代码点

数字 x30-39

字母 x23-24、x40-5A、x61-7A、xA6-DF (A6-DF 仅用于代码页 932 和 942)

特殊字符

所有其他有效的单字节字符代码点

编码 SQL 语句

SQL 语句的编码与语言无关。SQL 关键字可以以大写、小写或混合大小写输入。SQL 语句中数据库对象和主变量以及程序标号的名称不能包含以上所述的扩充字符集之外的字符。

双向 CCSID 支持

要正确处理不同平台上的双向数据, 下列 BiDi 属性是必需的:

- 文本类型 (LOGICAL 或 VISUAL)
- 整形 (SHAPED 或 UNSHAPED)
- 方向 (RIGHT-TO-LEFT 或 LEFT-TO-RIGHT)
- 数字形状 (ARABIC 或 HINDI)
- 对称交换 (YES 或 NO)

因为不同平台上的缺省值不一样, 因此在将 DB2 数据从一个平台移到另一个平台时会出现问题。例如, Windows 操作系统使用 LOGICAL UNSHAPED 数据, 而

OS/390 通常使用 SHAPED VISUAL 数据。因而，若对双向属性没有任何支持，则从 DB2 通用数据库 OS/390 版发送至 Windows 32 位操作系统工作站上的 DB2 UDB 的数据将不能正确显示。

双向特定的 CCSID

DB2 通过专门的双向“编码字符集标识符”(CCSID)来支持双向数据属性。下列双向 CCSID 已定义并在 DB2 UDB 中实现:

CCSID (dec)	CCSID (hex)	代码 页	字符串 类型
00420	x'01A4'	420	4
00424	x'01A8'	424	4
00612	x'21A4'	420	5
00616	x'21A8'	424	10
00856	x'0358'	856	5
00862	x'035E'	862	4
00864	x'0360'	864	5
00916	x'0394'	916	5
01046	x'0416'	1046	5
01089	x'0441'	1089	5
01255	x'04E7'	1255	5
01256	x'04E8'	1256	5
62208	x'F300'	856	4
62209	x'F301'	862	10
62210	x'F302'	916	4
62211	x'F303'	424	5
62213	x'F305'	862	5
62215	x'F307'	1255	4
62218	x'F30A'	864	4
62220	x'F30C'	856	6
62221	x'F30D'	862	6
62222	x'F30E'	916	6
62223	x'F30F'	1255	6
62224	x'F310'	420	6
62225	x'F311'	864	6
62226	x'F312'	1046	6
62227	x'F313'	1089	6
62228	x'F314'	1256	6
62229	x'F315'	424	8
62230	x'F316'	856	8
62231	x'F317'	862	8
62232	x'F318'	916	8
62233	x'F319'	420	8
62234	x'F31A'	420	9
62235	x'F31B'	424	6
62236	x'F31C'	856	10
62237	x'F31D'	1255	8
62238	x'F31E'	916	10
62239	x'F31F'	1255	10

62240	x'F320'	424	11
62241	x'F321'	856	11
62242	x'F322'	862	11
62243	x'F323'	916	11
62244	x'F324'	1255	11
62245	x'F325'	424	10
62246	x'F326'	1046	8
62247	x'F327'	1046	9
62248	x'F328'	1046	4
62249	x'F329'	1046	12
62250	x'F32A'	420	12

其中“CDRA 字符串类型”定义为:

字符串类型	文本类型	数字形状	方向	整形	对称交换
4	可视的	Passthru	LTR	已整形	OFF
5	隐式	阿拉伯	LTR	未整形	ON
6	隐式	阿拉伯	RTL	未整形	ON
7(*)	可视的	阿拉伯	上下文(*)	未整形	OFF
8	可视的	阿拉伯	RTL	已整形	OFF
9	可视的	Passthru	RTL	已整形	ON
10	隐式	Passthru	上下文-L	未整形	ON
11	隐式	Passthru	上下文-R	未整形	ON
12	隐式	阿拉伯	RTL	已整形	ON

注: (*) 当第一个字母字符为拉丁字符时, 字段方向为从左到右 (LTR); 当第一个字母字符为双向 (RTL) 字符时, 字段方向为从右到左 (RTL)。字符未整形, 但保留 LamAlef 连字, 而没有将其分开。

双向支持的 DB2 通用数据库实现

使用新的 CCSID 定义在 DB2 通用数据库中实现了双向格式转换。对于新的 BiDi 特定的 CCSID, 执行格式转换来代替代码页转换或在代码页转换之后再执行它。要使用此支持, 必须将 DB2BIDI 注册表变量设置为 YES。缺省情况下, 不设置此变量。服务器对所有转换设置此变量, 且只有在服务器启动后才能设置。因为存在附加检查和格式转换, 将 DB2BIDI 设置为 YES 可能会影响性能。

要在非 DRDA 环境中指定特定的双向 CCSID, (从上表中) 选择与您的客户机特性匹配的 CCSID, 并将 DB2CODEPAGE 设置为该值。若您已有与该数据库的一个连接, 您必须发出 TERMINATE 命令, 并再次连接以使 DB2CODEPAGE 的新设置生效。若您选择对于您的客户机平台的代码页或字符串类型不适当的 CCSID, 将得到不期望的结果。若您选择了不兼容的 CCSID (例如, 对于与阿拉伯语数据库的连接, 选择了希伯来语 CCSID), 或者若未为该服务器设置 DB2BIDI, 当您尝试连接时会收到出错信息。

对于 DRDA 环境，若 HOST EBCDIC 平台也支持这些双向的 CCSID，您只须设置 DB2CODEPAGE 值。然而，若 HOST 平台不支持这些 CCSID，则您还必须为将要连接的 HOST 数据库服务器指定一个 CCSID 替换。这是必需的，因为在 DRDA 环境中，代码页转换和格式转换由数据接收器执行。然而，若 HOST 服务器不支持这些双向 CCSID，它将不对从 DB2 UDB 接收的数据执行格式转换。若您使用 CCSID 替换，DB2 UDB 客户机一样对输出数据执行格式转换。有关设置 CCSID 替换的详情，参考 *DB2 Connect 用户指南*。

对于 HOST EBCDIC 平台是客户机，而 DB2 UDB 是服务器的情况，不支持 CCSID 替换。

双向支持的 DB2 Connect 实现

当在 DB2 Connect 和服务器上的数据库之间交换数据时，对入局数据执行转换的往往是接收器。除通常的代码页转换外，同样的约定通常也适用于双向格式转换。DB2 Connect 也能够对它准备发送到服务器数据库的数据和从服务器数据库接收到的数据执行双向格式转换。

为了 DB2 Connect 能够对服务器数据库的出局数据执行双向格式转换，必须替换服务器数据库的双向 CCSID。为此，可在服务器数据库的 DCS 数据库目录项的 PARMs 字段中使用 BIDI 参数。

注：若希望 DB2 Connect 对准备发送到 DB2 主机数据库的数据执行格式转换，即使不必替换其 CCSID，也必须将 BIDI 参数添加到 DCS 数据库目录的 PARMs 字段中。在这种情况下，应提供的 CCSID 是 DB2 主机数据库的缺省 CCSID。

将 BIDI 参数指定为 PARMs 字段中的第九个参数，同时指定双向 CCSID，希望替换缺省的服务器数据库双向 CCSID：

```
" , , , , , , , BIDI=xyz "
```

其中 xyz 是 CCSID 替换值

注：要使 BIDI 参数生效，必须将注册表变量 DB2BIDI 设置为 YES。

有关受支持的双向 CCSID 的列表和它们的字符串类型，可在第364页的『双向特定的 CCSID』中找到。

为了最好地描述如何使用此功能部件，举例如下。

假设您有一个运行 CCSID 62213（双向字符串类型 5）的希伯来语 DB2 客户机，而您又想存取运行 CCSID 00424（双向字符串类型 4）的 DB2 主机数据库。但您知道包含在 DB2 主机数据库中的数据是基于 CCSID 08616（双向字符串类型 6）。

这样，就出现两个问题：第一个问题是，DB2 主机数据库不能区分 CCSID 00424 和 08616 的双向字符串类型。第二个问题是，DB2 主机数据库不能识别 DB2 客户机的 CCSID (62213)。它只支持 CCSID 00862，它与 CCSID 62213 基于相同的代码页。

需要确保发送到 DB2 主机数据库的数据以双向字符串类型 6 格式开始，并且要让 DB2 Connect 知道它必须对从 DB2 主机数据库接收到的数据执行双向转换。需要对 DB2 主机数据库使用以下编目命令：

```
db2 catalog dcs database nydb1 as telaviv parms ",,,,,,,BIDI=08616"
```

此命令告诉 DB2 Connect 用 CCSID 08616 取代 DB2 主机数据库的 CCSID 00424。此取代包括下列处理：

1. 将 DB2 Connect 连接到使用 CCSID 00862 的 DB2 主机数据库。
2. DB2 Connect 对它准备发送给 DB2 主机数据库的数据执行双向格式转换即从 CCSID 62213（双向字符串类型 5）转换为 CCSID 62221（双向字符串类型 6）。
3. DB2 Connect 对它从 DB2 主机数据库接收到的数据执行双向格式转换。即从 CCSID 08616（双向字符串类型 6）转换为 CCSID 62213（双向字符串类型 5）。

注：有些情况下，使用双向 CCSID 可能造成 SQL 查询本身被修改，结果使 DB2 服务器不能识别该 SQL 查询。特别是，当可以使用另一种字符串类型时，应避免使用 IMPLICIT CONTEXTUAL CCSID 和 IMPLICIT RIGHT-TO-LEFT CCSID。若 SQL 查询包括引用字符串，则 CONTEXTUAL CCSID 可能产生不可预测的结果。避免在 SQL 语句中使用引用字符串；尽可能使用主变量。

若某个特定的双向 CCSID 导致不能按照下列建议改正的问题，则将 DB2BIDI 设置为 NO。

整理顺序

数据库管理程序使用整理顺序来比较字符数据这是对一组字符的排序，确定某个字符的 ASCII 码是大于、小于还是等于另一个字符的 ASCII 码。例如，可使用整理顺序来指示特定字符的小写与大写的 ASCII 码相等。

整理顺序在创建数据库时指定，且以后不能修改。

数据库管理程序允许使用应用程序接口 (API) 来通过定制整理顺序创建数据库。有关实现定制整理顺序表的详情, 参考 *Application Development Guide*。

注: 用 FOR BIT DATA 属性定义的字符串数据和 BLOB 数据使用二进制排序顺序进行排序。

一般注意事项

一旦定义了整理顺序, 就将用该整理顺序执行该数据库将来的所有字符比较。除了被定义为 FOR BIT DATA 或 BLOB 数据的字符数据以外, 整理顺序将被用于所有的 SQL 比较和 ORDER BY 子句, 以及建立索引和统计。要获取关于如何使用数据库整理顺序的详情, 参见 *SQL Reference* 中的“字符串比较”。

在下列情况下可能会出现潜在的问题:

- 应用程序将数据库中已排序的数据与应用程序的数据合并在一起, 应用程序数据是使用不同的整理顺序排序的。
- 应用程序将一个数据库中已排序的数据与另一个数据库中已排序的数据合并在一起, 但这两个数据库的整理顺序不同。
- 应用程序对排序的数据所作的假定不符合相关的整理顺序。例如, 数字排在字母之前对特定的整理顺序可能适用也可能不适用。

最后一点要记住的是, 对字符代码点进行直接比较所得到的任何排序的结果将只与使用等同的整理顺序排序的查询结果匹配。

联合体数据库注意事项

选择的数据库整理顺序可影响联合体系统性能。若数据源与 DB2 联合体数据库使用相同的整理顺序, DB2 可将涉及到字符数据的与次序有关的处理下推到数据源。若数据源整理顺序与 DB2 的整理顺序不匹配, 检索数据并在本地执行涉及到字符数据的与次序有关的处理 (这可能降低性能)。

要确定数据源和 DB2 是否有相同的整理顺序, 考虑下列因素:

- 国家语言支持。
整理顺序与服务器的受支持的语言相关。比较 DB2 NLS 信息和数据源 NLS 信息。
- 数据源特性。
某些数据源是用不区分大小写的整理顺序创建的, 在与次序相关的操作中这可能产生不同的结果。
- 定制。
某些数据源为整理顺序提供多个选项, 或允许定制整理顺序。

为一个 DB2 联合体数据库选择整理顺序，以将从该数据库存取的混合数据源为根据。例如：

- 若 DB2 数据库主要存取与 DB2 有相同代码页 (NLS) 的 Oracle 数据库，则在数据库创建时指定标识顺序 (Oracle 数据库使用一个等效的整理顺序)。
- 若 DB2 数据库只存取 DB2 UDB 数据库，确保与整理顺序值匹配。

有关设置 MVS 整理顺序的详情，参考 *Administrative API Reference*，您可以在 **sqlcrea** (创建数据库 API) 的说明下面找到有关示例。这些样本包含 EBCDIC 500、37 和 5026/5035 代码页的整理表。

设置 DB2 数据库的整理顺序之后，确保为每个数据源服务器设置 *collating_sequence* 服务器选项。此选项指定给定数据源服务器的整理顺序是否与 DB2 数据库的整理顺序匹配。

若整理顺序匹配，则将 *collating_sequence* 选项设置为 "Y"。此设置可使 DB2 优化器考虑在数据源进行与次序有关的处理，这可改进性能。但是，若数据源整理顺序与 DB2 数据库整理顺序不同，将收到不正确的结果。例如，若您的计划使用合并连接，DB2 优化器将把尽可能多的排序操作下推至数据源。若数据源整理顺序不同，则连接结果集可能不正确。

若整理顺序不匹配，则将 *collating_sequence* 选项设置为 "N"。当数据源整理顺序与 DB2 不同时，或当数据源整理操作可能不区分大小写时，使用此值。例如，在有英语代码页的不区分大小写的数据源中，将认为 TOLLESON、ToLLeSoN 和 tolleson 全部相同。若无法确定数据源的整理顺序与 DB2 整理顺序是否完全相同，可将 *collating_sequence* 选项设置为 "N"。

日期时间值

日期时间数据类型描述如下。尽管日期时间值可以在算术和字符串运算中使用，且与特定的字符串兼容，但它们既不是字符串也不是数字。

日期

日期值分为三个部分 (年、月以及日) 年部分的范围为 0001 至 9999。月部分的范围为 1 至 12。日部分的范围为 1 至 *x*，其中 *x* 取决于月。

日期的内部表示是 4 字节的字符串。每个字节由 2 个压缩十进制数字组成。前两个字节表示年，第三个字节表示月，最后一个字节表示日。

DATE 列的长度，如 SQLDA 中所描述的，是 10 字节，它是该值的字符串表示的适当长度。

时间

*时间*是一个三部分值（小时、分钟和秒），指定在 24 小时时钟下一天中的时间小时部分的范围是 0 至 24。其他部分的范围是 0 至 59。若小时是 24，则分钟和秒的说明为零。

时间的内部表示是 3 字节的字符串。每个字节由 2 个压缩十进制数字组成。第一个字节表示小时，第二个字节表示分钟，最后一个字节表示秒。

TIME 列的长度，如 SQLDA 中所描述的，是 8 字节，它是该值的字符串表示的适当长度。

时间戳记

*时间戳记*是一个七部分值（年、月、日、小时、分钟、秒和微秒），它指定以上定义的日期和时间，但不同的是这个时间包括微秒的说明。

时间戳记的内部表示是 10 字节的字符串。每个字节由 2 个压缩十进制数字组成。前 4 个字节表示日期，接着的 3 个字节表示时间，最后 3 个字节表示微秒。

TIMESTAMP 列的长度，如 SQLDA 中所描述的，是 26 字节，它是该值的字符串表示的适当长度。

日期时间值的字符串表示

其数据类型是 DATE、TIME 或 TIMESTAMP 的值是用对于 SQL 用户透明的一种内部形式表示的。但是，日期、时间和时间戳记也可用字符串表示，而这些表示直接涉及到 SQL 用户，因为没有其数据类型是 DATE、TIME 或 TIMESTAMP 的常数或变量。因此，要检索，必须将一个日期时间值分配给一个字符串变量。除非在该程序预编译或与数据库联编时被 "F" 格式的选项替换，否则，字符串表示通常是与该客户机的国家代码相关的日期时间值的缺省格式。有关各种国家代码的字符串格式的列表，参见第372页的表35。

当在使用一个内部日期时间值的操作中使用日期时间值的有效字符串表示时，在执行该操作前，将该字符串表示转换为日期、时间或时间戳记的内部形式。下面几节定义日期时间值的有效字符串表示。

日期字符串

日期的字符串表示是从一个数字开始且长度至少为 8 个字符的一个字符串。可以包括尾部空格；可以从日期的月部分和日部分中略去前导零。

表33中列出了日期的有效字符串格式。每个格式由名称标识，并包括一个相关的缩写和有关其使用的示例。

表 33. 日期的字符串表示的格式

格式名	缩写	日期格式	示例
国际标准化组织	ISO	yyyy-mm-dd	1991-10-27
IBM 美国标准	USA	mm/dd/yyyy	10/27/1991
IBM 欧洲标准	EUR	dd.mm.yyyy	27.10.1991
日本工业标准公元	JIS	yyyy-mm-dd	1991-10-27
当地定义的（本地）	LOC	取决于数据库国家代码	—

时间字符串

时间的字符串表示是从一个数字开始且长度至少为 4 个字符的一个字符串。可以包括尾部空格；可以从时间的小时部分中略去前导零，并可整个略去秒。若您选择略去秒，则认为是 0 秒的隐式说明。因此，13.30 等于 13.30.00。

表34中列出了时间的有效字符串格式每个格式由名称标识，并包括一个相关的缩写和有关其使用的示例。

表 34. 时间的字符串表示的格式

格式名	缩写	时间格式	示例
国际标准化组织	ISO	hh.mm.ss	13.30.05
IBM 美国标准	USA	hh:mm AM 或 PM	1:30 PM
IBM 欧洲标准	EUR	hh.mm.ss	13.30.05
日本工业标准公元	JIS	hh:mm:ss	13:30:05
当地定义的（本地）	LOC	取决于应用程序国家代码	—

注:

1. 在 ISO、EUR 或 JIS 时间字符串格式中，.ss（或 :ss）是可选的。
2. 对于 USA 时间字符串格式，可以略去分钟说明，指示 00 分钟的隐式说明。因此，1 PM 等效于 1:00 PM。
3. 在 USA 时间字符串格式中，小时说明不能大于 12 且不能是 0，除非是 00:00 AM 这种特殊情况。使用 24 小时时钟的 ISO 格式，在 USA 格式和 24 小时时钟之间的对应性如下所示：
 - 12:01 AM 至 12:59 AM 对应于 00.01.00 至 00.59.00。
 - 01:00 AM 至 11:59 AM 对应于 01.00.00 至 11.59.00。

- 12:00 PM (中午) 至 11:59 PM 对应于 12.00.00 至 23.59.00。
- 12:00 AM (午夜) 对应于 24.00.00, 而 00:00 AM (午夜) 对应于 00.00.00。

时间戳记字符串

时间戳记的字符串表示是从一个数字开始且长度至少为 16 个字符的一个字符串。一个时间戳记的完整字符串表示的格式为 `yyyy-mm-dd-hh.mm.ss.nnnnnn`。可以包括尾部空格; 可以从该时间戳记的月、日和小时部分略去前导零, 并且可以截断微秒或整个略去它。若您选择略去微秒部分的任何数字, 则认为是 0 的隐式说明。因此, 1991-3-2-8.30.00 等于 1991-03-02-08.30.00.000000。

MBCS 考虑事项

日期和时间戳记字符串只能包含单字节字符和数字。

日期和时间格式: 用字符串表示的日期和时间格式是与应用程序的国家代码相关的日期时间值的缺省格式。当预编译程序或将其联编至数据库时, 可指定 "F" 格式选项来覆盖此缺省格式。

以下是日期和时间的输入和输出格式的说明:

- 输入时间格式
 - 无缺省输入时间格式
 - 对于所有国家代码, 允许所有时间格式作为输入。
- 输出时间格式
 - 缺省输出时间格式等于本地时间格式。
- 输入日期格式
 - 无缺省输入日期格式
 - 在日期的本地格式与 ISO、JIS、EUR 或 USA 日期格式冲突之处, 将把本地格式作为日期输入格式。例如, 查看表35中的 UK 项。
- 输出日期格式
 - 在表35中显示了缺省输出日期格式

注: 表35也显示了与各种国家代码对应的该字符串格式的列表

表 35. 按国家代码划分的日期和时间格式

国家代码	本地日期格式	本地时间格式	缺省输出日期格式	输入日期格式
355 阿尔巴尼亚	yyyy-mm-dd	JIS	LOC	LOC, USA, EUR, ISO
785 阿拉伯	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO

表 35. 按国家代码划分的日期和时间格式 (续)

国家代码	本地日期格式	本地时间格式	缺省输出日期格式	输入日期格式
001 澳大利亚 (1)	mm-dd-yyyy	JIS	LOC	LOC, USA, EUR, ISO
061 澳大利亚	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
032 比利时	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
055 巴西	dd.mm.yyyy	JIS	LOC	LOC, EUR, ISO
359 保加利亚	dd.mm.yyyy	JIS	EUR	LOC, USA, EUR, ISO
001 加拿大	mm-dd-yyyy	JIS	USA	LOC, USA, EUR, ISO
002 加拿大 (法 语区)	dd-mm-yyyy	ISO	ISO	LOC, USA, EUR, ISO
385 克罗地亚	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
042 捷克共和国	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
045 丹麦	dd-mm-yyyy	ISO	ISO	LOC, USA, EUR, ISO
358 芬兰	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
389 FYR 马其 顿	dd.mm.yyyy	JIS	EUR	LOC, USA, EUR, ISO
033 法国	dd/mm/yyyy	JIS	EUR	LOC, EUR, ISO
049 德国	dd/mm/yyyy	ISO	ISO	LOC, EUR, ISO
030 希腊	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
036 匈牙利	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
354 冰岛	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
091 印度	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
972 以色列	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
039 意大利	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
081 日本	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO

表 35. 按国家代码划分的日期和时间格式 (续)

国家代码	本地日期格式	本地时间格式	缺省输出日期格式	输入日期格式
082 韩国	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
001 拉丁美洲 (1)	mm-dd-yyyy	JIS	LOC	LOC, USA, EUR, ISO
003 拉丁美洲	dd-mm-yyyy	JIS	LOC	LOC, EUR, ISO
031 荷兰	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
047 挪威	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
048 波兰	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
351 葡萄牙	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
086 中国	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
040 罗马尼亚	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
007 俄罗斯	dd/mm/yyyy	ISO	LOC	LOC, EUR, ISO
381 塞尔维亚 / 蒙的内哥罗	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
042 斯洛伐克	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
386 斯洛文尼亚	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
034 西班牙	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
046 瑞典	dd/mm/yyyy	ISO	ISO	LOC, EUR, ISO
041 瑞士	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
088 台湾	mm-dd-yyyy	JIS	ISO	LOC, USA, EUR, ISO
066 泰国 (2)	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
090 土耳其	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
044 英国	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
001 美国	mm-dd-yyyy	JIS	USA	LOC, USA, EUR, ISO
084 越南	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO

表 35. 按国家代码划分的日期和时间格式 (续)

国家代码	本地日期格式	本地时间格式	缺省输出日期格式	输入日期格式
<p>注:</p> <ol style="list-style-type: none"> 1. 使用缺省 C 本国语言环境的国家被赋予国家代码 001。 2. 佛教纪元的 yyyy 等价于罗马教的 yyyy + 543 年 (限于泰国)。 				

DB2 UDB 中的 Unicode / UCS-2 和 UTF-8 支持

下面描述了两个标准。

介绍

Unicode 字符编码标准是定长的字符编码模式，它包含了世界上几乎所有现用语言的字符。Unicode 字符经常显示为 "U+xxxx"，其中 xxxx 是字符的十六进制代码。

每个字符为 16 位 (2 字节)，与语言无关。所生成的 65000 个代码元素足以用于编码世界上主要语言的大多数字符，Unicode 标准还提供了一种扩展机制，可以编码一百多万字符。此扩展保留了一个代码值范围 (从 U+D800 到 U+D8FF，称为“代用码”)，用来将一些 32 位字符编码为两个连续的代码元素。

“国际标准化组织” (ISO) 和“国际电工委员会” (IEC) 标准 10646 (ISO/IEC 10646) 指定了“通用多个八位字节编码字符集” (UCS)，该编码字符集有一个双字节版本 (UCS-2) 和一个四字节版本 (UCS-4)。此 ISO 标准的双字节版本相当于无代用码的 Unicode。ISO 10646 还定义了一种扩展技术，用于在 UCS-2 编码的字符串中编入一些 UCS-4 代码。此扩展技术称为 UTF-16，它相当于有代用码的 Unicode。

DB2 UDB 支持 UCS-2; 即无代用码的 Unicode。

不支持 UTF-8 (代码页 1208) 客户机与非 Unicode 数据库连接。

UTF-8

在 UCS-2 或 Unicode 编码中，ASCII 和控制字符也是双字节，且前导字节为 0。例如，NULL 是 U+0000，而大写字母 "A" 由 U+0041 表示。这可能是基于 ASCII 的应用程序和 ASCII 文件系统的主要问题，因为在 UCS-2 字符串中，多余的 NULL 可能在字符串中随处出现。对于依赖于不变的 ASCII 代码的程序，可使用转换算法 UTF-8 避免这个问题。

UTF-8 (UCS 转换格式 8) 是一种算法转换，它将定长 UCS-4 字符转换成变长字节字符串。在 UTF-8 中，ASCII 字符由通常的单字节代码表示，但 UCS-2 中的

非 ASCII 字符变为双字节或三字节。换言之，即 UTF-8 将 UCS-2 字符转换为多字节代码集，而 ASCII 不变。使用 UTF-8 格式的每个 UCS-2 字符的字节数可根据下表来确定：

UCS-2 (十六进制)	UTF-8 (二进制)	说明
0000 - 007F	0xxxxxxx	ASCII
0080 - 07FF	110xxxxx 10xxxxxx	上限为 U+07FF
0800 - FFFF	1110xxxx 10xxxxxx 10xxxxxx	其他 UCS-2

注意：在此表的第三行中要排除范围 D800 至 DFFF，它控制 UCS-4 的范围 0000 0800 至 0000 FFFF。

在以上每一项中，x 串都是字符的 UCS 位表示法。例如，U0080 转换为 1100001010000000。

DB2 UDB 中的 UCS-2/UTF-8 实现

代码页 / CCSID 号码

在 IBM 内，已将 UCS-2 代码页注册为代码页 1200。所有代码页是用增长的字符集定义的；也就是说当向一个代码页添加新字符时，代码页号码不变。代码页 1200 始终引用 Unicode / UCS-2 的当前版本，并已用于在 DB2 UDB 中支持 UCS-2。

UCS 标准的特定版本由 Unicode 2.0 和 ISO/IEC 10646-1 定义，在 IBM 内已将它注册为 CCSID 13488。此 CCSID 已在 DB2 UDB 内部使用，用于存储 euc-Japan 和 euc-Taiwan 数据库中的字符串数据。CCSID 13488 和代码页 1200 都引用 UCS-2，除了它们的“双字节” (DBCS) 空格的值不同外，其处理方式是相同的：

CP/CCSID	单字节 (SBCS) 空间	双字节 (DBCS) 空间
1200	N/A	U+0020
13488	N/A	U+3000

注意：在 UCS-2 数据库中，U+3000 没有任何特殊意义。

关于转换表，由于代码页 1200 是 CCSID 13488 的超集，所以对它们使用了相同的 (超集) 表。

在 IBM 中，已将 UTF-8 注册为具有增长的字符集的 CCSID 1208 (有时也称为代码页 1208)。当向标准集中添加新字符，此号码 (1208) 不会改变。数 1208 用作多字节代码页号码，以支持 DB2 的 UCS-2/UTF-8。

DB2 UDB 支持 UCS-2 作为一个新的多字节代码页。MBCS 代码页号码为 1208，它是数据库代码页号码以及该数据库中字符串的代码页。UCS-2 的双字节代码页号码为 1200，它是数据库中的图形字符串数据的代码页。当在 UCS-2/UTF-8 中创建数据库时，CHAR、VARCHAR、LONG VARCHAR 和 CLOB 数据以 UTF-8

存储，而 GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC 和 DBCLOB 数据以 UCS-2 存储。我们称之为 UCS-2 数据库。

创建 UCS-2 数据库

缺省情况下，用创建数据库的应用程序的代码页创建数据库。因此，若从 UTF-8 客户机（例如，AIX 的 UNIVERSAL 本国语言环境）创建数据库，或者若将客户机上的 DB2CODEPAGE 注册表变量设置为 1208，您的数据库将创建为 UCS-2 数据库。或者，可显式地将“UTF-8”指定为 CODESET 名，并使用 DB2 UDB 支持的任何有效的双字母 TERRITORY 代码。

例如，要使用美国区域代码创建 UCS-2 数据库，发出如下命令：

```
DB2 CREATE DATABASE dbname USING CODESET UTF-8 TERRITORY US
```

要使用 **sqlcrea** API 创建 UCS-2 数据库，应相应地在 *sqledbccountryinfo* 中设置值。例如，将 SQLDBCODESET 设置为 UTF-8，将 SQLDBLOCALE 设置为任何有效的区域代码（例如 US）。

UCS-2 数据库的缺省整理顺序是 IDENTITY，它提供 UCS-2 代码点次序。因此，缺省情况下，将根据 UCS-2 代码点顺序排序和比较所有 UCS-2/UTF-8 字符。

所有与文化相关的参数，如日期或时间格式、小数分隔符等等，基于客户机的当前区域。

UCS-2 数据库允许同 DB2 UDB 支持的任何单字节和多字节代码页连接。客户机代码页和 UTF-8 之间的代码页字符转换由数据库管理程序自动执行。图形字符串类型的数据通常使用 UCS-2，不必通过代码页转换。命令行处理器(CLP) 环境是个例外。若从 CLP 选择图形字符串 (UCS-2) 数据，则返回的图形字符串数据将从 UCS-2 转换为客户机环境的代码页。

任何客户机均受字符编码系统、输入方法和它的环境所支持的字体限制，但是 UCS-2 数据库本身接受并存储所有 UCS-2 字符。因此，每个客户机通常使用 UCS-2 字符的一个子集，但数据库管理程序允许使用 UCS-2 字符的整个编码系统。

当字符由本地代码页转换为 UTF-8 时，字节数可能增多。ASCII 字符不会增多，但其他 UCS-2 字符将增大两三倍。UTF-8 格式中的每个 UCS-2 字符的字节数可根据第375页的『UTF-8』中的表来确定。

数据类型

DB2 UDB 所支持的所有数据类型在 UCS-2 数据库中也受支持。特别是，UCS-2 数据库支持图形字符串数据，并以 UCS-2 / Unicode 存储。每个客户机（包括 SBCS 客户机）与 UCS-2 数据库连接时，可使用 UCS-2 / Unicode 格式的图形字符串数据类型。

UCS-2 数据库同任何 MBCS 数据库一样，其字符串数据以字节数计。当使用 UTF-8 格式的字符串数据时，不应当认为每个字符都是单字节。在多字节 UTF-8 编码中，每个 ASCII 字符都是单字节，但每个非 ASCII 字符为两个或三个字节。当定义 CHAR 字段时应考虑这点。根据 ASCII 与非 ASCII 字符的比率，一个大小为 n 字节的 CHAR 字段可包含 $n/3$ 到 n 之间的任意多个字符。

对图形字符串 UCS-2 数据类型使用字符串 UTF-8 编码也对总存储器需求有影响。对于大多数字符是 ASCII 但其间插入一些非 ASCII 字符的情况，存储 UTF-8 数据可能是比较好的替代方法，因为存储器需求接近每个字符一个字节。另一方面，对于大多数字符是扩充为三字节 UTF-8 序列的非 ASCII 字符（例如，表意字符）的情况，UCS-2 图形字符串格式可能是比较好的替代方法，因为每个 UCS-2 字符只需要两字节，而使用 UTF-8 格式的每个对应字符需要三字节。

在 MBCS 环境中处理字符串的 SQL 标量函数，如 LENGTH、SUBSTR、POSSTR、MAX 和 MIN 等，是基于“字节”数而不是“字符”数进行运算的。这在 UCS-2 数据库中也一样，但当为 UCS-2 数据库指定偏移量和长度时应格外小心，因为始终是在数据库代码页的场境中定义这些值。也就是说，对于 UCS-2 数据库，应在 UTF-8 中定义这些值。因为一些单字节字符需要 UTF-8 格式的多字节，因此对单字节数据库有效的 SUBSTR 索引可能对 UCS-2 数据库无效。若指定了不正确的索引，则将返回 SQLCODE -191 (SQLSTATE 22504)。有关这些函数行为的说明，参考 *SQL Reference*。

在用户程序中，（C 语言中的）char 数据类型支持 SQL CHAR 数据类型。在用户程序中，sqldbchar 支持 SQL GRAPHIC 数据类型。注意，对于 UCS-2 数据库，sqldbchar 数据始终使用大尾结构（高字节在前）格式。当将应用程序与 UCS-2 数据库连接时，由 DB2 UDB 在应用程序代码页和 UTF-8 之间转换字符串数据，而图形字符串数据始终是 UTF-2 格式。

标识符

在 UCS-2 数据库中，所有标识符都使用多字节 UTF-8。因此，在标识符中使用任何 UCS-2 字符是可能的，DB2 UDB 允许使用扩充字符集中的字符（例如，重音字符或多字节字符）。有关哪些标识符允许使用扩充字符的详情，参见第315页的『附录B. 命名规则』。

客户机可输入其 SBCS 或 MBCS 环境支持的任何字符，并且标识符中的所有字符都由数据库管理程序转换为 UTF-8。当在 UCS-2 数据库的标识符中指定国家语言字符时，必需考虑以下两点：

- 每个非 ASCII 字符需要两个或三个字节。因此，一个 n 字节标识符只能容纳 $n/3$ 至 n 个字符，这取决于 ASCII 与非 ASCII 字符的比率。若仅有一个或两个非 ASCII 字符（例如，重音符），该限制接近 n 个字符，而对于全部由非 ASCII 字符组成的标识符（例如，日语），只能使用 $n/3$ 个字符。
- 若要在不同的客户机环境中输入标识符，应使用这些客户机可用的公共字符子集来定义标识符。例如，若要从拉丁语系 1、阿拉伯语和日语环境中存取 UCS-2 数据库，则所有标识符应严格限制为 ASCII。

UCS-2 文字

UCS-2 文字可以下列两种方式指定：

- 作为图形字符串常量指定，使用 *SQL Reference* 中“语言元素”一章的“图形字符串常量”一节所描述的 G'...' 或 N'...' 格式。用这种方式指定的任何文字将由数据库管理程序由应用程序代码页转换为 UCS-2。
- 使用 UX'...' 或 GX'...' 格式定义 UCS-2 十六进制字符串。在 UX 或 GX 后面的引号内指定的常量必须是 4 个十六进制位的倍数。每个四位组表示一个 UCS-2 代码点。

当使用命令行处理器 (CLP) 时，若本地应用程序代码页中存在 UCS-2 字符，则第一种方法比较容易（例如，从使用代码页 850 的终端输入代码页 850 的任何字符）。对于在应用程序代码页编码系统之外的字符，应使用第二种方法（例如，从使用代码页 850 的终端指定日语字符）。

UCS-2 数据库中的模式匹配

模式匹配是现存的 MBCS 数据库的行为与 UCS-2 数据库的行为稍微不同的一个范畴。

对于 DB2 UDB 中的 MBCS 数据库，当前行为如下所示：若匹配表达式包含 MBCS 数据，则该模式可以包含 SBCS 和 MBCS 字符。该模式中的特殊字符解释如下：

- SBCS 下划线字符是一个 SBCS 字符。
- DBCS 下划线字符是一个 MBCS 字符。
- 百分比字符（可能是 SBCS 或 DBCS）是由多个 SBCS 或 MBCS 字符组成的字符串或为空值。

若匹配表达式包含图形字符串 DBCS 数据，则该表达式仅包含 DBCS 字符。该模式中的特殊字符解释如下：

- DBCS 下划线字符是一个 DBCS 字符。

- DBCS 百分比字符是由多个 DBCS 字符组成的字符串或为空值。

在 UCS-2 数据库中，“单字节”与“双字节”字符之间没有任何真正的差别；每个 UCS-2 字符占用两个字节。尽管 UTF-8 格式是 UCS-2 字符的“混合字节”编码，但 UTF-8 格式的 SBCS 和 MBCS 字符之间没有任何真正的区别。每个字符是一个 UCS-2 字符，而与 UTF-8 格式的字节数无关。当指定字符串或图形字符串表达式时，下划线字符是一个 UCS-2 字符，而百分比字符是由多个 UCS-2 字符组成的字符串或空值。

在客户端，字符串表达式使用客户机的代码页，将由数据库管理程序转换为 UTF-8。SBCS 客户机代码页没有 DBCS 百分比字符或 DBCS 下划线字符，但每个受支持的代码页都包含一个单字节百分比字符（与 U+0025 对应）和一个单字节下划线字符（与 U+005F 对应）。UCS-2 数据库的特殊字符解释如下：

- SBCS 下划线字符（与 U+0025 对应）在图形字符串表达式中是一个 UCS-2 字符，而在字符串表达式中是一个 UTF-8 字符。
- SBCS 百分比字符（与 U+005F 对应）在图形字符串表达式中是由多个 UCS-2 字符组成的字符串或空值，而在字符串表达式中是由多个 UTF-8 字符组成的字符串或空值。

DBCS 代码页还支持 DBCS 百分比符号（与 U+FF05 对应）和 DBCS 下划线字符（与 U+FF3F 对应）。对于 UCS-2 数据库，这些字符没有任何特殊的意义。

对于可选的“转义表达式”，它指定一个用来修改下划线字符和百分比字符特殊意义的字符，仅支持 ASCII 字符或扩充为两字节 UTF-8 序列的字符。若指定扩充为三字节 UTF-8 值的转义字符，则将返回出错信息（错误 SQL0130N, SQLSTATE 22019）。

调入 / 调出 / 装入考虑事项

正如本节中所述，UCS-2 数据库支持 DEL、ASC 和 PC/IXF 文件格式。不支持 WSF 格式。

当从 UCS-2 数据库调出到 ASCII 定界 (DEL) 文件时，所有字符数据都转换为应用程序代码页。字符串和图形字符串数据都转换为客户机的 SBCS 或 MBCS 代码页。这是调出任何数据库的期望行为，并且不得更改，因为整个 ASCII 定界文件只能有一个代码页。因此，若调出到 ASCII 定界文件中，只会保存应用程序代码页中存在的那些 UCS-2 字符。其他字符用该应用程序代码页的缺省替代字符替换。对于 UTF-8 客户机（代码页 1208），则不会丢失数据，因为 UTF-8 客户机支持所有 UCS-2 字符。

当从 ASCII 文件 (DEL 或 ASC) 调入到 UCS-2 数据库时，字符串数据由应用程序代码页转换为 UTF-8，而图形字符串数据由应用程序代码页转换为 UCS-2。不会

丢失数据。若想要调入以另一种代码页保存的 ASCII 数据，在发出 IMPORT 命令之前应更改数据文件代码页。一种方法是，将 DB2CODEPAGE 设置为 ASCII 数据文件的代码页。

对于 SBCS 和 MBCS 客户机，有效的 ASCII 定界符的范围就是那些客户机的 DB2 UDB 当前所能支持的范围。对于 UTF-8 客户机，有效定界符的范围是从 0x01 至 0x7F，并有通常使用的限制。有关这些限制的完整列表，参考 *Data Movement Utilities Guide and Reference* 中的“调出 / 调入 / 装入实用程序文件格式”附录。

当从 UCS-2 数据库调出到 PC/IXF 文件时，字符串数据转换为客户机的 SBCS/MBCS 代码页。图形字符串数据不转换，并以 UCS-2（代码页 1200）存储。不会丢失数据。

当从 PC/IXF 文件调入到 UCS-2 数据库时，假定字符串数据使用存储在 PC/IXF 文件头中的 SBCS/MBCS 代码页，而图形字符串数据使用存储在 PC/IXF 文件头中的 DBCS 代码页。字符串数据由“调入”实用程序从 PC/IXF 文件头中指定的代码页转换为客户机的代码页，然后由 INSERT 语句从客户机代码页转换为 UTF-8。图形字符串数据由“调入”实用程序从 PC/IXF 文件头中指定的 DBCS 代码页直接转换为 UCS-2（代码页 1200）。

“装入”实用程序直接将数据装入数据库，缺省情况下，假设 ASC 或 DEL 文件中的数据使用数据库的代码页。因此，缺省情况下，ASCII 文件不执行代码页转换。当（使用 codepage 修饰词）显式指定了数据文件的代码页时，“装入”实用程序在装入数据之前，使用此信息来将指定的代码页转换为数据库代码页。对于 PC/IXF 文件，“装入”实用程序始终从 IXF 文件头中指定的代码页转换为数据库代码页（对于 CHAR，为 1208；对于 GRAPHIC，为 1200）。

对于 UCS-2，DBCLOB 文件的代码页总是 1200。CLOB 文件的代码页与要调入、装入或调出的数据文件的代码页相同。例如，对于使用 PC/IXF 格式的装入或调入，将认为 CLOB 文件使用 PC/IXF 文件头指定的代码页。若该 DBCLOB 文件处于 ASC 或 DEL 格式，则“装入”实用程序假定 CLOB 数据使用的是数据库的代码页（除非使用 codepage 修饰词另有所指），而“调入”实用程序假定它使用的是客户机应用程序的代码页。

始终对 UCS-2 数据库指定 nochecklengths 修饰词，因为：

- 可将任何 SBCS 与无 DBCS 代码页的数据库连接
- UTF-8 格式中的字符串长度通常与客户机代码页中的字符串长度不同。

有关“装入”、“调入”和“调出”实用程序的详情，参考 *Data Movement Utilities Guide and Reference*。

不兼容性

在与 UCS-2 数据库连接的应用程序中，图形字符串数据始终使用 UCS-2（代码页 1200）。对于与非 UCS-2 数据库连接的应用程序，图形字符串数据使用应用程序的 DBCS 代码页；但如果应用程序代码页是 SBCS，则不允许这样做。例如，当将一个 932 客户机与一个使用日语的非 UCS-2 数据库连接时，图形字符串数据使用代码页 301。对于与 UCS-2 数据库连接的 932 客户机应用程序，图形字符串数据使用 UCS-2 格式。

附录F. 注意事项

IBM 可能未在所有国家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说明或暗示只能使用 IBM 的产品、程序或服务。凡是同等功能的产品、程序或服务，只要不侵犯 IBM 的知识产权，都可以用来替代 IBM 产品、程序或服务。当然，评估和验证非 IBM 产品、程序或服务均由用户自行负责。

本文档的议题可能涉及 IBM 的某些专利或正在申请中的专利的应用。提供本文档，并不表示允许您使用这些专利。您可以将许可证查询以书面形式寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

关于双字节 (DBCS) 许可证查询的信息，请与您所在国家的 IBM 知识产权部门联系，将查询以书面形式寄往：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

以下段落不适用于英国与其它当地法律不允许这种供应方式的国家：国际商用机器公司『按原样』出版此书，不做任何明确或暗示的担保，包括但不限于有关非伪造、商业性或符合特殊目的的隐含保证。一些地区在某些事务中不允许否认拒绝明确或暗示的担保，因此本条款可能不适合您。

本信息中可能有技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些信息将包含在本书新的版本中。IBM 可以随时对本书中说明的产品和/或程序进行改进和/或改动，而不必通知您。

此信息中对非 IBM Web 站点的任何引用仅是为了方便起见，而不以任何方式为那些 Web 站点作保证。那些 Web 站点的资料并非此 IBM 产品资料的一部分，使用那些 Web 站点的风险由您自己承担。

对于您所提供的任何信息，IBM 有权利以任何她认为适当的方式使用或散发，而不必对您负任何责任。

为了以下目的：(1) 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 (2) 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合法用户请与下列地址联系：

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。

这些信息中描述的特许程序及其所有可用的特许资料，按 IBM 客户协议、IBM 国际程序许可证协议或任何等价的协议中的条款，由 IBM 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。

涉及非 IBM 产品的信息可从这些产品的供应商、其发行公告或其它公众可用源得到。IBM 未测试这些产品，因此不能确认性能的精确度、兼容性或其它对非 IBM 产品的索赔赔偿要求等。有关非 IBM 产品功能方面的问题可向它们的供应商提出。

所有关于 IBM 未来方向或意向的声明都可能随时更改或撤消，而不作任何通知，并且仅代表发展目标。

此信息包含了用于日常商业处理的数据和报表的示例。为了尽可能完整地说明问题，这些示例中包含了个人、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：

本信息中可能包含用源语言编写的示例应用程序，它们说明了各种不同的操作平台上的程序设计技术。您可以为了开发、使用、市场营销或分发应用程序(这些应用程序遵守编写这些示例程序的操作平台的应用程序接口)的目的，以任何形式复

制、修改和分发这些示例程序，不用向 IBM 付费。这些例子未经所有条件下的完整测试。因此，IBM 不能保证或暗示其可靠性、可用性或这些程序的功能。

这些样本程序或任何派生产品的每个副本或任何部分必须包含如下的版权公告：

©（您的公司名称）（年度）。此代码各部分派生自“IBM 公司样本程序”。© Copyright IBM Corp. _输入年份_。All rights reserved.

注册商标

以星号 (*) 标出的下列术语是 IBM 公司在美国和 / 或其他国家的商标。

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 公司的商标或注册商标。

Java 或所有基于 Java 的商标和标志以及 Solaris 是 Sun Microsystems 公司在美国和 / 或其他国家的商标。

Tivoli 和 NetView 是 Tivoli Systems 公司在美国和 / 或其他国家的商标。

UNIX 是经 X/Open 有限公司唯一许可的在美国和 / 或其它国家的注册商标。

以双星号 (**) 标出的其他公司、产品或服务名, 可能是其他公司的商标或服务标志。

索引

[A]

安全性 45, 101
安装
 Netscape 浏览器 310
按扇区分布的数据和奇偶性条纹化
 (RAID-5) 42

[B]

版本恢复 25
版本 6 的不兼容性
 当前解释方式 344
 对分级结构拥有 SELECT 特权
 343
 非双精度 SQLVAR 中的
 SQLNAME 340
 关联代码 336
 过时的配置关键字 341
 过时的数据库配置参数 345
 事件监控程序输出流格式 341
 字符名大小 339
 DATALINK 列 341
 FOR UPDATE 语法 338
 Java 程序设计 338
 OBJCAT 视图 336
 PC/IXF 格式更改 340
 RUMBA 344
 SYSFUN 字符串函数特征 342
 SYSIBM 基本目录 337
 SYSTABLE 列更改 342
 USING 和 SORT BUFFER 344
 VARCHAR 数据类型 338
保持活动的信息包 191
备份
 存储器考虑事项 36
 联机 35
 频率 35
 脱机 35
 用户出口程序 37
备份数据库向导 311
变换器步骤 70

标识列 89
标识入选关键字列 88
表
 并置 118
 重组 44
 概述 9
 估计大小需求 105
 关系 37
 规范化 90
 检查约束 100
 命名规则 318
 系统目录 106
 映射至表空间 130
 用户 106
表空间
 复原 27
 概述 13
 工作负荷考虑事项 134
 恢复 38
 临时 122, 132
 命名规则 318
 目录 121, 133
 前滚恢复 27
 设计 127
 设计和选择 120
 输入和输出 (I/O) 考虑事项 128
 数据库管理空间 (DMS) 126
 系统管理空间 (SMS) 122
 选择 SMS 或 DMS 135
 映射至缓冲池 129
 映射至节点组 130
 用户 121
 SYSCATSPACE 121
 TEMPSPACE1 122
 USERSPACE1 121
表之间的关系 37
别名
 命名规则 318
别名地址 200
并行处理能力
 概述 101

并行性
 查询 55
 分区间 56
 分区内 55
 概述 53
 和不同的硬件环境 65
 和索引创建 57
 类型 55
 实用程序 57
 数据库备份和复原实用程序 57
 “装入”实用程序 57
 “自动装入”实用程序 57
I/O 55
并行资源管理程序 181
并置, 表 118
不兼容性
 版本 6 332
 版本 7 329
 创建数据库 343
 计划的 328
 列不匹配 335
 列数据类型更改为 BIGINT 335
 说明 327
 外部关键字列名 333
 只读视图 (计划的) 328
 主关键字列名 333
 COLNAMES (计划的) 328
 FK_COLNAMES (计划的) 328
 PK_COLNAMES (计划的) 328
 SYSCAT.CHECKS 列 TEXT 335
 SYSCAT.INDEXES 列
 COLNAMES 334
 SYSCAT.STATEMENTS 列
 TEXT 334
 SYSCAT.VIEWS 列 TEXT 334
不可恢复的数据库 28
不确定事务 163
 恢复 150, 155
 人工恢复 163
 再同步 151
部分撤消群集 114

步骤（在入库中） 68

[C]

参考循环 97

参考约束 96

删除连接的关系 99

SQL 操作的意义 98

SQL DELETE 规则 98

SQL INSERT 规则 98

SQL UPDATE 规则 99

仓库 67

代理程序 68

过程 69

目标 68

源 68

仓库步骤 68

变换器 70

程序 70

用户定义程序 70

SQL 69

操作系统限制 38

插入规则 98

查看

联机信息 309

查询并行性 55

查询间并行性 55

查询内并行性 55

长整数字段数据

估计大小需求 108

超类型 86

撤消群集

部分 114

程序步骤 70

重组表 44

触发器 22, 100

命名规则 318

创建表空间向导 311

创建表向导 311

创建数据库向导 311

磁盘

条理化 41

阵列 41

RAID（独立磁盘冗余阵列） 41

磁盘故障

防止 41

磁盘镜像 43

磁盘镜像或磁盘双工 (RAID-1) 42

磁盘阵列

软件 43

硬件 42

磁盘组 266

从属表 97

从属行 97

存储器

备份和恢复所需的 36

媒体故障 37

存储器对象

表空间 13

概述 13

缓冲池 17

容器 16

[D]

打印 PDF 书籍 305

大对象 (LOB) 数据

估计大小需求 108

列定义 85

大陆分群 270

代理程序

仓库 68

高可用性 264

代理点 68

代码页

受支持的 Windows 代码页 362

DB2CODEPAGE 注册表变量 362

单处理器环境 59

单分区

单处理器环境 59

多处理器环境 60

第二种规范形式 91

地理编码 75

第三种规范形式 93

第四种规范形式 94

第一种规范形式 91

定义表列 85

独立磁盘冗余阵列 (RAID) 41

对目录表空间的建议 133

对象名, 联合体数据库 319

多分区节点组 53

多分区配置 61

多媒体对象 82

多站点更新 140, 142

存取 DB2 UDB 服务器的主机或

AS/400 应用程序 146

[F]

发行版间的不兼容性

说明 327

发行说明 305

方法

Sun Cluster 264

防止磁盘故障 41

非中断维护 209

分布式请求 49

分布式事务处理

安全性考虑事项 165

错误处理 163

对主机和 AS/400 数据库服务器的

支持 162

配置考虑事项 166

事务管理器 155

数据库连接考虑事项 162

应用程序 154

资源管理程序 156

RELEASE 语句 162

分布式数据库 139

分区

关键字 116

数据 114

数据库 53

映射 115

分区间并行性 56

配合分区内并行性使用 57

分区兼容性 118

分区内并行性 55

配合分区间并行性使用 57

分区数据库 53

分群

大陆 270

校园 270

父代

表 97

关键字 96, 97

行 97

复原

表空间 27

复原 (续)

数据库 25
复原向导 312
复制的摘要表 119

[G]

高可用性 181, 231, 261
高可用性群集多重处理
(HACMP) 181, 191
根类型 86
更改口令 316
更新规则 99
工作单元 139
远程 139
估计大小需求
表 105
长整数字段数据 108
大对象 (LOB) 数据 108
临时工作空间 112
日志文件空间 112
索引空间 109
故障恢复 191
概述 261
故障恢复时间 289
故障恢复支持 181, 191, 231, 261
并行存取方式 181
多个逻辑节点的故障恢复 184
分区故障恢复 183
分区故障相互恢复 187
故障恢复后的重新连接 188
热备份方式 181, 182
实例故障恢复 182
实例故障相互恢复 185
相互替换方式 181, 185
故障监控 281
故障容错 263
关键字 87
分区 116
关键字列
标识 88
关系
多对多 84
多对一 83
一对多 83
一对一 85

关系数据库概念

概述 7
归档记录 29
归档日志
联机 30
脱机 30
规则文件 191
限制 210
用于 HACMP 210
国家语言支持 (NLS)
日期时间值 369
双向 CCSID 支持 363
字符集 362

[H]

缓冲池
概述 17
IBMDEFAULTBP 129
恢复 23
必需的时间 36
操作系统限制 38
存储器考虑事项 36
点 35
减少工作表的记录 33
历史文件 12
日志末尾 27
日志文件 12
时间点 27
损坏的表空间 38
性能 39
影响的因素 27
恢复点 35
恢复对象
概述 12
历史文件 12
日志文件 12
活动日志 30

[J]

基地址 200
级联分配 192
记录
归档 29
循环 29

兼容性

分区 118
检查约束 22
减少工作表的记录 33
降低媒体故障的影响 41
降低事务故障的影响 43
交换别名地址 197
校园分群 270
节点
数据位置, 确定 115
节点同步 43
节点组 53
概述 9
设计 113
IBMCATGROUP 121
IBMDEFAULTGROUP 121
IBMTEMPGROUP 122
结构化类型 86
概述 101
进程 (在入库中) 69
具体值 81
具有多个处理器的分区 62
具有一个处理器的分区 61

[K]

可恢复的数据库 28
可扩充性 58
可缩放性 191
空间
数据 75
信息 73
控制方法 268
口令
更改 316
命名 316

[L]

类型
表 86, 101
视图 86
类型分级结构 86
概述 101
历史数据
概述 101

- 联合体数据库
 - 对象名 319
 - 区分大小写的名称 320
 - 认证 47
 - 设计考虑事项 137
 - 授权 47
 - 系统 49
 - 整理顺序, 准则 368
- 联机帮助 307
- 联机归档日志 30
- 联机信息
 - 查看 309
 - 搜索 313
- 连接缓冲, MTS 176
- 连接路径 90
- 连续可用性 263
- 两阶段落实 140, 142, 147
 - 错误处理 150
- 列
 - 命名规则 318
 - 为表定义 85
- 临时表空间 122, 132
- 临时工作空间
 - 估计大小需求 112
- 逻辑数据库分区 63
- 逻辑数据库设计 81
 - 定义表 83
 - 关系 83
 - 决定要记录的数据 81
- 逻辑网络接口 265
- 逻辑主机 265
- 落实
 - 两阶段 147
 - 两阶段期间的错误 150

[M]

- 媒体故障
 - 降低影响 41
 - 目录节点考虑事项 41
 - 日志 37
- 模式
 - 概述 11
- 目标
 - 表 86
 - 行 86

- 目标 (续)
 - 类型 86
 - 视图 86
- 目录
 - 数据库 103
- 目录表空间 121

[P]

- 配置
 - 多分区 61
- 配置参数
 - 概述 18
 - DB2 事务管理器考虑事项 145
- 配置多站点更新向导 311

[Q]

- 迁移
 - 数据库 323
- 前滚恢复 25
 - 表空间 27
 - 数据库 26
- 清除日志 32
- 区别大小写的名称, 联合体数据库 320
- 权限级 46
- 缺省代理点 68
- 确定数据的类型
 - 概述 101
- 群集
 - 管理 192
 - 监控 217
 - 配置 192

[R]

- 热备份配置 192
 - 示例 197
- 任务
 - 入库 70
- 认证 45
 - 联合体数据库概述 47
- 日期
 - 格式 372
- 日期时间值
 - 字符串表示 370

- 日期时间值 (续)
 - REORG 实用程序 369
- 日期字符串
 - 定义 370
- 日志
 - 必需的存储器 37
 - 活动的 30
 - 联机归档日志 30
 - 数据库 29
 - 用户出口程序 37
- 日志文件空间
 - 估计大小需求 112
- 容量 58
- 容器
 - 概述 16
- 入库
 - 对象 67
 - 概述 67
 - 任务 70
- 入库概述 67
- 软件磁盘阵列 43

[S]

- 删除规则 98
- 商业规则
 - 概述 20
- 商业元数据 70
- 设计
 - 表空间 127
 - 联合体数据库 137
 - 设计和选择表空间 120
 - 设置文档服务器 312
- 审查活动
 - 概述 101
- 时间
 - 格式 372
- 时间戳记
 - 定义 370
- 时间戳记字符串
 - 定义 372
- 时间字符串
 - 定义 371
- 实例
 - 概述 8
- 实体 81
- 实用程序并行性 57

- 使表规范化 90
 - 事件监控 210
 - 试探性操作 164
 - 视图
 - 概述 9
 - 命名规则 318
 - 事务
 - 存取分区数据库 162
 - 非 XA 155
 - 紧密耦合 155
 - 两阶段落实 155
 - 全局 155
 - 松散耦合 155
 - 事务处理
 - 配置 XA 事务管理器 169
 - 事务故障
 - 降低影响 43
 - 事务管理器
 - 使用 IBM TXSeries CICS 来实现 169
 - 使用 IBM TXSeries Encina 来实现 170
 - 从 Encina 应用程序中引用 DB2 数据库 171
 - 配置 DB2 170
 - 为每个资源管理程序配置 Encina 170
 - 使用 Tuxedo 来实现 172
 - 使用“Microsoft 事务服务器”来实现 174
 - 事务管理器 (TM) 142, 143
 - 授权 46
 - 概述 101
 - 联合体数据库概述 47
 - 书籍 297, 306
 - 输入和输出 (I/O) 考虑事项
 - 表空间 128
 - 数据
 - 长整数字段 108
 - 大对象 (LOB) 108
 - 分区 114
 - 运行 67
 - 资料式 67
 - 数据仓库中心 67
 - 数据库
 - 不可恢复的 28
 - 数据库 (续)
 - 对象命名规则 315
 - 分布式 139
 - 概述 9
 - 可恢复的 28
 - 命名规则 315
 - 目录 103
 - 文件 104
 - 在单个事务中使用数个数据库 140
 - 主机系统上 141
 - 数据库别名 315
 - 命名规则 315
 - 数据库对象
 - 表 9
 - 概述 7
 - 节点组 9
 - 命名规则 318, 363
 - 模式 11
 - 实例 8
 - 视图 9
 - 数据库 9
 - 索引 10
 - 系统目录表 11
 - 数据库分区 53
 - 同步 43
 - 数据库复原 25
 - 数据库管理程序
 - 命名规则 315
 - 数据库管理程序配置参数
 - 概述 18
 - 数据库管理空间 (DMS) 13, 126
 - 数据库恢复所需的时间 36
 - 数据库配置参数
 - 概述 19
 - 数据库迁移 323
 - 数据库前滚恢复 26
 - 数据库日志 29
 - 数据库设计
 - 逻辑 81
 - 物理 103
 - 数据库系统
 - 联合体 49
 - 数据块大小 17, 121
 - 选择 131
 - 数据源 49
 - 属性 81
 - 属性数据 74
 - 双向 CCSID 支持 363
 - CCSID 表 364
 - DB2 Connect 实现 366
 - DB2 UDB 实现 365
 - 搜索
 - 联机信息 310, 313
 - 损坏的表空间 38
 - 索引
 - 概述 10
 - 命名规则 318
 - 索引关键字 10
 - 索引空间
 - 估计大小需求 109
 - 索引向导 311
- [T]**
- 特权 46
 - 添加数据库向导 311, 312
 - 同步
 - 恢复考虑事项 43
 - 节点 43
 - 数据库分区 43
 - 同步点管理程序 (SPM) 143
 - 图形信息系统 (GIS) 73
 - 脱机归档日志 30
- [W]**
- 外部关键字 96
 - 外部关键字约束 21
 - 唯一标识实体 90
 - 唯一的
 - 关键字 87, 96
 - 索引 10
 - 约束 95
 - 唯一约束 20
 - 文件
 - 数据库 104
 - 无共享模型 181
 - 物理数据库设计 103
 - 物理文件
 - SMS 124

[X]

系统对象
 概述 18
 配置参数 18
系统管理空间 (SMS) 13, 122
系统记录设施
 XA 接口示例 169
系统临时表空间 122
系统目录表
 概述 11
 估计初始大小 106
系统网络结构 (SNA) 146
相互替换配置 192
 示例 198
向导
 备份数据库 311
 创建表 311
 创建表空间 311
 创建数据库 311
 复原数据库 312
 配置多站点更新 311
 索引 311
 添加数据库 311, 312
 完成任务 311
 性能配置 312
向 DMS 表空间添加容器 127
协调程序节点 53
心跳 191, 262
信息目录 70
信息中心 310
星形模式 71
性能
 恢复 39
性能配置向导 312
旋转分配 192
选择数据块大小 131
循环记录 29

[Y]

样本程序
 跨平台 303
 HTML 303
一致点 24
引用类型 86
 概述 101

应急恢复 24
应用程序设计
 整理顺序, 准则 368
硬件磁盘阵列 42
硬件环境 58
 并行性类型 65
 单分区, 单处理器 59
 单分区, 多处理器 60
 具有多个处理器的分区 62
 具有一个处理器的分区 61
 逻辑数据库分区 63
映射
 表空间至缓冲池 129
 表空间至节点组 130
 表至表空间 130
 分区 115
用户表
 页限制 106
用户表空间 121
用户出口程序
 备份 37
 日志 37
用户定义程序步骤 70
用户定义单值类型 (UDT)
 列定义 85
 命名规则 318
用户定义函数 (UDF) 86
 命名规则 318
用户定义事件 191, 210
用户脚本 279
用户临时表空间 122
用户 ID
 命名 316
用于 MTS 协调的事物的受支持的
 DB2 数据库服务器 175
语言标识符
 书籍 304
元数据 70
远程工作单元 139
约束 95
 检查 22
 外部关键字 21
 唯一的 20
 主关键字 20
 NOT NULL 20
运行数据 67

[Z]

灾难恢复 40
增强可缩放性 (ES) 191
摘要表
 复制的 119
 概述 101
整理顺序
 联合体数据库注意事项 368
 一般注意事项 368
 整理顺序选项 369
中断维护 209
主关键字 87, 96
主关键字约束 20, 97
主关键字值
 生成唯一的 89
主索引 87
主题区 67
资料式数据 67
资源管理程序
 将数据库设置成 157
子代表 97
子类型 86
自参考
 表 97
 行 97
 约束 97
组合关键字 88, 97
最新信息 305
作用域 86
坐标系 75

[特别字符]

“首个合适”次序 106
“系统数据仓库” (SDR) 200

C

cconsole 实用程序 270
ctelnet 实用程序 270

D

date
 定义 369
DB2 高可用性代理程序 276

DB2 高可用性代理程序 276 (续)
 控制方法, 用于 277
 注册 276
 hadb2tab 配置文件 277
DB2 库
 查看联机信息 309
 打印 PDF 书籍 305
 订购打印书籍 306
 联机帮助 307
 设置文档服务器 312
 书籍的语言标识符 304
 搜索联机信息 313
 向导 311
 信息中心 310
 最新信息 305
DB2 同步点管理程序 (SPM) 146
DB2 无共享模型 181
DB2 资料库
 结构 297
 书籍 297
DB2 Connect
 用于, 用于多站点数据库更新
 141
DB2CODEPAGE 注册表变量 362
DB2MSCS 实用程序
 概述 234
 设置单分区数据库系统 239
 设置分区数据库系统 240
 为相互替换设置两个单分区数据库
 系统 239
 DB2MSCS.CFG 参数 235
DMS (数据库管理空间) 13, 126
 添加容器至 127
DTP (分布式事务处理) 154

E

ES (增强可缩放性) 191

G

GIS (地理信息系统) 73

H

HACMP (高可用性群集多重处理)
 181, 191

HACMP ES 的恢复程序文件 212
HACMP ES 的恢复脚本 215
HACMP ES 的脚本文件 213
 安装 213
HACMP ES 的迁移任务 220
HACMP ES 配置示例 201
HA-NFS 270
HA.config 文件 283
HTML
 样本程序 303

I

IBMCATGROUP 121
IBMDEFAULTGROUP 121
IBMTEMPGROUP 122
I/O 并行性 55

L

LIST INDOUBT TRANSACTIONS 命
 令 163
LOB (大对象) 数据
 估计大小需求 108
 列定义 85

M

Microsoft 群集服务器 (MSCS) 231
Microsoft 事务服务器
 安装和配置 174
 调整 TCP/IP 通信 178
 连接缓冲 176
 软件的先决条件 174
 使用样本应用程序测试 DB2 178
 使用 ADO 2.1 和更新版本的连接
 缓冲 177
 事务超时和 DB2 连接特性 176
 受支持的 DB2 数据库服务器
 175
 验证安装 175
 再次使用 ODBC 连接 177
 在 DB2 中启用支持 174
MPP 环境 61
MSCS (Microsoft 群集服务器) 231

N

Netscape 浏览器
 安装 310
NFS 服务器节点 198
NFS 服务器替换配置示例 199
node_down 事件 191
node_up 事件 191
NOT NULL 约束 20
NULL 值 87

P

PDF 305

R

RAID (独立磁盘冗余阵列) 41, 42
RAID 设备
 优化性能, 使用 136
RAID-1 (磁盘镜像或磁盘双工) 42
RAID-5 (按扇区分布的数据和奇偶性
 条纹化) 42

S

schema
 命名规则 317
SDR (系统数据仓库) 200
SmartGuide
 向导 311
SMP 环境 60
SMP 群集环境 62
SMS (系统管理空间) 13, 122
SMS 物理文件 124
SNA (系统网络体系结构) 146
SP 计算机 192
SP 交换的 Eprimary 节点 201
SP 交换配置考虑事项 200
Spatial Extender
 概述 73
SPM (同步点管理程序) 143
SQL 步骤 69
SQL 优化器 10
Sun Cluster 2.2 上的高可用性
 故障排除 291
 连接 HA 实例的应用程序 271

Sun Cluster 2.2 上的高可用性 (续)
逻辑主机和 DB2 UDB EEE 274
设置 285
数据复制 276
数据库和数据库管理程序配置参数
275
应急恢复 276
DB2 安装位置和选项 275
DB2 高可用性代理程序 276
EE 和 EEE 实例的磁盘布局 272
EE 和 EEE 实例的主目录布局
273
hadb2_setup 命令 286
Sun Cluster 2.x 261
SYSCATSPACE 121

T

TEMPSPACE1 122
time
定义 370
TPM 和 TP_MON_NAME 值 159

U

UDF (用户定义函数) 86
USERSPACE1 121

W

Windows 代码页
受支持的代码页 362
DB2CODEPAGE 注册表变量 362
Windows 95 故障恢复
管理服务器考虑事项 257
控制中心考虑事项 257
Windows NT 故障恢复
管理 DB2 的考虑事项 251
回退考虑事项 242
计划 231
将 DB2 资源联机后运行脚本
254
将 DB2 资源联机前运行脚本
252
类型 232
启动和停止 DB2 资源 251
热备份 232

Windows NT 故障恢复 (续)
数据库考虑事项 255
通信考虑事项 256
维护 MSCS 系统 241
为相互替换设置分区数据库系统的
示例
初步任务 248
目标 247
运行 DB2MSCS 实用程序
249
注册 ClusterA 的数据驱动器映
射 250
注册 ClusterB 的数据驱动器映
射 250
为相互替换设置两个实例的示例
初步任务 245
目标 245
运行 DB2MSCS 实用程序
246
系统时间考虑事项 257
限制 259
相互替换 233
协调数据库驱动器映射 244
用户和组支持 256
运行脚本, 概述 252
在分区数据库环境中为相互替换设
置数据库驱动器映射 242
DB2MSCS 实用程序
概述 234
设置单分区数据库系统 239
设置分区数据库系统 240
为相互替换设置两个单分区数
据库系统 239
DB2MSCS.CFG 参数 235

X

XA 事务管理器
正在配置 169
xa_open 字符串 157
X/Open 事务管理器接口 (XA)
分布式事务处理 (DTP) 154

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。本指南对您可以收集哪些信息以使“DB2 客户支持中心”更好地为您服务提出了建议。

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，或与任何特许 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、产品说明、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、修订、书籍以及最新的 DB2 技术资料。

注：此资料可能只有英文版。

<http://www.elink.ibm.com/pbl/pbl/>

International Publications Ordering Web 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“专业认证程序”提供各种 IBM 产品（包括 DB2）的认证测试信息。

<ftp.software.ibm.com>

以匿名形式注册。可在目录 /ps/products/db2 中找到有关 DB2 和许多其他产品的演示程序、修订、信息和工具。

comp.databases.ibm-db2, bit.listserv.db2-l

这些 Internet 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，参见 *IBM Software Support Handbook* 的附录 A。要存取此文档，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，IBM 特许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。



Part Number: CT7XVSC

Printed in China

SB84-0219-00



CT7XVSC

