



数据恢复及高可用性指南与参考

注意

使用此信息及其支持的产品前，请先阅读第 317 页的附录 B、『声明』下的常规信息。

修订版声明

此文档包含 IBM 的所有权信息。它在许可协议中提供，且受版权法的保护。本出版物中包含的信息不包括对任何产品的保证，且提供的任何语句都不需要如此解释。

您可在线或通过当地的 IBM 代表处订购 IBM 出版物。

- 要在线订购出版物，请转至 IBM 出版物中心，网址为：www.ibm.com/shop/publications/order
- 要查找当地的 IBM 代表处，请转至 IBM 全球联系人目录，网址为：www.ibm.com/planetwide

要从美国或加拿大的 DB2 市场和销售部订购 DB2 出版物，请致电 1-800-IBM-4YOU (426-4968)。

当您向 IBM 发送信息时，即同意授予 IBM 独一无二的权力以它认为适当且不会对您造成任何影响的方式使用或分发该信息。

目录

关于本书	vii
----------------	-----

第 1 部分 高可用性 1

第 1 章 中断 3

中断特征符	3
中断成本	4
中断容错	4
恢复和避免策略	5

第 2 章 高可用性策略 7

通过冗余获取高可用性	7
通过故障转移获取高可用性	8
通过集群获取高可用性	8
数据库日志记录	8
循环日志记录	9
归档日志记录	9
日志控制文件	10

第 3 章 通过 IBM 数据服务器获取高可用性 11

客户机自动重新路由路线图	11
用于 Linux 和 UNIX 的 DB2 故障监视器工具	11
高可用性灾难恢复 (HADR)	12
DB2 高可用性 (HA) 功能部件	13
通过日志装入获取高可用性	14
日志镜像	14
通过暂挂 I/O 和联机分割镜像支持获取高可用性	15

第 4 章 为获取高可用性进行配置 17

描述和设置客户机自动重新路由	17
使用注册表变量来配置客户机自动重新路由重试行为	19
将客户机连接超时与客户机自动重新路由配合使用	19
配置客户机自动重新路由以将其与客户机连接分发器技术配合使用	20
标识用于客户机自动重新路由的备用服务器	21
使用 IBM 数据服务器 JDBC 和 SQLJ 驱动程序时的客户机重新路由设置	21
客户机自动重新路由的限制	21
DB2 故障监视器注册表文件	23
使用 db2fm 命令来配置 DB2 故障监视器	24
使用 db2fmc 和系统命令来配置 DB2 故障监视器	25
初始化高可用性灾难恢复 (HADR)	26
配置客户机自动重新路由和高可用性灾难恢复 (HADR)	28
索引记录和高可用性灾难恢复 (HADR)	29
高可用性灾难恢复 (HADR) 的数据库配置	29
DB2 高可用性灾难恢复 (HADR) 的日志归档配置	34
高可用性灾难恢复 (HADR) 的性能	35

集群管理器和高可用性灾难恢复 (HADR)	37
初始化备用数据库	37
配置 DB2 高可用性灾难恢复 (HADR) 同步方式	39
高可用性灾难恢复 (HADR) 支持	41
为获取高可用性安排维护	45
使用 SYSPROC.AUTOMAINT_GET_POLICY 或 SYSPROC.AUTOMAINT_GET_POLICYFILE 来收集自动维护策略信息	46
使用 SYSPROC.AUTOMAINT_SET_POLICY 或 SYSPROC.AUTOMAINT_SET_POLICYFILE 来配置自动维护策略	46
配置数据库日志记录选项	47
用于数据库日志记录的配置参数	48
使用 NOT LOGGED INITIALLY 参数减少日志记录	54
日志目录已满时分块事务	55
通过日志归档管理日志文件	55
为获取高可用性配置集群环境	57
集群管理器与 DB2 高可用性 (HA) 功能部件集成	58
IBM Tivoli System Automation for Multiplatforms (SA MP)	59
使用 DB2 高可用性 (HA) 功能部件来自动配置集群	59
使用 DB2 高可用性实例配置实用程序 (db2haicu) 来配置集群环境	60
DB2 集群管理器 API	100
受支持的集群管理软件	100
使分区数据库环境中的时钟同步	115
客户机/服务器时间戳记转换	116

第 5 章 管理和维护高可用性解决方案 117

日志文件管理	117
按需进行的日志归档	119
使用 db2tapemgr 来进行日志归档	119
使用用户出口程序使日志文件归档和检索自动进行	121
分配和除去日志文件	124
使用备份映像包括日志文件	125
防止日志文件意外丢失	126
尽量降低维护对可用性的影响	127
停止 DB2 高可用性灾难恢复 (HADR)	127
在 DB2 高可用性灾难恢复 (HADR) 环境中激活和取消激活数据库	128
在 DB2 高可用性灾难恢复 (HADR) 环境中执行卷动升级	129
使用分割镜像来克隆数据库	130
同步主数据库和备用数据库	131
DB2 高可用性灾难恢复 (HADR) 复制的操作	132
DB2 高可用性灾难恢复 (HADR) 不复制的操作	132
DB2 高可用性灾难恢复 (HADR) 备用数据库状态	133

使用 GET SNAPSHOT 命令来确定 HADR 备用数据库状态	136
DB2 高可用性灾难恢复 (HADR) 管理	136
DB2 高可用性灾难恢复 (HADR) 命令	137

第 6 章 在高可用性解决方案中检测和响应系统中断 139

管理通知日志	139
检测意外中断	141
监视高可用性灾难恢复 (HADR)	141
响应意外中断	142
客户机自动重新路由示例	142
执行 HADR 故障转移操作	145
切换高可用性灾难恢复 (HADR) 中的数据库角色	147
在执行接管操作之后重新集成数据库	148

第 2 部分 数据恢复 149

第 7 章 开发备份和恢复策略 151

确定备份频率	153
恢复的存储器注意事项	154
将相关数据保存在一起	154
不同操作系统和硬件平台之间的备份和复原操作	155

第 8 章 恢复历史记录文件 157

恢复历史记录文件条目状态	158
使用 DB_HISTORY 管理视图查看恢复历史记录文件条目	161
修剪恢复历史记录文件	163
使恢复历史记录文件修剪自动进行	163
防止恢复历史记录文件条目被修剪	164

第 9 章 管理恢复对象 167

使用 PRUNE HISTORY 命令或 db2Prune API 来删除数据库恢复对象	167
自动管理数据库恢复对象	168
防止恢复对象被删除	168
管理快照备份对象	169

第 10 章 监视备份、复原和恢复操作的进度 171

表空间状态	171
-----------------	-----

第 11 章 backup 概述 173

使用备份	175
执行快照备份	177
将分割镜像用作备份映像	178
备份到磁带	178
备份到命名管道	180
备份分区数据库	180
使用 IBM Tivoli Space Manager 分层存储管理来备份分区表	181
启用自动备份	182
自动备份数据库	183
优化备份性能	183

使用 backup 所需的特权、权限和授权	184
联机备份与其他实用程序的兼容性	184
备份示例	186

第 12 章 recover 概述 187

恢复数据	187
使用 db2adutl 来恢复数据	188
恢复已删除的表	193
崩溃恢复	194
恢复已损坏的表空间	195
恢复可恢复数据库中的表空间	196
恢复不可恢复数据库中的表空间	196
减少介质故障的影响	197
减少事务故障的影响	199
从分区数据库环境中的事务故障进行恢复	199
从数据库分区服务器的故障恢复	201
恢复大型机或中型服务器上的不确定事务	202
灾难恢复	203
版本恢复	204
前滚恢复	205
增量备份与恢复	207
从增量备份映像复原	209
自动增量复原的局限性	210
优化恢复性能	211
使用 recover 所需的特权、权限和授权	212

第 13 章 restore 概述 213

使用复原	213
从快照备份映像复原	214
复原到现有的数据库	215
复原到新的数据库	216
在测试和生产环境中使用增量复原	216
执行重定向复原操作	218
通过使用自动生成的脚本复原数据库来重新定义表空间容器	219
使用自动生成的脚本来执行重定向复原	221
数据库重建	222
重建和表空间容器	225
重建和临时表空间	225
为数据库重建选择目标映像	226
重建所选表空间	229
重建和增量备份映像	230
重建分区数据库	231
数据库重建的限制	232
优化复原性能	232
使用 restore 所需的特权、权限和授权	233
复原示例	233
重定向复原会话 - CLP 示例	233
重建会话 - CLP 示例	235

第 14 章 rollforward 概述 245

使用前滚	246
前滚表空间中的更改	247
rollforward 所需的授权	250
前滚会话 - CLP 示例	250

第 15 章 使用 IBM Tivoli Storage Manager (TSM) 进行数据恢复	255
配置 Tivoli Storage Manager 客户机	255
使用 Tivoli Storage Manager 时的注意事项	256
第 16 章 DB2 高级副本服务 (ACS)	257
启用 DB2 高级副本服务 (ACS)	257
安装 DB2 高级副本服务 (ACS)	258
激活 DB2 高级副本服务 (ACS)	259
配置 DB2 高级副本服务 (ACS)	259
DB2 高级副本服务 (ACS) 设置脚本 setup.sh	261
DB2 高级副本服务 (ACS) API	261
DB2 高级副本服务 (ACS) API 函数	262
DB2 高级副本服务 (ACS) API 数据结构	287
DB2 高级副本服务 (ACS) 最佳实践	301
DB2 高级副本服务 (ACS) 限制	301
DB2 高级副本服务 (ACS) API 返回码	301
DB2 高级副本服务 (ACS) 支持的操作系统和硬件	302
第 3 部分 附录	305

附录 A. DB2 技术信息概述	307
硬拷贝或 PDF 格式的 DB2 技术库	307
订购印刷版的 DB2 书籍	310
从命令行处理器显示 SQL 状态帮助	311
访问不同版本的 DB2 信息中心	311
在 DB2 信息中心中以您的首选语言显示主题	311
更新安装在您的计算机或内部网服务器上的 DB2 信息中心	312
手动更新安装在您的计算机或内部网服务器上的 DB2 信息中心	313
DB2 教程	314
DB2 故障诊断信息	315
条款和条件	315
附录 B. 声明	317
索引	321

关于本书

《数据恢复及高可用性指南与参考》描述了如何使 DB2[®] 数据库 Linux[®] 版、UNIX[®] 版和 Windows[®] 版数据库解决方案高度可用以及如何避免数据丢失。

《数据恢复及高可用性指南与参考》具有两部分：

- 第一部分：高可用性。它描述了用于帮助使数据库解决方案高度可用的策略和 DB2 数据库功能部件和功能。
- 第二部分：数据恢复。它描述了如何使用 DB2 备份和恢复功能以避免数据丢失。

第 1 部分 高可用性

数据库解决方案的可用性是对用户应用程序执行所需数据库任务的效果进行衡量的一种方法。如果用户应用程序无法连接到数据库，或者如果其事务因错误而失败或因系统上的负载而超时，那么该数据库解决方案可用性不高。如果用户应用程序能成功连接到数据库并正常工作，那么该数据库解决方案具有高可用性。

设计高可用性数据库解决方案或者提高现有解决方案的可用性需要了解访问数据库的应用程序的需求。要从额外的存储空间、更快的处理器或更多的软件许可证获得最大利益，重点是使数据库解决方案在企业最重要的应用程序最需要它时尽可能可用。

意外中断

可能影响数据库解决方案的用户可用性的意外系统故障包括：电源中断、网络中断、硬件故障、操作系统错误或其他软件错误，以及出现灾难事件后的系统崩溃。如果出现这样的故障后用户希望仍可使用数据库，高可用性数据库解决方案必须执行以下任务：

- 对用户应用程序屏蔽该故障，使之不受该故障影响。例如，如果某个数据库服务器出现故障，DB2 数据服务器可以将数据库客户机连接重新路由至备用数据库服务器。
- 响应故障以抑制其影响。例如，如果集群中的一台机器出现故障，那么集群管理器可从集群中除去该机器，从而不会再有事务路由至故障机器上进行处理。
- 从故障中恢复以使系统回复正常运行。例如，如果主数据库出现故障之后由备用数据库接管其数据库操作，而原来的主数据库可能会重新启动并恢复过来，再次成为主数据库，重新接管其操作。

执行以上 3 项任务时，要尽量不影响解决方案提供给用户应用程序的可用性。

计划的中断

在高可用性数据库解决方案中，维护活动对用户应用程序的数据库可用性的影响也必须尽量降低。

例如，如果数据库解决方案用于营业时间为上午 9 点至下午 5 点的传统商店柜台，那么维护活动可以在营业时间外脱机进行，而不影响用户应用程序的数据库可用性。如果数据库解决方案用于期望一天 24 小时供客户通过互联网访问的联机银行业务，那么维护活动必须联机运行，或者避开活动高峰期，以尽量减少对客户的数据可用性的影响。

当您就数据库解决方案的可用性作出商务决策或设计选择时，必须权衡以下两个因素：

- 数据库业务对客户不可用的代价
- 实现某种程度的可用性的成本

例如，假定某一基于互联网的企业的数据库解决方案每向客户服务一小时可获取特定收益 X。每年节省 10 小时停机时间的高可用性策略每年可为企业获得 10X 额外收益。如果实现此高可用性策略的成本低于预期的额外收益，那么就值得实现。

第 1 章 中断

中断是对数据库解决方案为用户应用程序服务的功能的任何破坏。中断可以分为下列两组：意外中断和计划的中断。

意外中断

意外中断的示例包括：

- 系统的一个组件的故障，其中包括硬件或软件故障。
- 无效管理或用户应用程序操作，例如，无意中删除对业务至关重要的事务所需的表。
- 由于非最佳配置或者硬件或软件不足导致的性能低下。

计划的中断

计划的中断的示例包括：

- 维护。一些维护活动要求您执行完全中断；其他维护活动可以在不停止数据库的情况下执行，但可对性能产生不良影响。后者是计划的中断的最常见类型。
- 升级。升级软件或硬件有时可能需要执行部分或完全中断。

在有关可用性的讨论中，焦点通常集中在灾难情况或组件故障上。但是，为了设计可靠的高可用性解决方案，需要讨论所有这些类型的中断。

中断特征符

中断特征符是作为中断特征的症状和行为的集合。根据导致最终用户使站点故障结束的响应时间很长的临时性能问题，中断的特征符可能会有变化。当设计用于避免和最小化中断以及从中进行恢复的策略时，请考虑这些变化会如何影响业务。

断电

当系统完全不可用于其最终用户时遇到断电类型中断。此类型中断可能由于硬件、操作系统或数据库级别的问题导致。当断电发生时，必须立即识别中断的作用域。中断完全发生在数据库级别上吗？中断发生在实例级别上吗？或者，它发生在操作系统或硬件级别上吗？

掉电

当系统性能低下至最终用户无法高效地完成其工作时遇到掉电类型中断。系统作为整体可能还处于接通和运行状态，但基本上，从最终用户角度来看，它没有正常运行。在系统维护和高峰使用时间段期间，可发生此类型中断。通常，在此类中断期间，CPU 和内存都接近容量。未适当调整或已过度使用的服务器经常导致掉电。

中断的频率和持续时间

在有关数据库可用性的对话中，焦点通常集中在给定时间段的总量或停机时间百分比（或相反，数据库系统可用的时间量）。但是，计划的中断或意外中断的频率和持续时间在这些中断对业务的影响方面有重大差别。

请考虑以下情况：必须对数据库系统进行将用七小时来执行的一些升级，并且可以选择在用户活动很少的时间段期间每天使数据库系统脱机一小时，也可以选择在最忙日的最忙时间期间使数据库脱机七小时。显然，若干次短时间中断将比一次中断七小时的成本要低并且对业务活动的损害也要小。现在，请考虑以下情况：发生了间歇的网络故障（每星期可能总计几分钟）这导致少量事务以固定频率失败。这些很短时间的中断导致以下局面：损失大量收益并损害客户对您业务的信任，难以再次恢复，这又对将来的收益造成更大损害。

请不要仅关注总的中断（或可用）时间。当对维护活动进行决策或响应意外中断时，请对成本较少时间较长的中断与成本数倍时间较短的中断进行权衡。在中断期间，进行此类决定可能很困难，因此，请创建公式或方法来使用这些中断特征符计算业务成本，以便可进行最佳选择。

多个和级联故障

当设计数据库解决方案来避免和最小化中断以及从中断进行恢复时，请注意多个组件同时发生故障或甚至一个组件的故障导致另一个组件发生故障的可能性。

中断成本

根据业务不同，中断成本会变化。作为最佳实践，每项业务都应该分析其任务的关键业务流程中断的成本。此分析的结果用来制订复原计划。如果标识了多个流程，那么此计划包括复原活动之间的优先级排序。

中断成本

可以估计客户面对不可用来处理客户事务的数据库系统时业务的成本。例如，可以计算在该数据库系统不可用期间每小时或每分钟的失销收益中的平均成本。计算收益中因客户信任减少而导致的预计损失则要困难许多，但是当评估业务的可用性要求时，应该考虑此成本。

还要考虑内部数据库系统不可用于业务流程时的成本。简单如电子邮件或日历软件的一些内容不可用一小时会使业务逐渐停止，因为职员无法完成其工作。

中断容错

根据业务不同，中断容错会变化。作为最佳实践，每项业务都应该分析其任务的关键业务流程中断的影响。此分析的结果用来制订复原计划。如果标识了多个流程，那么此计划包括复原的优先级顺序。

中断容错

确定可用性需要时的关键因素是了解业务容错性如何，或对于业务中的特定系统，则是中断的发生。例如，一个主要运行 Web 站点以发布菜单信息的饭店将不会由于服务器的偶尔中断损失很多收益。另一方面，记录交易的股票交易所服务器的任何中断都将导致重大损失。因此，使用大量资源来确保饭店服务器的可用性为 99.99% 将不合算，但对于股票交易所则肯定合算。

在讨论容错时，应该注意两个概念：恢复时间和恢复点。

恢复时间是使业务流程或系统恢复为联机状态所需的时间。

恢复点是对业务流程或系统进行复原的历史点。在数据库术语中，计划将权衡以下两种复原的优点：丢失一些事务的快速复原与不丢失任何事务但用较长时间来执行的完整复原。

恢复和避免策略

当就可用性方面考虑选择购买还是对系统进行设计时，很容易陷入查阅高可用性功能和技术的冗长列表的困境。但是，有关使系统具有并保持高可用性的最佳实践与下列任务的最佳实践一样多，因为它们与购买技术有关：进行很好的设计和配置选择以及设计并实行合理的管理过程和应急计划。

通过首先确定最适合业务需求的高可用性策略，您将为您的投资获取最全面的可用性。然后，可以选择最适合的技术来实现您的策略。

当设计或配置数据库解决方案以获取高可用性时，请考虑可如何避免中断、最大程度降低其影响以及使系统快速恢复。

避免中断

请尽可能避免中断。例如，除去单个故障点以避免意外中断，或研究用于执行联机维护活动的方法以避免计划的中断。监视数据库系统以识别对问题进行指示的系统行为中的趋势，并在这些问题导致中断之前将其解决。

最小化中断的影响

可以设计并配置数据库解决方案，以最小化计划的中断和意外中断的影响。例如，对数据库解决方案进行分配，以便组件和功能局部化，从而允许某些用户应用程序甚至在一个组件脱机时都继续处理事务。

快速地从意外中断进行恢复

制定恢复计划：创建清楚且存档完好的过程，在发生意外中断的情况下，管理员可以方便快速地遵循这些过程；创建对所涉及系统的所有组件进行描述且层次清楚的文档；就近放置服务协议以及组织良好的联系信息。虽然，快速地进行恢复非常重要，但是还请了解要收集哪些诊断信息，以便确定导致中断的根本原因并避免将来发生此中断。

第 2 章 高可用性策略

对于用户而言，数据库请求失败的原因并不重要。无论是因性能不佳而导致事务超时，还是解决方案的组件出现故障，还是管理员已使数据库脱机以执行维护，对用户而言，结果是一样的：数据库对进程请求不可用。

提高数据库解决方案可用性的策略包括：

冗余 具有解决方案的每个组件的辅助副本，在出现故障时可以接管工作负载。

系统监视

收集有关解决方案的组件的统计信息，以帮助均衡工作负载或检测组件是否出现故障。

负载均衡

将部分工作负载从超负荷的解决方案组件转移到负载较轻的另一解决方案组件。

故障转移

将所有工作负载从出现故障的解决方案组件转移到辅助组件。

使性能最大化

降低事务花很长时间才完成或超时的机率。

尽量降低维护的影响

安排自动维护活动和手动维护活动，尽可能降低对用户应用程序的影响。

通过冗余获取高可用性

如果数据库解决方案的任何一个元素或组件（从电源或网线到操作系统或应用软件）出现故障，最终结果都一样：数据库解决方案对用户应用程序不可用。保持高可用性的重要策略是使用冗余系统，这样，如果一个组件出现故障，该组件的辅助或备份副本可接管故障组件，使数据库保持对用户应用程序可用。

冗余在系统设计中很常见：

- 不间断或备用的电源
- 在每个组件之间运行多个网络纤程
- 网卡的结合度或负载均衡
- 在冗余阵列中使用多个硬盘驱动器
- 使用 CPU 集群

如果系统的这些组件有任何一个不是冗余的，那么该组件将是整个系统中的单一故障点。

可以在数据库级别创建冗余，这需要两个数据库：一是主数据库，通常情况下处理所有或大部分应用程序工作负载；二是辅助数据库，在主数据库出现故障时可接管其工作负载。在 DB2 高可用性灾难恢复（HADR）环境中，此辅助数据库被称为备用数据库。

通过故障转移获取高可用性

故障转移是指在主系统出现故障时将工作负载从主系统转移到辅助系统。当工作负载以这种方式转移后，我们称辅助系统已接管故障主系统的工作负载。

示例 1

在集群环境中，如果集群中的一个机器出现故障，集群管理软件可以将正在故障机器上运行的进程移至集群中的另一机器。

示例 2

在具有多个 IBM® 数据服务器的数据库解决方案中，一旦某个数据库不可用，那么不再可用的数据库服务器上所连接的数据库应用程序可由数据库管理器重新路由至辅助数据库服务器。

市场上最常见的两种故障转移策略称为空闲备用和相互接管：

空闲备用

在这种配置中，主系统处理所有工作负载，而辅助或备用系统则处于空闲或备用方式，准备在主系统上出现故障时接管工作负载。在 HADR 设置中，还可为备用系统配置为允许只读工作负载。

相互接管

在这种配置中，有多个系统，每个系统都是另一系统的指定辅助系统。一旦某个系统出现故障，该系统所指定的辅助系统必须为出现故障的系统接管工作负载，同时继续处理自己的工作负载，这对总体性能有负面影响。

通过集群获取高可用性

一个集群就是一组互相连接的机器，可一起工作，相当于单个系统。当集群的一个机器出现故障时，集群管理软件会将故障机器上的工作负载转移到其他机器上。

脉动信号监视

要检测集群中一台机器上的故障，故障转移软件可以在机器之间使用脉动信号监视或保持活动包来确定可用性。脉动信号监视需要用到一些系统服务，以在集群中所有机器之间维持连续通信。如果未检测到脉动信号，就会对备份机器启动故障转移。

IP 地址接管

当集群中的一台机器上出现故障时，集群管理器可以通过将 IP 地址从一台机器转移到另一台机器上将工作负载从一台机器上转移到另一机器上。这被称为 IP 地址接管或 IP 接管。客户机应用程序是看不到这种转移的，所以仍会继续使用原来的 IP 地址，而并不知道与该 IP 地址映射的物理机器已更改。

DB2 高可用性 (HA) 功能部件允许 IBM 数据服务器与集群管理软件集成。

数据库日志记录

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。如果需要将数据库复原到上一完整、脱机备份之前的一个点，那么需要日志才能将数据前滚至故障点。数据库日志记录是高可用的数据库解决方案设计的重要部分，因为从故障中恢复过来，以及同步主数据库和辅助数据库都需要用到数据库日志。

IBM 数据服务器支持两种类型的日志记录：循环和归档。每种类型都提供了不同水平的恢复功能：

- 『循环日志记录』
- 『归档日志记录』

选择归档日志记录的好处是前滚恢复可以使用归档日志和活动日志将数据库复原到日志的结尾或特定时间点。归档日志文件可用于恢复备份完成之后所进行的更改。这不同于循环日志记录，在其中只能恢复至备份时间，并且在该时间之后进行的所有更改都会丢失。

循环日志记录

当创建新数据库时，循环日志记录是缺省行为。（将 *logarchmeth1* 和 *logarchmeth2* 数据库配置参数设置为 OFF。）对于这种类型的日志记录，只允许完整的脱机数据库备份。进行完整备份时，数据库必须脱机（用户不可访问）。

正如它的名称所表示的那样，循环日志记录使用一个联机日志“环”，提供对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。循环日志记录不允许将数据库在上次完整备份操作后执行的事务中前滚。上次备份操作后发生的所有更改都将丢失。因为这种类型的复原操作将数据恢复至进行完整备份的特定时间点，所以它称为版本恢复。

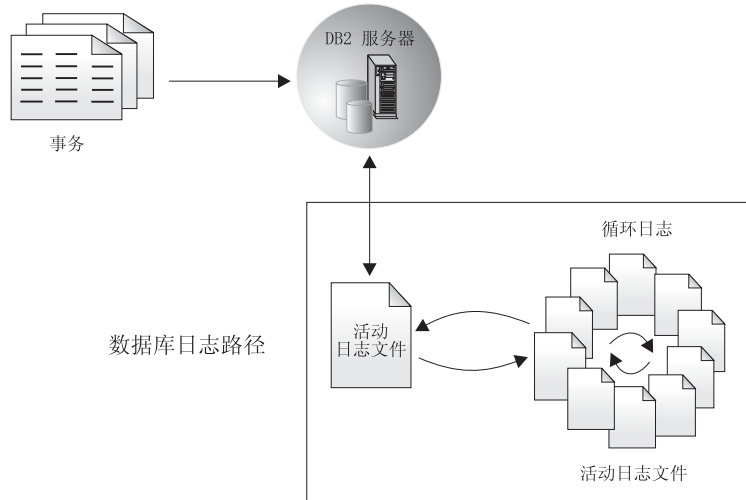
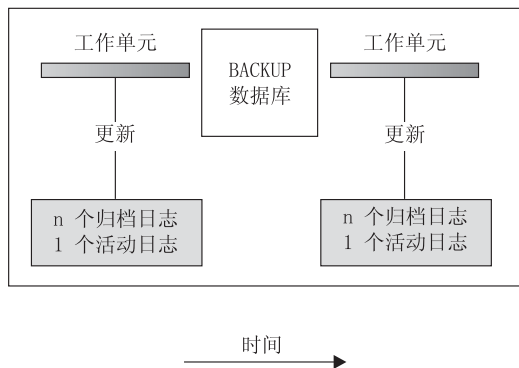


图 1. 循环日志记录

崩溃恢复期间，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。活动日志位于数据库日志路径目录中。

归档日志记录

归档日志记录是针对前滚恢复使用的。归档日志是曾经活动的，但不再是崩溃恢复所必需的日志。使用 *logarchmeth1* 数据库配置参数来启用归档日志记录。



在备份之间使用日志以跟踪对数据库的更改。

图 2. 前滚恢复中活动的和已归档的数据库日志. 在运行时间较长的事务中, 可以有多个活动日志。

仅当将数据库配置为归档日志记录时, 才支持进行联机备份。在联机备份操作期间, 将记录对数据库的所有活动。复原联机备份映像时, 必须至少将日志前滚至完成备份操作的时间点。为此, 日志在复原数据库时必须已归档并可用。在联机备份完成后, DB2 将强制关闭当前活动日志, 从而使日志归档。这样就可确保您的联机备份有一组完整的归档日志可用于恢复。

以下数据库配置参数允许您更改归档日志的存储位置: *newlogpath* 参数和 *logarchmeth1* 与 *logarchmeth2* 参数。更改 *newlogpath* 参数还会影响活动日志的存储位置。

要确定数据库日志日志路径目录中的哪些日志扩展数据块是归档日志, 应检查数据库配置参数 *loghead* 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”。关于此配置参数的更多信息, 请参阅 *Performance Guide* 一书。

日志控制文件

当数据库在出现故障后重新启动时, 数据库管理器应用存储在日志文件中的事务信息将数据库返回至一致状态。要确定需要将日志文件中的哪些记录应用于该数据库, 数据库管理器使用日志控制文件中记录的信息。

数据库弹性冗余

数据库管理器维护日志控制文件的两个副本, `SQLLOGCTL.LFH.1` 和 `SQLLOGCTL.LFH.2`, 以便当一个副本损坏时, 数据库管理器还可使用另一副本。

性能注意事项

应用日志控制文件中包含的事务信息会增加出现故障后重新启动数据库的开销。可以使用《数据服务器、数据库和数据库对象指南》中的『*softmax* - 恢复范围和软检查点时间间隔配置参数』来配置数据库管理器将事务写入磁盘的频率, 以减少崩溃恢复期间需要处理的日志记录数。

第 3 章 通过 IBM 数据服务器获取高可用性

IBM Data Server 支持多种高可用性策略。

客户机自动重新路由路线图

客户机自动重新路由是 IBM 数据服务器的一种特有功能，可将客户机应用程序从出现故障的服务器重定向至备用服务器，这样可以尽量减少中断，让应用程序得以继续运行。仅当在故障前已指定了备用服务器时，才能实现客户机自动重新路由。

表 1 列示了每个类别中的相关主题。

表 1. 客户机自动重新路由信息指南

类别	相关主题
一般信息	<ul style="list-style-type: none">第 21 页的『客户机自动重新路由的限制』第 17 页的『描述和设置客户机自动重新路由』《DB2[®] Connect[™] 服务器快速入门》中的『客户机自动重新路由的描述和设置 (DB2 Connect)』
配置	<ul style="list-style-type: none">第 21 页的『标识用于客户机自动重新路由的备用服务器』第 19 页的『使用注册表变量来配置客户机自动重新路由重试行为』第 21 页的『使用 IBM 数据服务器 JDBC 和 SQLJ 驱动程序时的客户机重新路由设置』
示例	<ul style="list-style-type: none">第 142 页的『客户机自动重新路由示例』
与其他 DB2 功能部件进行交互	<ul style="list-style-type: none">第 28 页的『配置客户机自动重新路由和高可用性灾难恢复 (HADR)』第 19 页的『将客户机连接超时与客户机自动重新路由配合使用』《开发 Java[™] 应用程序》中的『IBM 数据服务器 JDBC 和 SQLJ 驱动程序客户机重新路由支持』
故障诊断	<ul style="list-style-type: none">第 20 页的『配置客户机自动重新路由以将其与客户机连接分发器技术配合使用』

注: IBM 数据服务器客户机和非 Java IBM 数据服务器驱动程序中也提供了用于 DB2 z/OS[®] 版综合系统 (sysplex) 的客户机自动重新路由功能。借助此支持, 访问 DB2 z/OS 版综合系统 (sysplex) 的应用程序可以使用客户机提供的客户机自动重新路由功能, 而不需要检查 DB2 Connect 服务器。有关此功能的更多信息, 请参阅“DB2 信息中心”中有关客户机自动重新路由 (客户端) 功能的主题。

用于 Linux 和 UNIX 的 DB2 故障监视器工具

DB2 故障监视器工具仅在基于 UNIX 的系统上可用, 通过监视 DB2 数据库管理器实例并重新启动任何过早退出的实例来使 DB2 数据服务器数据库正常运行。

故障监视器协调程序 (FMC) 是在 UNIX 引导序列中启动的“故障监视器工具”的进程。*init* 守护程序启动 FMC, 并在 FMC 异常终止时重新启动 FMC。FMC 对每个 DB2 实例启动一个故障监视器。每个故障监视器都作为一个守护进程来运行, 并且具有与 DB2 实例相同的用户特权。

一旦启动了故障监视器, 就会监视 DB2 实例以确保它不会过早退出。如果故障监视器发生故障, 那么将通过 FMC 将它重新启动。每个故障监视器将依次负责监视一个 DB2 实例。如果 DB2 实例过早退出, 那么故障监视器会将其重新启动。仅当发出 `db2stop` 命令时, 故障监视器才变为不活动状态。如果 DB2 实例以任何其他方式关闭, 故障监视器都会将其再次启动。

DB2 故障监视器限制

由于实例的启动和关闭是由集群产品控制的, 所以, 如果正在使用高可用性集群产品 (例如, HACMP™、MSCS 或 IBM Tivoli® System Automation for titolatforms), 那么必须关闭故障监视器工具。

DB2 故障监视器与 DB2 运行状况监视器之间的差别

运行状况监视器和故障监视器是作用于单个数据库实例的工具。运行状况监视器使用运行状况指示器来评估数据库管理器性能或数据库性能特定方面的运行状况。运行状况指示器测量特定种类数据库对象 (例如表空间) 某些方面的运行状况。可以针对特定条件评估运行状况指示器以确定该类数据库对象的运行状况。此外, 当某个运行状况指示器超出阈值或指示某个数据库对象处于非正常状态时, 运行状况指示器可以生成警报以通知您。

与之对比, 故障监视器仅仅负责保持它所监视的实例正常运行。如果它所监视的 DB2 实例意外终止, 那么故障监视器就会重新启动该实例。故障监视器在 Windows 上不可用。

高可用性灾难恢复 (HADR)

DB2 数据服务器高可用性灾难恢复 (HADR) 功能是一种数据库复制功能, 它提供针对部分站点故障和整个站点故障的高可用性解决方案。HADR 通过将数据更改从源数据库 (称为主数据库) 复制到目标数据库 (称为备用数据库) 来防止数据丢失。

如果大多数或所有数据库需要保护, 或者如果执行必须在备用数据库上自动复制的 DDL 操作时, HADR 可能是最佳选择。

应用程序只能访问当前主数据库。通过前滚在主数据库上生成并交付给备用数据库的日志数据来对备用数据库进行更新。

部分站点故障可能是由硬件、网络或软件 (DB2 数据库系统或操作系统) 故障引起的。如果没有 HADR, 发生部分站点故障时就需要重新启动数据库所在的数据库管理系统 (DBMS) 服务器。重新启动数据库和数据库所在的服务器所需的时间长度是不可预测的。可能在几分钟时间后, 数据库才会恢复为一致状态并可用。使用 HADR, 那么备用数据库可在数秒内接管。另外, 还可以通过使用客户机自动重新路由功能, 或重试应用程序中的逻辑, 将正在使用原始主数据库的客户机重定向至备用数据库 (新的主数据库)。

当由于灾难 (例如, 火灾) 而导致整个站点被破坏时, 就可能会发生整个站点故障。因为 HADR 使用 TCP/IP 在主数据库和备用数据库之间进行通信, 所以数据库可以位

于不同位置。例如，主数据库可能位于某个城市的总部，而备用数据库位于另一城市的销售办事处。如果在主要站点发生了灾难，那么可以通过让远程备用数据库接管具有所有 DB2 功能的主数据库来维护数据可用性。执行接管操作之后，可以备份原始主数据库，并将其返回至主数据库状态；这即是所谓的故障回退。

使用 HADR，通过指定三种同步方式之一：同步、接近同步或异步，可以选择要对可能丢失的数据所给予保护的级别。

如果数据库的两个备份可以一致，发生故障的原始主服务器修复之后，可以作为备用数据库重新加入 HADR 数据库对。原始主数据库作为备用数据库重新集成到 HADR 数据库对之后，可以切换数据库角色，以再次将原始主数据库启用为主数据库。

HADR 只是 DB2 产品系列中提供的若干复制解决方案之一。WebSphere® Federation Server 和 DB2 数据库系统包括 SQL 复制和 Q 复制解决方案，在某些配置中也可以使用这些解决方案来提供高可用性。这些功能在多个位置维护逻辑上一致的数据库表副本。另外，它们还提供灵活性和复杂功能，如支持列和行过滤、数据变换、任何表副本的更新，并且可在分区数据库环境中使用。

DB2 高可用性 (HA) 功能部件

DB2 高可用性 (HA) 功能部件允许 IBM 数据服务器与集群管理软件集成。

当在集群环境中停止数据库管理器实例时，必须使集群管理器知道实例已停止。如果集群管理器不知道实例已停止，那么集群管理器可能尝试操作，例如对已停止的实例进行故障转移。当诸如停止数据库管理器实例之类的实例配置更改需要集群更改时，DB2 高可用性 (HA) 功能部件提供了基础结构，用于使数据库管理器能够与集群管理器通信。

每当实例更改需要集群更改时，如果数据库管理器与集群管理器通信，那么您在执行实例配置更改后无需执行单独的集群操作。

DB2 HA 功能部件由下列元素组成：

- IBM Tivoli System Automation for Multiplatforms (SA MP) 与 AIX® 和 Linux 上的 IBM 数据服务器捆绑在一起，作为 DB2 高可用性 (HA) 功能部件的一部分，并且与 DB2 安装程序集成。您可以使用 DB2 安装程序或 IBM 数据服务器安装介质中包含的 installSAM 和 uninstallSAM 脚本来安装、升级或卸载 SA MP。
- 在集群环境中，某些数据库管理器实例配置和管理操作需要相关集群配置更改。每当您执行某些数据库管理器实例配置和管理操作时，DB2 高可用性 (HA) 功能部件将允许数据库管理器自动请求集群管理器配置更改。请参阅：第 59 页的『使用 DB2 高可用性 (HA) 功能部件来自动配置集群』
- DB2 高可用性实例配置实用程序 (db2haicu) 是基于文本的实用程序，您可以使用它在集群环境中配置与管理高可用性数据库。db2haicu 通过查询系统来收集有关数据库实例、集群环境和集群管理器的信息。您可通过参数向 db2haicu 调用、输入文件提供更多信息，或在运行时通过在 db2haicu 提示符处提供信息来实现上述目的。请参阅：第 66 页的『DB2 高可用性实例配置实用程序 (db2haicu)』
- DB2 集群管理器 API 定义了一组功能，它们使数据库管理器能够向集群管理器通知配置更改。请参阅：第 100 页的『DB2 集群管理器 API』

通过日志装入获取高可用性

日志装入是将整个日志文件复制到备用机器上的过程，可以从归档设备上复制，也可以通过主数据库运行的用户出口程序来复制。

备用数据库根据生产机器生成的日志文件不断前滚。当生产机器发生故障时，发生故障转移并出现下列情况：

- 余下日志被传送到备用机器。
- 备用数据库前滚至日志的末尾，然后停止。
- 客户机与备用数据库重新连接，并继续运行。

备用机器有它自己的资源（例如磁盘），但必须具有与生产数据库相同的物理和逻辑定义。当使用这种方法时，通过使用 **RESTORE** 实用程序或分割镜像功能将主数据库复原到备用机器上。

要确保能够在灾难恢复情况下恢复数据库，请考虑下列事宜：

- 归档位置应该在地理上独立于主位置。
- 在备用数据库位置远程镜像日志
- 使用同步镜像，以避免失去支持。可以使用现代磁盘子系统（例如 **ESS** 和 **EMC**）或其他远程镜像技术来完成此任务。同时建议使用 **NVRAM** 高速缓存（本地和远程）来最小化灾难恢复情况下对性能的影响。

注：

1. 当备用数据库处理指示在主数据库上发生索引重建的日志记录时，备用服务器上的索引将不会自动重建。在备用服务器脱离前滚暂挂状态之后，与数据库的首次连接或首次尝试访问索引时，将在备用服务器上重建该索引。如果在主服务器上重建了任何索引，那么建议将备用服务器与主服务器重新同步。如果设置了 **LOGINDEXBUILD** 数据库配置参数，那么可以在前滚操作期间重建索引。
2. 如果 **LOAD** 实用程序在主数据库上运行并指定了 **COPY YES** 选项，备用数据库必须具有对副本映像的访问权。
3. 如果 **LOAD** 实用程序在主数据库上运行并指定了 **COPY NO** 选项，那么应该重新同步备用数据库，否则表空间将处于复原暂挂状态。
4. 有两种方法来初始化备用机器：
 - a. 通过从备份映像复原到备用机器。
 - b. 通过创建生产系统的分割镜像并发出指定了 **STANDBY** 选项的 **db2inidb** 命令。
仅在已初始化备用机器之后，才能在备用系统上发出 **ROLLFORWARD** 命令。
5. 未记录的操作将不会在备用数据库上重放。这样，建议在这类操作之后重新同步备用数据库。可以通过联机分割镜像和暂挂 I/O 支持来完成此任务。

日志镜像

IBM 数据服务器支持数据库级别的日志镜像。镜像日志文件有助于防止数据库在无意中删除活动日志，并且使数据库中的数据不会由于硬件故障而毁坏。

如果担心活动日志可能已损坏（由于磁盘崩溃所导致），应该考虑使用 **MIRRORLOGPATH** 配置参数指定数据库辅助路径来管理活动日志副本，从而镜像存储这些日志的卷。

MIRRORLOGPATH 配置参数允许数据库将完全相同的第二份日志文件写至另一路径。建议您将辅助日志路径设置到在物理上独立的磁盘（最好该磁盘也在另一磁盘控制器上）。在那种情况下，磁盘控制器不能是单个故障点。

首次启用 MIRRORLOGPATH 时，在下次数据库启动前并未实际使用。这与 NEWLOGPATH 配置参数相似。

如果写至活动日志路径或镜像日志路径时出错，数据库会将有问题的路径标记为“坏”，并将消息写至到管理通知日志，然后仅将后续的日志记录写至其他“好”日志路径中。DB2 将不会再次尝试使用“坏”路径，直到当前日志文件已满或被截断为止。当 DB2 需要打开下一个日志文件时，将会验证此路径是否有效，如果有效就开始使用。如果无效，DB2 将不会尝试再次使用该路径，直到下一个日志文件被第一次访问时为止。不会尝试使日志路径同步，但 DB2 保留了关于发生的访问错误的信息，因此在归档日志文件时可使用正确的路径。如果在写至余下“好”路径时出现故障，那么数据库关闭。

通过暂挂 I/O 和联机分割镜像支持获取高可用性

通过使用 IBM 数据服务器的暂挂 I/O 支持，可以分割主数据库的镜像副本，而不必使该数据库脱机。利用这种方法，可以在主数据库出现故障时迅速创建一个备用数据库来接管操作。

磁盘镜像是将数据同时写入两个单独的硬盘中的进程。一个数据副本是另一个数据副本的镜像。分割镜像是分离两个副本的进程。

可以使用磁盘镜像来维护主数据库的辅助副本。可以使用 IBM 数据服务器的暂挂 I/O 功能来分割数据库的主镜像副本和辅助镜像副本，而不必使该数据库脱机。分割主数据库副本和辅助数据库副本后，在主数据库出现故障时，辅助数据库可以接管操作。

如果不希望使用 IBM 数据服务器的 BACKUP 实用程序来备份大型数据库，可通过使用暂挂 I/O 和分割镜像功能根据镜像映像来建立副本。此方法还可以：

- 消除来自生产机器的备份操作开销
- 提供了克隆系统的一个快捷方式
- 提供了对空闲备用故障转移的快速实现。不需要初始复原操作，如果证实前滚操作太慢或遇到了错误，重新初始化会非常快。

db2inidb 命令初始化分割镜像，因此可用来：

- 作为克隆数据库
- 作为备用数据库
- 作为备份映像

只能对分割镜像发出此命令，必须首先运行该命令才能使用分割镜像。

在分区数据库环境中，不必同时对所有数据库分区暂挂 I/O 写操作。可以暂挂由一个或多个数据库分区组成的数据库分区子集来创建用于执行脱机备份的分割镜像。如果该子集包括目录分区，那么它必须是要暂挂的最后一个数据库分区。

在分区数据库环境中，必须先对每个数据库分区运行 db2inidb 命令，然后才能使用根据任何这些数据库分区创建的分割映像。可以使用 db2_all 命令来同时对所有数据库分区运行此工具。但是，如果使用了 RELOCATE USING 选项，那么无法使用 db2_all 命令来同时对所有数据库分区运行 db2inidb。必须为每个数据库分区提供独立的配置文件，

该配置文件包含所更改的数据库分区的 NODENUM 值。例如，如果要更改数据库的名称，这将影响每个数据库分区，并且必须在每个数据库分区上都有独立配置文件的情况下运行 `db2relocatedb` 命令。如果要移动属于单一数据库分区的容器，那么只需要在该数据库分区上运行一次 `db2relocatedb` 命令。

注： 确保分割镜像包含组成数据库的所有容器和目录（包括卷目录）。要收集此信息，请参阅 `DBPATHS` 管理视图，该视图显示了需要分割的数据库的所有文件和目录。

第 4 章 为获取高可用性进行配置

要配置 DB2 数据库解决方案以获取高可用性，您必须：安排数据库维护活动；配置主数据库服务器和备用数据库服务器，使它们相互了解并了解在出现故障时各自的角色；以及配置任何集群管理软件以从故障集群节点转移工作负载。

在配置数据库解决方案之前：

- 组合和安装组成解决方案的底层硬件和软件组件。这些底层组件可能包括：电源；网络连接；网卡；磁盘或其他存储设备；操作系统；以及集群管理软件。
- 在尝试使用这些底层组件进行数据库负载均衡、故障转移或恢复操作之前，在没有任何数据库工作负载的情况下测试这些底层组件，以确保它们工作正常。

冗余是高可用性解决方案的重要组成部分。但是，如果您不明智地安排维护，如果用完了恢复日志所需的存储空间，或者如果集群管理软件未正确配置，那么当用户需要使用数据库完成关键工作时，解决方案可能会不可用。

为获取高可用性进行配置包括：

- 配置客户机重新路由
 - 『描述和设置客户机自动重新路由』
- 配置故障监视器
 - 第 23 页的『DB2 故障监视器注册表文件』
- 配置 DB2 高可用性灾难恢复
 - 第 26 页的『初始化高可用性灾难恢复 (HADR)』
- 安排维护活动
 - 第 45 页的『为获取高可用性安排维护』
- 配置日志记录
 - 第 47 页的『配置数据库日志记录选项』
- 配置集群管理软件
 - 第 57 页的『为获取高可用性配置集群环境』

描述和设置客户机自动重新路由

客户机自动重新路由功能的主要目标是使 IBM 数据服务器客户机应用程序能够恢复通信，以便应用程序可以继续工作，并将中断减至最低。顾名思义，支持连续操作的核心在于重新路由。但是只有存在已向客户机连接标识的备用位置时，才能进行重新路由。

如果服务器是 Linux、UNIX 或 Windows 上的 DB2，那么可以在下列可配置环境中使用客户机自动重新路由功能：

1. 具有数据库分区功能 (DPF) 的企业服务器版 (ESE)

2. WebSphere Replication Server
3. 高可用性集群多处理器 (HACMP)
4. 高可用性灾难恢复 (HADR)。

客户机自动重新路由与 HADR 配合工作，允许客户机应用程序在访问数据库故障转移之后继续其工作，并将中断减至最少。

如果数据库服务器位于 System i[®] 或 System z[®] 上，那么也可以在下列配置中使用客户机自动重新路由功能：

1. IBM 数据服务器客户机通过具有备用服务器的 DB2 Connect 服务器连接至 z/OS 或 i5/OS[®] 系统。将在 IBM 数据服务器客户机与两台 DB2 Connect 服务器之间使用客户机自动重新路由功能。
2. 访问 DB2 z/OS 版综合系统 (sysplex) 数据共享环境的 DB2 Connect 个人或服务器产品。将在 DB2 Connect 与 z/OS 综合系统 (sysplex) 之间使用客户机自动重新路由功能。客户机自动重新路由功能支持 DB2 Connect 许可的客户机与综合系统 (sysplex) 之间的无缝故障转移。有关无缝故障转移的更多信息，请参阅“DB2 信息中心”中有关客户机自动重新路由 (客户端) 功能的主题。

对于 DB2 Connect 服务器及其备用服务器，因为不要求同步本地数据库，所以只需要确保原始和备用 DB2 Connect 服务器按以下方式编目目标主机或 System i 数据库，即可使用完全相同的数据库别名进行访问。

要使 DB2 数据库系统能够恢复通信，必须在通信中断之前指定备用服务器位置。使用 UPDATE ALTERNATE SERVER FOR DATABASE 命令来定义特定数据库上的备用服务器位置。

在服务器实例的特定数据库上指定备用服务器位置后，备用服务器位置信息将在连接过程中返回至 IBM 数据服务器客户机。在 DB2 Connect 个人或服务器产品与主机或 System i 数据库服务器之间使用客户机自动重新路由功能时，远程服务器必须为自己提供一个或多个备用地址。对于 DB2 z/OS 版，如果数据库是综合系统 (sysplex) 数据共享环境，那么多个地址已知。因此，不需要在 DB2 Connect 上编目备用服务器。如果客户机与服务器之间的通信因某种原因而中断，那么 IBM 数据服务器客户机将使用备用服务器信息尝试重新建立连接。IBM 数据服务器客户机将尝试与数据库服务器重新建立连接，该数据库服务器可能是原始服务器、服务器的数据库目录文件中列示的备用服务器或 z/OS 综合系统 (sysplex) 所返回的服务器列表中的备用服务器。最初可能尝试非常快速地重新建立连接，随后两次尝试之间的时间间隔逐渐增加，所以这些尝试的计时不同。

一旦连接成功，那么返回 SQLCODE -30108 以指示通信失败后已重新建立数据库连接。将返回主机名或 IP 地址以及服务名称或端口号。如果不能重新建立客户机与原始服务器或备用服务器之间的通信，那么 IBM 数据服务器客户机仅向应用程序返回表示原始通信故障的错误。

还应该注意下列在 DB2 Connect 服务器环境中进行备用服务器连接所涉及的注意事项：

- 使用 DB2 Connect 服务器代表远程和本地客户机访问主机或 System i 数据库时，系统数据库目录条目中的备用服务器连接信息可能会变得混乱。为使混乱程度降至最低，请考虑在系统数据库目录中编目两个条目来表示同一主机或 System i 数据库。为远程客户机编目一个条目，为本地客户机编目另一个条目。

- 从目标 DB2 z/OS 版服务器返回的任何综合系统 (sysplex) 信息仅保存在 DB2 Connect 服务器的高速缓存中。只有一个备用服务器写入磁盘。存在多个备用服务器或活动服务器时, 仅在内存中维护信息并且这些信息在进程终止时将丢失。

通常情况下, 如果指定备用服务器, 当检测到通信错误 (sqlcode -30081) 或 sqlcode -1224 时会启用客户机自动重新路由。但是, 在高可用性灾难恢复 (HADR) 环境中, 如果从 HADR 备用服务器返回 sqlcode -1776, 也会启用客户机自动重新路由。

使用注册表变量来配置客户机自动重新路由重试行为

当主数据库服务器出现故障时, DB2 客户机自动重新路由功能会将客户机应用程序请求传送至辅助数据库服务器。然后, DB2 客户机重新路由重复地尝试重新连接至主数据库。可以配置 DB2 客户机重新路由尝试重新连接的最大次数, 并配置连续两次重新连接尝试之间的休眠时间。

缺省情况下, 客户机自动重新路由功能最多 10 分钟后会重复地重新尝试连接数据库。但也可以使用下列两个注册表变量中的一个或全部来配置精确的重试行为:

- `DB2_MAX_CLIENT_CONNRETRIES`: 客户机自动重新路由尝试的最大连接重试次数。
- `DB2_CONNRETRIES_INTERVAL`: 连续连接重试之间的休眠时间 (以秒计)。

如果设置了 `DB2_MAX_CLIENT_CONNRETRIES`, 但未设置 `DB2_CONNRETRIES_INTERVAL`, 那么 `DB2_CONNRETRIES_INTERVAL` 缺省为 30。

如果未设置 `DB2_MAX_CLIENT_CONNRETRIES`, 但设置了 `DB2_CONNRETRIES_INTERVAL`, 那么 `DB2_MAX_CLIENT_CONNRETRIES` 缺省为 10。

如果既未设置 `DB2_MAX_CLIENT_CONNRETRIES` 也未设置 `DB2_CONNRETRIES_INTERVAL`, 那么客户机自动重新路由功能将恢复为采用先前所描述的缺省行为。

注:

- 带有 DB2 通用 JDBC 驱动程序的类型 4 连接的用户应使用下面两个数据源属性来配置客户机自动重新路由:
 - `maxRetriesForClientReroute`: 使用此属性来限制至服务器的主连接失败时重试的次数。仅当也设置了 `retryIntervalClientReroute` 属性时才使用此属性。
 - `retryIntervalForClientReroute`: 使用此属性来指定再次重试前休眠的时间 (以秒计)。仅当也设置了 `maxRetriesForClientReroute` 属性时才使用此属性。
- 当对访问 DB2 z/OS 版综合系统 (sysplex) 的客户机启用了客户机自动重新路由功能时 (当 `db2dsdriver` 配置文件中 `enableAcr` 为 `True` 时), 设置 `DB2_MAX_CLIENT_CONNRETRIES` 和 `DB2_CONNRETRIES_INTERVAL` 不生效。有关更多信息, 请参阅“DB2 信息中心”中有关客户机自动重新路由 (客户端) 功能的主题。

将客户机连接超时与客户机自动重新路由配合使用

在标准 DB2 数据库解决方案实现中, 客户机应用程序将尝试连接至数据库服务器, 直至成功或直到经过了所配置的连接超时时间。如果连接超时, 那么会有一个通信错误

返回给应用程序。如果正在使用 DB2 客户机自动重新路由，那么 DB2 会将客户机应用程序连接重定向至备用数据库服务器，而不是生成通信错误。

对于 CLI/ODBC、OLE DB 和 ADO.NET 应用程序，可以设置连接超时值并指定当客户机应用程序尝试与服务器建立连接时，在终止连接尝试并生成通信超时之前，它将等待该设置的秒数以获取应答。

如果启用了客户机重新路由，那么需要将连接超时值设置为一个大于或等于连接至服务器所用的最长时间的值。否则，连接可能会超时并通过客户机重新路由来重新路由至备用服务器。例如，在正常的一天里客户机连接至服务器需要 10 秒，在繁忙的一天需要 20 秒，那么连接超时值应至少设置为 20 秒。

配置客户机自动重新路由以将其与客户机连接分发器技术配合使用

在主数据库服务器出现故障时，分发器或分派器技术（如 WebSphere EdgeServer）会将客户机应用程序重新连接请求分发至一组已定义的系统。如果要将分发器技术与 DB2 客户机自动重新路由配合使用，那么必须将分发器本身标识为 DB2 客户机自动重新路由的备用服务器。

可以在类似下面的环境中使用分发器技术：

客户机 → 分发器技术 → (DB2 Connect 服务器 1 或 DB2 Connect 服务器 2) → DB2 z/OS

其中：

- 分发器技术组件具有 TCP/IP 主机名 **DThostname**
- DB2 Connect 服务器 1 具有 TCP/IP 主机名 **GWYhostname1**
- DB2 Connect 服务器 2 具有 TCP/IP 主机名 **GWYhostname2**
- DB2 z/OS 服务器具有 TCP/IP 主机名 **zOShostname**

客户机是使用 **DThostname** 来进行编目的，以便利用分发器技术来访问任何一个 DB2 Connect 服务器。加入分发器技术就可以决定是使用 **GWYhostname1** 还是 **GWYhostname2**。一旦作出了决定，客户机就可与这两个 DB2 Connect 网关的其中一个网关进行直接套接字连接。一旦与所选择的 DB2 Connect 服务器建立了套接字连接，您就建立了客户机至 DB2 Connect 服务器，该服务器至 DB2 z/OS 的典型连接。

例如，假定分发器选择 **GWYhostname2**。这将生成以下环境：

客户机 → DB2 Connect 服务器 2 → DB2 z/OS

如果发生任何通信故障，分发器不会重试任何连接。如果想在这样的环境中对数据库启用客户机自动重新路由功能，那么应将 DB2 Connect 服务器（DB2 Connect 服务器 1 或 DB2 Connect 服务器 2）中相关联的一个或多个数据库的备用服务器设置为分发器（**DThostname**）。然后，如果 DB2 Connect 服务器 1 由于任何原因而锁定，那么会触发客户机自动重新路由，并且在分发器同时作为主服务器和备用服务器的情况下重试客户机连接。此选项允许您使用 DB2 客户机自动重新路由功能来组合和维护分发器功能。将备用服务器设置为除了分发器主机名以外的主机仍然会为客户机提供客户机自动重新路由功能。但是，客户机将与已定义的备用服务器建立直接连接，从而未使用分发器技术，这就消除了分发器及其所带来的价值。

客户机自动重新路由功能拦截下列 SQL 代码：

- sqlcode -20157
- sqlcode -1768 (原因码 = 7)

注: 如果“TCP Keepalive”操作系统配置参数的设置太高, 那么可能无法将套接字故障及时通知客户机重新路由。(注意, 此配置参数的名称随平台而改变。)

标识用于客户机自动重新路由的备用服务器

每当 DB2 服务器或 DB2 Connect 服务器崩溃时, 与该服务器连接的每台客户机都会接收到通信错误, 该错误将终止连接, 并归结为应用程序错误。如果可用性极其重要, 那么应实现冗余设置或具备将服务器故障转移到备用节点的功能。在上述任一情况下, DB2 客户机节点尝试重新建立与原始服务器的连接, 此服务器可能正在故障转移节点上运行 (IP 地址也进行故障转移), 或者建立与新服务器的连接。

要定义新服务器或备用服务器:

使用 **UPDATE ALTERNATE SERVER FOR DATABASE** 或 **UPDATE ALTERNATE SERVER FOR LDAP DATABASE** 命令。

这些命令将在系统数据库目录中更新数据库别名的备用服务器信息。

使用 IBM 数据服务器 JDBC 和 SQLJ 驱动程序时的客户机重新路由设置

每当服务器崩溃时, 连接至该服务器的每台客户机都会接收到通信错误, 该错误将终止连接并导致应用程序错误。如果可用性很重要, 那么您应该具有冗余设置或故障转移支持。故障转移是在服务器出现故障时另一台服务器接管操作的功能。当连接终止时, IBM 数据服务器 JDBC 和 SQLJ 驱动程序客户机将尝试与原始服务器 (此服务器可能正在故障转移节点上运行) 重新建立连接或者与新服务器重新建立连接。重新建立连接时, 应用程序将接收到 `SQLException`, 此异常使应用程序知道事务出现故障, 但应用程序可以继续下一个事务。

有关详细信息, 请参阅下面的“相关概念”部分中有关 IBM 数据服务器 JDBC 和 SQLJ 驱动程序的客户机重新路由支持的主题。

客户机自动重新路由的限制

在设计高可用性 DB2 数据库解决方案时, 请考虑 DB2 数据库客户机重新路由限制。

以下是 DB2 数据库客户机自动重新路由功能的限制列表:

- 仅当用于连接 DB2 数据库服务器或 DB2 Connect 服务器的通信协议为 TCP/IP 时, 才支持客户机自动重新路由。这意味着, 如果连接使用的协议不是 TCP/IP, 将不会启用客户机自动重新路由功能。即使设置 DB2 数据库以便进行回送, 仍然必须使用 TCP/IP 通信协议以适应客户机自动重新路由功能。
- 在 DB2 Connect 个人或服务器产品与主机或 System i 数据库服务器之间使用自动重新路由功能时, 如果您处于下列情况中, 那么将受到一些相关影响:
 - 使用 DB2 Connect 服务器代表远程和本地客户机访问主机或 System i 数据库时, 系统数据库目录条目中的备用服务器连接信息可能会变得混乱。为使混乱程度降至最低, 请考虑在系统数据库目录中编目两个条目来表示同一主机或 System i 数据库。为远程客户机编目一个条目, 为本地客户机编目另一个条目。

- 从目标 DB2 z/OS 版服务器返回的任何综合系统 (sysplex) 信息仅保存在 DB2 Connect 服务器的高速缓存中。只有一个备用服务器写入磁盘。存在多个备用服务器或活动服务器时, 仅在内存中维护信息并且这些信息在进程终止时将丢失。
- 如果在替代服务器位置重新建立连接, 相同数据库别名的任何新连接将连接至替代服务器位置。如果要在解决原始位置上的问题之后建立与它的任何新连接, 下列几个选项可供选择:
 - 需要将替代服务器脱机并允许连接转移回原始服务器。(这假定已使用 UPDATE ALTERNATE SERVER 命令对原始服务器进行了编目, 以便将其设置为替代服务器的替代位置。)
 - 可以对新连接要使用的新数据库别名进行编目。
 - 可以对数据库条目取消编目, 然后再次对其进行编目。
- 如果客户机和服务器都支持客户机自动重新路由功能, 那么 DB2 数据库 Linux 版、UNIX 版和 Windows 版同时对客户机和服务器支持客户机自动重新路由功能。其他 DB2 数据库产品系列当前不支持此功能。
- 客户机自动重新路由功能的行为与在 DB2 z/OS 版综合系统 (sysplex) 环境中客户机自动重新路由的行为有所不同。尤其表现在下列方面:
 - 客户机自动重新路由功能需要主服务器指定一个备用服务器。这可通过在主服务器中发出 UPDATE ALTERNATE SERVER FOR DATABASE 或 UPDATE ALTERNATE SERVER FOR LDAP DATABASE 命令来完成。此命令将更新具有备用服务器信息的本地数据库目录, 以便同一客户机中的其他应用程序可以访问此信息。相比之下, 用于 DB2 z/OS 版的数据共享综合系统 (sysplex) 会维护客户机可以在内存中连接的一个或多个服务器的列表。如果发生通信故障, 那么客户机使用该服务器列表来确定适当的备用服务器的位置。
 - 在使用客户机自动重新路由功能的情况下, 每次专用寄存器设置更改时, 服务器都会告知客户机专用寄存器的最新设置。这将允许客户机在进行重新路由之后尽可能地重新建立运行时环境。相比之下, 用于 DB2 z/OS 版的综合系统 (sysplex) 在落实边界将专用寄存器设置返回给客户机, 因此需要重放在已重新路由的工作单元内更改的任何专用寄存器。所有其他专用寄存器将自动重放。

从 DB2® Universal Database™ 版本 8 修订包 7 开始, 仅在 Linux、UNIX 或 Windows 客户机与 Linux、UNIX 或 Windows 服务器之间提供完全客户机自动重新路由支持。在 Linux、UNIX 或 Windows 客户机与任何受支持版本的 DB2 z/OS 版综合系统 (sysplex) 服务器之间不提供此支持; 仅支持重新路由功能。

- 当与原始主机服务器上安装的 DB2 数据库实例比较时, 备用主机服务器中安装的 DB2 数据库服务器的版本必须相同 (但是可以具有更高版本的修订包)。
- 无论您是否具有在客户机上更新数据库目录的权限, 替代服务器信息始终保留在内存中。换言之, 如果没有更新数据库目录的权限 (或者因为它是只读数据库目录), 其他应用程序将不能确定和使用替代服务器, 原因是内存未在应用程序之间共享。
- 所有替代位置使用相同的认证方法。这意味着, 如果替代位置的认证类型与原始位置的认证类型不同, 客户机将无法重新建立数据库连接。
- 当发生通信故障时, 所有会话资源 (如用于联合处理和专用寄存器的全局临时表、标识、顺序、游标、服务器选项 (SET SERVER OPTION)) 均会丢失。应用程序负责重新建立会话资源以便继续处理工作。因为 DB2 数据库重放在通信错误之前发出的专用寄存器语句, 因此重新建立连接之后, 无需运行任何专用寄存器语句。但是, 某些专用寄存器不能重放。它们是:

- SET ENCRYPTPW
- SET EVENT MONITOR STATE
- SET SESSION AUTHORIZATION
- SET TRANSFORM GROUP

如果您在使用 DB2 Connect 时遇到问题，那么您应该参阅特定于数据服务器上的 DB2 Connect 产品的受限制专用寄存器列表。

- 如果在通信故障后重新建立了连接，并且客户机正在使用 CLI 或 JCC 2 类或 4 类驱动程序，那么将通过新服务器重新隐式预编译那些已对原始服务器预编译的 SQL 和 XQuery 语句。但是，将不通过新服务器重新预编译嵌入式 SQL 例程（例如，SQC 或 SQX 应用程序）。
- 不要对支持客户机重新路由功能的数据库别名运行高可用性灾难恢复（HADR）命令。实现了 HADR 命令来使用数据库别名标识目标数据库。因此，如果为目标数据库定义了替代数据库，HADR 命令就难以确定命令实际上作用于哪个数据库。虽然客户机可能需要使用支持客户机重新路由功能的别名进行连接，但 HADR 命令必须应用于特定的数据库。为了适应这种情况，可以定义特定于主数据库和备用数据库的别名，并且只对那些别名运行 HADR 命令。

另一种实现客户机自动重新路由的方法是使用 DNS 条目来指定 DNS 条目的替代 IP 地址。其思想是在 DNS 条目中指定另一个 IP 地址（备用服务器位置）；客户机虽然不了解备用服务器，但 DB2 数据库系统进行连接时将在此 DNS 条目的 IP 地址之间进行切换。

DB2 故障监视器注册表文件

在启动故障监视器守护程序时，会为每台物理机器上的每个 DB2 数据库管理器实例创建故障监视器注册表文件。此文件中的关键字和值指定故障监视器的行为。

可在 /sql1lib/ 目录中找到故障监视器注册表文件，文件名为 fm.<machine_name>.reg。通过使用 db2fm 命令可以改变此文件。

如果故障监视器注册表文件不存在，那么将使用缺省值。

以下是故障监视器注册表文件的内容示例：

```
FM_ON = no
FM_ACTIVE = yes
START_TIMEOUT = 600
STOP_TIMEOUT = 600
STATUS_TIMEOUT = 20
STATUS_INTERVAL = 20
RESTART_RETRIES = 3
ACTION_RETRIES = 3
NOTIFY_ADDRESS = <instance_name>@<machine_name>
```

故障监视器注册表文件关键字

FM_ON

指定是否应启动故障监视器。如果该值设置为 NO，那么将不启动故障监视器守护程序，或者如果已经启动，那么将关闭该程序。缺省值是 NO。

FM_ACTIVE

指定故障监视器是否处于活动状态。仅当 FM_ON 和 FM_ACTIVE 都设置为 YES 时，故障监视器才执行操作。如果 FM_ON 设置为 YES 并且 FM_ACTIVE 设置为 NO，那么将启动故障监视器守护程序，但故障监视器不活动。这意味着如果 DB2 关闭，那么不会尝试将它重新联机。缺省值为 YES。

START_TIMEOUT

指定故障监视器必须在其内启动它监视的服务的时间长短。缺省值为 600 秒。

STOP_TIMEOUT

请指定一个时间段，故障监视器必须在该时间段内关闭它所监视的服务。缺省值为 600 秒。

STATUS_TIMEOUT

指定故障监视器必须在其内获取它监视的服务的状态的时间长短。缺省值为 20 秒。

STATUS_INTERVAL

指定用来获取受监视的服务的状态的两次连续调用之间的最小时间。缺省值为 20 秒。

RESTART_RETRIES

指定在尝试失败之后故障监视器将尝试获取受监视的服务的状态的次数。一旦达到此数字，故障监视器将执行操作以使服务回到联机状态。缺省值为 3。

ACTION_RETRIES

指定故障监视器将尝试使服务回到联机状态的次数。缺省值为 3。

NOTIFY_ADDRESS

指定故障监视器将通知消息发送至的电子邮件地址。缺省值为 <instance_name>@<machine_name>。

使用 db2fm 命令来配置 DB2 故障监视器

可以使用 db2fm 命令来改变 DB2 故障监视器注册表文件。

以下是使用 db2fm 命令来更新故障监视器注册表文件的一些示例：

示例 1: 更新 START_TIMEOUT

要将实例 DB2INST1 的 START_TIMEOUT 值更新为 100 秒，请从 DB2 数据库命令窗口输入以下命令：

```
db2fm -i db2inst1 -T 100
```

示例 2: 更新 STOP_TIMEOUT

要将实例 DB2INST1 的 STOP_TIMEOUT 值更新为 200 秒，请输入以下命令：

```
db2fm -i db2inst1 -T /200
```

示例 3: 更新 START_TIMEOUT 和 STOP_TIMEOUT

要将实例 DB2INST1 的 START_TIMEOUT 值更新为 100 秒，并将 STOP_TIMEOUT 值更新为 200 秒，请输入以下命令：

```
db2fm -i db2inst1 -T 100/200
```

示例 4: 打开故障监视

要对实例 DB2INST1 打开故障监视，请输入以下命令：

```
db2fm -i db2inst1 -f yes
```

示例 5: 关闭故障监视

要对实例 DB2INST1 关闭故障监视，请输入以下命令：

```
db2fm -i db2inst1 -f no
```

在 UNIX 系统上，要确认已经不再为 DB2INST1 运行故障监视器，请输入以下命令：

```
ps -ef|grep -i fm
```

在 Linux 上，输入以下命令：

```
ps auxw|grep -i fm
```

显示了 db2fmd 和 DB2INST1 的条目表示仍在该实例上运行故障监视器。要关闭故障监视器，请作为实例所有者输入以下命令：

```
db2fm -i db2inst1 -D
```

使用 db2fmc 和系统命令来配置 DB2 故障监视器

可以使用 DB2 故障监视控制器实用程序 (FMCU) 命令 db2fmcu 或系统命令来配置 DB2 故障监视器。

以下是使用 db2fmcu 和系统命令来配置故障监视器的一些示例：

示例 1: 防止 FMC 启动

可以使用 DB2 故障监视控制器实用程序 (FMCU) 来防止 FMC 启动。由于 FMCU 要访问系统的 inittab 文件，所以它必须作为 root 用户运行。要禁止 FMC 运行，请作为 root 用户输入以下命令：

```
db2fmcu -d
```

注： 如果应用 DB2 数据服务器修订包，就会复位此命令并再次将 inittab 配置为包括 FMC。在应用修订包后，要防止 FMC 启动，必须重新发出以上命令。

示例 2: 包括要启动的 FMC

要撤销 db2fmcu -d 命令并将 inittab 重新配置为包括 FMC，请输入以下命令：

```
db2fmcu -u -p <fullpath>
```

其中 <fullpath> 是 db2fmcu 对象的完整路径，如 /opt/IBM/db2/bin/db2fmcu。

示例 3: 自动启动 DB2 数据库管理器实例

也可以启用 FMC 以便在系统第一次引导时自动启动实例。要对 DB2INST1 实例启用此功能，请输入以下命令：

```
db2iauto -on db2inst1
```

示例 4: 禁止自动启动实例

要关闭自动启动行为，请输入以下命令：

```
db2iauto -off db2inst1
```

示例 5: 防止故障监视器进程启动

通过针对系统上特定实例更改全局注册表记录字段，可以针对该实例阻止故障监视器进程启动。要更改全局注册表字段以便对 DB2INST1 实例禁用故障监视器，请作为 root 用户输入以下命令：

```
db2greg -updinstrec instancename=db2inst1!startatboot=0
```

要对 DB2INST1 实例撤销此命令并重新启用故障监视器，请作为 root 用户输入以下命令：

```
db2greg -updinstrec instancename=db2inst1!startatboot=1
```

初始化高可用性灾难恢复 (HADR)

使用以下过程来为 DB2 高可用性灾难恢复 (HADR) 设置和初始化主数据库和备用数据库。

可以使用命令行处理器 (CLP)、控制中心中的“设置高可用性灾难恢复” (HADR) 向导或通过调用 db2HADRStart API 来初始化 HADR。

要使用 CLP 以在系统上首次初始化 HADR：

1. 确定每个 HADR 数据库的主机名、主机 IP 地址和服务名称或端口号。

如果主机使用多个网络接口，请确保 HADR 主机名或 IP 地址映射至预期的接口。您需要在 /etc/services 中为每个受保护数据库分配不同的 HADR 端口。这些端口不能与分配给实例的端口相同。主机名只能映射至一个 IP 地址。

注：主数据库和备用数据库的实例名称无需相同。

2. 通过复原备份映像或初始化分割镜像，根据要作为主数据库的现有数据库来创建备用数据库。

在以下示例中，使用 BACKUP DATABASE 和 RESTORE DATABASE 命令将数据库 SOCKS 作为备用数据库进行初始化。在此情况下，NFS 已安装文件系统在这两处都是可访问的。

在主数据库上发出以下命令：

```
backup db socks to /nfs1/backups/db2/socks
```

在备用数据库上发出以下命令：

```
restore db socks from /nfs1/backups/db2/socks replace history file
```

以下示例举例说明，如何通过主数据库的分割镜像来使用 db2inidb 实用程序来初始化备用数据库。此过程是上述备份和复原过程的替代方案。

在备用数据库上发出以下命令：

```
db2inidb socks as standby
```

注：

- a. 主数据库和备用数据库的数据库名称必须相同。
- b. 复原操作或分割镜像初始化之后，建议不要在备用数据库上发出 ROLLFORWARD DATABASE 命令。使用前滚操作的结果可能与在备用数据库上使用 HADR 重放日志略有不同。如果数据库不相同，发出指定了 AS STANDBY 选项的 START HADR 命令将会失败。

- c. 当使用 RESTORE DATABASE 命令时，建议使用 REPLACE HISTORY FILE 选项。
 - d. 在使用 RESTORE DATABASE 命令创建备用数据库时，必须确保备用数据库保持处于前滚方式。这意味着不能发出带有 COMPLETE 选项或 STOP 选项的 ROLLFORWARD DATABASE 命令。前滚停止后，如果对数据库发出带有 AS STANDBY 选项的 START HADR 命令，就会返回错误。
 - e. 当设置备用数据库时，应该避免使用以下 RESTORE DATABASE 命令选项：TABLESPACE、INTO、REDIRECT 和 WITHOUT ROLLING FORWARD。
 - f. 当使用 db2inidb 实用程序来设置备用数据库时，不要使用 SNAPSHOT 或 MIRROR 选项。可以指定 RELOCATE USING 选项来更改以下一个或多个配置属性：实例名称、日志路径和数据库路径。但是，切勿更改数据库名或表空间容器路径。
3. 在主数据库和备用数据库上设置 HADR 配置参数。

注：创建备用数据库之后，设置以下配置参数至关重要：

- HADR_LOCAL_HOST
- HADR_LOCAL_SVC
- HADR_REMOTE_HOST
- HADR_REMOTE_SVC
- HADR_REMOTE_INST

如果在创建备用数据库之前设置了备用数据库上的设置，这些设置将反映在主数据库上设置的内容。

4. 与备用实例连接，并在备用数据库上启动 HADR，如以下示例中所示：

```
START HADR ON DB SOCKS AS STANDBY
```

注：通常，先启动备用数据库。如果先启动主数据库，并且未在 HADR_TIMEOUT 数据库配置参数指定的时间段内启动备用数据库，启动过程就会失败。

5. 与主实例连接，并在主数据库上启动 HADR，如以下示例中所示：

```
START HADR ON DB SOCKS AS PRIMARY
```

6. 现在在主数据库和备用数据库上启动 HADR。

要打开“安装高可用性灾难恢复（HADR）数据库”向导：

- a. 在控制中心中，展开对象树，直到找到要为其配置 HADR 的数据库为止。
- b. 用鼠标右键单击此数据库，然后在弹出菜单中单击“高可用性灾难恢复”→“安装”。“安装高可用性灾难恢复数据库”向导将打开。

控制中心中的上下文帮助工具提供了其他信息。

注：您可以在“设置高可用性灾难恢复数据库”向导中启动 HADR，也可以仅仅使用此向导来初始化 HADR，然后另选时间将它启动。要打开“启动 HADR”窗口：

- a. 在控制中心中，展开对象树，直到找到要为其管理 HADR 的数据库为止。用鼠标右键单击此数据库，然后在弹出菜单中单击“高可用性灾难恢复”→“管理”。这就打开了“管理高可用性灾难恢复”窗口。
- b. 单击“启动 HADR”。这就打开了“启动 HADR”窗口。

配置客户机自动重新路由和高可用性灾难恢复（HADR）

可以将客户机自动重新路由功能与高可用性灾难恢复（HADR）功能配合使用，以便将失败的数据库服务器发出的客户机应用程序请求传送至备用数据库服务器。

限制

- 仅当在服务器上指定了替代数据库位置时，才有可能重新路由。
- 只有使用 TCP/IP 协议，才支持客户机自动重新路由。

配置详细信息

- 使用 UPDATE ALTERNATE SERVER FOR DATABASE 命令来启用客户机自动重新路由。
- 缺省情况下，如果您在控制中心中使用“设置高可用性灾难恢复（HADR）数据库”向导来设置 HADR，就会启用客户机重新路由功能。
- 客户机自动重新路由不使用 HADR_REMOTE_HOST 和 HADR_REMOTE_SVC 数据库配置参数。
- 替代主机位置存储在服务器的系统数据库目录文件中。
- 如果未启用客户机自动重新路由，客户机应用程序将接收错误消息 SQL30081，而且不会进一步尝试建立与服务器的连接。

使用 UPDATE ALTERNATE SERVER FOR DATABASE 命令来设置具有 HADR 的客户机自动重新路由

系统设置如下：

- 您具有一台客户机，其中的数据库 MUSIC 的编目方式与位于主机 HORNET 上的数据库的编目方式相同。
- 数据库 MUSIC 是主数据库，它的相应备用数据库（也是 MUSIC）位于主机 MONTERO 上并且端口号为 456，该端口号是由 SVCENAME 配置参数指定的。

要启用客户机自动重新路由，为主机 HORNET 上的数据库 MUSIC 更新替代服务器：

```
db2 update alternate server for database music using hostname montero port 456
```

发出此命令后，客户机必须与主机 HORNET 成功连接以获取替代服务器信息。然后，如果客户机和主机 HORNET 上的数据库 MUSIC 之间发生通信错误，客户机将首先尝试重新与主机 HORNET 上的数据库 MUSIC 连接。如果失败，随后客户机将尝试与主机 MONTERO 上的备用数据库 MUSIC 建立连接。

索引记录和高可用性灾难恢复 (HADR)

您应该考虑为 DB2 高可用性灾难恢复 (HADR) 数据库设置 LOGINDEXBUILD 和 INDEXREC 数据库配置参数。

使用 LOGINDEXBUILD 数据库配置参数

建议：对于 HADR 数据库，将 LOGINDEXBUILD 数据库配置参数设置为 ON 以确保为索引的创建、重新创建和重组记录完整的信息。虽然，这意味着在主系统上的索引构建可能需要更长时间和更多日志空间，但是在 HADR 日志重放期间，将在备用系统上重建索引，而且当发生故障转移时索引仍可用。如果不记录主系统上的索引构建，而且发生故障转移，那么故障转移完成后所剩的任何无效索引必须重建，方可对其进行访问。当重新创建索引时，任何应用程序都不能对它们进行访问。

注：如果 LOG INDEX BUILD 表属性设置为缺省值 NULL，那么 DB2 将使用为 LOGINDEXBUILD 数据库配置参数指定的值。如果 LOG INDEX BUILD 表属性设置为 ON 或 OFF，为 LOGINDEXBUILD 数据库配置参数指定的值将被忽略。

可能会选择在一个或多个表上将 LOG INDEX BUILD 表属性设置为 OFF，原因如下：

- 没有足够的活动日志空间来支持索引构建的日志记录。
- 索引数据非常大而且不会经常访问表。因此，它适用于在接管操作结束时重新创建的索引。在此情况下，将 INDEXREC 配置参数设置为 RESTART。因为不经常访问表，所以此设置将导致系统在接管操作结束时，重新创建索引，而不是等待执行接管操作之后第一次访问表时创建。

如果 LOG INDEX BUILD 表属性在一个或多个表中设置为 OFF，那些表上的任何索引构建操作可能导致在执行接管操作时，随时重新创建索引。类似地，如果 LOG INDEX BUILD 表属性设置为缺省值 NULL，LOGINDEXBUILD 数据库配置参数设置为 OFF，表上的任何索引构建操作可能导致在执行接管操作时，随时重新创建该表上的索引。通过执行以下操作，可以防止重新创建索引：

- 在新的主数据库上重新创建所有无效索引之后，获得数据库的备份，并将其应用到备用数据库。执行此操作的结果是，备用数据库不必应用用于在主数据库上重新创建无效索引的日志，该数据库将那些索引标记为需要在备用数据库上重建。
- 在备用数据库上，将 LOG INDEX BUILD 表属性设置为 ON，或者将 LOG INDEX BUILD 表属性设置为 NULL，并将 LOGINDEXBUILD 配置参数设置为 ON，以确保记录索引的重新创建。

使用 INDEXREC 数据库配置参数

建议：在主数据库和备用数据库上，将 INDEXREC 数据库配置参数设置为 RESTART（缺省值）。这将导致完成接管操作之后，重建无效索引。如果未记录任何索引构建，那么此设置允许 DB2 检查无效索引并重建这些索引。此进程于后台执行，而且接管操作成功完成之后，可以访问数据库。

在通过后台重新创建索引进程重建索引之前，如果事务访问具有无效索引的表，那么将通过第一个访问无效索引的事务来重建该索引。

高可用性灾难恢复 (HADR) 的数据库配置

可以使用数据库配置参数来获得 DB2 高可用性灾难恢复 (HADR) 的最佳性能。

为了获得 DB2 高可用性灾难恢复 (HADR) 的最佳性能, 请确保数据库配置满足下列要求。

建议: 数据库配置参数和数据库管理器配置参数在主数据库和备用数据库所在系统上应尽可能一致。如果未在备用数据库上正确设置配置参数, 就可能会发生以下问题:

- 在备用数据库上, 当重放主数据库提供的日志文件时, 可能会返回错误消息。
- 接管操作完成后, 新的主数据库将无法处理工作负载, 这将引起性能问题, 或者, 在接收错误消息的应用程序中, 这些应用程序在与原始主数据库连接后无法接收到此类错误消息。

对主数据库配置参数所作的更改不会自动地传播至备用数据库, 必须在备用数据库上手动执行这些更改。对于动态配置参数来说, 不需要关闭并重新启动数据库管理系统 (DBMS) 或数据库就可以使更改生效。对于非动态配置参数来说, 更改将在备用数据库重新启动后生效。

备用数据库上的日志文件大小配置参数

上面描述的配置参数行为的一个例外情况是 LOGFILSIZ 数据库配置参数。尽管不会将此参数复制到备用数据库, 但是为了保证两个数据库上的日志文件一致, 备用数据库将忽略本地 LOGFILSIZ 配置并创建与主数据库上的日志文件大小匹配的本地日志文件。

接管完成后, 原始备用数据库 (新的主数据库) 将继续使用原始主数据库上设置的值, 直到该数据库重新启动为止。数据库重新启动后, 新的主数据库将改为使用本地配置的值。此外, 新的主数据库还将截断当前日志文件并调整任何预先创建的日志文件的大小。

如果数据库通过非强制接管操作来不断切换角色, 并且未将任何一个数据库取消激活, 那么使用的日志文件大小将始终是第一个主数据库确定的大小。但是, 如果将原始备用数据库 (新的主数据库) 取消激活并接着重新启动它, 它就会使用本地配置的日志文件大小。即使原始主数据库再次接管, 也将继续使用此日志文件大小。只有将原始主数据库取消激活并接着重新启动它之后, 日志文件大小才会恢复为原始主数据库上的设置。

备用数据库上的日志接收缓冲区大小

缺省情况下, 备用数据库上的日志接收缓冲区大小将是对主数据库 LOGBUFSZ 配置参数指定的值的两倍。当此大小不够大时, 也可能是数倍。例如, 当 HADR 同步方式为异步, 并且主数据库和备用数据库处于对等状态时, 如果主数据库正在处理大量事务负载, 那么备用数据库的日志接收缓冲区可能会达到最大容量, 而主数据库上的日志提供操作可能会停止。要管理这些临时峰值, 可以通过修改 DB2_HADR_BUF_SIZE 注册表变量来增大备用数据库上的日志接收缓冲区大小。

装入操作和 HADR

如果在主数据库上执行指定了 COPY YES 选项的装入操作, 只要可以通过 LOAD 命令指定的路径或设备访问副本, 就可以对主数据库执行此命令, 并将数据复制到备用数据库。如果备用数据库无法访问数据, 在备用数据库中就会将用来存储表的表空间标记为无效。备用数据库将跳过与此表空间相关的未来日志记录。要确保装入操作能够访问备用数据库上的副本, 建议您使用共享位置来存储 COPY YES 选项产生的输出文件。此外, 可以在装入操作执行期间将备用数据库取消激活, 对主数据库执行装入, 将输出文件副本放到备用数据库路径中, 然后激活备用数据库。

如果在主数据库上执行指定了 `NONRECOVERABLE` 选项的装入操作，此命令将对主数据库执行，并且将备用数据库上的表标记为无效。备用数据库将跳过与此表相关的未来日志记录。您可以选择发出指定了 `COPY YES` 和 `REPLACE` 选项的 `LAOD` 命令来恢复表，也可以删除该表以恢复空间。

因为 `HADR` 不支持执行指定了 `COPY NO` 选项的装入操作，所以自动将此命令转换为指定了 `NONRECOVERABLE` 选项的装入操作。要将指定了 `COPY NO` 选项的装入操作转换为指定了 `COPY YES` 选项的装入操作，请在主数据库中设置 `DB2_LOAD_COPY_NO_OVERRIDE` 注册表变量。备用数据库将忽略此注册表变量。确保备用数据库可通过使用相同的路径、设备或装入库来访问主数据库上指定的设备或目录。

如果正在使用 `Tivoli Storage Manager (TSM)` 来执行指定了 `COPY YES` 选项的装入操作，那么可能需要在主数据库和备用数据库上设置 `VENDOROPT` 配置参数。根据 `TSM` 的配置方法，主数据库和备用数据库上的值可以不同。并且，当使用 `TSM` 来执行指定了 `COPY YES` 选项的装入操作时，必须发出指定了 `GRANT` 选项的 `db2adutl` 命令来授权备用数据库对装入的文件进行读访问。

如果表数据是由指定了 `COPY YES` 选项的装入操作复制的，就会按以下方式复制索引：

- 如果建立索引方式设置为 `REBUILD` 并且表属性设置为 `LOG INDEX BUILD`，或者表属性设置为 `DEFAULT` 并且 `LOGINDEXBUILD` 数据库配置参数设置为 `ON`，那么主数据库将包含复制文件中的重建索引对象，以使备用数据库能够复制索引对象。如果备用数据库上的索引对象在装入操作执行前已被标记为无效，但由于重建了索引，所以此索引对象在装入操作完成后将再次可用。
- 如果建立索引方式设置为 `INCREMENTAL` 且表属性设置为 `LOG INDEX BUILD`，或者表属性设置为 `NULL` 且主数据库上的 `LOGINDEXBUILD` 数据库配置参数设置为 `ON`，那么仅当装入操作之前，索引未标记为无效时，才能更新备用数据库上的索引对象。否则，在备用数据库上将索引标记为无效。

注册表变量 `DB2_HADR_PEER_WAIT_LIMIT`

设置了注册表变量 `DB2_HADR_PEER_WAIT_LIMIT` 时，如果由于将日志复制到备用数据库而导致 `HADR` 主数据库上的记录被阻塞指定秒数，那么该主数据库将脱离对等状态。达到此限制时，主数据库将断开与备用数据库的连接。如果对等时间被禁用，那么主数据库将进入断开状态，并且记录将继续进行。如果对等时间已启用，那么主数据库将进入断开对等状态，此时记录继续被阻塞。在重新连接或对等时间到期时，主数据库将脱离断开对等状态。在主数据库脱离断开对等状态后，记录就立即继续进行。

在脱离对等状态时执行对等时间转换可确保对等时间语义在所有情况下都适用于安全接管。如果主数据库在转换期间失败，那么一般对等时间保护仍适用（只要备用数据库仍处于断开对等状态，就可以从备用数据库安全接管）。

对于备用数据库，在断开连接后，它将继续重放已接收到的日志。一旦重放了接收到的日志，备用数据库就将重新连接至主数据库。重新连接时会进行一般状态过渡：首先是远程同步复制状态，然后是对等状态。

与 `HADR_TIMEOUT` 的关系：

如果在阻塞期间主数据库不断地接收来自备用数据库的脉动信号消息，那么数据库配置参数 **HADR_TIMEOUT** 不会使主数据库脱离对等状态。对于 HADR 网络层来说，**HADR_TIMEOUT** 表示超时。如果 HADR 数据库在 **HADR_TIMEOUT** 时间段内未从其伙伴数据库接收到任何消息，那么该数据库将断开与其伙伴数据库的连接。该参数不控制更高层操作（例如，日志装入和 ACK）的超时。如果在备用数据库上重放的日志在较大的操作（如装入或重组）处被卡住，那么 HADR 组件仍将按正常时间表发送脉动信号消息给主数据库。在此情况下，除非设置了 **DB2_HADR_PEER_WAIT_LIMIT**，否则只要备用重放被阻塞，主数据库就将阻塞。

无论连接状态如何，**DB2_HADR_PEER_WAIT_LIMIT** 都将取消阻塞主数据库记录。请注意，即使未设置 **DB2_HADR_PEER_WAIT_LIMIT**，在检测到网络错误或者连接关闭时（可能是由于 **HADR_TIMEOUT**），主数据库也总是会脱离对等状态。

HADR 配置参数

若干个新数据库配置参数可用于支持 HADR。设置这些参数不会更改数据库的角色。必须发出 **START HADR** 或 **STOP HADR** 命令来更改数据库的角色。

HADR 配置参数不是动态的。对 HADR 配置参数所作的任何更改只有在关闭并重新启动数据库后才会生效。在分区数据库环境中，HADR 配置参数可视并且可以被更改，但是这些参数将被忽略。

主数据库的本地主机名必须与备用数据库的远程主机名相同，而且备用数据库的本地主机名必须与主数据库的远程主机名相同。使用 **HADR_LOCAL_HOST** 和 **HADR_REMOTE_HOST** 配置参数来设置每个数据库的本地主机和远程主机。建立连接时将检查本地主机名与远程主机名的配置是否一致，从而确保指定的远程主机就是期望的数据库。

可以将 HADR 数据库配置为使用 IPv4 或 IPv6 来找到它的伙伴数据库。如果主机服务器不支持 IPv6，数据库将使用 IPv4。如果服务器确实支持 IPv6，那么数据库是使用 IPv4 还是 IPv6 取决于对 **HADR_LOCAL_HOST** 和 **HADR_REMOTE_HOST** 配置参数指定的地址格式。数据库会尝试将这两个参数解析为相同的 IP 格式。下表说明了如何确定支持 IPv6 的服务器的 IP 方式：

用于 HADR_LOCAL_HOST 的 IP 方式	用于 HADR_REMOTE_HOST 的 IP 方式	用于 HADR 通信的 IP 方式
IPv4 地址	IPv4 地址	IPv4
IPv4 地址	IPv6 地址	出错
IPv4 地址	主机名，仅映射至 v4	IPv4
IPv4 地址	主机名，仅映射至 v6	出错
IPv4 地址	主机名，映射至 v4 和 v6	IPv4
IPv6 地址	IPv4 地址	出错
IPv6 地址	IPv6 地址	IPv6
IPv6 地址	主机名，仅映射至 v4	出错
IPv6 地址	主机名，仅映射至 v6	IPv6
IPv6 地址	主机名，映射至 v4 和 v6	IPv6
主机名，仅映射至 v4	IPv4 地址	IPv4

用于 HADR_LOCAL_HOST 的 IP 方式	用于 HADR_REMOTE_HOST 的 IP 方式	用于 HADR 通信的 IP 方式
主机名, 仅映射至 v4	IPv6 地址	出错
主机名, 仅映射至 v4	主机名, 仅映射至 v4	IPv4
主机名, 仅映射至 v4	主机名, 仅映射至 v6	出错
主机名, 仅映射至 v4	主机名, 映射至 v4 和 v6	IPv4
主机名, 仅映射至 v6	IPv4 地址	出错
主机名, 仅映射至 v6	IPv6 地址	IPv6
主机名, 仅映射至 v6	主机名, 仅映射至 v4	出错
主机名, 仅映射至 v6	主机名, 仅映射至 v6	IPv6
主机名, 仅映射至 v6	主机名, 映射至 v4 和 v6	IPv6
主机名, 映射至 v4 和 v6	IPv4 地址	IPv4
主机名, 映射至 v4 和 v6	IPv6 地址	IPv6
主机名, 映射至 v4 和 v6	主机名, 仅映射至 v4	IPv4
主机名, 映射至 v4 和 v6	主机名, 仅映射至 v6	IPv6
主机名, 映射至 v4 和 v6	主机名, 映射至 v4 和 v6	IPv6

仅当主数据库与备用数据库使用同一种格式时, 它们才能建立 HADR 连接。如果一台服务器支持 IPv6 (但也支持 IPv4), 而另一台服务器只支持 IPv4, 那么在 HADR_LOCAL_HOST 和 HADR_REMOTE_HOST 参数中至少要有一个参数必须指定 IPv4 地址。这指示数据库使用 IPv4, 即使服务器支持 IPv6 亦如此。

在准备 update database configuration 命令期间, 在对高可用性灾难恢复 (HADR) 本地服务参数和远程服务参数 (HADR_LOCAL_SVC 和 HADR_REMOTE_SVC) 指定值时, 指定的值必须是尚未用于任何其他服务 (包括其他 DB2 组件或其他 HADR 数据库) 的端口。特别是, 不能将任何一个参数值设置为服务器在等待来自远程客户机的通信时使用的 TCP/IP 端口 (SVCENAME 数据库管理器配置参数) 或下一个端口 (SVCENAME + 1)。

如果主数据库与备用数据库在不同机器上, 它们就可以使用同一个端口号或服务名称; 否则, 应该使用不同的值。可以将 HADR_LOCAL_SVC 和 HADR_REMOTE_SVC 参数设置为端口号或服务名称。

同步方式 (HADR_SYNCMODE) 和超时周期 (HADR_TIMEOUT) 在主数据库和备用数据库上必须一致。当 HADR 数据库对建立连接时, 将检查这些配置参数的一致性。

主数据库与备用数据库之间的通信使用 TCP 连接。如果主数据库由于正在启动或者断开连接而未与备用数据库建立连接, 它就会使用其本地端口来侦听新连接。未与主数据库连接的备用数据库将继续向其远程主机发出连接请求。

尽管本地主机参数和本地服务参数 (HADR_LOCAL_HOST 和 HADR_LOCAL_SVC) 仅在主数据库上使用, 但您仍应该在备用数据库上设置这些参数, 以确保当备用数据库必须作为主数据库接管时, 这些参数已就绪。

当主数据库启动时，它将等待备用数据库建立连接，等待时间至少为 30 秒钟或者 HADR_TIMEOUT 数据库配置参数值指定的秒数（以较大者为准）。如果备用数据库未在指定的时间内建立连接，启动将失败。（发出指定了 BY FORCE 选项的 START HADR 命令时的情况除外。）

在 HADR 数据库对建立连接后，它们将交换脉动信号消息。脉动信号间隔是 HADR_TIMEOUT 数据库配置参数值的 1/4 或者 30 秒（以较小者为准）。HADR_HEARTBEAT 监视元素显示数据库期望从另一数据库接收但未接收到的脉动信号数。如果一个数据库在 HADR_TIMEOUT 指定的秒数内未从另一数据库接收到任何消息，它就会启动断开连接操作。这意味着它最多等待 HADR_TIMEOUT 指定的秒数以允许主数据库检测备用数据库故障或它们之间的网络故障。如果将 HADR_TIMEOUT 配置参数设置得太小，您就会接收到虚假的警报，并且将频繁地断开连接。

如果 HADR_PEER_WINDOW 数据库配置参数设置为零，那么当主数据库和备用数据库处于对等状态时，备用数据库或网络的问题最多只会使主数据库无法在 HADR_TIMEOUT 配置参数指定的秒数内处理事务。如果将 HADR_PEER_WINDOW 设置为非零值，那么主数据库将在恢复与备用数据库的连接后或经过 HADR_PEER_WINDOW 指定的时间后（以先发生者为准）才落实事务。

注：为获得最大可用性，HADR_PEER_WINDOW 数据库配置参数的缺省值设置为零。当 HADR_PEER_WINDOW 设置为零时，一旦主数据库与备用数据库之间的连接关闭（无论是由于备用数据库关闭了连接，检测到网络错误，还是达到超时限制），主数据库将脱离对等状态，以避免妨碍事务处理。为增强数据一致性（但会降低可用性），可以将 HADR_PEER_WINDOW 数据库配置参数设置为非零值，这样将使主数据库在 HADR_PEER_WINDOW 值指定的时间内保持断开对等状态。

以下是主数据库和备用数据库的样本配置。

在主数据库上:

```
HADR_LOCAL_HOST      host1.ibm.com
HADR_LOCAL_SVC       hadr_service
HADR_REMOTE_HOST     host2.ibm.com
HADR_REMOTE_SVC      hadr_service
HADR_REMOTE_INST     dbinst2
HADR_TIMEOUT         120
HADR_SYNCMODE        NEARSYNC  HADR_PEER_WINDOW      120
```

在备用数据库上:

```
HADR_LOCAL_HOST      host2.ibm.com
HADR_LOCAL_SVC       hadr_service
HADR_REMOTE_HOST     host1.ibm.com
HADR_REMOTE_SVC      hadr_service
HADR_REMOTE_INST     dbinst1
HADR_TIMEOUT         120
HADR_SYNCMODE        NEARSYNC  HADR_PEER_WINDOW      120
```

DB2 高可用性灾难恢复 (HADR) 的日志归档配置

要将日志归档与 DB2 高可用性灾难恢复 (HADR) 配合使用，请将主数据库和备用数据库配置为能够从所有日志归档位置自动检索日志。

如果备用数据库或主数据库无法访问所有日志归档位置，那么必须将日志文件从日志归档手动复制到下列位置:

- 用于本地同步复制的备用数据库日志路径或归档位置

- 用于远程同步复制的主数据库日志路径或归档位置

只有当前主数据库能够执行日志归档。如果对主数据库和备用数据库设置了不同的归档位置，那么会只将日志归档到主数据库的归档位置。执行接管时，备用数据库成为新的主数据库，从该刻起归档的任何日志都将被保存到原始备用数据库的归档位置。在此配置中，日志被归档到其中一个位置，而不是同时被归档到两个位置；有一种例外情况，即执行接管后，新的主数据库可能会归档一小部分已被原始主数据库归档的日志。

执行接管后，如果新的主数据库（原始备用数据库）遇到介质故障并需要执行复原和前滚，它可能需要访问仅存在于原始主数据库归档位置的日志。

在主数据库通知备用数据库它已归档一个日志文件之前，备用数据库不会从它的本地日志路径中删除该日志文件。此行为防止了日志文件的丢失。如果主数据库发生故障，并且它的日志磁盘已损坏，但尚未在主数据库上归档特定日志文件，备用数据库就不会从它自己的磁盘中删除该日志文件，这是因为它尚未接收到指示主数据库已成功归档该日志文件的通知。如果备用数据库接着执行了接管，从而成为新的主数据库，它就会先归档该日志文件，然后再重新开始使用该文件。如果同时使用了 *logarchmeth1* 和 *logarchmeth2* 配置参数，那么在主数据库使用这两种方法归档日志文件前，备用数据库不会复用该日志文件。

高可用性灾难恢复（HADR）的性能

配置数据库系统的各个方面（包括网络带宽、CPU 电源和缓冲区大小）可以提高 DB2 高可用性灾难恢复（HADR）数据库的性能。

为了最大程度地提高 HADR 性能，在管理系统时请考虑以下建议：

- 网络带宽必须大于数据库日志生成速率。
- 网络延迟只有在 SYNC 和 NEARSYNC 方式下才会影响主数据库。
- 使用 SYNC 方式对系统性能产生的影响明显大于其他同步方式的影响。在 SYNC 方式下，主数据库只有成功地将日志页写入主数据库日志磁盘后才会将日志页发送至备用数据库。为了保护系统完整性，主数据库先等待备用数据库确认，然后再将事务已准备或落实完毕的情况通知应用程序。备用数据库只有在将接收到的日志页写入备用数据库磁盘后才会发送确认消息。产生的开销是：备用数据库上的日志写操作开销以及来回消息传递开销。
- 在 NEARSYNC 方式下，主数据库以并行方式写日志页并发送它们。然后，主数据库等待备用数据库确认。备用数据库一旦将日志页接收到它的内存中，就发出确认消息。在快速网络中，此工作方式能够将主数据库上的开销降至最低。当主数据库完成本地日志写操作时，确认消息可能已到达。
- 对于 ASYNC 方式来说，日志的写操作和发送操作是并行进行的；但是，在此方式下，主数据库不等待来自备用数据库的确认消息。因此，不存在网络延迟问题。与 NEARSYNC 方式相比，ASYNC 方式的性能开销更低。
- 对于在主数据库上写的每个日志，还将相同的日志页发送至备用数据库。每一次写操作都称为清空。清空大小不能超过主数据库上的日志缓冲区大小，后者由数据库配置参数 *logbufsz* 控制。每次清空的确切大小是不确定的。即使日志缓冲区较大，也不一定表示清空大小较大。
- 备用数据库的功能应该足够强大，从而以主数据库生成数据库操作记录的速度来重放那些记录。建议主数据库与备用数据库使用完全相同的硬件。

- 在大部分系统中，记录功能不会达到其极限。即使在 SYNC 方式下，在主数据库上也不会有明显的性能下降情况。例如，如果启用 HADR 后记录限制是每秒 40 Mb，但系统在启用 HADR 前的运行速度仅仅是每秒 30 Mb，那么您可能看不出整体系统性能有任何变化。
- 要加速同步复制进程，可以使用共享日志归档设备。但是，如果共享设备是串行设备（如，磁带机），由于混合读写操作，主数据库和备用数据库上的性能可能会降低。

网络阻塞

如果备用数据库重放日志页的速度太慢，它的日志接收缓冲区就可能会填满，从而导致缓冲区无法接收更多的日志页。在 SYNC 和 NEARSYNC 方式下，如果主数据库再次清空其日志缓冲区，数据就有可能缓存在由主数据库机器、网络和备用数据库组成的网络管道中。由于备用数据库没有可用缓冲区来接收数据，因此它无法进行确认，从而导致主数据库在等待备用数据库确认时阻塞。

在 ASYNC 方式下，主数据库将持续发送日志页直到管道填满为止，之后，它就无法发送更多的日志页。这种情况称为**拥塞**。拥塞由 **hadr_connect_status** 监视元素报告。对于 SYNC 和 NEARSYNC 方式来说，管道通常可以吸收一次清空内容，因此不会发生拥塞。但是，执行清空操作时，主数据库在等待备用数据库确认时将处于阻塞状态。

如果备用数据库长时间重放日志记录，例如数据库或表重组日志记录，也可能发生拥塞。

增大备用数据库日志接收缓冲区大小尽管无法消除所有拥塞原因，但有助于减少拥塞情况。缺省情况下，备用数据库日志接收缓冲区大小是主数据库日志写缓冲区大小的两倍。数据库配置参数 *logbufsz* 指定主数据库日志写缓冲区的大小。DB2 注册表变量 **DB2_HADR_BUF_SIZE** 可用于调整备用数据库日志接收缓冲区大小。

注册表变量 DB2_HADR_PEER_WAIT_LIMIT

当设置了注册表变量 **DB2_HADR_PEER_WAIT_LIMIT** 时，如果由于将日志复制到备用数据库而导致 HADR 主数据库上的记录被阻塞指定秒数，那么该主数据库将脱离对等状态。达到此限制时，主数据库将断开与备用数据库的连接。如果对等时间被禁用，那么主数据库将进入断开状态，并且记录将继续进行。如果对等时间已启用，那么主数据库将进入断开对等状态，此时记录继续被阻塞。在重新连接或对等时间到期时，主数据库将脱离断开对等状态。在主数据库脱离断开对等状态后，记录就立即继续进行。

在脱离对等状态时执行对等时间转换可确保对等时间语义在所有情况下都适用于安全接管。如果主数据库在转换期间失败，那么一般对等时间保护仍适用（只要备用数据库仍处于断开对等状态，就可以从备用数据库安全接管）。

注册表变量 DB2_HADR_SOSNDBUF 和 DB2_HADR_SORCVBUF

要使网络和 HADR 的性能最大化，TCP 套接字缓冲区大小可能需要调整。HADR 日志装入工作负载、网络带宽和传输延迟是调整 TCP 套接字缓冲区大小时要考虑的重要因素。如果在系统级别更改 TCP 套接字缓冲区大小，那么会将这些设置应用于机器上的所有 TCP 连接。设置大型系统级别套接字缓冲区大小将占用大量内存。

两个注册表变量 **DB2_HADR_SOSNDBUF** 和 **DB2_HADR_SORCVBUF** 只允许您为 HADR 连接调整 TCP 套接字发送和接收缓冲区大小。这两个变量的值介于 1024 至 4294967295 范围内，并且是操作系统的套接字缓冲区大小的缺省设置，具体情况因操作系统的不同而异。一些操作系统将自动对值进行四舍五入或以静默方式对用户指定的值设置上限。

集群管理器和高可用性灾难恢复（HADR）

可以在集群中的节点上实现 DB2 高可用性灾难恢复（HADR）数据库，并使用集群管理器来提高数据库解决方案的可用性。可以让同一个集群管理器同时管理主数据库和备用数据库，也可以让不同集群管理器来管理主数据库和备用数据库。

在同一集群管理器维护的主数据库和备用数据库中设置 HADR 数据库对

此配置最适合两种环境，即主数据库和备用数据库位于相同站点，以及要求尽可能最快速的故障转移环境。这些环境受益于使用 HADR 而不是使用崩溃恢复或其他恢复方法来维持 DBMS 可用性。

可以使用集群管理器来快速发现问题并启动接管操作。因为 HADR 要求单独存储 DBMS，所以应该使用单独卷控件配置集群管理器。在备用系统上使用 DBMS 之前，此配置防止集群管理器等待故障转移出现在卷上。可以使用客户机自动重新路由功能以将客户机应用程序重定向到新的主数据库。

在并非由同一集群管理器维护的主数据库和备用数据库中设置 HADR 数据库对

此配置最适合两种环境，即主数据库和备用数据库位于不同站点，以及整个站点发生故障时灾难恢复要求高可用性的环境。可以通过多种方法实现此配置。当 HADR 主数据库或备用数据库是集群的一部分时，可以使用两种可能的故障转移方案。

- 如果发生部分站点故障，并且 DBMS 可以将故障转移到其中的节点仍然可用，那么可以选择执行集群故障转移。在此情况下，使用集群管理器来执行 IP 地址和卷故障转移；HADR 不受影响。
- 如果在主数据库所处的位置发生整个站点故障，可以通过启动接管操作来使用 HADR 以保持 DBMS 可用性。如果在备用数据库所处的位置发生整个站点故障，可以修复站点或将备用数据库移至其他站点。

初始化备用数据库

要让数据库解决方案具有高可用性，一种策略是同时维护一个主数据库和一个辅助或备用数据库，前者响应用户应用程序请求，后者在前者出现故障时可以接管其数据库操作。初始化备用数据库需要将主数据库复制到备用数据库。

有几种方法可用来初始化备用数据库。例如：

- 使用磁盘镜像来复制主数据库，并使用 DB2 数据库暂挂的 I/O 支持来分割镜像以创建另一个数据库。
- 创建主数据库的备份映像并将该映像恢复到备用数据库。
- 使用 SQL 复制来从主数据库捕获数据并将该数据应用到备用数据库。

初始化备用数据库后，必须对数据库解决方案进行配置，使主数据库和备用数据库同步，从而当主数据库失败时，备用数据库可以接管主数据库。

将分割镜像用作备用数据库

使用以下过程来创建数据库的分割镜像，以用作备用数据库。如果在主数据库上发生故障且需要崩溃恢复，那么可以使用备用数据库来接管主数据库。

要将分割镜像用作备用数据库，请遵循下列步骤：

1. 暂挂主数据库上的 I/O ：

```
db2 set write suspend for database
```

数据库暂挂期间，您不应该运行其他实用程序或工具。您只应生成数据库副本。

2. 使用适当的操作系统级别的命令来从主数据库中分割镜像。

注： 确保复制整个数据库目录（包括卷目录）。还必须复制日志目录和存在于数据库目录之外的任何容器目录。要收集此信息，请参阅 DBPATHS 管理视图，该视图显示了需要分割的数据库的所有文件和目录。

3. 恢复主数据库上的 I/O:

```
db2 set write resume for database
```

4. 在辅助系统上对镜像数据库进行编目。

注： 缺省情况下，镜像数据库与主数据库不能存在于同一个系统上。它必须位于具有相同目录结构的辅助系统上并使用与主数据库相同的实例名。如果镜像数据库与主数据库必须存在于同一个系统上，那么可使用 `db2relocatedb` 实用程序或 `db2inidb` 命令的 `RELOCATE USING` 选项来实现这一点。

5. 在辅助系统上启动数据库实例:

```
db2start
```

6. 在辅助系统上，通过将镜像数据库置于前滚暂挂状态将其初始化:

```
db2inidb database_alias as standby
```

如果需要，那么指定 `db2inidb` 命令的 `RELOCATE USING` 选项来重定位备用数据库:

```
db2inidb database_alias as standby relocate using relocatedbcfg.txt
```

其中 `relocatedbcfg.txt` 文件包含重定位数据库所需的信息。

注：

- a. 如果您具有 DMS 表空间（数据库管理的空间）或自动存储器表空间，那么可以使用分割镜像建立完整的数据库备份。使用分割镜像建立备份可减少在生产数据库上建立备份的开销。
 - b. 必须将数据库目录（包括卷目录）、日志目录和容器目录移至期望的位置，然后才能使用 `RELOCATE USING` 选项。
7. 设置用户出口程序以从主系统中检索日志文件。
 8. 将数据库前滚至日志末尾或时间点。
 9. 继续检索日志文件，并通过日志前滚数据库，直到到达日志的末尾或备用数据库要求的时间点。
 10. 要使备用数据库联机，发出指定了 `STOP` 选项的 `ROLLFORWARD` 命令。

注： 在使镜像数据库离开前滚暂挂状态之后，就不能对其应用来自主数据库的日志。

配置 DB2 高可用性灾难恢复 (HADR) 同步方式

HADR_SYNCMODE 配置参数决定 DB2 高可用性灾难恢复 (HADR) 数据库解决方案防止事务丢失的程度。同步方式决定主数据库服务器根据备用数据库上记录的状态, 何时认为事务已完成。同步方式配置参数值越严格, 数据库解决方案就越能防止事务数据丢失, 但事务处理速度也越慢。您必须在防止事务丢失的需求与对性能需求之间取得平衡。

这些方式仅在主数据库与备用数据库处于对等或断开对等状态时适用。

使用 HADR_SYNCMODE 配置参数来设置同步方式。有效值为:

SYNC (同步)

此方式可以尽可能地避免事务丢失, 在三种方式中, 使用此方式会导致事务响应时间最长。

在此方式中, 仅当日志已写入主数据库上的日志文件, 而且主数据库已接收到来自备用数据库的应答, 确定日志也已写入备用数据库上的日志文件时, 方才认为日志写入是成功的。保证日志数据同时存储在这两处。

如果备用数据库在重放日志记录之前崩溃, 那么它下次启动时, 可从其本地日志文件中检索和重放这些记录。如果主数据库发生故障, 故障转移至备用数据库可以保证任何已在主数据库上落实的事务也在备用数据库上落实了。执行故障转移操作后, 当客户机重新与新的主数据库连接时, 可能会有在新的主数据库上已落实的事务, 对于原始主数据库却从未报告为已落实。当主数据库在处理来自备用数据库的应答消息之前出现故障时, 即会出现此种情况。客户机应用程序应考虑查询数据库以确定是否存在此类事务。

如果主数据库与备用数据库的连接断开, 接下来的操作将取决于 `hadr_peer_window` 数据库配置参数的配置。如果 `hadr_peer_window` 设置为非零时间值, 那么与备用数据库的连接断开时, 主数据库将转为断开对等状态, 并继续等待, 直到备用数据库确认后再落实事务。如果 `hadr_peer_window` 数据库配置参数设置为零, 主数据库和备用数据库将不在处于对等状态, 且不会保留事务而等待备用数据库确认。如果在数据库不处于对等或断开对等状态期间进行了故障转移操作, 那么将不能保证在主数据库上落实的所有事务都会出现在备用数据库上。

当数据库处于对等或断开对等状态时, 如果主数据库失败, 那么它可以在完成故障转移操作之后, 作为备用数据库重新加入 HADR 数据库对。因为在主数据库接收到来自备用数据库的应答, 确认日志已写入备用数据库上的日志文件之前, 不认为事务已落实, 所以主数据库上的日志顺序将与备用数据库上的日志顺序相同。原始主数据库 (现在是备用数据库) 只需要通过重放自从执行故障转移操作以来, 在新的主数据库上生成的新日志记录来进行同步复制。

如果主数据库发生故障时并未处于对等状态, 那么其日志顺序可能与备用数据库上的日志顺序不同。如果必须执行故障转移操作, 主数据库和备用数据库上的日志顺序可能不同, 因为在故障转移之后, 备用数据库启动自己的日志顺序。因为无法撤销某些操作 (例如, 删除表), 所以不可能将主数据库回复到创建新的日志顺序的时间点。如果日志顺序不同, 并且在原始主数据库发出了带有 `STANDBY` 选项的 `START HADR` 命令, 那么将收到命令成功的消息。但是, 此消息是在试图重新集成之前发出的。如果重新集成失败, HADR 对验证消息将发送至主数据库和备用数据库的管理日志和诊断日志。被重新集成的备用数据库将保持等候, 但在进行对验证期间, 主数据库将拒绝备用数据库,

导致备用数据库关闭。如果原始主数据库成功地重新加入 HADR 对，那么可以通过发出未指定 BY FORCE 选项的 TAKEOVER HADR 命令来完成数据库的故障回退。如果原始主数据库无法重新加入 HADR 对，那么可以通过复原新的主数据库的备份映像将其重新初始化为备用数据库。

NEARSYNC (接近同步)

此方式具有比同步方式更短的事务响应时间，但针对事务丢失提供的保护也较少。

在此方式中，仅当日志记录已写入主数据库上的日志文件，而且主数据库已接收到来自备用系统的应答，确定日志也已写入备用系统上的主存储器时，方才认为日志写入是成功的。仅当两处同时发生故障，并且目标位置未将接收到的所有日志数据转移至非易失性存储器时，才会出现数据的丢失。

如果备用数据库在其将日志记录从存储器复制到磁盘之前崩溃，那么备用数据库上将丢失日志记录。通常，当备用数据库重新启动时，它可以从主数据库中获取丢失的日志记录。但是，如果主数据库或网络上的故障使检索无法进行，并且需要故障转移，那么日志记录将不会出现在备用数据库上，而且与这些日志记录相关联的事务将不会出现在备用数据库上。

如果事务丢失，那么在故障转移操作之后，新的主数据库与原始主数据库不相同。客户机应用程序应该考虑重新提交这些事务，以便使应用程序状态保持最新。

当主数据库和备用数据库处于对等状态时，如果主数据库发生故障，那么在没有使用完全复原操作重新初始化的情况下，原始主数据库可能无法作为备用数据库重新加入 HADR 对。如果故障转移涉及丢失的日志记录（因为主数据库和备用数据库已发生故障），主数据库和备用数据库上的日志顺序将会不同，并且在未首先执行复原操作的情况下，重新启动原始主数据库以作为备用数据库的尝试将会失败。如果原始主数据库成功地重新加入 HADR 对，那么可以通过发出未指定 BY FORCE 选项的 TAKEOVER HADR 命令来完成数据库的故障回退。如果原始主数据库无法重新加入 HADR 对，那么可以通过复原新的主数据库的备份映像将其重新初始化为备用数据库。

ASYNCR (异步)

如果主系统发生故障，此方式发生事务丢失的机率最高。在三种方式之中，此方式的事务响应时间也是最短的。

在此方式中，仅当日志记录已写入主数据库上的日志文件，而且已将此记录传递给主系统主机的 TCP 层时，方才认为日志写入是成功的。因为主系统不会等待来自备用系统的应答，所以当事务仍处于正在传入备用系统的过程中时，可能会认为事务已落实。

主数据库主机上、网络上或备用数据库上的故障可能导致传送中的日志记录丢失。如果主数据库可用，那么会在重新建立“对”连接时，将丢失的日志记录重新发送至备用数据库。但是，如果在有丢失日志记录的情况下需要执行故障转移操作，那么那些日志记录将不会到达备用数据库，从而导致在故障转移中丢失关联的事务。

如果事务丢失，那么在故障转移操作之后，新的主数据库与原始主数据库不是完全相同的。客户机应用程序应该考虑重新提交这些事务，以便使应用程序状态保持最新。

当主数据库和备用数据库处于对等状态时，如果主数据库发生故障，那么在没使用完全复原操作重新初始化的情况下，原始主数据库可能无法作为备用数据库重新加入 HADR 对。如果故障转移涉及丢失的日志记录，主数据库和备用数据库上的日志顺序将会不同，并且重新启动原始主数据库以作为备用数据库的尝试将失败。因为，如果在异步方式中发生故障转移，日志记录更有可能丢失，所以主数据库将不能重新加入 HADR 对的可能性也更大。如果原始主数据库成功地重新加入 HADR 对，那么可以通过发出未指定 BY FORCE 选项的 TAKEOVER HADR 命令来完成数据库的故障回退。如果原始主数据库无法重新加入 HADR 对，那么可以通过复原新的主数据库的备份映像将其重新初始化为备用数据库。

高可用性灾难恢复（HADR）支持

为获得 DB2 数据库高可用性灾难恢复（HADR）功能的最佳性能，请在设计高可用性数据库解决方案时考虑系统要求和功能限制。

高可用性灾难恢复（HADR）的系统要求

为了获取高可用性灾难恢复（HADR）的最佳性能，请确保系统满足下列硬件、操作系统和 DB2 数据库系统要求。

建议：为获得更好的性能，应对主数据库所在的系统和备用数据库所在的系统使用相同的硬件和软件。如果备用数据库所在的系统比主数据库所在的系统拥有的资源少，那么备用数据库可能无法与主数据库所生成的事务装入保持一致。这会导致备用数据库落后，或者主数据库性能降低。在故障转移情况下，新的主数据库应该拥有足够的资源以服务于客户机应用程序。

硬件和操作系统要求

建议：HADR 主数据库和备用数据库使用相同的主机。即，它们应该来自相同的供应商，而且具有相同的体系结构。

主数据库和备用数据库上的操作系统版本（包括补丁）应该相同。在滚动升级过程中可以暂时违反此规则，但要极为谨慎对待。

TCP/IP 接口在 HADR 主机之间必须可用，并且建议使用高速、大容量网络。

DB2 数据库要求

主数据库和备用数据库的数据库系统的版本必须相同；例如，都必须为版本 8 或版本 9。在滚动升级期间，为了测试新的级别，备用数据库的数据库系统的修改级别（例如，修订包级别）在短时间内可以比主数据库的数据库系统的修改级别高。但是，不能长时间保持此配置。如果主数据库的数据库系统的修改级别比备用数据库的数据库系统的修改级别高，那么主数据库与备用数据库不会彼此连接。

主数据库和备用数据库的 DB2 数据库软件的位大小必须相同（32 位或 34 位）。主数据库和备用数据库上的表空间及其容器必须完全相同。必须相同的属性包括表空间类型（DMS 或 SMS）、表空间大小、容器路径、容器大小和容器文件类型（原始设备或文件系统）。在主数据库和备用数据库上，为日志文件分配的空间量也应该相同。

当您为主数据库发出表空间语句（如 CREATE TABLESPACE、ALTER TABLESPACE 或 DROP TABLESPACE）时，会对备用数据库重放此语句。在您为主数据库发出表空间语句之前，必须确保在这两个数据库上安装了所涉及的设备。

如果在主数据库上创建表空间，并且对备用数据库执行的日志重放由于容器不可用而失败，那么主数据库不会接收到指示日志重放失败的错误消息。

要检查日志重放错误，在创建新的表空间时必须监视 db2diag 日志以及备用数据库上的管理通知日志。

如果执行了接管操作，那么您创建的新表空间在新的主数据库上不可用。要从这种情况恢复，那么使用备份映像在新的主数据库上复原表空间。

在以下示例中，先在数据库 MY_DATABASE 上复原表空间 MY_TABLESPACE，然后再将该数据库用作新的主数据库：

1. db2 connect to my_database
2. db2 list tablespaces show detail

注：运行 db2 list tablespaces show detail 命令以显示所有表空间的状态以及获取步骤 5 所需的表空间标识号。

3. db2 stop hadr on database my_database
4. db2 "restore database my_database tablespace (my_tablespace) online redirect"
5. db2 "set tablespace containers for my_tablespace_ID_# ignore rollforward container operations using (path '/my_new_container_path/')" "
6. db2 "restore database my_database continue"
7. db2 rollforward database my_database to end of logs and stop tablespace "(my_tablespace)"
8. db2 start hadr on database my_database as primary

主数据库和备用数据库不需要相同的数据库路径。如果使用相对容器路径，那么同一个相对路径可能映射至主数据库和备用数据库上的不同绝对容器路径。

HADR 完全支持自动存储器数据库，包括指定了 ADD STORAGE ON 子句的 ALTER DATABASE 语句的复制。与表空间容器类似，存储路径在主数据库和备用数据库上都必须存在。

主数据库和备用数据库的数据库名必须相同。这表示它们必须在不同的实例中。

不支持重定向复原。即，HADR 不支持重定向表空间容器。但是，支持更改数据库目录和日志目录。使用相对路径创建的表空间容器将被复原到相对于新的数据库目录的路径。

缓冲池要求

因为也会在备用数据库上重放缓冲池操作，所以主数据库和备用数据库具有相同的内存量这一点至关重要。

高可用性灾难恢复 (HADR) 的安装和存储要求

为了获取高可用性灾难恢复 (HADR) 的最佳性能, 请确保系统满足下列安装和存储要求。

安装要求

对于 HADR, 实例路径在主数据库和备用数据库上应该相同。使用不同的实例路径在某些情况下可能会导致出现问题, 例如, 如果 SQL 存储过程调用用户定义的函数 (UDF) 并且期望 UDF 对象代码对于主服务器和备用服务器都在同一目录下。

存储要求

HADR 完全支持自动存储数据库, 包括指定了 `ADD STORAGE ON` 子句的 `ALTER DATABASE` 语句的复制。与表空间容器类似, 存储路径在主数据库和备用数据库上都必须存在。可以使用符号链接来创建完全相同的路径。主数据库和备用数据库可以在同一计算机上。即使它们的数据库存储器以相同的路径开头, 也不会发生冲突, 这是因为使用的实际目录在目录中嵌入了实例名 (由于主数据库与备用数据库的数据库名必须相同, 所以它们必须在不同的实例中)。存储路径的格式为 `storage_path_name/inst_name/dbpart_name/db_name/tbsp_name/container_name`。

主数据库和备用数据库上的表空间及其容器必须完全相同。属性必须完全相同, 包括: 表空间类型 (DMS 或 SMS)、表空间大小、容器路径、容器大小和容器文件类型 (原始设备或文件系统)。如果数据库启用了自动存储器, 那么存储路径必须完全相同。这包括路径名和每个路径上专用于数据库的空间量。在主数据库和备用数据库上, 为日志文件分配的空间量也应该相同。

当您对主数据库发出表空间语句 (如 `CREATE TABLESPACE`、`ALTER TABLESPACE` 或 `DROP TABLESPACE`) 时, 会对备用数据库重放此语句。在您对主数据库发出表空间语句之前, 必须确保在这两个数据库上安装了所涉及的设备。

如果主数据库和备用数据库上的表空间设置不同, 那么对备用数据库重放日志时可能会遇到错误, 如 `OUT OF SPACE` 或 `TABLE SPACE CONTAINER NOT FOUND`。同样, 如果数据库启用了自动存储器而存储路径不同, 那么将不会重放与 `ALTER DATABASE` 语句的 `ADD STORAGE ON` 子句相关联的日志记录。因此, 现有存储路径可能会提前用备用系统上的空间, 而自动存储器表空间将不能增加大小。如果发生任一上述情况, 就会将受影响的表空间置于前滚暂挂状态并在后面重放日志时忽略该表空间。如果执行接管操作, 那么表空间将不可用于应用程序。

如果在接管之前发现备用数据库上有问题, 那么解决方法是在解决存储器问题时重新建立备用数据库。执行此操作的步骤包括:

- 取消激活备用数据库。
- 删除备用数据库。
- 确保必需的文件系统存在并具有足够的可用空间用于后续复原和前滚。
- 使用主数据库的最新备份复原备用系统上的数据库 (或者, 使用 `db2inidb` 命令通过分割镜像或快速复制来进行重新初始化)。如果主数据库启用了自动存储器, 那么在复原期间不要重新定义存储路径。另外, 不应将表空间容器作为复原的一部分进行重定向。
- 在备用系统上重新启动 HADR。

但是，如果在执行接管之后才发现备用数据库有问题（或者如果选择了直到此时才解决存储器问题），那么解决方案取决于所遇到问题的类型。

如果数据库启用了自动存储器，但在与备用数据库相关联的存储路径上没有可用空间，那么请遵循以下步骤：

1. 通过扩展文件系统或除去文件系统中不必要的非 DB2 文件来使存储路径上的空间可用。
2. 执行表空间前滚至日志末尾。

在不会发生添加或扩展容器以作为日志重放的一部分的情况下，如果必需的备份映像和日志文件归档都可用，那么您可能可以恢复该表空间，方法是首先发出带有 IGNORE ROLLFORWARD CONTAINER OPERATIONS 选项的 SET TABLESPACE CONTAINERS 语句，然后发出 ROLLFORWARD 命令。

主数据库和备用数据库不需要相同的数据库路径。如果使用相对容器路径，那么同一个相对路径可能映射至主数据库和备用数据库上的不同绝对容器路径。因此，如果主数据库和备用数据库在同一台计算机上，那么必须使用相对路径来定义所有表空间容器，以使它们映射到主数据库和备用数据库的不同路径。

HADR 和网络地址转换 (NAT) 支持

NAT 有助于减少可用 IPv4 地址出现必然不足的情况。它被视为 IPv6 解决方案（可能用数十年时间实现）的快速修正备用解决方案。NAT 在 HADR 环境中受支持。

HADR 和 NAT 支持

HADR 始终对主和备用节点上本地和远程主机配置进行交叉检查。

在 NAT 环境中，主机通过一个 IP 地址为其本身所知，但通过另一个 IP 地址为其他主机所知。此行为将导致 HADR 主机交叉检查失败。

为了避免在 NAT 环境中发生这种情况，可以将注册表变量 **DB2_HADR_NO_IP_CHECK** 设置为 ON。这将绕过主机交叉检查，从而使主和备用节点能够在 NAT 环境中进行连接。

如果没有在 NAT 环境中运行，那么建议将 **DB2_HADR_NO_IP_CHECK** 注册表变量保留为缺省设置 OFF。禁用交叉检查会削弱 HADR 的配置验证。

高可用性灾难恢复 (HADR) 的限制

为了获取高可用性灾难恢复 (HADR) 的最佳性能，请在设计高可用性 DB2 数据库解决方案时考虑 HADR 的限制。

下表对高可用性灾难恢复 (HADR) 的限制进行了概述：

- 在分区数据库环境中，不支持 HADR。
- 主数据库和备用数据库的操作系统版本和 DB2 数据库系统版本都必须相同，但滚动升级期间可有短时不同。
- 主数据库和备用数据库的 DB2 数据库系统软件的位大小必须相同（32 位或 64 位）。
- 不支持备用数据库上的读操作。客户机无法与备用数据库连接。
- 只能由当前主数据库执行日志归档。

- 只能对当前主数据库运行自调整内存管理器（STMM）。在主数据库启动或备用数据库通过接管而转换为主数据库后，直到第一个客户机连接生效，STMM EDU 才可能启动。
- 在备用数据库上不支持备份操作。
- 未进行日志记录的操作（例如对数据库配置参数和恢复历史记录文件所作的更改）不会被复制到备用数据库。
- 不支持指定了 COPY NO 选项的装入操作。
- HADR 不支持对数据库日志文件使用原始 I/O（直接磁盘访问）。如果 HADR 是通过 START HADR 命令启动的，或者在配置了 HADR 的情况下激活（重新启动）数据库，并且检测到原始日志，那么相关联的命令将失败。
- 对于一阶段落实，HADR 数据库可以充当联合服务器（事务管理器）或数据源（资源管理器）。对于两阶段落实，HADR 数据库只能充当数据源。

为获取高可用性安排维护

DB2 数据库解决方案需要定期维护。您将需要执行诸如以下的维护：软件或硬件升级、数据库性能调整、数据库备份、统计信息收集和商业目的的监视。您必须尽量降低这些维护活动对数据库解决方案可用性的影响。

在安排维护活动之前，必须先确定您必须对数据库解决方案执行的维护活动。

要安排维护，请执行下列步骤：

1. 确定数据库活动较少的期间。

最好是将维护活动安排在使用率低的时间进行（对数据库系统发出请求的用户应用程序最少的时间段。）根据您创建的业务应用程序类型，甚至可能有没有用户应用程序访问数据库的时间段。

2. 按照以下准则对必须执行的维护活动分类：

- 维护可以自动进行
- 执行维护时必须将数据库解决方案脱机
- 可以在数据库解决方案联机时执行维护

3. 对于那些可以自动进行的维护活动，使用下列方法中的一种来配置自动维护：

- 使用 auto_maint 配置参数
- 使用“配置自动维护”向导
- 使用名为 AUTOMAINT_SET_POLICY 和 AUTOMAINT_SET_POLICYFILE 的系统存储过程之一

4. 如果必须执行的任何维护活动需要数据库服务器脱机，那么请安排在使用率较低的时间进行这些脱机维护活动。

5. 对于那些可以在数据库服务器联机时执行的维护活动：

- 确定运行这些联机维护活动对可用性的影响。
- 按照尽可能降低运行这些维护活动对数据库系统可用性的影响的原则安排这些联机维护活动。

例如：安排在使用率低的时间进行联机维护活动；以及使用调速机制来均衡维护活动使用的系统资源量。

使用 `SYSPROC.AUTOMAINT_GET_POLICY` 或 `SYSPROC.AUTOMAINT_GET_POLICYFILE` 来收集自动维护策略信息

可以使用系统存储过程 `AUTOMAINT_GET_POLICY` 和 `AUTOMAINT_GET_POLICYFILE` 来检索为数据库配置的自动维护策略。

要检索数据库的自动维护策略，请执行下列步骤：

1. 连接至数据库
2. 调用 `AUTOMAINT_GET_POLICY` 或 `AUTOMAINT_GET_POLICYFILE`
 - `AUTOMAINT_GET_POLICY` 所需的参数有：
 - a. 维护类型，指定要返回其信息的自动维护活动的类型。
 - b. 指向 `BLOB` 的指针，该过程将在 `BLOB` 中以 `XML` 格式返回自动维护策略信息。
 - `AUTOMAINT_GET_POLICYFILE` 所需的参数有：
 - a. 维护类型，指定要返回其信息的自动维护活动的类型。
 - b. 该过程将自动维护策略信息输出到的文件的名称。

有效维护类型值有：

- `AUTO_BACKUP` - 自动备份
- `AUTO_REORG` - 表和索引自动重组
- `AUTO_RUNSTATS` - 自动表 `runstats` 操作
- `MAINTENANCE_WINDOW` - 维护窗口

使用 `SYSPROC.AUTOMAINT_SET_POLICY` 或 `SYSPROC.AUTOMAINT_SET_POLICYFILE` 来配置自动维护策略

可以使用系统存储过程 `AUTOMAINT_SET_POLICY` 和 `AUTOMAINT_SET_POLICYFILE` 来为数据库配置自动维护策略。

要配置数据库的自动维护策略，请执行下列步骤：

1. 连接至数据库
2. 调用 `AUTOMAINT_SET_POLICY` 或 `AUTOMAINT_SET_POLICYFILE`
 - `AUTOMAINT_SET_POLICY` 所需的参数有：
 - a. 维护类型，指定要配置的自动维护活动的类型。
 - b. 指向以 `XML` 格式指定自动维护策略的 `BLOB` 的指针。
 - `AUTOMAINT_SET_POLICYFILE` 所需的参数有：
 - a. 维护类型，指定要配置的自动维护活动的类型。
 - b. 指定自动维护策略的 `XML` 文件的名称。

有效维护类型值有：

- `AUTO_BACKUP` - 自动备份
- `AUTO_REORG` - 表和索引自动重组
- `AUTO_RUNSTATS` - 自动表 `runstats` 操作
- `MAINTENANCE_WINDOW` - 维护窗口

AUTOMAINT_SET_POLICY 或 AUTOMAINT_SET_POLICYFILE 的自动维护策略规范 XML 样本

无论是使用 AUTOMAINT_SET_POLICY 还是使用 AUTOMAINT_SET_POLICYFILE 来指定自动维护策略，都必须使用 XML 来指定策略。SQLLIB/samples/automaintcfg 中有一些样本文件，演示如何用 XML 来指定自动维护策略。

传递到系统存储过程 AUTOMAINT_SET_POLICY 的第二个参数是包含 XML 的 BLOB，它指定所需的自动维护策略。传递到系统存储过程 AUTOMAINT_SET_POLICYFILE 的第二个参数是指定所需自动维护策略的 XML 文件的名称。传递到 AUTOMAINT_SET_POLICY 的 BLOB 中的有效 XML 元素与传递到 AUTOMAINT_SET_POLICYFILE 的 XML 文件中的有效 XML 元素相同。

在样本目录 SQLLIB/samples/automaintcfg 中，有四个包含自动维护策略规范示例的 XML 文件：

DB2MaintenanceWindowPolicySample.xml

演示如何指定数据库管理器应在其间安排自动维护的维护窗口。

DB2AutoBackupPolicySample.xml

演示指定数据库管理器应如何执行自动备份。

DB2AutoReorgPolicySample.xml

演示指定数据库管理器应如何执行表和索引自动重组（其中包括从多维集群（MDC）表收回扩展数据块）。

DB2DefaultAutoRunstatsPolicySample.xml

演示指定数据库管理器应如何执行自动表 runstats 操作。

可以通过从这些文件复制 XML 并根据系统要求修改该 XML 来创建自己的自动维护策略规范 XML。

配置数据库日志记录选项

使用数据库日志记录配置参数来指定数据库的数据日志记录选项，例如要使用的日志记录的类型、日志文件的大小以及日志文件的存储位置。

要配置数据库日志记录选项，您必须拥有 SYSADM、SYSCTRL 或 SYSMANT 权限。

可以通过在命令行处理器（CLP）中使用 UPDATE DATABASE CONFIGURATION 命令、在控制中心中使用“配置数据库日志记录”向导 GUI 或通过调用 db2CfgSet API 来配置数据库日志记录选项。

- 要在命令行处理器中使用 UPDATE DATABASE CONFIGURATION 命令来配置数据库日志记录选项：
 1. 指定是使用循环日志记录功能还是归档日志记录功能。如果要使用循环日志记录功能，那么必须将 LOGARCHMETH1 和 LOGARCHMETH2 数据库配置参数设置为 OFF。这是缺省设置。要使用归档日志记录功能，必须至少将这些数据库配置参数中的一个参数设置为非 OFF 值。例如，如果要使用归档日志记录功能，并且要将归档的日志保存到磁盘，请发出以下命令：

```
db2 update db configuration for mydb using logarchmeth1
disk:/u/dbuser/archived_logs
```

归档的日志将保存在 `/u/dbuser/archived_logs` 目录中。

2. 根据需要，对其他数据库日志记录配置参数指定值。其他用于数据库日志记录的配置参数如下所示：

- ARCHRETRYDELAY
- BLK_LOG_DSK_FUL
- FAILARCHPATH
- LOGARCHOPT1
- LOGARCHOPT2
- LOGBUFSZ
- LOGFILSIZ
- LOGPRIMARY
- LOGRETAIN
- LOGSECOND
- MAX_LOG
- MIRRORLOGPATH
- NEWLOGPATH
- MINCOMMIT
- NUMARCHRETRY
- NUM_LOG_SPAN
- OVERFLOWLOGPATH
- USEREXIT

要了解有关这些数据库日志记录配置参数的更多信息，请参阅『用于数据库日志记录的配置参数』。

- 要打开“配置数据库日志记录”向导：
 1. 在控制中心中，展开对象树，直到找到要为其设置日志记录选项的数据库为止。
 2. 用鼠标右键单击该数据库，然后从弹出菜单中选择“配置数据库日志记录”。这将打开“配置数据库日志记录”向导。
- 在控制中心中通过联机帮助工具提供了详细信息。

用于数据库日志记录的配置参数

任何高可用性策略的一个关键元素是数据库日志记录。可以使用数据库日志来记录事务信息、使主数据库与辅助或备用数据库同步并前滚已接管出现故障的主数据库的辅助数据库。要根据您的需求配置这些数据库日志记录活动，必须设置各种数据库配置参数。

归档重试延迟 (`archretrydelay`)

指定在上一次尝试失败之后，归档日志文件尝试之间等待的时间量（以秒计）。缺省值为 20。

日志磁盘已满时挂起 (`blk_log_dsk_ful`)

可以设置此配置参数以防止当 DB2 不能在活动日志路径中创建新日志文件时出现“磁盘已满”错误。相反，DB2 将尝试每隔五分钟就创建一次日志文件，直至成功。每次尝试之后，DB2 都会将一条消息写至管理通知日志。确认应用程序因为日志磁盘已满情况而挂起的唯一的方法就是监视管理通知日志。在成功创

建日志文件之前，尝试更新表数据的所有用户应用程序都不能落实事务。只读查询可能不会直接受影响；但是，如果查询需要访问被更新请求锁定的数据或者由更新应用程序在缓冲池中修正的数据页时，只读查询也将像挂起一样。

将 *blk_log_dsk_ful* 设置为 YES 导致 DB2 遇到日志磁盘已满错误时应用程序挂起。于是您就能够解决错误，而事务继续运行。磁盘已满情况可以通过将旧的日志文件移至另一文件系统或增加文件的大小以使挂起应用程序能够完成来解决。

如果 *blk_log_dsk_ful* 设置为 NO，那么接收到日志磁盘已满错误的事务将失败并被回滚。在某些情况下，如果事务导致日志磁盘已满错误，那么数据库将崩溃。

故障转移归档路径 (*failarchpath*)

如果指定的日志归档方法失败，那么为归档日志文件指定备用目录。在失败的日志归档方法再次可用之前，此目录是日志文件的临时存储器，此时日志文件将从此目录中移至日志归档方法。通过将日志文件移动至该临时位置，可以避免日志目录发生已满情况。此参数必须是一个标准现有目录。

日志归档方法 1 (*logarchmeth1*)、日志归档方法 2 (*logarchmeth2*)

这些参数使数据库管理器将日志文件归档至活动日志路径之外的位置。如果指定这两个参数，每个日志文件均归档两次。这意味着您将拥有两个位于不同位置的归档日志文件副本。

这些参数的有效值包括介质类型，且在某些情况下，包括目标字段。使用冒号 (:) 来分隔值。有效值为：

OFF 指定不使用日志归档方法。如果 *logarchmeth1* 和 *logarchmeth2* 都设置为 OFF，那么认为数据库正在使用循环日志记录，且不可前滚恢复。这是缺省值。

LOGRETAIN

此值仅可用于 *logarchmeth1*，且等价于将 *logretain* 配置参数设置为 RECOVERY。如果指定此值，将自动更新 *logretain* 配置参数。

USEREXIT

此值仅对 *logarchmeth1* 有效，且等价于将 *userexit* 配置参数设置为 ON。如果指定此值，将自动更新 *userexit* 配置参数。

DISK 此值后必须紧跟冒号 (:)，然后是现有标准路径名，日志文件将在其中归档。例如，如果将 *logarchmeth1* 设置为 DISK:/u/dbuser/archived_logs，那么将归档日志文件放入名为 /u/dbuser/archived_logs 的目录。

注：如果正在归档至磁带，可以使用 *db2tapemgr* 实用程序来存储和检索日志文件。

TSM 如果指定不带任何附加配置参数，此值指示应该使用缺省管理类，将日志文件归档在本地 TSM 服务器上。如果此值后紧跟冒号 (:) 和 TSM 管理类，那么使用指定的管理类来归档日志文件。

VENDOR

指定将使用供应商库来归档日志文件。此值后必须紧跟冒号 (:) 和库的名称。库中提供的 API 必须使用备份并复原供应商产品的 API。

注：

1. 如果将 *logarchmeth1* 或 *logarchmeth2* 设置为 OFF 以外的值，那么必须配置数据库以进行前滚恢复。
2. 如果更新 *userexit* 或 *logretain* 配置参数，将自动更新 *logarchmeth1*，反之亦然。但是，如果您要使用 *userexit* 或 *logretain*，必须将 *logarchmeth2* 设置为 OFF。

日志归档选项 1 (*logarchopt1*)、日志归档选项 2 (*logarchopt2*)

指定传递至 TSM 服务器或供应商 API 的字符串。对于 TSM，此字段用于允许数据库检索在不同 TSM 节点或通过不同 TSM 用户生成的日志。字符串必须以如下格式提供：

```
"-fromnode=nodename -fromowner=ownername"
```

其中 *nodename* 是最初归档日志文件的 TSM 节点的名称，*ownername* 是最初归档日志文件的 TSM 用户的名称。每个日志归档选项字段对应于一个日志归档方法：*logarchopt1* 与 *logarchmeth1* 配合使用，*logarchopt2* 与 *logarchmeth2* 配合使用。

日志缓冲区 (*logbufsz*)

此参数允许您指定在将日志记录写至磁盘之前用作这些记录的缓冲区的内存量。当发生下列任何一项事件时会将日志记录写入磁盘：

- 事务落实
- 日志缓冲区已满
- 发生了某些其他的内部数据库管理器事件。

增加日志缓冲区的大小可使与日志记录相关的输入/输出 (I/O) 活动更有效，因为将日志记录写到磁盘中的频率更低，而每次写入的记录却更多。但是，如果日志缓冲区大小值较大，执行恢复所需的时间就会较长。

日志文件大小 (*logfilsiz*)

此参数以 4 KB 的页数指定每个配置日志的大小。

对可配置的总活动日志空间有 1024 GB 的逻辑限制。此限制源自每个日志文件的上限（即 4 KB）以及最大主日志文件数与最大辅助日志文件数之和（即 256）。

日志文件的大小对性能有直接的影响。从一个日志切换至另一个日志需要付出性能代价。因此，从纯性能角度来说，日志文件大小越大越好。此参数还指示要归档的日志文件大小。这种情况下，日志文件大小越大并不一定越好，因为较大的日志文件大小增加了故障或导致日志装入方案中的延迟的发生机率。当考虑活动日志空间时，最好有较多的较小日志文件。例如，如果有 2 个很大的日志文件，并且事务启动接近一个日志文件的末尾，那么仅有一半日志空间仍然可用。

每当释放数据库（与该数据库的所有连接都终止）时，就截断当前正写入的日志文件。因此，如果一个数据库被频繁释放时，最好不要选择较大日志文件大小，因为 DB2 将创建较大文件而只是将其截断。可以使用 **ACTIVATE DATABASE** 命令来避免此成本，并且预先准备好缓冲池还有助于提高性能。

假定应用程序将数据库保持为打开以使打开数据库时的处理时间最短，日志文件大小应由建立脱机归档日志副本所花的时间确定。

将日志文件的丢失降低至最小程度，也是设置日志大小时的一个重要注意事项。归档会占用整个日志。如果使用单个的大日志，那么会增加归档之间的时

间。如果介质包含日志故障，某些事务信息将可能丢失。减小日志大小会增加归档的频率，但可以减少由于介质故障而丢失信息，因为可以使用丢失的日志之前的那些较小的日志。

日志保留 (logretain)

此配置参数已由 *logarchmeth1* 替换。为了与先前版本的 DB2 兼容，仍支持此参数。

如果 *logretain* 设置为 RECOVERY，归档日志将保留在数据库日志路径目录中，而将数据库视为可恢复的数据库则意味着启用了前滚恢复。

注：*logretain* 数据库配置参数的缺省值不支持前滚恢复。如果打算使用前滚恢复，必须更改此参数的值。

每个事务的最大日志数 (max_log)

此参数指示一个事务可以消耗的主日志空间的百分比。该值是为 *logprimary* 配置参数指定的值的百分比。

如果该值设置为 0，那么对一个事务可以消耗的总的主日志空间的百分比没有限制。如果应用程序违反了 *max_log* 配置，那么将强制该应用程序与数据库断开连接，事务将被回滚，并且将返回错误 SQL1224N。

可以通过将 *DB2_FORCE_APP_ON_MAX_LOG* 注册表变量设置为 *FALSE* 来重设此行为。这将导致违反了 *max_log* 配置的事务失败，并返回错误 SQL0964N。该应用程序仍然可以落实在工作单元中由先前语句完成的工作，它也可以回滚已完成的工作以撤销该工作单元。

当启用了无限活动日志空间时，此参数和 *num_log_span* 配置参数非常有用。如果打开了无限记录（即，*logsecond* 为 -1），那么事务数不受日志文件数的上限（*logprimary* + *logsecond*）限制。当到达 *logprimary* 的值时，DB2 将开始归档活动日志，而不是使事务失败。这样可能会导致问题，例如，有一个长期运行的事务，但一直未落实它（可能是由于应用程序不正确导致的）。如果出现这种情况，那么活动日志空间会不断增长，从而可能使得崩溃恢复性能很差。为了防止这样，可以为 *max_log* 和/或 *num_log_span* 配置参数指定值。

注：由 *max_log* 配置参数施加的限制不适用于下列 DB2 命令：ARCHIVE LOG、BACKUP DATABASE、LOAD、REORG、RESTORE DATABASE 和 ROLLFORWARD DATABASE。

镜像日志路径 (mirrorlogpath)

要防止主日志路径上的日志发生磁盘故障或被无意中删除的情况，可以指定在辅助（镜像）路径上维护完全相同的一组日志。要执行此操作，将此配置参数的值更改为指向另一目录。如果数据库被配置为进行前滚恢复，那么不要将当前存储在镜像日志路径目录中的归档日志移至新位置。

因为可以更改日志路径位置，因此前滚恢复所需的日志可以存在于不同的目录中。在前滚操作期间可更改此配置参数的值，以允许您访问位于多个位置的日志。

必须跟踪这些日志的位置。

直到数据库处于一致状态时才会应用所作的更改。配置参数 *database_consistent* 返回数据库的状态。

要关闭此配置参数，将它的值设置为 DEFAULT。

注:

1. 如果主日志路径是原始设备, 那么此配置参数不受支持。
2. 对此参数指定的值不能是原始设备。

新日志路径 (**newlogpath**)

数据库日志最初是在 `SQLLOGDIR` 中创建的, `SQLLOGDIR` 是数据库目录的子目录。通过将此配置参数的值更改为指向另一目录或另一设备, 可以更改放置活动日志及以后的归档日志的位置。如果数据库被配置为进行前滚恢复, 那么不要将当前存储在数据库日志路径目录中的归档日志移至新位置。

因为可以更改该日志路径的位置, 因此前滚恢复所需的日志可以存在于不同的目录中或不同的设备上。在前滚操作期间可更改此配置参数的值, 以允许您访问位于多个位置的日志。

必须跟踪这些日志的位置。

直到数据库处于一致状态时才会应用所作的更改。配置参数 `database_consistent` 返回数据库的状态。

对组的落实次数 (**mincommit**)

此参数允许您延迟将日志记录写入磁盘, 直到执行了最小数目的落实为止。此延迟可有助于减少与写入日志记录相关的数据库管理器开销, 这样如果您有多个应用程序对数据库运行, 且在很短的时间内该应用程序请求了许多落实, 那么可改进性能。

仅当此参数的值大于 1, 且与该数据库连接的应用程序的数量大于此参数的值时, 才会对落实进行这种分组。落实组合生效时, 保持应用程序落实请求, 直到经过 1 秒钟或落实请求数等于此参数的值为止。

出错时的归档重试次数 (**numarchretry**)

指定在日志文件归档到 `failarchpath` 配置参数指定的路径之前, 使用指定的归档方法归档日志文件的尝试次数。如果设置了 `failarchpath` 配置参数, 那么只能使用该参数。缺省值为 5。

跨越的活动日志文件数 (**num_log_span**)

此参数指示一个活动事务可以跨越的活动日志文件数。如果该值设置为 0, 那么对单个事务可以跨越的日志文件数没有限制。

如果应用程序违反了 `num_log_span` 配置, 那么将强制该应用程序与数据库断开连接, 并且将返回错误 `SQL1224N`。

当启用了无限活动日志空间时, 此参数和 `max_log` 配置参数非常有用。如果打开了无限记录 (即, `logsecond` 为 -1), 那么事务数不受日志文件数的上限 (`logprimary + logsecond`) 限制。当到达 `logprimary` 的值时, DB2 将开始归档活动日志, 而不是使事务失败。这样可能会导致问题, 例如, 有一个长期运行的事务, 但一直未落实它 (可能是由于应用程序不正确导致的)。如果出现这种情况, 那么活动日志空间会不断增长, 从而可能使得崩溃恢复性能很差。为了防止这样, 可以为 `max_log` 和/或 `num_log_span` 配置参数指定值。

注: 由 `num_log_span` 配置参数施加的限制不适用于下列 DB2 命令: `ARCHIVE LOG`、`BACKUP DATABASE`、`LOAD`、`REORG`、`RESTORE DATABASE` 和 `ROLLFORWARD DATABASE`。

溢出日志路径 (**overflowlogpath**)

此参数可以用于几种函数, 这取决于日志记录要求。可以指定位置来让 DB2 查

找前滚操作需要的日志文件。它与 ROLLFORWARD 命令的 OVERFLOW LOG PATH 选项相似，但是，不需要对发出的每个 ROLLFORWARD 命令指定 OVERFLOW LOG PATH 选项，可以只设置此配置参数一次。如果同时使用了这两个选项，那么 OVERFLOW LOG PATH 选项将覆盖该前滚操作的 *overflowlogpath* 配置参数。

如果 *logsecond* 设置为 -1，那么可以指定一个目录以让 DB2 存储从归档文件检索到的活动日志文件。（如果活动日志文件不再存在于活动日志路径中，那么必须检索它们以用于回滚操作）。

如果未指定 *overflowlogpath*，那么 DB2 会将日志文件检索到活动日志路径中。通过指定此参数，可以提供附加资源让 DB2 存储检索到的日志文件。好处包括将 I/O 成本分布到不同的磁盘上，以及允许将更多的日志文件存储在活动日志路径中。

例如，如果对复制使用 db2ReadLog API，那么可以使用 *overflowlogpath* 来指定一个位置让 DB2 搜索此 API 所需的日志文件。如果未找到日志文件（在活动日志路径或溢出日志路径中）且数据库配置为启用 *userexit*，那么 DB2 将检索日志文件。还可以使用此参数来指定目录以让 DB2 存储检索到的日志文件。好处包括降低活动日志路径上的 I/O 成本以及允许将更多的日志文件存储在活动日志路径中。

如果将原始设备配置为活动日志路径，那么在想要将 *logsecond* 设置为 -1 或想要使用 db2ReadLog API 时，必须配置 *overflowlogpath*。

要设置 *overflowlogpath*，指定一个最长 242 个字节的字符串。该字符串必须指向路径名，并且它必须为标准路径名，而不是相对路径名。该路径名必须是目录，而不是原始设备。

注：在分区数据库环境中，数据库分区号自动追加在路径后面。这样做是为了维护多逻辑节点配置中路径的唯一性。

主日志 (logprimary)

此参数指定将创建的大小为 *logfilsiz* 的主日志数。

主日志，无论是空的还是满的，都需要相同的磁盘空间容量。因此，如果配置的日志多于需要的日志，将会不必要地占用磁盘空间。如果配置的日志太少，可能会遇到日志满载的情况。当选择要配置的日志数时，必须考虑建立的每个日志的大小，以及应用程序是否可以处理日志满载的情况。对活动日志空间的总日志文件大小限制为 256 GB。

如果对现有数据库启用前滚恢复，那么将主日志数更改为主日志和辅助日志之和，再加 1。对其启用前滚恢复的数据库中的 LONG VARCHAR 和 LOB 字段记录了其他信息。

辅助日志 (logsecond)

此参数指定创建并用于恢复（如果需要）的辅助日志文件的数目。

如果主日志文件已满，可按需要一次分配一个辅助日志文件（大小为 *logfilsiz*），最多可分配由此参数指定的最大数目。如果此参数设置为 -1，那么将数据库配置为无限活动日志空间。对在数据库上运行的未完成事务的大小或数量没有任何限制。在必须容纳大型作业的环境中（这些作业需要的日志空间比通常分配给主日志的空间多），无限活动日志记录功能非常有用。

注：

1. 要将 *logsecond* 设置为 -1, 必须启用日志归档。
2. 如果此参数设置为 -1, 那么崩溃恢复时间可能会延长, 这是因为 DB2 可能需要检索归档日志文件。

用户出口 (*userexit*)

此配置参数已由 *logarchmeth1* 替换。为了与先前版本的 DB2 兼容, 仍支持此参数。

此参数使数据库管理器调用用户出口程序来归档和检索日志。归档日志文件的位置不是活动日志路径。如果 *userexit* 设置为 ON, 将启用前滚恢复。

用于存储脱机归档日志的设备和用于建立副本的软件的数据传输速度, 最少必须与数据库管理器日志中写入数据的平均速率匹配。如果传输速度跟不上生成新日志数据的速度, 而且日志记录活动持续足够长的时间, 那么可能会用尽磁盘空间。用尽磁盘空间的时间长短由可用磁盘空间量确定。如果发生这种情况, 数据库处理将停止。

当使用磁带或光盘介质时, 数据传输速度最为重要。某些磁带设备需要相同的时间来复制文件, 而与它的大小无关。必须确定归档设备的功能。

磁带设备还有其他注意事项。归档请求的频率是很重要的。例如, 如果完成任何复制操作所需的时间是 5 分钟, 那么日志应该足够大, 以便在工作量高峰期可保存 5 分钟的日志数据。磁带设备可能存在设计限制, 它会限制每天的操作数。当确定日志大小时, 必须考虑这些因素。

注:

1. 此值必须设置为 ON 才能启用无限活动日志空间。
2. *userexit* 数据库配置参数的缺省值不支持前滚恢复, 所以在使用之前必须对其进行更改。

使用 NOT LOGGED INITIALLY 参数减少日志记录

如果应用程序根据主表创建并填充工作表, 并且您不担心这些工作表的可恢复性 (因为可以根据主表方便地重新创建它们), 那么创建工作表时可以在 CREATE TABLE 语句上指定 NOT LOGGED INITIALLY 参数。这样可以减少日志记录并提高性能。

使用 NOT LOGGED INITIALLY 参数的优点是, 不记录在创建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)。这不仅降低了日志记录工作量, 还能提高应用程序性能。还可以对现有表使用带 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 表属性时应该考虑下列事宜:

- 落实时将表的所有更改清仓至磁盘。这意味着落实操作可能会耗用更长时间。
- 如果激活了 NOT LOGGED INITIALLY 属性且发生了未记录的活动, 那么当语句失败或执行 ROLLBACK TO SAVEPOINT 时将回滚整个工作单元 (SQL1476N)。

- 如果正在使用高可用性灾难恢复（HADR），那么不应该使用 NOT LOGGED INITIALLY 表属性。在指定了 NOT LOGGED INITIALLY 选项的情况下，在主数据库上创建的表不会被复制到备用数据库上。在 HADR 备用数据库接管主数据库后尝试访问这样的表将导致错误。
- 执行前滚操作时，不能恢复这些表。如果前滚操作遇到使用 NOT LOGGED INITIALLY 选项创建或改变的表，那么将该表被标记为不可用。在恢复了数据库之后，访问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。要利用不记录日志行为，必须在创建该表的同一个工作单元中填充该表。这就潜在地存在并行。

使用已声明的临时表减少日志记录

如果计划使用已声明的临时表作为工作表，注意下列各项：

- 已声明的临时表不是在目录中创建的。因此，不挂起锁定。
- 不对已声明的临时表执行日志记录，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。
- 只有创建已声明的临时表的应用程序才能访问表的那个实例。
- 当删除应用程序与数据库的连接时，隐式删除该表。
- 工作单元中使用一个已声明临时表的操作中的错误不会导致该工作单元完全回滚。但是，更改已声明的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）将删除已声明的临时表中被该工作单元（或保存点）修改过的所有行。

日志目录已满时分块事务

如果由于没有足够的空间用于新日志文件而导致 DB2 数据库管理器无法在活动日志路径中创建新文件，那么将会产生一些错误，表明磁盘已满。如果设置了 blk_log_dsk_ful 数据库配置参数，那么 DB2 数据库管理器将重复尝试创建新的日志文件，直到成功创建了该文件为止，而不是返回“磁盘已满”错误。

如果设置了 blk_log_dsk_ful 数据库配置参数，那么 DB2 数据库管理器将尝试每隔五分钟就创建一次日志文件，直至成功。如果指定了日志归档方法，那么 DB2 数据库管理器还会检查日志文件归档操作是否已完成。如果归档日志文件已成功归档，那么 DB2 数据库管理器将不活动日志文件重命名为新的日志文件名并继续。每次尝试之后，DB2 数据库管理器都会将一条消息写至管理通知日志。可以确认应用程序因为日志磁盘已满情况而挂起的唯一方法就是监视管理通知日志。

在成功创建日志文件之前，尝试更新表数据的任何用户应用程序都不能落实事务。只读查询可能不会直接受影响；但是，如果查询需要访问被更新请求锁定的数据或者由更新应用程序在缓冲池中修正的数据页时，只读查询也将像挂起一样。

通过日志归档管理日志文件

DB2 数据服务器日志文件归档包含各种操作系统文件处理和调度问题。例如：在某些情况下，DB2 数据库管理器可能尝试检索正在归档的日志文件；此外，如果磁盘正好在

DB2 数据库管理器归档一系列日志文件时出现故障，那么这些日志文件（以及它们包含的事务数据）可能会丢失。正确配置数据库记录可以防止这些类型的潜在问题损害可用性和恢复策略。

以下是适用于所有日志归档方法的一般注意事项：

- 为数据库配置参数 *logarchmeth1* 指定一个值表示您想让数据库管理器使用指定的方法在数据库前滚恢复期间归档文件或检索日志文件。当 Rollforward 实用程序需要一个在日志路径目录中找不到的日志文件时，会发出检索日志文件的请求。
- 如果您要执行下列任何操作，那么不应该使用本地连接的磁带机来存储日志文件：
 - 无限日志记录
 - 联机表空间级恢复
 - 复制
 - 异步读取日志 API (db2ReadLog)
 - 高可用性灾难恢复 (HADR)

任何这些事件都会导致检索日志文件，从而与日志归档操作相冲突。

- 如果正在使用日志归档，那么当活动日志写完时，日志管理器将尝试将它们归档。在某些情况下，如果数据库在日志管理器能够成功记录归档之前被取消激活，那么日志管理器可能会在该数据库被激活时尝试再次归档日志。因此，一个日志文件会多次归档。
- 在归档过程中，当一个日志文件满时，即使该日志文件仍是活动的且需要用于正常的处理，仍将它传送至日志管理器。这使得数据的副本可以尽快地从易丢失数据的介质中转移。传递至日志管理器的日志文件保留在日志路径目录中，直到不再需要它用于正常处理为止。这样就复用了磁盘空间。
- 当日志文件已归档且不包含任何打开事务时，DB2 数据库管理器不会删除该文件但在需要此类文件时将其重命名为下一个日志文件。这将改进性能，因为创建新的日志文件（而不是重命名文件）会写出所有页以保证有足够的磁盘空间。复用磁盘上的页比释放这些页再重新获得所需的页更有效率。
- 在崩溃恢复或回滚期间，DB2 数据库管理器将不检索日志文件，除非 *logsecond* 数据库配置参数设置为 -1。
- 配置日志归档不保证可前滚恢复至故障点，只试图将故障范围缩小。在日志文件写完时，日志管理器将异步归档日志。如果在一个日志文件写完之前包含该日志的磁盘发生故障，那么该日志文件中的数据将会丢失。另外，由于文件是排除等待归档的，磁盘可能会在备份了所有文件之前就发生故障，从而导致队列中的任何日志文件都丢失。

如果日志路径所在的磁盘或设备发生故障，那么只要镜像日志路径所在的磁盘或设备未发生故障，就可以使用 *MIRRORLOGPATH* 数据库配置参数来确保将日志写至辅助路径。

- 每个单独日志文件的配置大小对日志归档都有直接影响。如果每个日志文件都非常大，磁盘发生故障时将会丢失大量数据。如果配置数据库以使用较小的日志文件，那么日志管理器将更频繁地归档日志。

但是，如果您要将数据移到慢速设备（如磁带）上，您可能希望有较大的日志文件以防构建该队列。如果归档每个文件需要大量开销（例如，回绕磁带设备或建立与归档介质的连接），也建议您使用较大日志文件。

- 如果正在使用日志归档，那么当活动日志写完时，日志管理器将尝试将它们归档。在某些情况下，日志管理器将在日志满之前归档日志。如果日志文件由于下列任一原因而被截断，就会出现上述情况：数据库取消激活、发出 ARCHIVE LOG 命令、在联机备份结束时或发出 SET WRITE SUSPEND 命令。

注：要释放未用的日志空间，在将日志文件归档前要将它截断。

- 如果正在将日志和备份映像归档到作为日志和备份映像的存储设备的磁带机，那么需要确保备份映像和归档日志的目标位置不是同一磁带机。因为某些日志归档可能在备份操作进行时发生，所以当这两个进程尝试同时写至同一磁带机时可能发生错误。

在调用用户出口程序或供应商程序来归档或检索日志文件时应注意以下注意事项：

- DB2 数据库管理器在启动用户出口程序来归档日志文件时，以读方式打开该文件。在某些平台上，这可防止用户出口程序能删除日志文件。其他平台（例如，AIX）允许进程（包括用户出口程序）删除日志文件。用户出口程序在日志文件归档后永远不能删除它，因为该文件可能仍是活动的并是崩溃恢复所需的。DB2 数据库管理器管理在归档日志文件时复用的磁盘空间。
- 如果用户出口程序或供应商程序接收到对一个不存在的文件进行归档的请求（因为有多归档请求，而在第一次归档操作成功后即将该文件删除），或接收到对一个不存在的文件进行检索的请求（因为此文件位于另一个目录中，或已到达日志的末尾），那么应忽略此请求并传递一个成功的返回码。
- 在 Windows 操作系统上，不能使用 REXX™ 用户出口来归档日志。
- 用户出口程序或供应商程序应该允许时间点恢复之后存在具有相同名称的不同日志文件；用户出口程序或供应商程序应编写为保留这两个日志文件，并将这些日志文件与正确的恢复路径相关联。
- 如果对使用同一磁带设备归档日志文件的两个或多个数据库启用了用户出口程序或供应商程序，并且在其中一个数据库上发生了前滚操作，那么其他数据库不应该是活动的。如果另一数据库尝试在前滚操作进行时归档日志文件，那么可能找不到前滚操作所需的日志，或归档至磁带设备的新日志文件可能覆盖先前存储在该磁带设备上的日志文件。

为防止这两种情况发生，可以确保在前滚操作期间在调用用户出口程序的数据库分区上没有任何其他数据库是打开的，或者编写一个用户出口程序来处理这种情况。

为获取高可用性配置集群环境

创建机器集群并使用集群管理软件来均衡这些机器上的工作负载是设计高可用性解决方案的一种策略。如果在集群中的一个或几个机器上安装 IBM 数据服务器，那么必须配置集群管理器，以正确地对影响数据库的故障作出反应。另外，还必须配置数据库管理器实例，使它在集群环境中正常工作。

关于此任务

手动配置和管理数据库实例与集群管理器是一项复杂、耗时和易于出错的工作。当诸如停止数据库管理器实例之类的实例配置更改需要集群更改时，DB2 高可用性（HA）功能部件提供了基础结构，用于使数据库管理器能够与集群管理器通信。

过程

1. 安装集群管理软件。

在 AIX、Linux 和 Solaris SPARC 操作系统上，SA MP 与 DB2 企业服务器版、DB2 工作组服务器版、DB2 Connect 企业服务器版以及 DB2 Connect 应用程序服务器版集成在一起。在 Linux 操作系统上，它还与 DB2 Express-C 固定期限许可证（FTL）和 DB2 高可用性功能 Express™ 版集成在一起。在 Windows 操作系统上，SA MP 与所有这些 DB2 数据库产品和功能捆绑在一起，但未与 DB2 安装程序集成在一起。

2. 为集群管理器配置 IBM 数据服务器数据库管理器实例，并为 IBM 数据服务器配置集群管理器。

DB2 高可用性实例配置实用程序（db2haicu）是基于文本的实用程序，您可以使用它在集群环境中配置与管理高可用性数据库。db2haicu 通过查询系统来收集有关数据库实例、集群环境和集群管理器的信息。您可通过参数向 db2haicu 调用、输入文件提供更多信息，或在运行时通过在 db2haicu 提示符处提供信息来实现上述目的。

3. 随着时间的推移，当您的数据库需要更改，并且您需要在集群环境中修改数据库配置时，请继续保持数据库管理器实例配置与集群管理器配置同步。

当诸如停止数据库管理器实例之类的实例配置更改需要集群更改时，DB2 高可用性（HA）功能部件提供了基础结构，用于使数据库管理器能够与集群管理器通信。

不论您是将 db2haicu 与 SA MP 配合使用，还是使用另一个支持 DB2 集群管理器 API 的集群管理器，使用 DB2 HA 功能部件管理集群环境都会比分别维护数据库管理器配置和集群配置更为容易。

集群管理器与 DB2 高可用性（HA）功能部件集成

DB2 高可用性（HA）功能部件允许 IBM 数据服务器与集群管理软件集成。

当在集群环境中停止数据库管理器实例时，必须使集群管理器知道实例已停止。如果集群管理器不知道实例已停止，那么集群管理器可能尝试操作，例如对已停止的实例进行故障转移。当诸如停止数据库管理器实例之类的实例配置更改需要集群更改时，DB2 高可用性（HA）功能部件提供了基础结构，用于使数据库管理器能够与集群管理器通信。

DB2 HA 功能部件由下列元素组成：

- IBM Tivoli System Automation for Multiplatforms（SA MP）与 AIX 和 Linux 上的 IBM 数据服务器捆绑在一起，作为 DB2 高可用性（HA）功能部件的一部分，并且与 DB2 安装程序集成。您可以使用 DB2 安装程序或 IBM 数据服务器安装介质中包含的 installSAM 和 uninstallSAM 脚本来安装、升级或卸载 SA MP。
- 在集群环境中，某些数据库管理器实例配置和管理操作需要相关集群配置更改。每当您执行某些数据库管理器实例配置和管理操作时，DB2 高可用性（HA）功能部件将允许数据库管理器自动请求集群管理器配置更改。请参阅：第 59 页的『使用 DB2 高可用性（HA）功能部件来自动配置集群』
- DB2 高可用性实例配置实用程序（db2haicu）是基于文本的实用程序，您可以使用它在集群环境中配置与管理高可用性数据库。db2haicu 通过查询系统来收集有关数据库实例、集群环境和集群管理器的信息。您可通过参数向 db2haicu 调用、输入文件提供更多信息，或在运行时通过在 db2haicu 提示符处提供信息来实现上述目的。请参阅：第 66 页的『DB2 高可用性实例配置实用程序（db2haicu）』

- DB2 集群管理器 API 定义了一组功能，它们使数据库管理器能够向集群管理器通知配置更改。请参阅：第 100 页的『DB2 集群管理器 API』

IBM Tivoli System Automation for Multiplatforms (SA MP)

IBM Tivoli System Automation for Multiplatforms (SA MP) 为 AIX、Linux、Solaris SPARC 和 Windows 提供了高可用性及灾难恢复功能。

SA MP 与 AIX、Linux 以及 Solaris SPARC 操作系统上的 DB2 企业服务器版、DB2 工作组服务器版、DB2 Connect 企业服务器版和 DB2 Connect 应用程序服务器版集成在一起。在 Linux 操作系统上，它还与 DB2 Express-C 定期许可证 (FTL) 和 DB2 高可用性功能部件 Express 版集成到一起。在 Windows 操作系统上，SA MP 与所有这些 DB2 数据库产品和功能部件捆绑在一起，但是未与 DB2 安装程序集成在一起。

可以使用 SA MP 的此副本来管理 DB2 数据库系统的高可用性；如果不购买 SA MP 许可证的升级，就不能使用此副本来管理集群中的其他资源。

在 AIX 和 Linux 上，SA MP 是 IBM 数据服务器集群环境中的缺省集群管理器。

有关 SA MP 的更多信息，请参阅：<http://publib.boulder.ibm.com/tividd/td/IBMTivoliSystemAutomationforMultiplatforms2.2.html>。在以下 Web 站点上也提供了受支持的操作系统的列表：<http://www.ibm.com/software/tivoli/products/sys-auto-linux/platforms.html>。

使用 DB2 高可用性 (HA) 功能部件来自动配置集群

在集群环境中，某些数据库管理器实例配置和管理操作需要相关集群配置更改。每当您执行某些数据库管理器实例配置和管理操作时，DB2 高可用性 (HA) 功能部件将允许数据库管理器自动请求集群管理器配置更改。

开始前

要使数据库管理器能够执行数据库管理任务所需的集群配置，您必须为高可用性配置实例，方法是使用 db2haicu 来为该实例创建集群域。有关更多信息，请参阅：第 60 页的『使用 DB2 高可用性实例配置实用程序 (db2haicu) 来配置集群环境』。

过程

当执行下列数据库管理器实例配置和管理操作时，数据库管理器将为您自动执行相关的集群管理器配置：

- 使用 START DATABASE 或 db2start 来启动数据库
- 使用 STOP DATABASE 或 db2stop 来停止数据库
- 使用 CREATE DATABASE 来创建数据库
- 使用 CREATE TABLESPACE 来添加存储器
- 使用 ALTER TABLESPACE DROP 或 DROP TABLESPACE 来除去存储器
- 使用 ALTER DATABASE 来添加或除去存储路径
- 使用 DROP TABLESPACE 来删除数据库
- 使用 RESTORE DATABASE 或 db2Restore 来复原数据库
- 使用 SET TABLESPACE CONTAINERS 为重定向复原指定表空间容器

- 使用 ROLLFORWARD DATABASE 或 db2Rollforward 前滚数据库
- 使用 RECOVER DATABASE 或 db2Recover 来恢复数据库
- 使用 CREATE EVENT MONITOR 来创建事件监视器
- 使用 DROP EVENT MONITOR 来删除事件监视器
- 使用以下各项来创建和改变外部例程:
 - CREATE PROCEDURE
 - CREATE FUNCTION
 - CREATE FUNCTION
 - CREATE METHOD
 - ALTER PROCEDURE
 - ALTER FUNCTION
 - ALTER METHOD
- 使用以下各项来删除外部例程:
 - DROP PROCEDURE
 - DROP FUNCTION
 - DROP METHOD
- 使用 START HADR 来为数据库启动 DB2 高可用性灾难恢复 (HADR) 操作
- 使用 STOP HADR 来为数据库停止 HADR 操作
- 使用 TAKEOVER HADR 来促使 HADR 备用数据库作为 HADR 主数据库接管工作
- 设置数据库管理器配置参数 **DIAGPATH** 或 **SPM_LOG_PATH**
- 设置数据库配置参数 **NEWLOGPATH**、**OVERFLOWLOGPATH**、**MIRRORLOGPATH** 或 **FAILARCHPATH**
- 使用 db2idrop 来删除数据库管理器实例

结果

当数据库管理器为所列示的数据库管理任务协调集群配置更改时，您不必执行单独的集群管理器操作。

使用 DB2 高可用性实例配置实用程序 (db2haicu) 来配置集群环境

您可以在集群环境中使用 DB2 高可用性实例配置实用程序 (db2haicu) 来配置和管理数据库。当为 db2haicu 指定数据库管理器实例配置详细信息时，db2haicu 向集群管理软件通知所需的集群配置详细信息。

开始前

- 要在 SUSE Linux Enterprise Server (SLES) 11 上将 DB2 高可用性与 IBM Tivoli System Automation for Multiplatforms (SA MP) V3.1 配合使用，在运行 db2haicu 工具来创建 HA 环境之前，必须下载并安装 SA MP V3.1 修订包 4。要下载必需的修订包，请参阅：<http://www.ibm.com/software/tivoli/support/sys-auto-multi>

- 在使用 DB2 高可用性实例配置实用程序 (db2haicu) 之前, 必须执行一组任务。有关更多信息, 请参阅: 第 96 页的『DB2 高可用性实例配置实用程序 (db2haicu) 先决条件』。

限制

使用 DB2 高可用性实例配置实用程序 (db2haicu) 时存在一些限制。有关更多信息, 请参阅: 第 98 页的『DB2 高可用性实例配置实用程序 (db2haicu) 限制』。

关于此任务

可以采用交互方式运行 db2haicu 或使用 XML 输入文件:

交互方式

当通过运行 db2haicu 命令而未使用 **-f** 参数指定 XML 输入文件来调用 DB2 高可用性实例配置实用程序 (db2haicu) 时, 实用程序将以交互方式运行。在交互方式下, db2haicu 显示信息并询问您以获得基于文本格式的信息。有关更多信息, 请参阅: 第 68 页的『以交互方式运行 DB2 高可用性实例配置实用程序 (db2haicu)』

批处理方式和 XML 输入文件

您可以将 **-f <input-file-name>** 参数与 db2haicu 命令配合使用, 以运行 DB2 高可用性实例配置实用程序 (db2haicu), 同时 XML 输入文件指定配置详细信息。当必须多次执行配置任务, 例如具有多个要为高可用性而配置的数据库分区时, 运行 db2haicu 和 XML 输入文件是非常有用的。有关更多信息, 请参阅: 第 68 页的『使用 XML 输入文件来运行 DB2 高可用性实例配置实用程序 (db2haicu)』

过程

对每个数据库管理器实例执行下列步骤。

1. 创建新的集群域

当首次为数据库管理器实例运行 DB2 高可用性实例配置实用程序 (db2haicu) 时, db2haicu 会创建被称为集群域的集群模型。有关更多信息, 请参阅: 第 96 页的『使用 DB2 高可用性实例配置实用程序 (db2haicu) 来创建集群域』

2. 继续优化集群域配置以及管理和维护集群域

当您使用 db2haicu 来修改集群环境的集群域模型时, 数据库管理器将相关更改传播到数据库管理器实例和集群配置。有关更多信息, 请参阅: 第 97 页的『使用 DB2 高可用性实例配置实用程序 (db2haicu) 来维护集群域』

下一步任务

DB2 高可用性实例配置实用程序 (db2haicu) 没有单独的诊断日志。可以使用数据库管理器诊断日志、db2diag 日志文件和 db2pd 工具来调查与诊断 db2haicu 错误。有关更多信息, 请参阅: 第 98 页的『对 DB2 高可用性实例配置实用程序 (db2haicu) 进行故障诊断』

集群域

集群域是包含有关集群元素 (如数据库、安装点和故障转移策略) 的信息的模型。可以使用 DB2 高可用性实例配置实用程序 (db2haicu) 创建集群域。db2haicu 使用集群

域中的信息来启用配置和维护集群管理任务。此外，作为 DB2 高可用性（HA）功能部件的一部分，数据库管理器还使用集群域中的信息来执行自动化集群管理任务。

如果将集群元素添加至集群域，那么任何后续 db2haicu 配置操作或由 DB2 HA 功能部件中数据库管理器执行的任何自动化集群管理操作都会包含该元素。如果从集群域中除去集群元素，那么 db2haicu 操作或数据库管理器自动化集群管理操作将不再包含该元素。db2haicu 和数据库管理器只能与使用 db2haicu 创建的集群域中集群元素的集群管理器协同操作。

可以使用 db2haicu 来创建和配置下列集群域元素：

- 计算机或机器（在集群域上下文中，它们称为**集群域节点**）
- 网络接口卡或 NIC（在 db2haicu 中称为**网络接口、接口、网络适配器或适配器**）
- IP 地址
- 数据库，包括高可用性灾难恢复（HADR）主/备用数据库对
- 数据库分区
- 安装点和路径，包括在出现故障时对故障转移不重要的那些路径
- 故障转移策略
- 定额设备

集群管理软件：

集群管理软件使一组计算机能够执行的工作量达到最大。集群管理器平衡工作负载以减少瓶颈、监视集群元素的运行状况并在元素出现故障时管理故障转移。集群管理器还可以帮助系统管理员对集群中的元素执行管理任务（例如，将计算机中需要处理的工作负载重新路由到该计算机外）。

集群元素

要正常运行，集群管理器必须了解许多与集群元素相关的详细信息以及集群元素之间的关系。

以下是集群管理器必须了解的一些集群元素示例：

- 集群中的物理或虚拟计算机、机器或设备（在集群上下文中，它们称为**集群节点**）
- 连接集群节点的网络
- 将集群节点与网络连接在一起的网络接口卡
- 集群节点的 IP 地址
- 虚拟或服务 IP 地址

以下是集群管理器必须了解的一些关系示例：

- 安装了相同软件并且相互之间可以进行故障转移的**集群节点对**
- 具有相同属性并且相互之间可以进行故障转移的**网络**
- 当前与虚拟 IP 地址关联的**集群节点**

添加或修改集群元素

要让集群管理器了解集群元素和这些元素之间的关系，系统管理员必须向集群管理器注册这些元素。如果系统管理员更改集群元素，那么该管理员必须将所作的更改告知集群管理器。集群管理器具有可帮助完成这些任务的界面。

集群管理是一项富有挑战性的任务，因为可能的集群元素数量庞大且种类繁多。管理员必须精通集群节点的硬件和操作系统、网络协议和配置，以及集群节点上安装的软件（如数据库软件）。向集群管理软件注册集群元素或者在系统更改后更新集群管理器是一项复杂且耗时的工作。

使用 db2haicu 添加或修改集群元素

在 DB2 数据库解决方案中，可以使用 DB2 高可用性实例配置实用程序 (db2haicu) 向集群管理器注册集群元素，并且在对集群进行管理更改后更新集群管理器。使用 db2haicu 可简化这些任务，因为在您了解 db2haicu 用于封装集群元素和这些元素之间的关系模型之后，您不必精通硬件、操作系统和集群管理器界面的特征即可执行这些任务。

资源和资源组:

资源是指任何已向集群管理器注册的集群元素，如集群节点、数据库、安装点或网络接口卡。如果某个元素未向集群管理器注册，那么集群管理器不会知道该元素并且不会将它包括在集群管理操作中。资源组是资源的逻辑集合。资源组是一种功能非常强大的结构，这是因为可以在资源组上定义关系和约束，而这些关系和约束能够简化对这些组中的资源执行的复杂管理任务。

当集群管理器将资源收集成组时，它可以统一对这些资源进行操作。例如，如果两个数据库 database-1 和 database-2 属于资源组 resource-group-A，那么当集群管理器对 resource-group-A 执行启动操作时，database-1 和 database-2 都会被这个集群管理器操作启动。

限制

- 资源组不能包含等值，等值也不能包含资源组（等值是指一组资源，它们彼此提供相同的功能，并且相互之间可以进行故障转移。）
- 一个资源只能位于一个资源组中
- 一个资源不能同时位于资源组和等值中
- 资源组可以包含其他资源组，但最大嵌套级别为 50
- 一个资源组中可以收集的最大资源数为 100

定额设备:

当集群管理器的一般决策过程不能作出明确选择时，定额设备可帮助集群管理器作出集群管理决策。当集群管理器必须在多个潜在操作之间进行选择时，它计算每个潜在操作各有多少个集群域节点支持，然后选择大多数集群域节点支持的操作。如果正好有相同数目的集群域节点支持多个选项，那么集群管理器将引用定额设备来作出选择。

db2haicu 支持下表中列示的定额设备。

表 2. db2haicu 支持的定额设备类型

定额设备	描述
网络	网络定额设备是每个集群域节点在任何时候都可连接至的 IP 地址。

集群域中的网络:

要配置与网络相关的集群域元素，可以使用 DB2 高可用性实例配置实用程序（db2haicu）在集群域中添加物理网络。物理网络由以下几部分组成：网络接口卡、IP 地址和子网掩码。

网络接口卡

网络接口卡（NIC）是使计算机（也称为集群节点）与网络连接在一起的硬件。NIC 有时称为接口、网络适配器或适配器。使用 db2haicu 在集群域中添加物理网络时，应指定至少一个 NIC，包括该 NIC 所属的计算机的主机名、该主机上 NIC 的名称以及 NIC 的 IP 地址。

IP 地址

因特网协议地址（IP 地址）是网络上的唯一地址。在 IP V4 中，IP 地址为 32 位，并且通常用点分十进制表示法表示如下：129.30.180.16。IP 地址由网络部分和主机部分组成。

db2haicu 不支持 IP V6。

子网掩码

使用子网掩码可以将一个网络分为多个逻辑子网。子网掩码是一种机制，用于将 IP 地址的主机部分中的某些位移至 IP 地址的网络部分。使用 db2haicu 在集群域中添加 IP 地址时，有时需要指定该 IP 地址的子网掩码。例如，使用 db2haicu 添加 NIC 时，必须指定该 NIC 的 IP 地址的子网掩码。

网络等值

等值是指一组资源，它们彼此提供相同的功能，并且相互之间可以进行故障转移。使用 db2haicu 创建网络时，该网络中的 NIC 之间可以进行故障转移。这种网络也称为网络等值。

网络协议

使用 db2haicu 在集群域中添加网络时，必须指定要使用的网络协议的类型。当前仅支持 TCP/IP 网络协议。

集群域中的故障转移策略:

故障转移策略指定在网络接口卡或数据库服务器等集群元素出现故障时集群管理器应如何响应。通常，集群管理器会将工作负载从故障元素转移到备用元素，先前已向集群管理器将该备用元素标识为故障元素的适当替换元素。这种将工作负载从故障元素转移到另一个元素的操作称为故障转移。

循环故障转移策略

使用循环故障转移策略时，如果集群中的一台计算机（也称为集群域节点或简单地称为节点）出现故障，那么在集群域中的任何其他节点上，数据库管理器将重新开始出现故障的集群域节点中的工作。

相互故障转移策略

要配置相互故障转移策略，应将集群域中的一个计算机对（也称为集群域节点或简单地称为节点）作为系统对关联。如果此计算机对中的一个节点出现故障，那么故障节点上的数据库分区将故障转移到该计算机对中的另一个节点。仅当具有多个数据库分区时，相互故障转移才可用。

N 加 M 故障转移策略

使用 N 加 M 故障转移策略时，如果集群域中的一台计算机（也称为集群域节点或简单地称为节点）出现故障，那么故障节点上的数据库分区将故障转移到集群域中的任何其他节点。仅当有多个数据库分区时，N 加 M 故障转移才可用。

本地重新启动故障转移策略

使用本地重新启动故障转移策略时，如果集群域中的一台计算机（也称为集群域节点或简单地称为节点）出现故障，那么数据库管理器将在出现故障的那个节点上的适当位置（或本地）重新启动数据库。

HADR 故障转移策略

配置 HADR 故障转移策略时，您正在启用 DB2 高可用性灾难恢复（HADR）功能部件来管理故障转移。如果 HADR 主数据库出现故障，那么数据库管理器会将出现故障的数据库中的工作负载移至 HADR 备用数据库。

定制故障转移策略

配置定制故障转移策略时，您将在集群域中创建一系列计算机（也称为集群域节点或简单地称为节点），数据库管理器可以故障转移到这些计算机上。如果集群域中的一个节点出现故障，那么数据库管理器会将故障节点中的工作负载移至您指定的列表中的一个节点。

集群域中的安装点:

安装文件系统后，可以使用 DB2 高可用性实例配置实用程序（db2haicu）将该安装点添加至集群域。

安装点

在 UNIX、Linux 和 AIX 操作系统上，安装文件系统意味着使该文件系统可用于操作系统。在安装操作期间，操作系统将执行一些任务（例如，读取索引或导航数据结构），并使一个目录路径与安装的文件系统关联。可用于访问安装的文件系统的关联目录路径称为安装点。

非重要安装点或路径

集群中可能存在出现故障时不需要进行故障转移的安装点或路径。可以使用 db2haicu 将这些非重要安装点或路径的列表添加至集群域。集群管理器不会将该非重要列表中的安装点或路径包括在故障转移操作中。

例如，请考虑以下情况：您在集群中的计算机 node1 上的 /mnt/driveA 中安装了一个硬盘驱动器。如果您认为让 /mnt/driveA 可用是至关重要的，那么在 node1 出现故障时集群管理器会进行故障转移以保持 /mnt/driveA 可用。但是，如果您认为可以接受在

node1 出现故障时 /mnt/driveA 不可用，那么您可以通过将 /mnt/driveA 添加至非重要路径列表来向集群管理器指示 /mnt/driveA 对于故障转移不是重要的。如果 /mnt/driveA 标识为对于故障转移不是重要的，那么在 node1 出现故障时该驱动器可能不可用。

DB2 高可用性实例配置实用程序 (db2haicu)

DB2 高可用性实例配置实用程序 (db2haicu) 是基于文本的实用程序，您可以使用它在集群环境中配置与管理高可用性数据库。db2haicu 通过查询系统来收集有关数据库实例、集群环境和集群管理器的信息。您可通过参数向 db2haicu 调用、输入文件提供更多信息，或在运行时通过在 db2haicu 提示符处提供信息来实现上述目的。

语法

```
db2haicu [ -f <XML-input-file-name> ]
          [ -disable ]
          [ -delete [ dbpartitionnum <db-partition-list> |
                    hadrdb <database-name> ] ]
```

参数

传递到 db2haicu 命令的参数区分大小写，并且它们必须采用小写形式。

-f <XML-input-file-name>

可以使用 **-f** 参数来指定 XML 输入文件 <XML-input-file-name> 中的集群域详细信息。有关更多信息，请参阅：第 68 页的『使用 XML 输入文件来运行 DB2 高可用性实例配置实用程序 (db2haicu)』。

-disable

一旦使用 db2haicu 来为数据库管理器实例创建集群域，该实例被视为已配置用于获取高可用性。当配置数据库管理器实例以获取高可用性时，那么每当您执行需要相关集群配置更改的某些数据库管理器管理操作时，数据库管理器将向集群管理器通知那些集群配置更改。当数据库管理器将这些集群管理任务与集群管理器进行协调时，您不必为那些管理任务执行单独的集群管理器操作。数据库管理器与集群管理器之间的集成是 DB2 高可用性 (HA) 功能部件的功能。

可以使用 **-disable** 参数来使数据库管理器实例停止以被配置用于获取高可用性。如果不再配置数据库管理器实例以获取高可用性，那么在您执行需要相关集群配置更改的任何数据库管理器管理操作时，数据库管理器将不与集群管理器进行协调。

要重新配置数据库管理器实例以获取高可用性，您可以再次运行 db2haicu。

-delete

可以使用 **-delete** 参数来删除当前数据库管理器实例的资源组。

如果不使用 **dbpartitionnum** 参数或 **hadrdb** 参数，那么 db2haicu 将除去与当前数据库管理器实例相关联的所有资源组。

dbpartitionnum <db-partition-list>

可以使用 **dbpartitionnum** 参数来删除与 <db-partition-list> 中列示的数据库分区相关联的资源组。<db-partition-list> 是用逗号分隔的编号列表，它标识了数据库分区。

hadrdb <database-name>

可以使用 **hadrdb** 参数来删除与名为 <database-name> 的 DB2 高可用性灾难恢复 (HADR) 数据库相关联的资源组。

如果在 db2haicu 除去资源组之后集群域中没有剩余资源组，那么 db2haicu 也将除去集群域。

运行带 **-delete** 参数的 db2haicu 将导致当前数据库管理器实例停止配置用于获取高可用性。如果不再配置数据库管理器实例以获取高可用性，那么在您执行需要相关集群配置更改的任何数据库管理器管理操作时，数据库管理器将不与集群管理器进行协调。

要重新配置数据库管理器实例以获取高可用性，您可以再次运行 db2haicu。

DB2 高可用性实例配置实用程序 (db2haicu) 启动方式:

首次为给定数据库管理器实例运行 DB2 高可用性实例配置实用程序 (db2haicu) 时，db2haicu 在启动方式下工作。

当运行 db2haicu 时，db2haicu 检查数据库管理器实例与系统配置，并且搜索现有集群域。集群域是包含有关集群元素（如数据库、安装点和故障转移策略）的信息的模型。可以使用 DB2 高可用性实例配置实用程序 (db2haicu) 创建集群域。db2haicu 使用集群域中的信息来启用配置和维护集群管理任务。此外，作为 DB2 高可用性 (HA) 功能部件的一部分，数据库管理器还使用集群域中的信息来执行自动化集群管理任务。

为给定数据库管理器实例运行 db2haicu 且没有为该实例创建和配置集群域时，db2haicu 将立即开始创建和配置新集群域。db2haicu 通过提示您输入诸如新集群域名称与当前机器主机名之类的信息来创建新的集群域。

如果创建集群域，但未完成配置集群域任务，那么下次运行 db2haicu 时，db2haicu 将继续执行配置集群域任务。

在为数据库管理器实例创建和配置集群域后，db2haicu 将在维护方式下运行。

DB2 高可用性实例配置实用程序 (db2haicu) 维护方式:

当运行 DB2 高可用性实例配置实用程序 (db2haicu) 且存在为当前数据库管理器实例创建的集群域时，db2haicu 在维护方式下工作。

当 db2haicu 在维护方式下运行时，db2haicu 向您展示可执行的配置和管理任务的列表。

db2haicu 维护任务包括将诸如数据库或集群节点之类的集群元素添加到集群域，并且从集群域中除去元素。db2haicu 维护任务还包括修改集群域元素的详细信息，例如数据库管理器实例的故障转移策略。

当在维护方式下运行 db2haicu 时，db2haicu 向您展示可以对集群域执行的操作的列表:

- 添加或除去集群节点（主机名标识的机器）
- 添加或除去网络接口（网络接口卡）
- 添加或除去数据库分区（仅限于分区数据库环境）
- 添加或除去 DB2 高可用性灾难恢复 (HADR) 数据库
- 添加或除去高可用性数据库
- 添加或除去安装点
- 添加或除去 IP 地址
- 添加或除去非重要路径

- 移动数据库分区和 HADR 数据库以进行预定维护
- 更改当前实例的故障转移策略
- 为集群域创建新的定额设备
- 破坏集群域

以交互方式运行 **DB2 高可用性实例配置实用程序 (db2haicu)** :

当通过运行 `db2haicu` 命令而未使用 `-f` 参数指定 XML 输入文件来调用 DB2 高可用性实例配置实用程序 (db2haicu) 时, 实用程序将以交互方式运行。在交互方式下, `db2haicu` 显示信息并询问您以获得基于文本格式的信息。

开始前

- 要在 SUSE Linux Enterprise Server (SLES) 11 上将 DB2 高可用性与 IBM Tivoli System Automation for Multiplatforms (SA MP) V3.1 配合使用, 在运行 `db2haicu` 工具来创建 HA 环境之前, 必须下载并安装 SA MP V3.1 修订包 4。要下载必需的修订包, 请参阅: <http://www.ibm.com/software/tivoli/support/sys-auto-multi>
- 在使用 DB2 高可用性实例配置实用程序 (db2haicu) 之前, 必须执行一组任务。有关更多信息, 请参阅: 第 96 页的『DB2 高可用性实例配置实用程序 (db2haicu) 先决条件』。

关于此任务

以交互方式运行 `db2haicu` 时, 您将看到信息和问题以文本格式出现在屏幕上。您可以在屏幕底部的提示符处输入 `db2haicu` 请求的信息。

过程

要以交互方式运行 `db2haicu`, 调用 `db2haicu` 命令 (不带 `-f <input-file-name>`)。

下一步任务

DB2 高可用性实例配置实用程序 (db2haicu) 没有单独的诊断日志。可以使用数据库管理器诊断日志、`db2diag` 日志文件和 `db2pd` 工具来调查与诊断 `db2haicu` 错误。有关更多信息, 请参阅: 第 98 页的『对 DB2 高可用性实例配置实用程序 (db2haicu) 进行故障诊断』

使用 **XML 输入文件来运行 DB2 高可用性实例配置实用程序 (db2haicu)** :

您可以将 `-f <input-file-name>` 参数与 `db2haicu` 命令配合使用, 以运行 DB2 高可用性实例配置实用程序 (db2haicu), 同时 XML 输入文件指定配置详细信息。当必须多次执行配置任务, 例如具有多个要为高可用性而配置的数据库分区时, 运行 `db2haicu` 和 XML 输入文件是非常有用的。

开始前

- 要在 SUSE Linux Enterprise Server (SLES) 11 上将 DB2 高可用性与 IBM Tivoli System Automation for Multiplatforms (SA MP) V3.1 配合使用, 在运行 `db2haicu` 工具来创建 HA 环境之前, 必须下载并安装 SA MP V3.1 修订包 4。要下载必需的修订包, 请参阅: <http://www.ibm.com/software/tivoli/support/sys-auto-multi>

- 在使用 DB2 高可用性实例配置实用程序 (db2haicu) 之前, 必须执行一组任务。有关更多信息, 请参阅: 第 96 页的『DB2 高可用性实例配置实用程序 (db2haicu) 先决条件』。

关于此任务

在 sqllib 目录的 samples 子目录中存在一组您可以修改且与 db2haicu 配合使用的样本 XML 输入文件, 可用于配置集群环境。有关更多信息, 请参阅: 第 86 页的『DB2 高可用性实例配置实用程序 (db2haicu) 的样本 XML 输入文件』

过程

1. 创建 XML 输入文件。
2. 调用 db2haicu (带 `-f <input-file-name>`)。

要使用 db2haicu 和您创建的名为 db2haicu-input.xml 的输入文件来为当前数据库管理器实例配置集群环境, 请使用下列命令:

```
db2haicu -f db2haicu-input.xml
```

下一步任务

DB2 高可用性实例配置实用程序 (db2haicu) 没有单独的诊断日志。可以使用数据库管理器诊断日志、db2diag 日志文件和 db2pd 工具来调查与诊断 db2haicu 错误。有关更多信息, 请参阅: 第 98 页的『对 DB2 高可用性实例配置实用程序 (db2haicu) 进行故障诊断』

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 XML 模式定义:

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 XML 模式定义 (XSD) 用于定义可以在 db2haicu XML 输入文件中指定的集群域对象。此 db2haicu XSD 位于 sqllib/samples/ha/xml 目录中的文件 db2ha.xsd 中。

DB2ClusterType

db2haicu XML 模式定义 (XSD) 的根元素是 DB2Cluster, 其类型为 DB2ClusterType。db2haicu XML 输入文件必须以 DB2Cluster 元素开头。

『XML 模式定义』

第 70 页的『子元素』

第 71 页的『属性』

第 71 页的『使用说明』

XML 模式定义

```
<xs:complexType name='DB2ClusterType'>
  <xs:sequence>
    <xs:element name='DB2ClusterTemplate'
      type='DB2ClusterTemplateType'
      minOccurs='0'
      maxOccurs='unbounded' />
    <xs:element name='ClusterDomain'
      type='ClusterDomainType'
      maxOccurs='unbounded' />
    <xs:element name='FailoverPolicy'
      type='FailoverPolicyType'
      minOccurs='0' />
  </xs:sequence>
</xs:complexType>
```

```

<xs:element name='DB2PartitionSet'
  type='DB2PartitionSetType'
  minOccurs='0'
  maxOccurs='unbounded' />
<xs:element name='HADRDBSet'
  type='HADRDBType'
  minOccurs='0'
  maxOccurs='unbounded' />
<xs:element name='HADBSet'
  type='HADBType'
  minOccurs='0'
  maxOccurs='unbounded' />
</xs:sequence>
<xs:attribute name='clusterManagerName' type='xs:string' use='optional' />
</xs:complexType>

```

子元素

DB2ClusterTemplate

类型: DB2ClusterTemplateType

使用说明:

不要在 db2haicu XML 输入文件中包括 DB2ClusterTemplateType 元素。当前保留 DB2ClusterTemplateType 元素以供将来使用。

ClusterDomain

类型: ClusterDomainType

ClusterDomainType 元素包含关于下列内容的规范: 集群域中的机器或计算机 (也称为集群域节点)、网络等值 (相互之间可以进行故障转移的多组网络), 以及定额设备 (决定性机制)。

出现规则:

必须在 DB2ClusterType 元素中包括一个或多个 ClusterDomain 元素。

FailoverPolicy

类型: FailoverPolicyType

FailoverPolicyType 元素指定集群管理器应该对集群域使用的故障转移策略。

出现规则:

可以在 DB2ClusterType 元素中包括零个或一个 FailoverPolicy 元素。

DB2PartitionSet

类型: DB2PartitionSetType

DB2PartitionSetType 元素包含关于数据库分区的信息。DB2PartitionSetType 元素仅适用于分区数据库环境。

出现规则:

根据 db2haicu XML 模式定义, 可以在 DB2ClusterType 元素中包括零个或多个 DB2PartitionSet 元素。

HADRDBSet

类型: HADRDBType

HADRDBType 元素包含一列高可用性灾难恢复 (HADR) 主/备用数据库对。

出现规则:

根据 db2haicu XML 模式定义, 可以在 DB2ClusterType 元素中包括零个或多个 HADRDBSet 元素。

使用说明:

- 在分区数据库环境中不能包括 HADRDBSet。
- 如果包括 HADRDBSet, 那么必须在 FailoverPolicy 元素中指定故障转移策略 HADRFailover。

HADBSet

类型: HADBType

HADBType 元素包含要包括在集群域中并使其具有高可用性的数据库的列表。

出现规则:

根据 db2haicu XML 模式定义, 可以在 DB2ClusterType 元素中包括零个或多个 HADBSet 元素。

属性

clusterManagerName (可选)

clusterManagerName 属性指定集群管理器。

下表中指定了此属性的有效值:

表 3. clusterManager 属性的有效值

clusterManagerName 值	集群管理器产品
TSA	IBM Tivoli System Automation for Multiplatforms (SA MP)

使用说明

在单一分区数据库环境中, 您通常只为每个数据库管理器实例创建单个集群域。

以下是多分区数据库环境的一种可能配置:

- 将 FailoverPolicy 元素设置为 Mutual
- 在 DB2PartitionSet 的 DB2Partition 子元素中, 使用 MutualPair 元素指定位于一个集群域中的两个集群域节点

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 ClusterDomainType XML 模式定义:

ClusterDomainType 元素包含关于下列内容的规范: 集群域中的机器或计算机 (也称为集群域节点)、网络等值 (相互之间可以进行故障转移的多组网络), 以及定额设备 (决定性机制)。

第 72 页的『超元素』

第 72 页的『XML 模式定义』

第 72 页的『子元素』

第 73 页的『属性』

超元素

下列类型的元素包含 ClusterDomainType 子元素:

- DB2ClusterType

XML 模式定义

```
<xs:complexType name='ClusterDomainType'>
  <xs:sequence>
    <xs:element name='Quorum'
      type='QuorumType'
      minOccurs='0' />
    <xs:element name='PhysicalNetwork'
      type='PhysicalNetworkType'
      minOccurs='0'
      maxOccurs='unbounded' />
    <xs:element name='ClusterNode'
      type='ClusterNodeType'
      maxOccurs='unbounded' />
  </xs:sequence>
  <xs:attribute name='domainName' type='xs:string' use='required' />
</xs:complexType>
```

子元素

Quorum

类型: QuorumType

QuorumType 元素为集群域指定定额设备。

出现规则:

可以在 ClusterDomainType 元素中包括零个或一个 Quorum 元素。

PhysicalNetwork

类型: PhysicalNetworkType

PhysicalNetworkType 元素包含相互之间可以进行故障转移的网络接口卡。这种网络也称为网络等值。

出现规则:

可以在 ClusterDomainType 元素中包括零个或多个 PhysicalNetwork 元素。

ClusterNode

类型: ClusterNodeType

ClusterNodeType 元素包含关于集群中的特定计算机或机器（也称为集群域节点）的信息。

出现规则:

必须在 ClusterDomainType 元素中至少指定一个 ClusterNode 元素。

使用说明

IBM Tivoli System Automation for Multiplatforms (SA MP) 最多支持 32 个集群域节点。如果集群管理器是 SA MP, 那么最多可以在 ClusterDomainType 元素中包括 32 个 ClusterNode 元素。

属性

domainName (必需)

必须为 ClusterDomainType 元素指定唯一名称。

如果要使用可靠的可缩放集群技术 (RSCT) 来管理集群, 那么下列限制适用于 domainName:

- domainName 只能包含字符 A-Z、a-z、数字 0-9、句号 (.) 和下划线 (_)
- domainName 不能是“IW”

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 QuorumType XML 模式定义:

QuorumType 元素为集群域指定定额设备。

- 『超元素』
- 『XML 模式定义』
- 『子元素』
- 『属性』

超元素

下列类型的元素包含 QuorumType 子元素:

- ClusterDomainType

XML 模式定义

```
<xs:complexType name='QuorumType'>
  <xs:attribute name='quorumDeviceProtocol'
type='QuorumDeviceProtocolType' use='required' />
<xs:attribute name='quorumDeviceName'
              type='xs:string'
              use='required' />
</xs:complexType>
```

子元素

无。

属性

quorumDeviceProtocol (必需)

quorumDeviceProtocol 指定要使用的定额类型。

定额设备可帮助集群管理器作出集群管理决策。当集群管理器必须在多个潜在操作之间进行选择时, 它计算每个潜在操作各有多少个集群域节点支持, 然后选择大多数集群域节点支持的操作。如果正好有相同数目的集群域节点支持多个选项, 那么集群管理器将引用定额设备来作出选择。

quorumDeviceProtocol 属性的类型为 QuorumDeviceProtocolType。

以下是 QuorumDeviceProtocolType 的 XML 模式定义:

```
<xs:simpleType name='QuorumDeviceProtocolType'>
  <xs:restriction base='xs:string'>
    <xs:enumeration value='disk' />
    <xs:enumeration value='scsi' />
    <xs:enumeration value='network' />
  </xs:restriction>
</xs:simpleType>
```

```

        <xs:enumeration value='eckd' />
        <xs:enumeration value='mns' />
    </xs:restriction>
</xs:simpleType>

```

下表中指定了当前支持的此属性值:

表 4. *quorumDeviceProtocol* 属性的有效值

quorumDeviceProtocol 值	含义
网络	网络定额设备是每个集群域节点在任何时候都可连接至的 IP 地址。

quorumDeviceName (必需)

quorumDeviceName 的值取决于在 quorumDeviceProtocol 中指定的定额设备的类型。

下表中指定了此属性的有效值:

表 5. *quorumDeviceName* 属性的有效值

quorumDeviceProtocol 的值	quorumDeviceName 的有效值
网络	包含格式正确的 IP 地址的字符串。例如: 12.126.4.5 要使您指定的 IP 地址成为有效的网络定额设备, 每个集群域节点都必须能够访问此 IP 地址 (例如, 使用 ping 实用程序)。

DB2 高可用性实例配置实用程序 (*db2haicu*) 输入文件的 *PhysicalNetworkType* XML 模式定义:

PhysicalNetworkType 元素包含相互之间可以进行故障转移的网络接口卡。这种网络也称为网络等值。

『超元素』

『XML 模式定义』

第 75 页的『子元素』

第 75 页的『属性』

超元素

下列类型的元素包含 *PhysicalNetworkType* 子元素:

- *ClusterDomainType*

XML 模式定义

```

<xs:complexType name='PhysicalNetworkType'>
  <xs:sequence>
    <xs:element name='Interface'
      type='InterfaceType'
      minOccurs='1'
      maxOccurs='unbounded' />
    <xs:element name='LogicalSubnet'
      type='IPAddressType'
      minOccurs='0'
      maxOccurs='unbounded' />
  </xs:sequence>
</xs:complexType>

```

```

</xs:sequence>
<xs:attribute name='physicalNetworkName'
              type='xs:string'
              use='required' />
<xs:attribute name='physicalNetworkProtocol' type='PhysicalNetworkProtocolType'
              use='required' />
</xs:complexType>

```

子元素

Interface

类型: InterfaceType

InterfaceType 元素由 IP 地址、网络中的计算机或机器（也称为集群域节点）的名称以及该集群域节点上的网络接口卡（NIC）的名称组成。

出现规则:

必须在 PhysicalNetworkType 元素中指定一个或多个 Interface 元素。

LogicalSubnet

类型: IPAddressType

IPAddressType 元素包含一个 IP 地址的所有详细信息，例如，基地址、子网掩码和该 IP 地址所属的网络的名称。

出现规则:

可以在 PhysicalNetworkType 元素中包括一个或多个 LogicalSubnet 元素。

属性

physicalNetworkName (必需)

必须为每个 PhysicalNetworkType 元素指定唯一的 physicalNetworkName。

physicalNetworkProtocol (必需)

physicalNetworkProtocol 属性的类型为 PhysicalNetworkProtocolType。

以下是 PhysicalNetworkProtocolType 元素的 XML 模式定义:

```

<xs:simpleType name='PhysicalNetworkProtocolType'>
  <xs:restriction base='xs:string'>
    <xs:enumeration value='ip' />
    <xs:enumeration value='rs232' />
    <xs:enumeration value='scsi' />
    <xs:enumeration value='ssa' />
    <xs:enumeration value='disk' />
  </xs:restriction>
</xs:simpleType>

```

下表中指定了当前支持的此属性值:

表 6. physicalNetworkProtocol 属性的有效值

physicalNetworkProtocol 值	含义
ip	TCP/IP 协议

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 InterfaceType XML 模式定义:

InterfaceType 元素由 IP 地址、网络中的计算机或机器（也称为集群域节点）的名称以及该集群域节点上的网络接口卡（NIC）的名称组成。

- 『超元素』
- 『XML 模式定义』
- 『子元素』
- 『属性』

超元素

下列类型的元素具有 InterfaceType 子元素:

- PhysicalNetworkType

XML 模式定义

```
<xs:complexType name='InterfaceType'>
  <xs:sequence>
    <xs:element name='IPAddress' type='IPAddressType' />
  </xs:sequence>
  <xs:attribute name='interfaceName' type='xs:string' use='required' />
  <xs:attribute name='clusterNodeName' type='xs:string' use='required' />
</xs:complexType>
```

子元素

IPAddress

类型: IPAddressType

IPAddressType 元素包含一个 IP 地址的所有详细信息，例如，基地址、子网掩码和该 IP 地址所属的网络的名称。

出现规则:

必须在 InterfaceType 元素中正好指定一个 IPAddress。

属性

interfaceName (必需)

必须在 interfaceName 属性中指定 NIC 的名称。在 interfaceName 中指定的 NIC 必须存在于您在 clusterNodeName 属性中指定的集群域节点上。

clusterNodeName (必需)

必须指定位于您在 IPAddress 元素中指定的 IP 地址处的集群域节点的名称。

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 IPAddressType XML 模式元素:

IPAddressType 元素包含一个 IP 地址的所有详细信息，例如，基地址、子网掩码和该 IP 地址所属的网络的名称。

- 第 77 页的『超元素』
- 第 77 页的『XML 模式定义』
- 第 77 页的『子元素』
- 第 77 页的『属性』

超元素

下列类型的元素具有 IPAddressType 子元素:

- PhysicalNetworkType
- InterfaceType
- DB2PartitionType

XML 模式定义

```
<xs:complexType name='IPAddressType'>  
  <xs:attribute name='baseAddress' type='xs:string' use='required' />  
  <xs:attribute name='subnetMask' type='xs:string' use='required' />  
  <xs:attribute name='networkName' type='xs:string' use='required' />  
</xs:complexType>
```

子元素

无。

属性

baseAddress (必需)

必须使用具有以下有效 IP 地址格式的字符串指定基本 IP 地址: 范围在 0 到 255 之间的四组数字, 中间用句点分隔。例如:

162.148.31.101

subnetMask (必需)

必须使用具有有效 IP 地址格式的字符串指定基本 IP 地址。

networkName (必需)

此处对 networkName 指定的值必须与对包含此 IPAddress 元素的 PhysicalNetworkType 元素的 physicalNetworkName 属性所指定的值相同。

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 ClusterNodeType XML 模式定义:

ClusterNodeType 元素包含关于集群中的特定计算机或机器 (也称为集群域节点) 的信息。

『超元素』

『XML 模式定义』

第 78 页的『子元素』

第 78 页的『属性』

超元素

下列类型的元素具有 ClusterNodeType 元素:

- ClusterDomainType

XML 模式定义

```
<xs:complexType name='ClusterNodeType'>  
  <xs:attribute name='clusterNodeName' type='xs:string' use='required' />  
</xs:complexType>
```

子元素

无。

属性

clusterNodeName (必需)

必须指定集群域节点的名称。

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 *FailoverPolicyType* XML 模式定义:

FailoverPolicyType 元素指定集群管理器应该对集群域使用的故障转移策略。

『超元素』

『XML 模式定义』

『子元素』

『可能的值』

超元素

下列类型的元素包含 *InterfaceType* 子元素:

- *DB2ClusterType*

XML 模式定义

```
<xs:complexType name='FailoverPolicyType'>
  <xs:choice>
    <xs:element name='RoundRobin'
      type='xs:string'
      minOccurs='0' />
    <xs:element name='Mutual'
      type='xs:string'
      minOccurs='0'
      maxOccurs='unbounded' />
    <xs:element name='NplusM'
      type='xs:string'
      minOccurs='0'
      maxOccurs='unbounded' />
    <xs:element name='LocalRestart'
      type='xs:string'
      fixed='' />
    <xs:element name='HADRFailover'
      type='xs:string'
      fixed='' />
    <xs:element name='Custom'
      type='xs:string'
      minOccurs='0' />
  </xs:choice>
</xs:complexType>
```

子元素

无。

可能的值

选择下列其中一个选项，以指定在集群域中的任何地方出现故障时集群管理器应使用的故障转移策略的类型。

故障转移策略指定在网络接口卡或数据库服务器等集群元素出现故障时集群管理器应如何响应。通常，集群管理器会将工作负载从故障元素转移到备用元素，先前已向集群管理器将该备用元素标识为故障元素的适当替换元素。这种将工作负载从故障元素转移到另一个元素的操作称为故障转移。

RoundRobin

使用循环故障转移策略时，如果集群中的一台计算机（也称为集群域节点或简单地称为节点）出现故障，那么在集群域中的任何其他节点上，数据库管理器将重新开始出现故障的集群域节点中的工作。

Mutual

要配置相互故障转移策略，应将集群域中的一个计算机对（也称为集群域节点或简单地称为节点）作为系统对关联。如果此计算机对中的一个节点出现故障，那么故障节点上的数据库分区将故障转移到该计算机对中的另一个节点。仅当具有多个数据库分区时，相互故障转移才可用。

NPlusM

使用 N 加 M 故障转移策略时，如果集群中的一台计算机（也称为集群域节点或简单地称为节点）出现故障，那么故障节点上的数据库分区将故障转移到集群域中的任何其他节点。仅当有多个数据库分区时， N 加 M 故障转移才可用。

LocalRestart

使用本地重新启动故障转移策略时，如果集群中的一台计算机（也称为集群域节点或简单地称为节点）出现故障，那么数据库管理器将在出现故障的那个节点上的适当位置（或本地）重新启动数据库。

HADRFailover

配置 *HADR* 故障转移策略时，您正在启用 DB2 高可用性灾难恢复（HADR）功能部件来管理故障转移。如果 HADR 主数据库出现故障，那么数据库管理器会将出现故障的数据库中的工作负载移至 HADR 备用数据库。

Custom

配置定制故障转移策略时，您将在集群域中创建一系列计算机（也称为集群域节点或简单地称为节点），数据库管理器可以故障转移到这些计算机上。如果集群域中的一个节点出现故障，那么数据库管理器会将故障节点中的工作负载移至您指定的列表中的一个节点。

DB2 高可用性实例配置实用程序 (*db2haicu*) 输入文件的 *DB2PartitionSetType XML* 模式定义:

DB2PartitionSetType 元素包含关于数据库分区的信息。*DB2PartitionSetType* 元素仅适用于分区数据库环境。

『超元素』

第 80 页的『XML 模式定义』

第 80 页的『子元素』

第 80 页的『属性』

超元素

InterfaceType 是下列各项的子元素:

- *PhysicalNetworkType*

XML 模式定义

```
<xs:complexType name='DB2PartitionSetType'>
  <xs:sequence>
    <xs:element name='DB2Partition'
      type='DB2PartitionType'
      maxOccurs='unbounded' />
  </xs:sequence>
</xs:complexType>
```

子元素

DB2Partition

类型: DB2PartitionType

DB2PartitionType 元素指定数据库分区（包括该数据库分区所属的 DB2 数据库管理器实例）和数据库分区号。

出现规则:

必须在 DB2PartitionSetType 元素中指定一个或多个 DB2Partition 元素。

属性

无。

DB2 高可用性实例配置实用程序 (*db2haicu*) 输入文件的 *DB2PartitionType* XML 模式元素:

DB2PartitionType 元素指定数据库分区（包括该数据库分区所属的 DB2 数据库管理器实例）和数据库分区号。

『超元素』

『XML 模式定义』

第 81 页的『子元素』

第 81 页的『属性』

超元素

InterfaceType 是下列各项的子元素:

- DB2PartitionSetType

XML 模式定义

```
<xs:complexType name='DB2PartitionType'>
  <xs:sequence>
    <xs:element name='VirtualIPAddress'
      type='IPAddressType'
      minOccurs='0'
      maxOccurs='unbounded' />
    <xs:element name='Mount'
      type='MountType'
      minOccurs='0'
      maxOccurs='unbounded' />
    <xs:element name='HADRDB'
      type='HADRDBType'
      minOccurs='0'
      maxOccurs='unbounded' />
    <xs:element name='MutualPair'
      type='MutualPolicyType' />
  </xs:sequence>
</xs:complexType>
```

```

        minOccurs='0'
        maxOccurs='1' />
<xs:element name='NPlusMNode'
            type='NPlusMPolicyType'
            minOccurs='0'
            maxOccurs='unbounded' />
</xs:sequence>
<xs:attribute name='instanceName' type='xs:string' use='required' />
<xs:attribute name='dbpartitionnum' type='xs:integer' use='required' />
</xs:complexType>

```

子元素

VirtualIPAddress

类型: IPAddressType

IPAddressType 元素包含一个 IP 地址的所有详细信息, 例如, 基地址、子网掩码和该 IP 地址所属的网络的名称。

您可以不包括 VirtualIPAddress, 也可以在 DB2PartitionType 元素中包括无数个 VirtualIPAddress 元素。

Mount 类型: MountType

MountType 元素包含关于安装点的信息, 例如, 标识所安装文件的位置的文件路径。

您可以不包括 Mount, 也可以在 DB2PartitionType 元素中包括无数个 Mount 元素。

HADRDB

类型: HADRDBType

HADRDBType 元素包含一系列高可用性灾难恢复 (HADR) 主/备用数据库对。

您可以不包括 HADRDB, 也可以在 DB2PartitionType 元素中包括无数个 HADRDB 元素。

MutualPair

类型: MutualPolicyType

MutualPolicyType 元素包含关于相互之间可以进行故障转移的集群域节点对的信息。

您可以不包括 MutualPair, 也可以在 DB2PartitionType 元素中正好包括一个 MutualPair 元素。

NPlusMNode

类型: NPlusMPolicyType

您可以不包括 NPlusMNode, 也可以在 DB2PartitionType 元素中包括无数个 NPlusMNode 元素。

属性

instanceName (必需)

在 instanceName 属性中, 必须指定与此 DB2PartitionType 元素关联的 DB2 数据库管理器实例。

dbpartitionnum (必需)

在 dbpartitionnum 属性中, 必须指定唯一标识数据库分区的数据库分区号 (例如, 在 db2nodes.cfg 文件中指定的 dbpartitionnum 号码。)

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 *MountType* XML 模式定义:

MountType 元素包含关于安装点的信息, 例如, 标识所安装文件的位置的文件路径。

- 『超元素』
- 『XML 模式定义』
- 『子元素』
- 『属性』

超元素

下列类型的元素包含 *MountType* 子元素:

- DB2PartitionType

XML 模式定义

```
<xs:complexType name='MountType'>  
  <xs:attribute name='filesystemPath' type='xs:string' use='required' />  
</xs:complexType>
```

子元素

无。

属性

filesystemPath (必需)

指定在安装文件系统时与安装点关联的路径。

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 *MutualPolicyType* XML 模式定义:

MutualPolicyType 元素包含关于相互之间可以进行故障转移的集群域节点对的信息。

- 『超元素』
- 『XML 模式定义』
- 『子元素』
- 第 83 页的『属性』

超元素

下列类型的元素包含 *MutualPolicyType* 子元素:

- DB2PartitionType

XML 模式定义

```
<xs:complexType name='MutualPolicyType'>  
  <xs:attribute name='systemPairNode1' type='xs:string' use='required' />  
  <xs:attribute name='systemPairNode2' type='xs:string' use='required' />  
</xs:complexType>
```

子元素

无。

属性

systemPairNode1 (必需)

在 systemPairNode1 中, 必须指定可以故障转移到 systemPairNode2 中指定的集群域节点的集群域节点名称。

systemPairNode2 (必需)

在 systemPairNode2 中, 必须指定可以故障转移到 systemPairNode1 中指定的集群域节点的集群域节点名称。

DB2 高可用性实例配置实用程序 (*db2haicu*) 输入文件的 *NPlusMPolicyType* XML 模式定义:

『超元素』
『XML 模式定义』
『子元素』
『属性』

超元素

下列类型的元素包含 *NPlusMPolicyType* 子元素:

- *DB2PartitionType*

XML 模式定义

```
<xs:complexType name='NPlusMPolicyType'>  
  <xs:attribute name='standbyNodeName' type='xs:string' use='required' />  
</xs:complexType>
```

子元素

无。

属性

standbyNodeName (必需)

在 standbyNodeName 元素中, 必须指定包含此 *NPlusMPolicyType* 元素的分区可以故障转移至的集群域节点的名称。

DB2 高可用性实例配置实用程序 (*db2haicu*) 输入文件的 *HADRDBType* XML 模式定义:

HADRDBType 元素包含一列高可用性灾难恢复 (HADR) 主/备用数据库对。

『超元素』
第 84 页的『XML 模式定义』
第 84 页的『子元素』
第 84 页的『属性』
第 84 页的『使用说明』
第 84 页的『限制』

超元素

下列类型的元素包含 *HADRDBType* 子元素:

- *DB2ClusterType*

- DB2PartitionType

XML 模式定义

```
<xs:complexType name='HADRDBType'>
  <xs:sequence>
    <xs:element name='VirtualIPAddress'
      type='IPAddressType'
      minOccurs='0'
      maxOccurs='unbounded' />
    <xs:element name='HADRDB'
      type='HADRDBDefn'
      maxOccurs='unbounded' />
  </xs:sequence>
</xs:complexType>
```

子元素

VirtualIPAddress

类型: IPAddressType

IPAddressType 元素包含一个 IP 地址的所有详细信息, 例如, 基地址、子网掩码和该 IP 地址所属的网络的名称。

出现规则:

可以在 HADRDBType 元素中包括零个或多个 VirtualIPAddress 元素。

HADRDB

类型: HADRDBDefn

HADRDBDefn 元素包含关于高可用性灾难恢复 (HADR) 主/备用数据库对的信息。

出现规则:

可以在 HADRDBType 元素中包括一个或多个 VirtualIPAddress 元素。

属性

无。

使用说明

如果在给定集群域的规范中包括 HADRDBType 元素, 那么还必须在相同集群域规范中包括指定 HADRFailover 的 FailoverPolicy 元素。

限制

在分区数据库环境中不能使用 HADRDBType 元素。

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 HADRDBDefn XML 模式定义:

HADRDBDefn 元素包含关于高可用性灾难恢复 (HADR) 主/备用数据库对的信息。

第 85 页的『超元素』

第 85 页的『XML 模式定义』

第 85 页的『子元素』

第 85 页的『属性』

超元素

下列类型的元素包含 HADRDBDefn 子元素:

- HADRDBType

XML 模式定义

```
<xs:complexType name='HADRDBDefn'>
  <xs:attribute name='databaseName' type='xs:string' use='required' />
  <xs:attribute name='localInstance' type='xs:string' use='required' />
  <xs:attribute name='remoteInstance' type='xs:string' use='required' />
  <xs:attribute name='localHost' type='xs:string' use='required' />
  <xs:attribute name='remoteHost' type='xs:string' use='required' />
</xs:complexType>
```

子元素

无。

属性

databaseName (必需)

输入 HADR 数据库的名称。

localInstance (必需)

localInstance 是 HADR 主数据库的数据库管理器实例。

remoteInstance (必需)

remoteInstance 是 HADR 备用数据库的数据库管理器实例。

localHost (必需)

localHost 是 HADR 主数据库所在的集群域节点的主机名。

remoteHost (必需)

remoteHost 是 HADR 备用数据库所在的集群域节点的主机名。

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 HADBType XML 模式定义:

HADBType 元素包含要包括在集群域中并使其具有高可用性的数据库的列表。

『超元素』

『XML 模式定义』

第 86 页的『子元素』

第 86 页的『属性』

超元素

下列类型的元素包含 HADBType 子元素:

- DB2ClusterType

XML 模式定义

```
<xs:complexType name='HADBType'>
  <xs:sequence>
    <xs:element name='HADB' type='HADRDBDefn' maxOccurs='unbounded' />
  </xs:sequence>
  <xs:attribute name='instanceName' type='xs:string' use='required' />
</xs:complexType>
```

子元素

HADB

类型: HADBDefn

HADBDefn 元素描述要包括在集群域中并使其具有高可用性的数据库。

出现规则:

必须在 HADBType 元素中包括一个或多个 HADB 元素。

属性

instanceName (必需)

在 instanceName 属性中, 必须指定 HADB 元素中指定的数据库所属的 DB2 数据库管理器实例。

DB2 高可用性实例配置实用程序 (db2haicu) 输入文件的 HADBDefn XML 模式元素:

HADBDefn 元素描述要包括在集群域中并使其具有高可用性的数据库。

『超元素』

『XML 模式定义』

『子元素』

『属性』

超元素

HADBDefn 是下列各项的子元素:

- HADRDBType

XML 模式定义

```
<xs:complexType name='HADBDefn'>  
  <xs:attribute name='databaseName' type='xs:string' use='required' />  
</xs:complexType>
```

子元素

无。

属性

databaseName (必需)

必须在 databaseName 属性中正好指定一个数据库名称。

DB2 高可用性实例配置实用程序 (db2haicu) 的样本 XML 输入文件:

在 sqllib 目录的 samples 子目录中存在一组您可以修改且与 db2haicu 配合使用的样本 XML 输入文件, 可用来配置集群环境。

db2ha_sample_sharedstorage_mutual.xml:

样本文件 db2ha_sample_sharedstorage_mutual.xml 是一个 XML 输入文件示例, 您将它传递至 DB2 高可用性实例配置实用程序 (db2haicu) 来指定新的集群域。db2ha_sample_sharedstorage_mutual.xml 位于 sqllib/samples/ha/xml 目录中。

功能部件

db2ha_sample_sharedstorage_mutual.xml 样本演示如何将 db2haicu 与 XML 输入文件配合使用来定义具有下列详细信息的集群域:

- 定额设备: 网络
- 集群中的计算机数 (集群域节点数): 两台
- 故障转移策略: 相互
- 数据库分区数: 一个
- 虚拟 (服务) IP 地址数: 一个
- 用于故障转移的共享安装点数: 一个

XML 源代码

```
<!-- ===== -->
<!-- = Use the DB2 High Availability Instance Configuration Utility = -->
<!-- = (db2haicu) XML schema definition, db2ha.xsd, and specify = -->
<!-- = IBM Tivoli System Automation for Multiplatforms (SA MP) = -->
<!-- = Base Component as the cluster manager. = -->
<!-- ===== -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:noNamespaceSchemaLocation="db2ha.xsd"
             clusterManagerName="TSA"
             version="1.0">

  <!-- ===== -->
  <!-- = Create a cluster domain named db2HADomain. = -->
  <!-- ===== -->
  <ClusterDomain domainName="db2HADomain">

    <!-- ===== -->
    <!-- = Specify a network quorum device (IP address: 19.126.4.5). = -->
    <!-- = The IP must be pingable at all times by each of the cluster = -->
    <!-- = domain nodes. = -->
    <!-- ===== -->
    <Quorum quorumDeviceProtocol="network" quorumDeviceName="19.126.4.5"/>

    <!-- ===== -->
    <!-- = Create a network named db2_public_network_0 with an IP = -->
    <!-- = network protocol. = -->
    <!-- = This network contains two computers: hasys01 and hasys02. = -->
    <!-- = Each computer has one network interface card (NIC) called = -->
    <!-- = eth0. = -->
    <!-- = The IP address of the NIC on hasys01 is 19.126.52.139 = -->
    <!-- = The IP address of the NIC on hasys02 is 19.126.52.140 = -->
    <!-- ===== -->
    <PhysicalNetwork physicalNetworkName="db2_public_network_0"
                    physicalNetworkProtocol="ip">

      <Interface interfaceName="eth0" clusterNodeName="hasys01">
        <IPAddress baseAddress="19.126.52.139"
                  subnetMask="255.255.255.0"
                  networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys02">
        <IPAddress baseAddress="19.126.52.140"
                  subnetMask="255.255.255.0"
                  networkName="db2_public_network_0"/>
      </Interface>

    </PhysicalNetwork>

  </ClusterDomain>

</DB2Cluster>
```

```

<!-- ===== -->
<!-- = List the computers (cluster nodes) in the cluster domain. = -->
<!-- ===== -->
<ClusterNode clusterNodeName="hasys01"/>
<ClusterNode clusterNodeName="hasys02"/>

</ClusterDomain>

<!-- ===== -->
<!-- = The failover policy specifies the order in which the cluster = -->
<!-- = domain nodes should fail over. = -->
<!-- ===== -->
<FailoverPolicy>
  <Mutual />
</FailoverPolicy>

<!-- ===== -->
<!-- = Specify all the details of the database partition = -->
<!-- ===== -->
<DB2PartitionSet>

  <DB2Partition dbpartitionnum="0" instanceName="db2inst1">
    <VirtualIPAddress baseAddress="19.126.52.222"
      subnetMask="255.255.255.0"
      networkName="db2_public_network_0"/>
    <Mount filesystemPath="/home/db2inst1"/>
    <MutualPair systemPairNode1="hasys01" systemPairNode2="hasys02" />
  </DB2Partition>

</DB2PartitionSet>

</DB2Cluster>

```

db2ha_sample_DPF_mutual.xml:

样本文件 `db2ha_sample_DPF_mutual.xml` 是一个 XML 输入文件示例，您将它传递至 DB2 高可用性实例配置实用程序 (`db2haicu`) 来指定新的集群域。`db2ha_sample_DPF_mutual.xml` 位于 `sql1lib/samples/ha/xml` 目录中。

功能部件

`db2ha_sample_DPF_mutual.xml` 样本演示如何将 `db2haicu` 与 XML 输入文件配合使用来定义具有下列详细信息的集群域:

- 定额设备: 网络
- 集群中的计算机数 (集群域节点数): 四台
- 故障转移策略: 相互
- 数据库分区数: 两个
- 虚拟 (服务) IP 地址数: 一个
- 用于故障转移的共享安装点数: 两个
- 为实现高可用性配置的数据库数: 两个

XML 源代码

```

<!-- ===== -->
<!-- = Use the DB2 High Availability Instance Configuration Utility = -->
<!-- = (db2haicu) XML schema definition, db2ha.xsd, and specify = -->
<!-- = IBM Tivoli System Automation for Multiplatforms (SA MP) = -->
<!-- = Base Component as the cluster manager. = -->

```

```

<!-- ===== -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:noNamespaceSchemaLocation="db2ha.xsd"
             clusterManagerName="TSA"
             version="1.0">

  <!-- ===== -->
  <!-- = Create a cluster domain named db2HADomain. = -->
  <!-- ===== -->
  <ClusterDomain domainName="db2HADomain">

    <!-- ===== -->
    <!-- = Specify a network quorum device (IP address: 19.126.4.5). = -->
    <!-- = The IP must be pingable at all times by each of the cluster = -->
    <!-- = domain nodes. = -->
    <!-- ===== -->
    <Quorum quorumDeviceProtocol="network" quorumDeviceName="19.126.4.5"/>

    <!-- ===== -->
    <!-- = Create a network named db2_public_network_0 with an IP = -->
    <!-- = network protocol. = -->
    <!-- = This network contains four computers: hasys01, hasys02, = -->
    <!-- = hasys03, and hasys04. = -->
    <!-- = Each computer has a network interface card called eth0. = -->
    <!-- = The IP address of eth0 on hasys01 is 19.126.124.30 = -->
    <!-- = The IP address of eth0 on hasys02 is 19.126.124.31 = -->
    <!-- = The IP address of eth0 on hasys03 is 19.126.124.32 = -->
    <!-- = The IP address of eth0 on hasys04 is 19.126.124.33 = -->
    <!-- ===== -->
    <PhysicalNetwork physicalNetworkName="db2_public_network_0"
                    physicalNetworkProtocol="ip">

      <Interface interfaceName="eth0" clusterNodeName="hasys01">
        <IPAddress baseAddress="19.126.124.30"
                  subnetMask="255.255.255.0"
                  networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys02">
        <IPAddress baseAddress="19.126.124.31"
                  subnetMask="255.255.255.0"
                  networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys03">
        <IPAddress baseAddress="19.126.124.32"
                  subnetMask="255.255.255.0"
                  networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys04">
        <IPAddress baseAddress="19.126.124.33"
                  subnetMask="255.255.255.0"
                  networkName="db2_public_network_0"/>
      </Interface>

    </PhysicalNetwork>

    <!-- ===== -->
    <!-- = Create a network named db2_private_network_0 with an IP = -->
    <!-- = network protocol. = -->
    <!-- = This network contains four computers: hasys01, hasys02, = -->
    <!-- = hasys03, and hasys04 (same as db2_public_network_0.) = -->
    <!-- = In addition to eth0, each computer has a network interface = -->
    <!-- = card called eth1. = -->
    <!-- = The IP address of eth1 on hasys01 is 192.168.23.101 = -->
    <!-- = The IP address of eth1 on hasys02 is 192.168.23.102 = -->

```

```

<!-- = The IP address of eth1 on hasys03 is 192.168.23.103      = -->
<!-- = The IP address of eth1 on hasys04 is 192.168.23.104      = -->
<!-- =====>
<PhysicalNetwork physicalNetworkName="db2_private_network_0"
    physicalNetworkProtocol="ip">

    <Interface interfaceName="eth1" clusterNodeName="hasys01">
        <IPAddress baseAddress="192.168.23.101"
            subnetMask="255.255.255.0"
            networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys02">
        <IPAddress baseAddress="192.168.23.102"
            subnetMask="255.255.255.0"
            networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys03">
        <IPAddress baseAddress="192.168.23.103"
            subnetMask="255.255.255.0"
            networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys04">
        <IPAddress baseAddress="192.168.23.104"
            subnetMask="255.255.255.0"
            networkName="db2_private_network_0"/>
    </Interface>

</PhysicalNetwork>

<!-- =====>
<!-- = List the computers (cluster nodes) in the cluster domain. = -->
<!-- =====>
<ClusterNode clusterNodeName="hasys01"/>
<ClusterNode clusterNodeName="hasys02"/>
<ClusterNode clusterNodeName="hasys03"/>
<ClusterNode clusterNodeName="hasys04"/>

</ClusterDomain>

<!-- =====>
<!-- = The failover policy specifies the order in which the cluster = -->
<!-- = domain nodes should fail over.                               = -->
<!-- =====>
<FailoverPolicy>
    <Mutual />
</FailoverPolicy>

<!-- =====>
<!-- = Specify all the details of the database partitions.         = -->
<!-- =====>
<DB2PartitionSet>

    <DB2Partition dbpartitionnum="0" instanceName="db2inst1">
        <VirtualIPAddress baseAddress="19.126.124.251"
            subnetMask="255.255.255.0"
            networkName="db2_public_network_0"/>
        <Mount filesystemPath="/hafs/db2inst1/NODE0000"/>
        <MutualPair systemPairNode1="hasys01" systemPairNode2="hasys02" />
    </DB2Partition>

    <DB2Partition dbpartitionnum="1" instanceName="db2inst1">
        <Mount filesystemPath="/hafs/db2inst1/NODE0001"/>

```

```

        <MutualPair systemPairNode1="hasys02" systemPairNode2="hasys01" />
    </DB2Partition>

    <DB2Partition dbpartitionnum="2" instanceName="db2inst1">
        <Mount filesystemPath="/hafs/db2inst1/NODE0002"/>
        <MutualPair systemPairNode1="hasys03" systemPairNode2="hasys04" />
    </DB2Partition>

    <DB2Partition dbpartitionnum="3" instanceName="db2inst1">
        <Mount filesystemPath="/hafs/db2inst1/NODE0003"/>
        <MutualPair systemPairNode1="hasys04" systemPairNode2="hasys03" />
    </DB2Partition>

</DB2PartitionSet>

<!-- ===== -->
<!-- = List of databases to be configured for High Availability = -->
<!-- ===== -->
<HADBSet instanceName="db2inst1">
    <HADB databaseName = "SAMPLE" />
    <HADB databaseName = "MYDB" />
</HADBSet>

</DB2Cluster>

```

db2ha_sample_DPF_NPlusM.xml:

样本文件 `db2ha_sample_DPF_NPlusM.xml` 是一个 XML 输入文件示例，您将它传递至 DB2 高可用性实例配置实用程序 (`db2haicu`) 来指定新的集群域。`db2ha_sample_DPF_NPlusM.xml` 位于 `sql1lib/samples/ha/xml` 目录中。

功能部件

`db2ha_sample_DPF_NPlusM.xml` 样本演示如何将 `db2haicu` 与 XML 输入文件配合使用来定义具有下列详细信息的集群域:

- 定额设备: 网络
- 集群中的计算机数 (集群域节点数): 四台
- 故障转移策略: N 加 M
- 数据库分区数: 两个
- 虚拟 (服务) IP 地址数: 一个
- 用于故障转移的共享安装点数: 四个

XML 源代码

```

<!-- ===== -->
<!-- = Use the DB2 High Availability Instance Configuration Utility = -->
<!-- = (db2haicu) XML schema definition, db2ha.xsd, and specify = -->
<!-- = IBM Tivoli System Automation for Multiplatforms (SA MP) = -->
<!-- = Base Component as the cluster manager. = -->
<!-- ===== -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="db2ha.xsd"
    clusterManagerName="TSA"
    version="1.0">

    <!-- ===== -->
    <!-- = Create a cluster domain named db2HADomain. = -->
    <!-- ===== -->

```

```

<ClusterDomain domainName="db2HADomain">

  <!-- ===== -->
  <!-- = Specify a network quorum device (IP address: 19.126.4.5). = -->
  <!-- = The IP must be pingable at all times by each of the cluster = -->
  <!-- = domain nodes. = -->
  <!-- ===== -->
  <Quorum quorumDeviceProtocol="network" quorumDeviceName="19.126.4.5"/>

  <!-- ===== -->
  <!-- = Create a network named db2_public_network_0 with an IP = -->
  <!-- = network protocol. = -->
  <!-- = This network contains four computers: hasys01, hasys02, = -->
  <!-- = hasys03, and hasys04. = -->
  <!-- = Each computer has a network interface card called eth0. = -->
  <!-- = The IP address of eth0 on hasys01 is 19.126.124.30 = -->
  <!-- = The IP address of eth0 on hasys02 is 19.126.124.31 = -->
  <!-- = The IP address of eth0 on hasys03 is 19.126.124.32 = -->
  <!-- = The IP address of eth0 on hasys04 is 19.126.124.33 = -->
  <!-- ===== -->
  <PhysicalNetwork physicalNetworkName="db2_public_network_0"
    physicalNetworkProtocol="ip">

    <Interface interfaceName="eth0" clusterNodeName="hasys01">
      <IPAddress baseAddress="19.126.124.30"
        subnetMask="255.255.255.0"
        networkName="db2_public_network_0"/>
    </Interface>

    <Interface interfaceName="eth0" clusterNodeName="hasys02">
      <IPAddress baseAddress="19.126.124.31"
        subnetMask="255.255.255.0"
        networkName="db2_public_network_0"/>
    </Interface>

    <Interface interfaceName="eth0" clusterNodeName="hasys03">
      <IPAddress baseAddress="19.126.124.32"
        subnetMask="255.255.255.0"
        networkName="db2_public_network_0"/>
    </Interface>

    <Interface interfaceName="eth0" clusterNodeName="hasys04">
      <IPAddress baseAddress="19.126.124.33"
        subnetMask="255.255.255.0"
        networkName="db2_public_network_0"/>
    </Interface>

  </PhysicalNetwork>

  <!-- ===== -->
  <!-- = Create a network named db2_private_network_0 with an IP = -->
  <!-- = network protocol. = -->
  <!-- = This network contains four computers: hasys01, hasys02, = -->
  <!-- = hasys03, and hasys04 (same as db2_public_network_0.) = -->
  <!-- = In addition to eth0, each computer has a network interface = -->
  <!-- = card called eth1. = -->
  <!-- = The IP address of eth1 on hasys01 is 192.168.23.101 = -->
  <!-- = The IP address of eth1 on hasys02 is 192.168.23.102 = -->
  <!-- = The IP address of eth1 on hasys03 is 192.168.23.103 = -->
  <!-- = The IP address of eth1 on hasys04 is 192.168.23.104 = -->
  <!-- ===== -->
  <PhysicalNetwork physicalNetworkName="db2_private_network_0"
    physicalNetworkProtocol="ip">

    <Interface interfaceName="eth1" clusterNodeName="hasys01">
      <IPAddress baseAddress="192.168.23.101"
        subnetMask="255.255.255.0"

```

```

        networkName="db2_private_network_0"/>
</Interface>

<Interface interfaceName="eth1" clusterNodeName="hasys02">
  <IPAddress baseAddress="192.168.23.102"
    subnetMask="255.255.255.0"
    networkName="db2_private_network_0"/>
</Interface>

<Interface interfaceName="eth1" clusterNodeName="hasys03">
  <IPAddress baseAddress="192.168.23.103"
    subnetMask="255.255.255.0"
    networkName="db2_private_network_0"/>
</Interface>

<Interface interfaceName="eth1" clusterNodeName="hasys04">
  <IPAddress baseAddress="192.168.23.104"
    subnetMask="255.255.255.0"
    networkName="db2_private_network_0"/>
</Interface>

</PhysicalNetwork>

<!-- ===== -->
<!-- = List the computers (cluster nodes) in the cluster domain. = -->
<!-- ===== -->
<ClusterNode clusterNodeName="hasys01"/>
<ClusterNode clusterNodeName="hasys02"/>
<ClusterNode clusterNodeName="hasys03"/>
<ClusterNode clusterNodeName="hasys04"/>

</ClusterDomain>

<!-- ===== -->
<!-- = The failover policy specifies the order in which the cluster = -->
<!-- = domain nodes should fail over. = -->
<!-- ===== -->
<FailoverPolicy>
  <NPlusM />
</FailoverPolicy>

<!-- ===== -->
<!-- = Specify all the details of the database partitions = -->
<!-- ===== -->
<DB2PartitionSet>

  <DB2Partition dbpartitionnum="0" instanceName="db2inst1">
    <VirtualIPAddress baseAddress="19.126.124.250"
      subnetMask="255.255.255.0"
      networkName="db2_public_network_0"/>
    <Mount filesystemPath="/ha_dpf1/db2inst1/NODE0000"/>
    <Mount filesystemPath="/hafs/NODE0000"/>
    <NPlusMNode standbyNodeName="hasys03" />
  </DB2Partition>

  <DB2Partition dbpartitionnum="1" instanceName="db2inst1">
    <Mount filesystemPath="/ha_dpf1/db2inst1/NODE0001"/>
    <Mount filesystemPath="/hafs/NODE0001"/>
    <NPlusMNode standbyNodeName="hasys04" />
  </DB2Partition>

</DB2PartitionSet>

</DB2Cluster>

```

db2ha_sample_HADR.xml:

样本文件 *db2ha_sample_DPF_HADR.xml* 是一个 XML 输入文件示例，您将它传递至 DB2 高可用性实例配置实用程序 (*db2haicu*) 来指定新的集群域。*db2ha_sample_HADR.xml* 位于 *sql1lib/samples/ha/xml* 目录中。

功能部件

db2ha_sample_HADR.xml 样本演示如何将 *db2haicu* 与 XML 输入文件配合使用来定义具有下列详细信息的集群域:

- 定额设备: 网络
- 集群中的计算机数 (集群域节点数): 两台
- 故障转移策略: HADR
- 数据库分区数: 一个
- 虚拟 (服务) IP 地址数: 无
- 用于故障转移的共享安装点数: 无

XML 源代码

```
<!-- ===== -->
<!-- = Use the DB2 High Availability Instance Configuration Utility = -->
<!-- = (db2haicu) XML schema definition, db2ha.xsd, and specify = -->
<!-- = IBM Tivoli System Automation for Multiplatforms (SA MP) = -->
<!-- = Base Component as the cluster manager. = -->
<!-- ===== -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:noNamespaceSchemaLocation="db2ha.xsd"
             clusterManagerName="TSA"
             version="1.0">

  <!-- ===== -->
  <!-- = Create a cluster domain named db2HADomain. = -->
  <!-- ===== -->
  <ClusterDomain domainName="db2HADomain">

    <!-- ===== -->
    <!-- = Specify a network quorum device (IP address: 19.126.4.5). = -->
    <!-- = The IP must be pingable at all times by each of the cluster = -->
    <!-- = domain nodes. = -->
    <!-- ===== -->
    <Quorum quorumDeviceProtocol="network" quorumDeviceName="19.126.4.5"/>

    <!-- ===== -->
    <!-- = Create a network named db2_public_network_0 with an IP = -->
    <!-- = network protocol. = -->
    <!-- = This network contains two computers: hasys01 and hasys02. = -->
    <!-- = Each computer has a network interface card called eth0. = -->
    <!-- = The IP address of eth0 on hasys01 is 19.126.52.139 = -->
    <!-- = The IP address of eth0 on hasys01 is 19.126.52.140 = -->
    <!-- ===== -->
    <PhysicalNetwork physicalNetworkName="db2_public_network_0"
                    physicalNetworkProtocol="ip">

      <Interface interfaceName="eth0" clusterNodeName="hasys01">
        <IPAddress baseAddress="19.126.52.139"
                  subnetMask="255.255.255.0"
                  networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys02">
```



```

        <IPAddress baseAddress="19.126.52.140"
                  subnetMask="255.255.255.0"
                  networkName="db2_public_network_0"/>
    </Interface>

</PhysicalNetwork>

<!-- ===== -->
<!-- = Create a network named db2_private_network_0 with an IP      = -->
<!-- = network protocol.                                          = -->
<!-- = This network contains two computers: hasys01 and hasys02.  = -->
<!-- = In addition to eth0, each computer has a network interface = -->
<!-- = card called eth1.                                          = -->
<!-- = The IP address of eth1 on hasys01 is 192.168.23.101      = -->
<!-- = The IP address of eth1 on hasys02 is 192.168.23.102      = -->
<!-- ===== -->
<PhysicalNetwork physicalNetworkName="db2_private_network_0"
                  physicalNetworkProtocol="ip">

    <Interface interfaceName="eth1" clusterNodeName="hasys01">
        <IPAddress baseAddress="192.168.23.101"
                  subnetMask="255.255.255.0"
                  networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys02">
        <IPAddress baseAddress="192.168.23.102"
                  subnetMask="255.255.255.0"
                  networkName="db2_private_network_0"/>
    </Interface>

</PhysicalNetwork>

<!-- ===== -->
<!-- = List the computers (cluster nodes) in the cluster domain.  = -->
<!-- ===== -->
<ClusterNode clusterNodeName="hasys01"/>
<ClusterNode clusterNodeName="hasys02"/>

</ClusterDomain>

<!-- ===== -->
<!-- = The failover policy specifies the order in which the cluster = -->
<!-- = domain nodes should fail over.                               = -->
<!-- ===== -->
<FailoverPolicy>
    <HADRFailover />
</FailoverPolicy>

<!-- ===== -->
<!-- = Specify all the details of the database partitions          = -->
<!-- ===== -->
<DB2PartitionSet>
    <DB2Partition dbpartitionnum="0" instanceName="db2inst1" />
</DB2PartitionSet>

<!-- ===== -->
<!-- = List of HADR databases                                      = -->
<!-- ===== -->
<HADRDBSet>
    <HADRDB databaseName="HADRDB"
            localInstance="db2inst1"
            remoteInstance="db2inst1"
            localHost="hasys01">

```

```
        remoteHost="hasys02" />
</HADRDBSet>

</DB2Cluster>
```

DB2 高可用性实例配置实用程序 (db2haicu) 先决条件

在使用 DB2 高可用性实例配置实用程序 (db2haicu) 之前, 必须执行一组任务。

一般信息

在数据库管理器实例所有者可以运行 db2haicu 之前, 具有 root 用户权限的用户必须运行 preprnode 命令。

preprnode 是适用于 AIX 的 Reliable Scalable Cluster Technology (RSCT) 文件集与适用于 Linux 的 RSCT 程序包的一部分。preprnode 负责为集群内通信初始化节点。设置集群时, 需要运行 preprnode 命令。有关 preprnode 的更多信息, 请参阅:

- preprnode 命令 (AIX)
- RSCT for Linux Technical Reference - preprnode

有关 RSCT 的更多信息, 请参阅 RSCT Administration Guide - What is RSCT?

在运行 db2haicu 之前, 数据库管理器实例所有者必须执行下列任务:

- 在将被添加到集群的所有机器上使服务文件同步。
- 为将用于创建集群域的数据库管理器实例运行 db2profile 脚本。
- 使用 db2start 命令启动数据库管理器。

DB2 高可用性灾难恢复 (HADR)

如果将使用 HADR 功能, 那么执行下列任务:

- 确保按照各自主数据库和备用数据库角色来启动所有 DB2 高可用性灾难恢复 (HADR) 数据库, 并且确保所有主/备用 HADR 数据库对处于对等状态。
- 将所有 HADR 数据库的 **hadr_peer_window** 配置为至少 120 秒的值。
- 禁用 DB2 故障监视器。

分区数据库环境

如果具有多个数据库分区需要配置以获取高可用性, 那么执行下列步骤:

- 在将被添加到集群域的所有机器上配置 DB2_NUM_FAILOVER_NODES 注册表变量。
- (可选) 在运行 db2haicu 之前激活数据库。

使用 DB2 高可用性实例配置实用程序 (db2haicu) 来创建集群域

当首次为数据库管理器实例运行 DB2 高可用性实例配置实用程序 (db2haicu) 时, db2haicu 会创建被称为集群域的集群模型。

DB2 高可用性实例配置实用程序 (db2haicu) 自动检测到的数据库路径:

首次运行 DB2 高可用性实例配置实用程序 (db2haicu) 时, db2haicu 将搜索数据库系统以查找与集群配置相关的数据库配置信息。

单一数据库分区环境

在单一数据库分区环境中，db2haicu 将自动检测下列路径：

- 实例主目录路径
- 审计日志路径
- 审计归档日志路径
- 同步点管理器（SPM）日志路径
- DB2 诊断日志（db2diag 日志文件）路径
- 数据库相关路径：
 - 数据库日志路径
 - 数据库表空间容器路径
 - 数据库表空间目录路径
 - 本地数据库目录

多数据库分区环境

在多数据库分区环境中，db2haicu 只自动检测下列路径：

- 数据库日志路径
- 数据库表空间容器路径
- 数据库表空间目录路径
- 本地数据库目录

使用 DB2 高可用性实例配置实用程序（db2haicu）来维护集群域

当您使用 db2haicu 来修改集群环境的集群域模型时，数据库管理器将相关更改传播到数据库管理器实例和集群配置。

开始前

在可以使用 db2haicu 来配置集群环境之前，必须创建和配置集群域。有关更多信息，请参阅第 96 页的『使用 DB2 高可用性实例配置实用程序（db2haicu）来创建集群域』

关于此任务

db2haicu 维护任务包括将诸如数据库或集群节点之类的集群元素添加到集群域，并且从集群域中除去元素。db2haicu 维护任务还包括修改集群域元素的详细信息，例如数据库管理器实例的故障转移策略。

过程

1. 运行 db2haicu

当在维护方式下运行 db2haicu 时，db2haicu 向您展示可以对集群域执行的操作的列表：

- 添加或除去集群节点（主机名标识的机器）
- 添加或除去网络接口（网络接口卡）
- 添加或除去数据库分区（仅限于分区数据库环境）
- 添加或除去 DB2 高可用性灾难恢复（HADR）数据库

- 添加或删除高可用性数据库
- 添加或删除安装点
- 添加或删除 IP 地址
- 添加或删除非重要路径
- 移动数据库分区和 HADR 数据库以进行预定维护
- 更改当前实例的故障转移策略
- 为集群域创建新的定额设备
- 破坏集群域

2. 选择要执行的任务并回答 db2haicu 提出的后续问题。

结果

数据库管理器在集群域中使用该信息来与集群管理器进行协同工作。当使用 db2haicu 来配置数据库和集群元素时，DB2 高可用性（HA）功能部件提供的集成和自动化集群配置与管理中将包含那些元素。使用 db2haicu 来进行数据库管理器实例配置更改时，数据库管理器将为您执行所需的集群管理器配置更改，因此您随后不必调用集群管理器。

下一步任务

DB2 高可用性实例配置实用程序（db2haicu）没有单独的诊断日志。可以使用数据库管理器诊断日志、db2diag 日志文件和 db2pd 工具来调查与诊断 db2haicu 错误。有关更多信息，请参阅：『对 DB2 高可用性实例配置实用程序（db2haicu）进行故障诊断』

对 DB2 高可用性实例配置实用程序（db2haicu）进行故障诊断

DB2 高可用性实例配置实用程序（db2haicu）没有单独的诊断日志。可以使用数据库管理器诊断日志、db2diag 日志文件和 db2pd 工具来调查与诊断 db2haicu 错误。

DB2 高可用性实例配置实用程序（db2haicu）限制

使用 DB2 高可用性实例配置实用程序（db2haicu）时存在一些限制。

- 『软件和硬件』
- 第 99 页的『配置任务』
- 第 99 页的『使用说明』
- 第 99 页的『建议』

软件和硬件

•

目前，IBM Tivoli System Automation for Multiplatforms (SA MP) 版本 2.2 修订包 3 是 db2haicu 支持的唯一的集群管理器。

•

另外，还需要参阅 Reliable Scalable Cluster Technology (RSCT) V2.4.7.3。有关 RSCT 的更多信息，请参阅：RSCT Administration Guide - What is RSCT?

•

db2haicu 不支持 IP V6。

配置任务

您无法使用 `db2haicu` 来执行下列任务:

- 无法使用 `db2haicu` 来配置自动客户机重新路由。
- 当从 DB2 数据库 Linux 版、UNIX 版和 Windows 版 版本 9 升级到 IBM 数据服务器版本 9.5 或从版本 9.5 升级到更高版本时，无法使用 `db2haicu` 来迁移集群配置。要迁移集群配置，必须执行下列步骤：
 1. 删除现有集群域（如果存在集群域）
 2. 升级数据库服务器
 3. 使用 `db2haicu` 创建新的集群域

使用说明

当规划集群配置和管理活动时，请考虑下列 `db2haicu` 使用说明:

- 即使 `db2haicu` 可执行某些通常需要 `root` 用户权限的管理任务，但 `db2haicu` 仍然要使用数据库管理器实例所有者的特权来运行。尽管 `db2haicu` 只有实例所有者特权，但由 `root` 用户执行的 `db2haicu` 初始化仍然使它能够执行所需的配置更改。
- 创建新的集群域时，`db2haicu` 不会验证您为新集群域指定的名称是否有效。例如，`db2haicu` 不会确认该名称长度是否有效，或是否包含有效的字符，或是否与现有集群域不同名。
- `db2haicu` 不会验证或确认用户指定或传递到集群管理器的信息。因为 `db2haicu` 无法知道所有集群管理器关于集群对象名的限制，例如，`db2haicu` 将文本传递到集群管理器，而没有确认它的字符或长度等是否有效。
- 如果在创建和配置新的集群域时发生错误且 `db2haicu` 失败，那么必须执行下列步骤：
 1. 通过运行使用 **-delete** 参数的 `db2haicu` 来除去创建了一部分的集群域的资源组
 2. 通过再次调用 `db2haicu` 来重新创建新的集群域。
- 当运行带 **-delete** 参数的 `db2haicu` 时，`db2haicu` 会立即删除与当前数据库管理器实例相关联的资源组，而不确认这些资源组是否被锁定。
- 要除去与 DB2 高可用性灾难恢复（HADR）主数据库与备用数据库对的数据库管理器实例相关联的资源组，请执行下列步骤：
 1. 首先针对 HADR 备用数据库的数据库管理器实例运行带 **-delete** 参数的 `db2haicu`。
 2. 另外，还针对 HADR 主数据库的数据库管理器实例运行带 **-delete** 参数的 `db2haicu`。
- 如果您使用 `db2haicu` 尝试执行的集群操作超时，那么 `db2haicu` 将不会返回错误。除非在进行 `db2haicu` 调用后查看诊断日志；或除非后续集群操作失败，且在调查后续故障时，您才能确定原始的集群操作超时，否则当集群操作超时时，您将不知道操作已超时。
- 如果尝试将给定数据库实例的故障转移策略更改为主动 - 被动，那么存在一种情况，在此情况下配置操作将失败，但 `db2haicu` 不会返回错误。如果将当前脱机的机器指定为活动机器，那么 `db2haicu` 将不会使该机器成为活动机器，但 `db2haicu` 将不会返回错误，指示该更改失败了。

建议

以下是使用 `db2haicu` 时配置集群和数据库管理器实例的的建议的列表。

- 当通过将条目添加到 `/etc/fstab` 来为集群添加新的安装点时，请使用 **noauto** 选项来防止安装点自动安装在集群中的多个机器上。例如：

```
dev/vpatha1      /db/svtpdb/NODE0010    ext3 noauto 0 0
```

DB2 集群管理器 API

DB2 集群管理器 API 定义了一组功能，它们使数据库管理器能够向集群管理器通知配置更改。

受支持的集群管理软件

集群管理软件使 DB2 数据库操作能从集群中一个节点上的故障主数据库转移到集群中另一节点上的辅助数据库。

DB2 数据库支持以下集群管理软件：

- 适用于 AIX 的高可用性集群多处理（HACMP）

有关 HACMP/ES 的详细信息，请参阅标题为“IBM DB2 Universal Database Enterprise Edition for AIX and HACMP/ES”的白皮书，可以从 IBM 软件库 Web 站点（<http://www.ibm.com/software/sw-library/>）获得该白皮书。

- Tivoli System Automation for Linux。

有关 Tivoli System Automation 的详细信息，请参阅标题为“Highly Available DB2 Universal Database using Tivoli System Automation for Linux”的白皮书，可以从 IBM 软件库 Web 站点（<http://www.ibm.com/software/sw-library/>）获得该白皮书。

- Microsoft® Cluster Server，用于 Windows 操作系统

有关 Microsoft Cluster Server 的信息，请参阅以下白皮书，可以从 IBM 软件库 Web 站点（<http://www.ibm.com/software/sw-library/>）获得该白皮书：“Implementing IBM DB2 Universal Database V8.1 Enterprise Server Edition with Microsoft Cluster Server”。

- Sun Cluster 或 VERITAS Cluster Server，用于 Solaris 操作系统。

有关 Sun Cluster 的信息，请参阅标题为“DB2 Universal Database and High Availability on Sun Cluster 3.X”的白皮书，可以从 IBM 软件库 Web 站点（<http://www.ibm.com/software/sw-library/>）获得该白皮书。有关 VERITAS Cluster Server 的信息，请参阅标题为“DB2 UDB and High Availability with VERITAS Cluster Server”的白皮书，可以从“IBM 支持与下载”Web 站点（<http://www.ibm.com/support/docview.wss?uid=swg21045033>）获得该白皮书。

- 惠普公司的 Multi-Computer/ServiceGuard

高可用性集群多处理 AIX 版

高可用性集群多处理（HACMP）AIX 版是一个集群管理软件。HACMP 集群中的节点交换称为脉动信号或保持活动信息包的消息。如果某个节点停止发送这些消息，那么 HACMP 将在集群中的其他节点间调用故障转移；一旦修复了出现故障的节点后，HACMP 会将该节点重新集成到集群中。

有两种类型的事件：一种是在 HACMP 的操作内期望的标准事件，另一种是与硬件和软件组件中的参数的监视相关联的用户定义的事件。

其中一个标准事件是 `node_down` 事件。这是当集群中的某个节点出现故障，并且 HACMP 在集群中的其他节点间启动了故障转移时发生的事件。当计划应该执行什么操作来作为恢复过程的一部分时，HACMP 允许两个故障转移选项：热（或空闲）备用和相互接管。

注：当使用 HACMP 时，通过使用 `db2iauto` 实用程序确保在引导时不启动 DB2 实例：

```
db2iauto -off InstName
```

其中，`InstName` 是实例的登录名。

集群配置

在热备用配置中，作为接管节点的 AIX 处理器节点不运行任何其他工作负载。在相互接管配置中，作为接管节点的 AIX 处理器节点要运行其他工作负载。

在分区数据库环境中，对于每个节点上的数据库分区，DB2 数据库通常以相互接管的方式来运行。以下情况是一个例外：当目录分区是热备用配置的一部分时。

一个计划的注意事项是如何管理大型集群。管理一个小的集群比管理一个大的集群容易得多；但是，管理一个大的集群又比管理许多小的集群容易得多。在计划过程中，要考虑在集群环境中如何使用应用程序。例如，如果有一个单独的、大型的、同类应用程序在 16 个节点上运行，那么将配置作为单个集群管理可能比作为 8 个两节点集群管理更容易些。如果相同的 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，那么可能最好将这些节点分组到更小的集群中。记住，一次只能将节点集成到一个 HACMP 集群中；启动多个集群的一个配置比启动一个大集群快得多。只要某个节点及其备份在同一集群中，HACMP 就可以同时支持单个和多个集群。

HACMP 故障转移恢复允许资源组对物理节点的预定义（也称作级联）赋值。该故障转移恢复过程还允许资源组对物理节点的浮动（也称作旋转）赋值。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程序服务器指定应用程序或应用程序组件，它们可由 HACMP 通过故障转移和重新集成在物理节点间进行操纵。故障转移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）。如果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，那么其他那些节点可能可以访问这些磁盘并重新启动该数据库分区（在接管节点上）。这是 HACMP 自动执行的操作类型。HACMP 也可用来复原 DB2 实例主用户目录所使用的 NFS 文件系统。

在分区数据库环境中计划使用 DB2 数据库进行恢复时，完整阅读 HACMP 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 集群，以及恢复节点（热备用或相互接管）。

强烈建议在外部分盘配置中建立磁盘和适配器的镜像。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。在相互接管配置中，这种方案需要某些附加计划，以便配对的节点可相互访问对方的卷组而不造成冲突。在分区数据库环境中，这表示所有容器名在所有数据库中都必须唯一的。

实现唯一性的一种方法是将数据库分区号包括在名称中。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，如果指定了附加自变量，那么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。自变量必须在容器字符串的末尾，并且只能使用下列其中一种格式：

表 7. 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. % 是模数。		
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
```

- 创建在有四个节点的系统上使用的容器。

```
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
```

为 HACMP 配置 DB2 数据库分区

一旦完成配置，HACMP 就启动实例中的每个数据库分区，一次启动一个物理节点。建议使用多个集群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 集群比启动四个 16 节点集群要快。

脚本文件与 DB2 企业服务器版封装在一起，以帮助配置热备用或相互接管节点中的 HACMP 故障转移或恢复。对于单节点，该脚本文件称为 rc.db2pe.ee，对于多节点，

那么称为 rc.db2pe.eee。它们位于 sqllib/samples/hacmp/es 目录中。将正确的文件复制到 HACMP 集群中每个系统上的 /usr/bin, 并将其重命名为 rc.db2pe。

另外, 可在相互接管配置中的故障转移阶段从 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, 可以配置缓冲池大小以确保正确的资源分配。)

HACMP 事件监视和用户定义的事件

进程在给定节点上中断时启动故障转移操作是用户定义的事件的一个示例。作为集群设置的一部分, 必须将事件手动配置为用户定义的事件。

有关高可用性的 IBM DB2 数据库环境的实现和设计的详细信息, 请参阅 IBM 软件库 Web 站点 (<http://www.ibm.com/software/sw-library/>)。

IBM Tivoli System Automation for Multiplatforms (Linux 和 AIX)

IBM Tivoli System Automation for Multiplatforms (Tivoli SAM) 是集群管理软件, 便于将用户、应用程序和数据从一个数据库系统自动切换至集群中的另一个系统。Tivoli SAM 自动控制 IT 资源 (如进程、文件系统和 IP 地址)。

Tivoli SAM 提供了一个框架来自动管理资源的可用性。资源的示例包括:

- 可以通过编写其启动、监视和停止脚本来进行控制的任何软件
- Tivoli SAM 已有权访问的任何网络接口卡 (NIC)。也就是说, Tivoli SAM 通过在它有权访问的 NIC 之间浮动 IP 地址来管理用户要使用的任何 IP 地址的可用性。

例如, DB2 实例和“高可用性灾难恢复”功能都有启动、停止和监视命令。因此, 可以编写 Tivoli SAM 脚本来自动管理这些资源。紧密相关的资源 (例如, 同时一起在同一个节点上运行的资源) 称为资源组。

DB2 资源

在单一分区的 DB2 环境中, 单个 DB2 实例正在服务器上运行。此 DB2 实例具有对数据的本地访问权 (数据的可执行映像和数据库由此实例拥有)。如果要使此 DB2 实例可供远程客户机访问, 那么必须为此 DB2 实例指定一个未使用的 IP 地址。

DB2 实例、本地数据和 IP 地址都属于资源, 它们必须由 Tivoli SAM 自动管理。因为这些资源紧密相关 (例如, 它们同时一起在同一个节点上运行), 所以它们称为资源组。

整个资源组被放在集群中的一个节点上。如果进行故障转移, 那么整个资源组在另一个节点上启动。

组中的资源之间存在下列依赖性:

- 必须先启动本地磁盘, 然后再启动 DB2 实例
- 必须先停止 DB2 实例, 然后再停止本地磁盘
- 必须将 HA IP 地址与实例一起使用

磁盘存储器

DB2 数据库可以使用下列资源作为本地数据存储器:

- 原始磁盘 (例如, /dev/sda1)
- 逻辑卷管理器 (LVM) 管理的逻辑卷
- 文件系统 (例如, ext3 和 jfs)

DB2 数据可以全部存储在一个或多个原始磁盘上, 全部存储在逻辑卷上, 全部存储在文件系统上或同时存储在上述三种资源上。任何可执行文件都需要驻留在某种类型的文件系统上。

DB2 数据库对 HA IP 地址的要求

DB2 数据库对于 IP 地址没有任何特殊要求。不必为了使实例具有高可用性而定义高可用的 IP 地址。但是, 一定要记住, 受保护的 IP 地址 (如果有) 是数据的客户机访问点, 因此所有客户机都必须知道此地址。实际上, 建议将客户机在它们的 CATALOG TCPIP NODE 命令中使用的那个 IP 地址作为此 IP 地址。

Tivoli SAM 资源组

IBM Tivoli System Automation for Multiplatforms 是一个产品, 它通过自动控制基于 Linux 的集群中诸如进程、应用程序和 IP 地址之类的资源和其他资源来提供高可用性。要自动控制 IT 资源 (如 IP 地址), 必须对 Tivoli SAM 定义此资源。此外, 这些资源还必须包含在至少一个资源组中。如果总是要求在同一台机器上主管这些资源, 那么应将它们放置在相同的资源组中。

需要将每个应用程序都定义为资源, 以便通过 Tivoli SAM 进行管理和自动控制。应用程序资源通常在一般资源类 IBM.Application 中定义。在此资源类中, 有几个定义资源的属性, 但其中至少三个属性是特定于应用程序的:

- StartCommand
- StopCommand
- MonitorCommand

这些命令可以是脚本或二进制可执行文件。

设置 DB2 环境的 Tivoli SAM

有关可帮助您设置 Tivoli SAM 以使用 DB2 环境的详细配置信息, 请在 IBM 软件库 Web 站点 (<http://www.ibm.com/software/sw-library/>) 中搜索“Tivoli System Automation”。

Microsoft 故障转移集群支持 (Windows)

Microsoft 故障转移集群支持 Windows 操作系统上的服务器的集群。它自动检测并响应服务器或应用程序故障, 并且可以平衡服务器工作负载。

简介

Microsoft 故障转移集群是 Windows 服务器操作系统的功能。此软件支持集群中两个服务器的连接 (在 DataCenter Server 中最多为 4 个服务器) 以获取数据和应用程序的高

可用性和易于管理。故障转移集群还可自动检测服务器或应用程序故障并进行恢复。它可用来转移服务器工作负载以平衡机器利用率，以及在无需停机的情况下执行规划的维护工作。

下列 DB2 产品支持故障转移集群：

- DB2 工作组服务器版
- DB2 企业服务器版 (DB2 ESE)
- DB2 Connect 企业版 (DB2 CEE)

DB2 故障转移集群组件

集群是两个或多个节点的配置，其中每个节点都是独立的计算机系统。集群对网络客户机而言就好像是单个服务器。

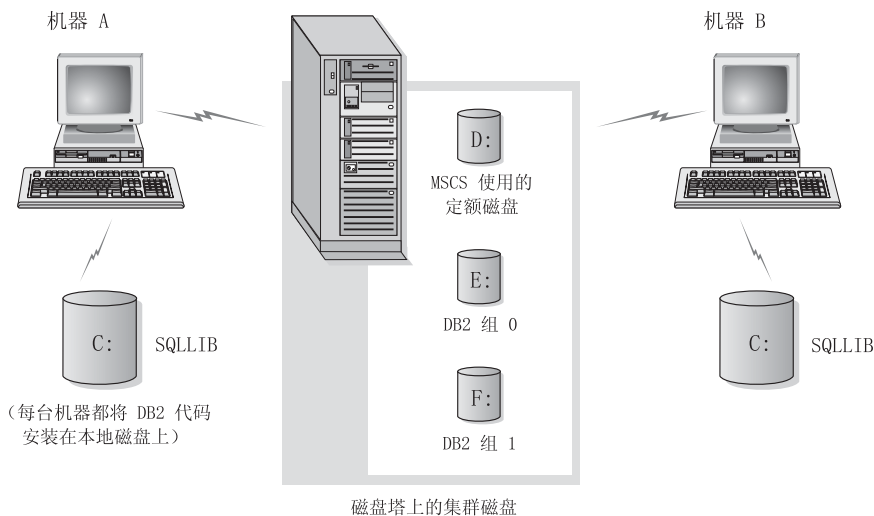


图 3. 故障转移集群配置示例

进行故障转移集群的集群中的节点是通过使用一个或多个共享存储器总线和一个或多个物理上独立的网络来连接的。仅将服务器与集群连接而未将客户机与集群连接的网络称为专用网络。支持客户机连接的网络称为公共网络。每个节点上有一个或多个本地磁盘。每个共享存储器总线与一个或多个磁盘连接。共享总线上的每个磁盘一次只能由集群的一个节点拥有。DB2 软件位于本地磁盘上。DB2 数据库文件（例如，表、索引和日志文件）位于共享磁盘上。因为故障转移集群不支持在集群中使用原始分区，所以无法在故障转移集群环境中将 DB2 配置为使用原始设备。

DB2 资源

在故障转移集群环境中，资源是由集群软件管理的实体。例如，磁盘、IP 地址或类属服务可以作为资源来管理。DB2 通过创建它自己的资源类型（称为“DB2 服务器”）来与故障转移集群集成。每个“DB2 服务器”资源管理一个 DB2 实例，并且在分区数据库环境中运行时，每个“DB2 服务器”资源管理一个数据库分区。“DB2 服务器”资源的名称就是实例名，虽然在分区数据库环境中，“DB2 服务器”资源的名称由实例名和数据库分区（或节点）号组成。

联机前和联机后脚本

在将 DB2 资源联机前和联机后都可运行脚本。这些脚本分别被称为联机前和联机后脚本。联机前和联机后脚本是可以运行 DB2 和系统命令的 .BAT 文件。

在 DB2 的多个实例可以在同一机器上运行的情况下，可以使用联机前和联机后脚本来调整配置，以便可以成功启动实例。即使发生故障转移，也可以使用联机后脚本来执行手动数据库恢复。联机后脚本还可用来启动从属于 DB2 的任何应用程序或服务。

DB2 组

相关或从属资源被组织成资源组。组中的所有资源作为一个单元在集群节点之间移动。例如，在典型 DB2 单一分区集群环境中，将存在包含下列资源的 DB2 组：

1. DB2 资源。DB2 资源管理 DB2 实例（或节点）。
2. IP 地址资源。IP 地址资源允许客户机应用程序与 DB2 服务器连接。
3. “网络名”资源。“网络名”资源允许客户机应用程序通过使用名称而不使用 IP 地址与 DB2 服务器连接。“网络名”资源具有与 IP 地址资源的依赖性。“网络名”资源是可选的。（配置“网络名”资源可能影响故障转移性能。）
4. 一个或多个“物理磁盘”资源。每个“物理磁盘”资源管理集群中的一个共享磁盘。

注： DB2 资源的配置取决于同一组中的所有其他资源，因此仅当所有其他资源联机之后才能启动 DB2 服务器。

故障转移配置

可使用两种类型的配置：

- 热备用
- 相互接管

在一个分区数据库环境中，集群不必全部具有相同类型的配置。您可将一些集群设置为使用热备用，而将其他集群设置为使用相互接管。例如，如果您的 DB2 实例由五个工作站组成，可以将两台机器设置为使用相互接管配置，将另外两台机器设置为使用热备用配置，而将剩余的一台机器配置为不支持故障转移。

热备用配置

在热备用配置中，故障转移集群中的一台机器提供专用的故障转移支持，而另一台机器则参与数据库系统。如果参与该数据库系统的机器发生故障，那么该机器上的数据库服务器将在故障转移机器上启动。如果在一个分区数据库环境中，您正在一台机器上运行多逻辑节点，而该机器发生故障，那么这些逻辑节点将在故障转移机器上启动。第 107 页的图 4 显示了一个热备用配置的示例。

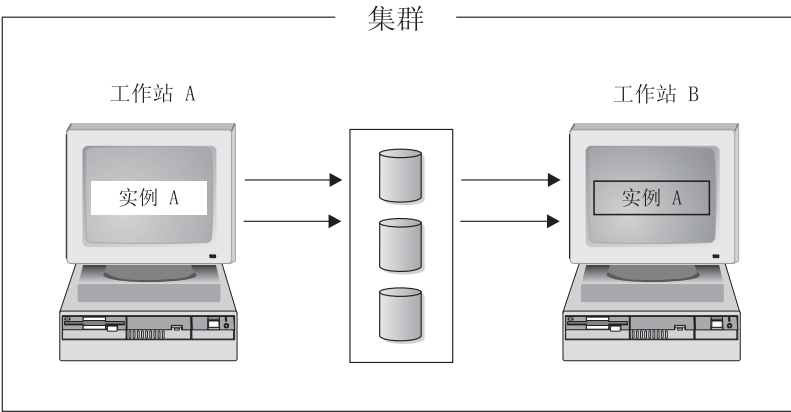


图 4. 热备用配置

相互接管配置

在一个相互接管配置中，两个工作站都参与数据库系统（即，每台机器上至少有一个数据库服务器在运行）。如果故障转移集群中的一个工作站发生故障，那么发生故障的机器上的数据库服务器将在另一台机器上启动以运行。在一个相互接管配置中，一台机器上的数据库服务器发生故障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处于活动状态。图 5 显示了一个相互接管配置的示例。

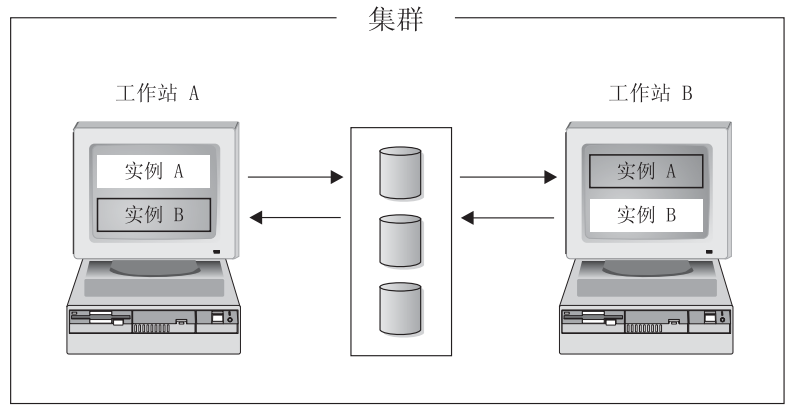


图 5. 相互接管配置

有关在 Windows 操作系统上高可用的 IBM DB2 数据库环境的实现和设计的详细信息，请参阅 IBM 软件库 Web 站点 (<http://www.ibm.com/software/sw-library/>)。

Solaris 操作系统集群支持

DB2 支持下列两个可用于 Solaris 操作系统的集群管理器: Sun Cluster 和 Veritas Cluster Server (VCS)。

有关 Sun Cluster 的信息, 请参阅标题为“DB2 Universal Database and High Availability on Sun Cluster 3.X”的白皮书, 可以从 IBM 软件库 Web 站点 (<http://www.ibm.com/software/sw-library/>) 获得该白皮书。

注: 当使用 Sun Cluster 3.0 或 Veritas Cluster Server 时, 确保在引导时通过使用 db2iauto 实用程序不启动 DB2 实例, 如下所示:

```
db2iauto -off InstName
```

其中, *InstName* 是实例的登录名。

高可用性

提供数据服务的计算机系统包含许多不同的组件, 每个组件都有一个关联的“平均无故障时间”(MTBF)。MTBF 是组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)。虽然这看起来像是很长一段长时间, 但每 200 个磁盘中, 就有一个有可能在 6 个月内发生故障。

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 集群。当用于高可用性时, 集群由两台或更多机器、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集群中的另一机器, 应该能够继续提供对其数据的访问。将数据服务从一台机器移至另一机器称为故障转移, 图 6 对其进行了说明。

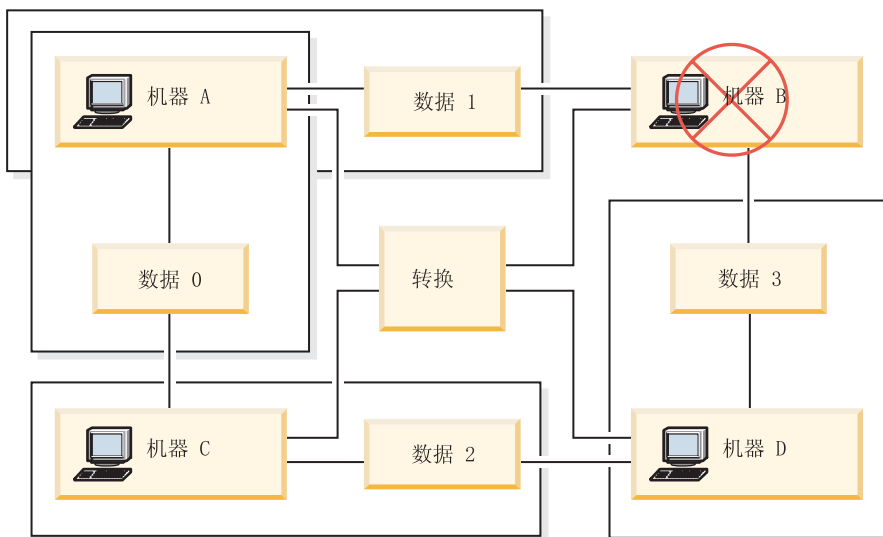


图 6. 故障转移. 当机器 B 故障时, 它的数据服务被移到集群中的另一台机器上, 以便仍可访问这些数据。

专用网络接口用来在集群中的机器之间发送脉动信号消息和控制消息。公用网络接口用来直接与 HA 集群的客户机通信。HA 集群中的磁盘与集群中的两台或更多机器相连, 因此, 当一台机器发生故障时, 另一机器可以对它们进行访问。

在 HA 集群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。如果执行故障转移，那么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 集群的其中一个好处是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。集群中的所有部件都应该是冗余的，包括机器本身。集群应当能够经受住任何单个故障点。

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要求。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，它正在访问高可用的 Web 服务器。请求是使用 URL（统一资源定位符）发出的，该 URL 包含主机名和指向 Web 服务器上的文件的路径。浏览器期望该主机名和路径在 Web 服务器的故障转移之后保持不变。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转移，那么该浏览器将需要重新发出请求。

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 集群时：

1. 确保集群管理员熟悉系统并了解发生故障转移时应发生的情况。
2. 确保集群的每一部分都确实是冗余的，当它发生故障时，可以被快速替换。
3. 强制一个测试系统在受控环境中发生故障，并确保它每次都能正确地进行故障转移。
4. 跟踪每次故障转移的原因。虽然这应该不会经常发生，但找出任何导致集群不稳定的问题非常重要。例如，如果集群的一段一个月内导致了 5 次故障转移，那么找出原因并将它解决。
5. 当发生故障转移时，务必通知该集群的支持人员。
6. 不能使集群超负荷。确保其余的系统在发生故障转移之后，仍能够处理适量的工作负载。
7. 经常检查容易引起故障的组件（如磁盘），以便可以在发生问题之前进行替换。

故障容错

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 集群还有一个优点，即可以从仅影响那些位置的一部分的灾难中恢复。

由于 HA 集群可伸缩、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 3.0（和更高版本）支持：

如果您打算在 Solaris 操作系统集群上运行 DB2 数据库解决方案，那么可以使用 Sun Cluster 3.0 来管理该集群。高可用性代理程序充当 DB2 数据库与 Sun Cluster 3.0 之间的介体。

在关于 Sun Cluster 3.0 支持的此主题中所做的声明也适用于 3.0 以上更高版本的 Sun Cluster。



图 7. DB2 数据库、Sun Cluster 3.0 和高可用性. DB2 数据库、Sun Cluster 3.0 与高可用性代理程序之间的关系。

故障转移

Sun Cluster 3.0 通过启用应用程序故障转移来提供高可用性。定期监视每个节点，集群软件自动将集群感知应用程序从失败的主节点重定位至指定的辅助节点。当故障转移发生时，客户机可能遇到短时间中断服务的情况，可能必须重新连接至服务器。但是，它们将不知道它们正从其中访问应用程序和数据的物理服务器。通过允许主节点故障时由集群中的其他节点自动主管工作负载，Sun Cluster 3.0 可以显著地减少停机时间并提高工作效率。

多主机磁盘

Sun Cluster 3.0 需要多主机磁盘存储器。这表示磁盘一次可与多个节点连接。在 Sun Cluster 3.0 环境中，多主机存储器能使磁盘机变得高可用。位于多主机存储器上的磁盘机允许单个节点故障，因为通过备用服务器节点，仍然有一条物理路径指向数据。通过主节点可以全局访问多主机磁盘。如果客户机请求正通过一个节点来访问数据，且该节点故障，那么将这些请求转换至具有与相同磁盘的直接连接的另一个节点。卷管理器为多主机磁盘的数据冗余提供镜像或 RAID 5 配置。通常，Sun Cluster 3.0 支持 Solstice DiskSuite 和 VERITAS Volume Manager 作为卷管理器。将多主机磁盘与磁盘镜像和组合分割区组合在一起防止节点故障和各个磁盘故障。

全局设备

使用全局设备，可以提供从任何节点对集群范围内的集群中的任何设备的高可用访问，而不必考虑设备的物理位置。所有磁盘都包括在带有指定设备标识 (DID) 的全局名称空间中，并且被配置为全局设备。因此，从所有集群节点上看，磁盘本身是可视的。

文件系统/全局文件系统

集群或全局文件系统是内核（在一个节点上）与底层文件系统卷管理器（在具有与一个或多个磁盘的物理连接的节点上）之间的代理。集群文件系统依赖于具有与一个或多个节点的物理连接的全局设备。它们与底层文件系统和卷管理器无关。通常，可以通过使用 Solstice DiskSuite 或 VERITAS Volume Manager 在 UFS 上构建集群文件系统。仅当磁盘上的文件系统整体上作为集群文件系统装上时，数据才变得对所有节点都可用。

设备组

所有多主机磁盘都必须受控于 Sun Cluster 框架。由 Solstice DiskSuite 或 VERITAS Volume Manager 管理的磁盘组首先是在多主机磁盘上创建的。然后，将它们注册为 Sun Cluster 磁盘设备组。磁盘设备组是一种类型的全局设备。多主机设备组是高可用的。如果当前管理设备组的节点失败，那么可以通过备用通道访问磁盘。管理设备组的节点的故障不影响对设备组的访问，但执行恢复和一致性校验所需的时间除外。在此时间段，所有请求停止（对应用程序是透明的），直到系统使设备组可用为止。

资源组管理器（RGM）

RGM 提供高可用性的机制并在每个集群节点上作为守护程序运行。它根据预先配置策略在选择的节点上自动启动和停止资源。即使在发生节点故障的情况下，RGM 允许资源高可用，或通过在受影响的节点上停止并在另一个节点上启动它来重新引导。RGM 还会自动启动和停止特定于资源的监视器，以便检测资源故障并将失败的资源重定位至另一个节点。

数据服务

术语“数据服务”用来描述已配置为在集群而不是单个服务器上运行的第三方应用程序。数据服务包括应用软件和启动、停止以及监视该应用程序的 Sun Cluster 3.0 软件。Sun Cluster 3.0 提供用来控制和监视集群内的应用程序的数据服务方法。这些方法在“资源组管理器”（RGM）的控制下运行，它使用这些方法来启动、停止和监视集群节点上的应用程序。这些方法与集群框架软件和多主机磁盘一起，使应用程序成为高可用的数据服务。作为高可用的数据服务，它们可以在集群内任何单个故障后防止重要的应用程序中断，而不考虑故障是在节点上、接口组件上还是在应用程序本身中发生的。RGM 还管理集群中的资源，包括网络资源（逻辑主机名和共享地址）和应用程序实例。

资源类型、资源和资源组

资源类型是由下列部分组成：

1. 要在集群上运行的软件应用程序。
2. RGM 用作回调方法以管理作为集群资源的应用程序的控制程序。
3. 组成集群的静态配置的部分的一组属性。

RGM 使用资源类型属性来管理特定类型的资源。

资源继承其资源类型的属性和值。这是在集群上运行的基础应用程序的实例。每个实例在集群内都需要有唯一的名称。每个资源都必须在资源组中配置。RGM 将同一节点上的某个组的所有资源同时联机 and 脱机。当 RGM 将资源组联机或脱机时，它对该组中的各个资源调用回调方法。

资源组当前在其上联机的节点被其主节点或主体调用。资源组由它的每一个主体管理。每个资源组都有相关联的 Nodelist 属性，由集群管理员设置，以指定资源组的所有潜在主体或主方。

有关 Sun Cluster 平台上高可用的 IBM DB2 数据库环境的实现和设计的详细信息，请参阅标题为“DB2 and High Availability on Sun Cluster 3.X”的白皮书，可以在 IBM 软件库 Web 站点 (<http://www.ibm.com/software/sw-library/>) 上获得该白皮书。

VERITAS Cluster Server 支持:

如果您打算在 Solaris 操作系统集群上运行 DB2 数据库解决方案，那么可以使用 VERITAS Cluster Server 来管理该集群。VERITAS Cluster Server 可以管理异种环境中大量应用程序；并且在存储区域网络 (SAN) 和传统客户机/服务器环境中，它最多支持 32 个节点集群。

硬件要求

以下是当前受 VERITAS Cluster Server 支持的硬件的列表:

- 对于服务器节点:
 - Sun Microsystems 提供的运行 Solaris 2.6 或更新版本的任何 SPARC/Solaris 服务器，RAM 不得低于 128 MB。
- 对于磁盘存储器:
 - EMC Symmetrix、IBM Enterprise Storage Server[®]、HDS 7700 和 9xxx、Sun T3、Sun A5000、Sun A1000、Sun D1000 以及 VCS 2.0 或更新版本支持的任何其他磁盘存储器；VERITAS 代表可以确认哪些磁盘子系统是受支持的，您也可以参阅 VCS 文档。
 - 典型环境将需要镜像专用磁盘（在每个集群节点中）以用于 DB2 二进制文件和 DB2 数据的节点之间的共享磁盘。
- 对于网络互连:
 - 对于公用网络连接，支持基于 IP 寻址的任何网络连接。
 - 对于脉动信号连接（集群内部），冗余脉动信号连接是必需的；这种要求可通过每个服务器使用两个附加以太网控制器或者每个服务器使用一个附加以太网控制器且每个集群使用一个共享 GABdisk 来满足

软件要求

下列 VERITAS 软件组件是限定配置:

- VERITAS Volume Manager 3.2 或更新版本、VERITAS File System 3.4 或更新版本以及 VERITAS Cluster Server 2.0 或更新版本。
- DB Edition for DB2 Solaris 版 1.0 或更新版本。

当 VERITAS Cluster Server 不需要卷管理器时，强烈建议使用 VERITAS Volume Manager 来轻松安装、配置和管理。

故障转移

VERITAS Cluster Server 是一个可用性集群解决方案，它通过启用应用程序故障转移来管理应用程序服务（如 DB2 数据库）的可用性。会定期监视每个单独的集群节点的状态及其相关联的软件服务通常。当发生中断应用程序服务（在此情况下是 DB2 数据库

服务)的故障时, VERITAS Cluster Server 和/或 VCS HA-DB2 代理程序将检测到该故障并自动执行一些步骤来复原该服务。用来复原该服务的步骤包括在同一节点上重新启动 DB2 数据库系统, 或者将 DB2 数据库系统移至集群中的另一个节点并在该节点上将其重新启动。如果需要将应用程序迁移到新节点, 那么 VERITAS Cluster Server 将与该应用程序相关联的一切(即, 网络 IP 地址和基础存储器的所有权)移至新节点, 以使用户不会意识到实际上服务在另一个节点上运行。他们将仍然通过使用相同的 IP 地址来访问该服务, 但是这些地址现在将指向另一个集群节点。

当 VERITAS Cluster Server 发生故障转移时, 用户可能看到也可能看不到服务中断。这将取决于客户机与应用程序服务的连接的类型(有状态或无状态)。在带有有状态连接的应用程序环境(如 DB2 数据库)中, 用户可能会发现服务短时间中断, 并且在故障转移完成之后可能需要重新连接。在带有无状态连接的应用程序环境(如 NFS)中, 用户可能发现服务的短时间延迟, 但通常将不会发现中断并且不需要再次登录。

通过支持应用程序作为可以在集群节点之间自动迁移的服务, VERITAS Cluster Server 不仅可以减少未计划的停机时间, 而且还可以缩短与计划停机时间(维护和升级)相关联的运行中断持续时间。还可以手动启动故障转移。如果硬件或操作系统升级必须在特定节点上执行, 那么 DB2 数据库系统可以迁移至集群中的另一个节点, 并且可以执行更新, 然后将 DB2 数据库系统迁移回原始节点。

建议在这些类型的集群环境中使用的应用程序应允许崩溃。允许崩溃的应用程序在仍然保持落实数据的完整性的同时可以从意外崩溃中恢复。允许崩溃的应用程序有时被称为集群友好应用程序。DB2 数据库系统是允许崩溃的应用程序。

有关如何减少使用 VERITAS CFS、CVM 和 VCS 解决方案来执行故障转移所需的时间的信息, 请参阅标题为“DB2 Universal Database Version 8 and VERITAS Database Edition/HA for DB2”的白皮书, 可以从 IBM 软件库 Web 站点(<http://www.ibm.com/software/sw-library/>)获得该白皮书。

共享存储器

与“VCS HA-DB2 代理程序”配合使用时, Veritas Cluster Server 需要共享存储器。共享存储器是具有与集群中的多个节点的物理连接的存储器。位于共享存储器上的磁盘机允许节点故障, 因为通过一个或多个备用集群节点, 指向磁盘机的物理路径仍然存在。

在 VERITAS Cluster Server 的控制下, 集群节点可以通过称为“磁盘组”的逻辑构造来访问共享存储器。磁盘组表示逻辑定义的存储设备的集合, 这些存储设备的所有权是可在集群中的节点之间自动迁移。在给定时间, 只能将磁盘组导入至单个节点。例如, 如果“磁盘组 A”导入至“节点 1”并且“节点 1”发生故障, 那么可以从故障节点导出“磁盘组 A”, 并将它导入至集群中的另一节点。VERITAS Cluster Server 可以同时控制单个集群中的多个磁盘组。

除了允许磁盘组定义之外, 卷管理器还可以通过使用镜像或 RAID 5 在共享存储器上提供冗余数据配置。VERITAS Cluster Server 支持 VERITAS Volume Manager 和 Solstice DiskSuite 作为逻辑卷管理器。将共享存储器与磁盘镜像和分割组合区组合在一起以防止节点故障和各个磁盘或控制器故障。

VERITAS Cluster Server 全局原子广播 (GAB) 和低等待时间传输 (LLT)

集群配置中需要节点间的通信机制，以便节点可以根据硬件和软件状态交换信息，记录集群成员信息并使所有集群节点上的此信息同步。通过低等待时间传输（LLT）运行的“全局原子广播”（GAB）工具提供由 VERITAS Cluster Server 使用的高速、低等待时间机制来完成此任务。GAB 被作为内核模块装入到每个集群节点上，并提供原子广播机制，以确保所有节点同时获取状态更新信息。

通过利用内核间通信功能，LLT 对需要在集群节点间交换和同步的所有信息提供高速和低等待时间传输。GAB 在 LLT 顶部运行。VERITAS Cluster Server 不将 IP 用作脉动信号机制，但提供两个其他可靠选项。带有 LLT 的 GAB 可配置为充当脉动信号机制，或者可将 GABdisk 配置为基于磁盘的脉动信号。脉动信号必须通过冗余连接运行。这些连接可以是集群节点间两个专用以太网连接或一个专用以太网连接和一个 GABdisk 连接。使用两个 GABdisks 的配置不受支持，因为节点间集群状态的交换需要专用以太网连接。

有关 GAB 或 LLT 或有关如何在 VERITAS Cluster Server 配置中配置它们的更多信息，请参阅 VERITAS Cluster Server 2.0 User's Guide for Solaris。

捆绑和企业代理程序

代理程序是用来管理特定资源或应用程序的可用性的程序。当启动代理程序时，它从 VCS 获取必要的配置信息，然后定期监视资源或应用程序并用状态来更新 VCS。通常，使用代理程序来使资源联机、使资源脱机、或监视资源，同时提供四种类型的服务：启动、停止、监视和清除。启动和停止用来使资源联机或脱机，监视用来测试特定资源或应用程序以了解它的状态，而在恢复过程中使用清除。

将各种捆绑代理程序包括为 VERITAS Cluster Server 的一部分，并在安装 VERITAS Cluster Server 时安装这些代理程序。捆绑代理程序是这样的 VCS 进程，它们管理通常在集群配置中发现的预定义资源类型（即，IP、安装、处理和共享），并且它们有助于极大地简化集群安装和配置。超过 20 个捆绑代理程序与 VERITAS Cluster Server 捆绑在一起。

企业代理程序想要将重点放在特定应用程序（例如，DB2 数据库应用程序）上。“VCS HA-DB2 代理程序”可视作“企业代理程序”，它通过 VCS Agent 框架与 VCS 交互。

VCS 资源、资源类型和资源组

资源类型是用来定义 VCS 集群中将要监视的资源对象定义。资源类型包括资源类型名和一组与该资源相关联的属性，这些属性从高可用性方面来看是非常重要的。资源继承其资源类型的属性和值，且资源名称在集群范围内必须是唯一的。

有两种类型的资源：持久和标准（非持久）。持久资源是如网络接口控制器（NIC）之类的资源，它们受监视但未被 VCS 联机或脱机。标准资源是其联机和脱机状态受 VCS 控制的资源。

受监视的最低级别对象是资源，有各种资源类型（即，共享和安装）。每个资源必须配置到资源组中，且 VCS 将使特定资源组中的所有资源同时联机和脱机。要使资源组联机或脱机，VCS 将调用启动或停止方法以用于组中每个资源。有两种类型的资源组：故障转移和并行。高可用的 DB2 数据库配置，不论它是否为分区数据库环境，都将使用故障转移资源组。

“主体”或“主方”节点是可以潜在主管资源的节点。名为 `systemlist` 的资源组属性用来指定集群中哪些节点可以是特定资源组中的主体。在双节点集群中，通常两个节点都包括在 `systemlist` 中，但是在主管几个高可用的应用程序的较大节点集群中，可能需要确保某些应用程序服务（由它们的资源在最低级别定义）永远不可能故障转移至某些节点。

可在资源组之间定义依赖性，在评估各种资源故障和管理恢复时，VERITAS Cluster Server 从属于此资源组依赖性层次结构。例如，如果不能使资源组 `ClientApp1` 联机（除非资源组 `DB2` 已成功启动），资源组 `ClientApp1` 将被视作从属于资源组 `DB2`。

有关带有 VERITAS Cluster Server 的高可用 IBM DB2 数据库环境的实现和设计的详细信息，请参阅标题为“DB2 UDB and High Availability with VERITAS Cluster Server”的技术手册，可以从 <http://www.ibm.com/support/docview.wss?rs=71&uid;=swg21045033> 获得该技术手册。

使分区数据库环境中的时钟同步

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差加上事务的任何潜在操作和通信延迟，应小于对 `max_time_diff`（节点之间的最大时差）数据库管理器配置参数指定的值。

为确保日志记录时间戳记反映分区数据库环境中的事务顺序，DB2 使用每台机器上的系统时钟以及文件 `SQLLOGCTL.LFH` 中存储的虚拟时间戳记作为日志记录时间戳记的基准。但是，如果将系统时钟设置得提前，就会自动将日志时钟设置得提前。虽然可以将系统时钟往后调，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匹配为止。于是，这两个时钟便同步了。这意味着一个数据库节点上的短期系统时钟错误可能会对数据库日志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 2005 年 11 月 7 日，而当前年份是 2003 年，并假定在该数据库分区服务器上的数据库分区中落实了更新事务之后更正了此错误。如果继续使用该数据库，并且随着时间的推移定期对其进行更新，那么 2003 年 11 月 7 日至 2005 年 11 月 7 日之间的任何时间点实际上是无法通过前滚恢复到达的。当数据库分区服务器 A 上的 `COMMIT` 完成时，数据库日志中的时间戳记被设置为 2005，而日志时钟会停留在 2005 年 11 月 7 日，直到系统时钟与此时间相匹配为止。如果尝试前滚到这个时间范围内的某个时间点，那么操作将在指定停止点后的第一个时间戳记（即 2003 年 11 月 7 日）处停止。

虽然 DB2 无法控制对系统时钟的更新，但是 `max_time_diff` 数据库管理器配置参数减少了发生此类问题的机会：

- 此参数的可配置值的范围是 1 分钟至 24 小时。
- 当对非目录分区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分区。该目录分区就会检查请求连接的数据库分区上的时间与它自己的时间是否在 `max_time_diff` 参数指定的范围之内。如果超出此范围，那么拒绝该连接。
- 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。如果两个或更多个数据库分区服务器的时差超出 `max_time_diff` 允许的限制，那么会回滚该事务，以防止将不正确的时间传播至其他数据库分区服务器。

客户机/服务器时间戳记转换

本节说明客户机/服务器环境中时间戳记的生成:

- 如果对前滚操作指定了本地时间, 那么返回的所有消息也以本地时间表示。

注: 在服务器上和 (在分区数据库环境中) 目录数据库分区上所有时间都会转换。

- 在服务器上时间戳记字符串转换为 GMT, 所以时间表示服务器的时区而不是客户机的时区。如果客户机与服务器在不同的时区, 那么应该使用服务器的本地时间。
- 如果时间戳记字符串接近由夏令时导致的时间更改, 那么了解停止时间是在时间更改之前还是之后是很重要的, 这样才能正确地将它指定。

第 5 章 管理和维护高可用性解决方案

一旦创建、配置和开始运行 DB2 数据库高可用性解决方案后，您必须执行一些后续活动。您必须监视、维护和修复数据库解决方案，使它保持对客户机应用程序可用。

当数据库系统运行时，需要监视和响应下列各种情况：

1. 管理日志文件。

日志文件越来越大，需要归档；有些日志文件需要复制和移动，以便可用于复原操作。

2. 执行维护活动：

- 安装软件
- 升级硬件
- 重组数据库表
- 数据库性能调整
- 数据库备份

3. 同步主数据库和辅助或备用数据库，以便故障转移平稳进行。

4. 标识和响应意外的硬件或软件故障。

日志文件管理

DB2 数据库管理器使用编号方案来命名日志文件。此命名策略会影响日志文件复用和日志序列。此外，没有客户机应用程序连接的 DB2 数据库在下一个客户机应用程序连接至该数据库服务器时会使用新的日志文件。DB2 Data Server 数据库日志记录行为的这两个方面，对于您在日志文件管理上进行选择会有影响。

管理数据库日志时，请考虑以下事项：

- 归档日志的编号方案以 S0000000.LOG 开始，直到 S9999999.LOG，符合日志文件的最大潜在大小 10000000。如果发生以下情况，数据库管理器将复位到 S0000000.LOG：
 - 数据库配置文件更改为启用前滚恢复
 - 数据库配置文件更改为禁用前滚恢复
 - 已使用了 S9999999.LOG。

在复原数据库之后（执行或未执行前滚恢复），DB2 数据库管理器复用日志文件名。数据库管理器会确保在前滚恢复期间不会应用错误的日志。在复原操作后，如果 DB2 数据库管理器复用日志文件名，那么将新的日志文件归档到独立的目录，以便可以归档多个同名的日志文件。日志文件的位置记录在恢复历史记录文件中，以便在前滚恢复期间应用这些位置记录。必须确保前滚恢复可以找到正确的日志。

前滚操作成功完成后，使用的最后一个日志会截断，而日志记录会从下一个按顺序的日志开始。日志路径目录中，序号大于用于前滚恢复的最后一个日志的任何日志

都将复用。截断的日志中，跟在截断点后的任何条目都将用 0 覆盖。确保在调用 Rollforward 实用程序前建立了日志的副本。（可以调用用户出口程序将日志复制到另一位置。）

- 如果某个数据库尚未激活（通过 ACTIVATE DATABASE 命令），那么 DB2 数据库管理器会在所有应用程序已从该数据库断开连接后截断当前日志文件。下次有应用程序与该数据库连接时，DB2 数据库管理器开始将日志记录到一个新日志文件中。如果系统上产生了很多小的日志文件，那么可能需要考虑使用 ACTIVATE DATABASE 命令。使用该命令不仅能节省应用程序连接时初始化数据库所用的开销，还可节省分配大日志文件、截断它然后再分配新的大日志文件所用的开销。
- 因为日志文件名是重复使用的（请参阅图 8），所以归档日志可能会与某个数据库的两个或多个不同的日志序列相关。例如，如果您要恢复“备份 2”，可以使用两种可能的日志序列。如果在完整的数据库恢复期间前滚至某个时间点，然后在到达日志末尾前停止，那么您已创建了一个新的日志序列。两个日志序列不能合在一起。如果联机备份映像跨越了第一个日志序列，那么必须使用此日志序列来完成前滚恢复。

如果在恢复之后创建了新的日志序列，那么旧日志序列中的所有表空间备份映像都是无效的。这通常是在复原时识别的，但是如果在数据库复原操作之后立即执行表空间复原操作，那么 RESTORE 实用程序无法识别旧日志序列上的表空间备份映像。在实际上前滚了数据库前，将使用的日志序列都是未知的。如果表空间在旧日志序列上，它一定会由表空间前滚操作“捕获”。使用无效备份映像的复原操作可能会成功完成，但该表空间的表空间前滚操作将失败，而表空间将仍处于复原暂挂状态。

例如，假设一个表空间级的备份操作“备份 3”在顶级日志序列的 S0000013.LOG 和 S0000014.LOG 之间完成（请参阅图 8）。如果要使用数据库级别的备份映像“备份 2”进行复原和前滚，您将需要前滚 S0000012.LOG。然后，可以继续前滚到顶端日志序列或（较新的）底端日志序列。如果前滚底端日志序列，将不能使用表空间级别的备份映像“备份 3”来执行表空间复原和前滚恢复。

要使用表空间级别的备份映像“备份 3”完成至日志末尾的表空间前滚操作，需要复原数据库级别的备份映像“备份 2”，然后使用顶端日志序列前滚。一旦复原了表空间级的备份映像“备份 3”，就可以启动至日志末尾的前滚操作。

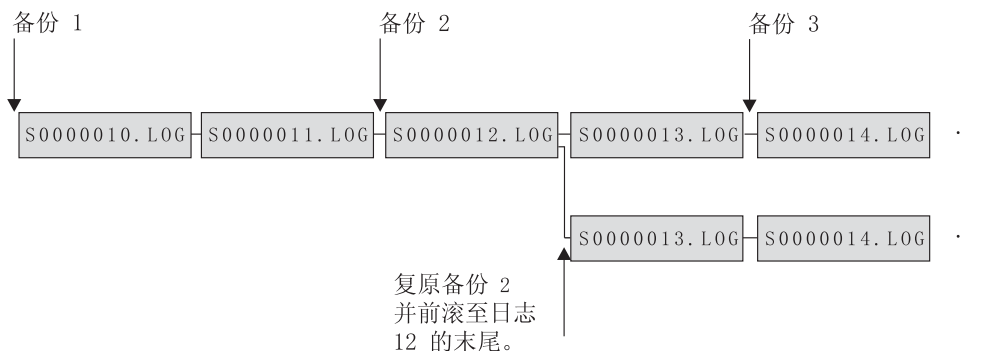


图 8. 复用日志文件名

按需进行的日志归档

IBM 数据服务器支持随时关闭（如果启用，还包括归档）可恢复数据库的活动日志。因此您可以到某个已知点为止的一组完整的日志文件，然后使用这些日志文件来更新一个备用数据库。

通过调用 ARCHIVE LOG 命令或通过调用 db2ArchiveLog API，可以启动按需进行日志归档操作。

使用 db2tapemgr 来进行日志归档

可以使用 db2tapemgr 实用程序将已归档的日志文件存储到磁带设备。db2tapemgr 实用程序将日志文件从磁盘复制到指定的磁带设备，并使用已复制的日志文件的新位置更新恢复历史记录文件。

配置

将数据库配置参数 LOGARCHMETH1 设置为要复制到磁带的日志文件在磁盘上的位置。db2tapemgr 实用程序读取此 LOGARCHMETH1 值以找到要复制的日志文件。在分区数据库环境中，必须在包含要复制的日志文件的每个数据库分区上设置 LOGARCHMETH1 配置参数。

db2tapemgr 实用程序不使用 LOGARCHMETH2 数据库配置参数。

STORE 和 DOUBLE STORE 选项

发出指定了 STORE 或 DOUBLE STORE 选项的 DB2TAPEMGR 命令将归档的日志从磁盘传送到磁带。

- STORE 选项将某个范围的日志文件或所有日志文件从日志归档目录存储到指定的磁带设备，并从磁盘中删除那些文件。
- DOUBLE STORE 选项扫描历史记录文件以查看先前是否已将那些日志存储到磁带。
 - 如果先前从未存储过某个日志，DB2TAPEMGR 就会将该日志文件存储到磁带，但不从磁盘中删除该文件。
 - 如果先前已存储过某个日志，DB2TAPEMGR 就会将该日志文件存储到磁带并从磁盘中删除该文件。

如果要在磁带和磁盘上都保存归档日志的副本，或者要将相同的日志存储在两盘不同的磁带上，请使用 DOUBLE STORE。

当发出指定了 STORE 或 DOUBLE STORE 选项的 DB2TAPEMGR 命令时，db2tapemgr 实用程序首先扫描历史记录文件以查找 LOGARCHMETH1 配置参数设置为磁盘的条目。如果它未找到原本应该在磁盘上的文件，将发出警告。如果 db2tapemgr 实用程序找不到要存储的日志文件，它将停止操作并发出消息以通知您没有要执行的操作。

RETRIEVE 选项

发出指定了 RETRIEVE 选项的 DB2TAPEMGR 命令将文件从磁带传送到磁盘。

- 使用 RETRIEVE ALL LOGS 或 LOGS n TO n 选项来检索所有符合指定条件的归档日志，并将它们复制到磁盘。
- 使用 RETRIEVE FOR ROLLFORWARD TO POINT-IN-TIME 选项来检索执行前滚操作所需的所有归档日志，并将它们复制到磁盘。

- 使用 RETRIEVE HISTORY FILE 选项来从磁带检索历史记录文件并将其复制到磁盘。

行为

•

如果 db2tapemgr 实用程序在磁盘上找到日志文件，它就会读取磁带头以确保可以将日志文件写入磁带。它还将更新那些当前在磁带上的文件的历史记录。如果更新失败，操作将停止，并且将显示错误消息。

•

如果磁带是可写的，db2tapemgr 实用程序就会将日志复制到磁带。复制文件后，将从磁盘中删除日志文件。最后，db2tapemgr 实用程序将历史记录文件复制到磁带并从磁盘中删除该文件。

•

db2tapemgr 实用程序不会将日志文件追加到磁带。如果存储操作未写满整盘磁带，未使用的空间就会浪费掉。

•

对于任何给定磁带，db2tapemgr 实用程序只能在该磁带上存储一次日志文件。作此限制的目的是避免与写磁带介质相关的固有问题，例如导致磁带拉长。

•

在分区数据库环境中，db2tapemgr 实用程序每次仅对一个数据库分区执行。必须对每个数据库分区运行适当的命令，并使用 DB2TAPEMGR 命令的 ON DBPARTITIONNUM 选项来指定数据库分区号。还必须确保每个数据库分区都有权访问磁带设备。

示例

以下示例显示如何使用 DB2TAPEMGR 命令，以将数据库分区号 0 中 sample 数据库的主归档日志路径中的所有日志文件存储到磁带设备，并从归档日志路径中除去它们：

```
db2tapemgr db sample on dbpartitionnum 0 store on /dev/rmt0.1 all logs
```

以下示例显示如何将主归档日志路径中的前 10 个日志文件存储到磁带设备，并从归档日志路径中除去它们：

```
db2tapemgr db sample on dbpartitionnum store on /dev/rmt0.1 10 logs
```

以下示例显示如何将主归档日志路径中的前 10 个日志文件存储到磁带设备、将这些日志文件存储到第二盘磁带并从归档日志路径中除去它们：

```
db2tapemgr db sample on dbpartitionnum double store on /dev/rmt0.1 10 logs  
db2tapemgr db sample on dbpartitionnum double store on /dev/rmt1.1 10 logs
```

以下示例显示如何从磁带检索所有日志文件并将它们存储到目录中：

```
db2tapemgr db sample on dbpartitionnum retrieve all logs from /dev/rmt1.1  
to /home/dbuser/archived_logs
```

使用用户出口程序使日志文件归档和检索自动进行

通过创建 DB2 数据库管理器调用来执行归档或检索操作的用户出口程序，可以自动执行日志文件归档和检索。

当 DB2 数据库管理器调用用户出口程序时，会执行下列操作：

- 数据库管理器将控制权交给用户出口程序；
- 数据库管理器将参数传递给用户出口程序；并且
- 在完成时，用户出口程序再将返回码发送回数据库管理器。

配置

在调用用户出口程序进行日志文件的归档或检索前，确保 `logarchmeth1` 数据库配置参数已设置为 `USEREXIT`。它还使数据库可进行前滚恢复。

用户出口程序的要求

•

用户出口程序的可执行文件的名称必须为 `db2uext2`。

•

用户出口程序必须将日志文件从活动日志路径复制到归档日志路径，而不移动这些文件。不要从活动日志路径中除去日志文件。如果从活动日志路径中除去日志文件，那么在出现故障时，DB2 数据库可能无法成功恢复。

在恢复期间，DB2 数据库要求日志文件位于活动日志路径中。DB2 数据库服务器将从活动日志路径中除去恢复操作不再需要的归档日志文件。

•

用户出口程序必须处理错误情况。由于 DB2 数据库管理器只能处理有限的返回情况，所以用户出口程序必须处理错误。

请参阅第 123 页的『用户出口错误处理』。

•

每个 DB2 数据库管理器实例只能调用一个用户出口程序。由于数据库管理器实例只能调用一个用户出口程序，所以必须将您的用户出口程序设计成有一个部分包含它必须执行的操作。

样本用户出口程序

对所有受支持的平台都提供了样本用户出口程序。您可以修改这些程序以适应您的特定要求。样本程序中有很好的注释信息，它有助于您进行更有效的使用。

您应该知道用户出口程序必须从活动日志路径复制日志文件到归档日志路径中。不要从活动日志路径中除去日志文件。（这可能会在数据库恢复期间导致问题。）DB2 从活动日志路径中除去恢复操作不再需要的归档日志文件。

以下是 DB2 数据服务器附带的样本用户出口程序的描述。

- 基于 **UNIX** 的系统

DB2 数据服务器基于 UNIX 版的系统中的用户出口样本程序位于 `sqllib/samples/c` 子目录中。虽然提供的样本是用 C 语言编码的，您的用户出口程序仍可以不同的编程语言编写。

您的用户出口程序必须是可执行文件，名称为 `db2uext2`。

这里有四个可用于基于 UNIX 的系统的样本用户出口程序：

– `db2uext2.ctsm`

此样本使用 Tivoli Storage Manager 来归档和检索数据库日志文件。

– `db2uext2.ctape`

该样本使用磁带介质来归档和检索数据库日志文件。

– `db2uext2.cdisk`

该样本使用操作系统的 COPY 命令以及磁盘介质来归档和检索数据库日志文件。

– `db2uext2.cxbsa`

此样本使用 X/Open group 发布的 XBSA Draft 0.8。可将它用于归档和检索数据库日志文件。此样本只在 AIX 上受支持。

• Windows 操作系统

DB2 数据服务器 Windows 版操作系统中的用户出口样本程序位于 `sqllib\samples\c` 子目录中。虽然提供的样本是用 C 语言编码的，您的用户出口程序仍可以不同的编程语言编写。

您的用户出口程序必须是可执行文件，名称为 `db2uext2`。

对于 Windows 操作系统，有两个样本用户出口程序：

– `db2uext2.ctsm`

此样本使用 Tivoli Storage Manager 来归档和检索数据库日志文件。

– `db2uext2.cdisk`

该样本使用操作系统的 COPY 命令以及磁盘介质来归档和检索数据库日志文件。

用户出口程序的调用格式

DB2 数据库管理器调用用户出口程序时，它将一组参数（数据类型为 CHAR）发送到该程序。

命令语法

```
db2uext2 -OS<os> -RL<db2rel> -RQ<request> -DB<dbname>  
          -NN<nodenum> -LP<logpath> -LN<logname> -AP<tsmpasswd>  
          -SP<startpage> -LS<logsize>
```

os 指定运行实例的平台。有效值包括：AIX、Solaris、HP-UX、SCO、Linux 和 NT。

db2rel 指定 DB2 发行版级别。例如，SQL07020。

request

指定请求类型。有效的值是：ARCHIVE 和 RETRIEVE。

dbname

指定数据库名称。

nodenum

指定本地节点号，例如 5。

logpath

指定日志文件的标准路径。该路径必须包含尾部路径分隔符。例如 /u/data-base/log/path/ 或 d:\logpath\。

logname

指定要归档或检索的日志文件的名称，例如 S0000123.LOG。

tsmpasswd

指定 TSM 密码。（如果先前已指定了数据库配置参数 *tsm_password* 的值，该值将发送到用户出口程序。）

startpage

指定日志扩展数据块开始的那个设备的 4 KB 偏移页的数量。

logsize

指定日志扩展数据块的大小，以 4 KB 页为单位。只有将原始设备用于日志记录时，此参数才有效。

用户出口错误处理

如果创建了一个用户出口程序来自动执行日志文件归档和检索，那么该用户出口程序会将返回码传递给调用它的 DB2 数据库管理器。DB2 数据库管理器只能处理数目有限的特定错误代码列表。但是，用户出口程序可能会遇到许多不同类型的错误情况，例如，操作系统错误。用户出口程序必须将它遇到的错误情况映射至数据库管理器可以处理的错误代码。

表 8 显示了可由用户出口程序返回的代码，并描述数据库管理器如何解释这些代码。如果返回码未在该表中列出，那么将其值当作 32 处理。

表 8. 用户出口程序返回码。只适用于归档和检索操作。

返回码	说明
0	成功。
4	遇到临时资源错误。 ^a
8	需要操作员介入。 ^a
12	硬件错误。 ^b
16	用户出口程序或该程序使用的软件功能出错。 ^b
20	传送给用户出口程序一个或多个参数出错。验证用户出口程序是否正确地处理指定的参数。 ^b
24	找不到用户出口程序。 ^b
28	输入/输出 (I/O) 故障或操作系统导致的错误。 ^b
32	用户出口程序由用户终止。 ^b
255	由用户出口程序无法为可执行文件装入库文件而导致的错误。 ^c

表 8. 用户出口程序返回码 (续). 只适用于归档和检索操作。

返回码	说明
	<p>^a 对于归档与检索请求，返回码 4 或 8 导致在五分钟后重试。如果用户出口程序继续对向同一日志文件发出的检索请求返回 4 或 8，那么 DB2 将继续重试，直到成功为止。（这适用于前滚操作或对 db2ReadLog API（由复制实用程序使用）的调用。）</p> <p>^b 用户出口请求将暂挂 5 分钟。在此期间，将忽略所有的请求，包括导致错误状态的请求。在这个时间为 5 分钟的暂挂后，再处理下一个请求。如果处理此请求时没有错误，将继续处理新的用户出口请求，且 DB2 会重新发出失败过的或先前被暂挂的归档请求。如果在重试期间生成一个大于 8 的返回码，那么会将请求再暂挂 5 分钟。此 5 分钟的暂挂将继续，直到该校正该问题或直到停止并重新启动该数据库。一旦从该数据库断开了与所有应用程序的连接，DB2 会对先前可能未成功归档的任何日志文件发出归档请求。如果用户出口程序对日志文件归档失败，磁盘可能会填满了日志文件，从而可能使性能降低。一旦磁盘变满，数据库管理器将不再接受应用程序对更新数据库的请求。如果调用了用户出口程序来检索日志文件，那么前滚恢复会暂挂但不停止，除非指定了 ROLLFORWARD STOP 选项。如果未指定 STOP 选项，可以更正问题并继续恢复。</p> <p>^c 如果用户出口程序返回错误代码 255，可能是程序无法装入可执行文件所需的库文件。要进行验证，可手动调用用户出口程序。会显示更多信息。</p> <p>注：在归档和检索操作期间，对 0 和 4 之外的所有返回码都发出警报消息。该警报消息包含来自用户出口程序的返回码和提供给用户出口程序的输入参数的副本。</p>

分配和除去日志文件

如果崩溃恢复可能需要数据库日志目录中的日志文件，就不能将它们除去。崩溃恢复需要的日志文件称为活动日志。崩溃恢复不需要的日志文件称为归档日志。

如果已启用无限日志记录，那么成功归档日志文件后，将删除这些日志文件。当 logarchmeth1 数据库配置参数未设置为 OFF 时，仅在崩溃恢复不再需要完整日志文件之后，才考虑除去该日志文件。

分配新日志文件和除去旧日志文件的过程取决于 logarchmeth1 数据库配置参数的设置：

Logarchmeth1 和 Logarchmeth2 都设置为 OFF

将使用循环日志记录。对于循环日志记录，不支持前滚恢复，但支持崩溃恢复。

在循环日志记录期间，不会生成新日志文件（辅助日志除外）且不删除旧日志文件。日志文件以循环的方式来处理。即，当最后一个日志文件已满时，DB2 开始写入第一个日志文件。

如果所有日志文件都是活动的且循环日志记录进程不能回绕至第一个日志文件，那么日志已满情况就可能发生。当所有主日志文件处于活动状态且已满时，会创建辅助日志文件。当取消激活数据库，或活动日志文件需要辅助日志文件所使用的空间时，将删除辅助日志文件。

Logarchmeth1 设置为 LOGRETAIN

使用归档日志。数据库是可恢复的数据库。启用前滚恢复和崩溃恢复。归档日志文件之后，必须从当前日志路径中删除它们，以便新的日志文件可以复用磁盘空间。每当日志文件已满，DB2 就开始将记录写至另一个日志文件，并且（如果尚未达到主辅日志的最大数目）创建新日志文件。

Logarchmeth1 设置为 **OFF** 或 **LOGRETAIN** 以外的值。

使用归档日志。数据库是可恢复的数据库。启用前滚恢复和崩溃恢复。当日志文件满时，自动对它进行归档。

通常不会删除日志文件。相反，当需要新的日志文件而没有可用的日志文件时，那么重命名归档日志文件并再次使用该文件。一旦归档日志文件已关闭并被复制到日志归档目录中，就不会被删除或重命名。DB2 等待直到需要新日志文件，然后重命名最旧的归档日志。在恢复期间已移至数据库目录的日志文件，当不再需要该文件时就会在恢复进程期间将它除去。在 DB2 用尽日志空间之前，可在数据库目录中看到旧的日志文件。

如果在归档日志文件时发生错误，归档暂挂一段时间，此时间由 **ARCHRETRYDELAY** 数据库配置参数指定。还可以使用 **NUMARCHRETRY** 数据库配置参数来指定 DB2 尝试将日志文件归档到主要或辅助归档目录的次数，然后它再尝试将日志文件归档到故障转移目录（由 **FAILARCHPATH** 数据库配置参数指定）。只有在设置 **FAILARCHPATH** 数据库配置参数之后，才使用 **NUMARCHRETRY**。如果 **NUMARCHRETRY** 设置为 0，DB2 将继续从主日志路径或辅助日志路径重试归档。

除去旧日志文件的最简单方法是重新启动数据库。一旦重新启动了数据库，就只有新日志文件和用户出口程序归档失败的日志文件将在数据库目录中找到。

当数据库重新启动时，数据库日志目录中日志的最小数目将等于主日志的数目，这些主日志是可以通过使用 *logprimary* 数据库配置参数来配置的。可在日志目录中找到多于主日志数目的日志。如果关闭数据库时日志目录中空日志数大于重新启动数据库时 *logprimary* 配置参数的值，那么可能发生这种情况。如果在关闭数据库与重新启动数据库之间更改了 *logprimary* 配置参数的值，或者如果分配了辅助日志且从未使用，那么将发生这种情况。

当重新启动数据库时，如果空日志数小于 *logprimary* 配置参数指定的主日志的数目，那么分配附加日志文件以弥补所差的日志数目。如果数据库目录中有多于主日志的空日志可用，那么在重新启动数据库，可在数据库目录中找到所有可用的空日志。数据库关闭之后，已创建的辅助日志文件在重新启动数据库时将保留在活动日志路径中。

使用备份映像包括日志文件

执行联机备份操作时，可以指定在备份映像中包括复原和恢复数据库所需的日志文件。这意味着，如果需要将备份映像交付至灾难恢复位置，那么无需单独发送日志文件或自行将其打包。此外，不必决定需要哪些日志文件来保证联机备份的一致性。这为成功恢复提供某种保护，以防止删除所需的日志文件。

要使用此功能，指定 **BACKUP DATABASE** 命令的 **INCLUDE LOGS** 选项。当指定此选项时，**BACKUP** 实用程序将截断当前活动日志文件并将必要的日志扩展数据块集合复制到备份映像中。

要从备份映像复原日志文件，可使用 **RESTORE DATABASE** 命令的 **LOGTARGET** 选项，并指定存在于 DB2 服务器上的标准路径。然后，复原数据库实用程序将日志文件从映像写至目标路径。如果目标路径中已存在名称相同的日志文件，复原操作将失败并返回一个错误。如果未指定 **LOGTARGET** 选项，那么不会从备份映像中复原任何日志文件。

如果指定 LOGTARGET 选项，且备份映像不包含任何日志文件，那么在尝试复原任何表空间数据之前，将返回一个错误。如果指定无效或只读路径，复原操作也将失败。在指定 LOGTARGET 选项条件下，复原数据库或表空间的过程中，如果不能抽取一个或多个日志文件，那么复原操作将失败并返回错误。

还可以选择只复原保存在备份映像中的日志文件。要执行此操作，可使用 RESTORE DATABASE 命令的 LOGTARGET 选项指定 LOGS 选项。以此方式复原日志文件时，如果复原操作遇到任何问题，那么复原操作将失败并返回错误。

自动增量复原操作期间，将仅从备份映像中检索复原操作的目标映像中包括的日志。不会从那些备份映像中抽取增量复原操作期间所引用的中间映像中包括的任何日志。手动增量复原期间，当复原包括日志文件的备份映像时，如果指定日志目标目录，那么会复原此备份映像中的日志文件。

如果前滚一个数据库，而该数据库是使用包含日志文件的联机备份映像复原的，就会发生错误 SQL1268N，此错误表示前滚恢复已由于检索日志时接收到错误而停止。如果您尝试将备份映像复原到的目标系统无权访问源系统归档其事务日志时使用的设施，就会发生此错误。

如果备份数据库时指定了 BACKUP DATABASE 命令的 INCLUDE LOGS 选项，接着执行使用该备份映像的复原操作和前滚操作，那么虽然备份映像包含日志，DB2 在前滚数据库时也仍然会搜索其他事务日志。标准前滚行为会一直搜索其他事务日志，直到找不到更多日志为止。有可能有多个时间戳记相同的日志文件。因此，DB2 在找到第一个与数据库前滚目标时间点匹配的时间戳记时不会立即停止，因为其他日志文件也可能拥有该时间戳记。DB2 将继续查找事务日志，直到找到大于指定时间点的时间戳记为止。

当找不到任何其他日志时，前滚操作将成功地结束。但是，如果搜索其他事务日志文件时出错，就会返回错误 SQL1268N。在初始复原期间，可能会由于某些数据库配置参数被复位或覆盖而发生错误 SQL1268N。在这些数据库配置参数中，有三个是 TSM 参数，即 TSM_NODENAME、TSM_OWNER 和 TSM_PASSWORD。它们全都被复位为 NULL。要前滚至日志末尾，在执行前滚操作前，需要将这些数据库配置参数复位为与源系统相对应。此外，可以在发出 ROLLFORWARD DATABASE 命令时指定 NORETRIEVE 选项。这将防止 DB2 数据库系统尝试获取可能已在别处丢失的事务日志。

注：

1. 脱机备份不支持此功能。
2. 当联机备份映像中包括日志时，在版本 8.2 之前的 DB2 数据库发行版中无法复原生成的映像。

防止日志文件意外丢失

在需要删除数据库或执行时间点前滚恢复的情况下，可能会丢失以后恢复操作所需的日志文件。在这些情况下，对当前数据库日志路径目录中的所有日志进行复制是很重要的。

请考虑以下方案：

-

如果计划在复原操作之前删除数据库，那么需要在发出 DROP DATABASE 命令之前将日志文件保存在活动日志路径中。因为删除数据库前其中一些日志文件可能尚未归档，所以，复原数据库后，前滚恢复可能需要这些日志文件。通常，不需要在发出 RESTORE 命令之前删除数据库。但是，如果该数据库已损坏到会导致 RESTORE 命令失败的程度，那么可能必须删除该数据库（或者通过指定 DROP DATABASE 命令的 AT[®] NODE 选项来在一个数据库分区中删除该数据库）。您也可能决定在执行复原操作前删除数据库以便从头开始。

如果将数据库前滚至特定时间点，那么指定时间戳记之后的日志数据将被覆盖。在完成时间点前滚操作并重新连接到数据库之后，如果确定实际上需要将该数据库前滚至稍后的时间点，将无法实现此操作，这是因为日志已被覆盖。或许原始日志文件集已归档，但是 DB2 可能调用用户出口程序来自动归档新生成的日志文件。视编写的用户出口程序的不同，这可能导致归档日志目录中的原始日志文件集被覆盖。即使原始日志文件集和新日志文件集都存在于归档日志目录中（相同文件的不同版本），也必须确定应对以后的恢复操作使用的那组日志。

尽量降低维护对可用性的影响

您将需要对 DB2 数据库解决方案执行维护，例如，软件或硬件升级、数据库性能调整、数据库备份、统计信息收集和商业目的的监视。为了尽量降低执行该维护对解决方案可用性的影响，需要执行以下工作：谨慎安排脱机维护，以及使用可降低联机维护对可用性的影响的 DB2 功能部件和功能。

在可以使用下列步骤来尽量降低维护对 DB2 数据库解决方案的可用性的影响之前，必须：

- 配置自动维护；以及
 - 安装高可用性灾难恢复（HADR）功能部件。
1. 允许自动维护为您进行维护。

DB2 数据库可以使许多数据库维护活动自动进行。一旦配置了自动维护，将不需要您采取任何其他步骤就能执行该维护。

2. 使用 DB2 高可用性灾难恢复（HADR）卷动升级来尽量降低其他维护活动的影响。

如果要升级软件或硬件，或者如果要修改某些数据库管理器配置参数，可以使用 HADR 功能部件在对可用性中断最小的情况下完成这些更改。由 HADR 实现的这种无缝更改被称为卷动升级。

即使是在 HADR 环境中，有些维护活动也需要在执行维护之前关闭数据库。在某些情况下，关闭 HADR 数据库的过程与关闭标准数据库的过程会稍微不同：如果 HADR 数据库是由与之连接的客户机应用程序启动的，那么必须使用 DEACTIVATE DATABASE 命令。

停止 DB2 高可用性灾难恢复（HADR）

如果正在使用 DB2 高可用性灾难恢复（HADR）功能，那么在两个 DB2 数据库系统（主数据库系统和备用数据库系统）上执行维护比在一个单独的数据库服务器上执行维护要复杂得多。如果需要停止 HADR 以执行维护，可使用 STOP HADR 命令来停

止 HADR 功能。如果仅在备用数据库系统上执行维护，那么只需要停止备用数据库上的 HADR。要完全停止使用 HADR，可以停止两个数据库上的 HADR。

警告：如果要停止指定的数据库，但仍然想保持其 HADR 主数据库角色或备用数据库角色，那么不要发出 STOP HADR 命令。如果发出 STOP HADR 命令，该数据库将变成标准数据库，并且可能要求进行重新初始化才能继续作为 HADR 数据库工作。而是，请发出 DEACTIVATE DATABASE 命令。

可以仅针对主数据库或备用数据库发出 STOP HADR 命令。如果针对标准数据库发出此命令，那么会返回错误。

可以使用命令行处理器（CLP）、控制中心中的“管理高可用性灾难恢复”（HADR）窗口或 db2HADRStop 应用程序编程接口（API）停止 HADR。

要使用 CLP 停止主数据库或备用数据库上的 HADR 操作，在要停止 HADR 操作的数据库上发出 STOP HADR 命令。

在以下示例中，停止数据库 SOCKS 上的 HADR 操作：

```
STOP HADR ON DATABASE SOCKS
```

如果针对不活动的主数据库发出此命令，数据库切换至标准数据库并保持脱机。

如果针对不活动的备用数据库发出此命令，数据库切换至标准数据库、置于前滚暂挂状态，并保持脱机。

如果在活动的主数据库上发出此命令，会停止将日志装入备用数据库，并且在主数据库上关闭所有 HADR 引擎分派单元（EDU）。数据库切换至标准数据库并保持联机。可以继续从事务处理。可以发出带有 AS PRIMARY 选项的 START HADR 命令将数据库角色重新切换为主数据库。

如果在活动的备用数据库上发出此命令，那么返回错误消息，指示必须取消激活备用数据库，然后才能尝试将其转换至标准数据库。

要打开“停止 HADR”窗口：

1. 在控制中心中，展开对象树，直到找到要为其管理 HADR 的数据库为止。用鼠标右键单击此数据库，然后在弹出菜单中单击“高可用性灾难恢复”→“管理”。这就打开了“管理高可用性灾难恢复”窗口。
2. 单击“停止 HADR”。这就打开了“停止 HADR”窗口。
3. 如果只想对一个数据库停止 HADR，那么清除其他数据库的复选框。
4. 如果只启动了一个数据库（主数据库或备用数据库），那么该数据库的名称将显示在“停止 HADR”窗口中。
5. 单击“确定”。该窗口将关闭。可能会显示进度指示器以指示正在运行命令。该指示器完成后，您将看到指示命令成功与否的通知。

控制中心中的上下文帮助工具提供了其他信息。

在 DB2 高可用性灾难恢复（HADR）环境中激活和取消激活数据库

如果标准数据库是由客户机连接启动的，那么最后一个客户机断开连接时，该数据库将关闭。如果 HADR 主数据库是由客户机连接启动的，那么等同于使用 ACTIVATE

DATABASE 命令来启动数据库。要关闭由客户机连接启动的 HADR 主数据库，需要显式地发出 DEACTIVATE DATABASE 命令。

对于处于前滚暂挂状态的标准数据库，ACTIVATE DATABASE 和 DEACTIVATE DATABASE 命令不适用。只能继续前滚、停止前滚或者使用 START HADR 来启动该数据库以使其成为 HADR 备用数据库。一旦数据库作为 HADR 备用数据库启动，您就可以使用 ACTIVATE DATABASE 和 DEACTIVATE DATABASE 命令来启动和停止该数据库。

使用下列方法来激活主数据库：

- 客户机连接
- ACTIVATE DATABASE 命令
- 指定了 AS PRIMARY 选项的 START HADR 命令

使用下列方法来主数据库取消激活：

- DEACTIVATE DATABASE 命令
- 指定了 FORCE 选项的 db2stop 命令

使用下列方法来激活备用数据库：

- ACTIVATE DATABASE 命令
- 指定了 AS STANDBY 选项的 START HADR 命令

使用下列方法将备用数据库取消激活：

- DEACTIVATE DATABASE 命令
- 指定了 FORCE 选项的 db2stop 命令

在 DB2 高可用性灾难恢复（HADR）环境中执行滚动升级

当升级软件/硬件、更新 DB2 数据库系统或更改数据库配置参数时，在高可用性灾难恢复（HADR）环境中使用此过程。此过程保持数据库服务在整个升级过程中可用，只在处理从一个数据库切换至另一数据库时有短暂的服务中断。因为当主数据库和备用数据库在同一系统上时 HADR 执行最理想，所以应该尽快在两种系统上应用更改。

注：应该在测试环境中实现所有 DB2 数据库系统修订包和升级，然后再应用到生产系统中。

在启动滚动升级之前，HADR 数据库对应该处于对等状态。

此过程不能用于从 DB2 数据库系统的较低版本升级到较高版本；例如，不能使用此过程从版本 8 升级到版本 9 数据库系统。只能使用此过程来对数据库系统执行滚动更新（从一个修改级别滚动更新为另一个修改级别），例如，通过应用修订包。

如果更新 DB2 HADR 配置参数，此过程就不起作用。对 HADR 配置参数的更新应该单独进行。由于 HADR 要求主数据库与备用数据库上的参数相同，所以这可能要求同时将主数据库和备用数据库取消激活并进行更新。

要在 HADR 环境中执行滚动升级：

1. 升级备用数据库所在的系统：
 - a. 使用 DEACTIVATE DATABASE 命令来关闭备用数据库。

- b. 如有必要，请关闭备用数据库上的实例。
- c. 更改下列一项或多项：软件、硬件或 DB2 配置参数。

注：在执行卷动升级时，不能更改任何 HADR 配置参数。

- d. 如有必要，请重新启动备用数据库上的实例。
 - e. 使用 `ACTIVATE DATABASE` 命令来重新启动备用数据库。
 - f. 确保备用数据库进入对等状态。使用 `GET SNAPSHOT` 命令来检查此状态。
2. 切换主数据库和备用数据库的角色：
 - a. 在备用数据库上发出 `TAKEOVER HADR` 命令。
 - b. 将客户机导向新的主数据库。可以使用客户机自动重新路由来实现此操作。

注：因为备用数据库作为主数据库接管，所以现在升级新的主数据库。如果正在应用 DB2 数据库系统修订包，那么 `TAKEOVER HADR` 命令会将原始主数据库的角色更改为备用数据库。但是，此命令将不允许新的备用数据库连接至最近更新的主数据库。因为新的备用数据库使用较低版本的 DB2 数据库系统，所以它不可能理解由更新的主数据库生成的新日志记录，因而将关闭此备用数据库。为了使新备用数据库与新主数据库重新连接（即，针对要改正的 HADR 对），还必须对新备用数据库进行更新。

3. 使用与上述步骤 1 中相同的过程升级原始主数据库（现在是备用数据库）。执行此操作之后，两个数据库均已升级且以 HADR 对等状态相互连接。HADR 系统提供完整的数据库服务和充分的高可用性保护。
4. 可选。要返回至原始配置，如步骤 2 中所述，切换主数据库和备用数据库的角色。

使用分割镜像来克隆数据库

使用以下过程来创建克隆数据库。尽管您可以写入克隆数据库，但通常此类数据库用于只读活动，例如，运行报告。

不能备份克隆的数据库，不能在原始系统上复原备份映像，也不能通过原始系统上生成的日志文件前滚。可以使用 `AS SNAPSHOT` 选项，但它仅提供在暂挂 I/O 时数据库的瞬时副本；在对克隆执行 `db2inidb` 命令之后，将回滚所有其他未完成的落实工作。

要克隆一个数据库，应遵循以下步骤：

1. 暂挂主数据库上的 I/O：

```
db2 set write suspend for database
```

数据库暂挂期间，您不应该运行其他实用程序或工具。您只应生成数据库副本。

2. 使用适当的操作系统级别的命令来从主数据库中分割镜像。

注意：确保复制整个数据库目录（包括卷目录）。还必须复制日志目录和存在于数据库目录之外的任何容器目录。要收集此信息，请参阅 `DBPATHS` 管理视图，该视图显示了需要分割的数据库的所有文件和目录。

3. 恢复主数据库上的 I/O：

```
db2 set write resume for database
```

4. 在辅助系统上对镜像数据库进行编目。

注意：缺省情况下，镜像数据库与主数据库不能存在于同一个系统上。它必须位于具有相同目录结构的辅助系统上并使用与主数据库相同的实例名。如果镜像数据库

与主数据库必须存在于同一个系统上，那么可使用 `db2relocatedb` 实用程序或 `db2inidb` 命令的 `RELOCATE USING` 选项来实现这一点。

5. 在辅助系统上启动数据库实例:

```
db2start
```

6. 在辅助系统上初始化镜像数据库:

```
db2inidb database_alias as snapshot
```

如果需要，那么指定 `db2inidb` 命令的 `RELOCATE USING` 选项来重定位克隆数据库:

```
db2inidb database_alias as snapshot relocate using relocatedbcfg.txt
```

其中 `relocatedbcfg.txt` 文件包含重定位数据库所需的信息。

注意:

- a. 此命令将回滚分割发生时未完成的事务，启动新的日志链序列以使主数据库的所有日志都不能在克隆数据库上重放。
- b. 必须将数据库目录（包括卷目录）、日志目录和容器目录移至期望的位置，然后才能使用 `RELOCATE USING` 选项。

同步主数据库和备用数据库

有一种高可用性策略是配备一个主数据库，一个辅助或备用数据库，前者出现故障时由后者接管操作。如果备用数据库必须接管故障主数据库的数据库操作，那么它必须包含完全相同的数据，了解所有未完成事务并且以与主数据库服务器不发生故障时完全相同的方式继续数据库处理。更新备用数据库，使之成为主数据库副本的进行过程称为同步。

在同步主数据库和备用数据库之前，必须:

- 创建并配置主数据库和备用数据库。
- 配置主数据库和备用数据库之间的通信。
-

选择同步策略（例如，日志装入、日志镜像、暂挂 I/O 和磁盘镜像或 HADR。）

有几种保持主数据库服务器和备用数据库服务器同步的策略:

- 将日志从主数据库装入备用数据库并在备用数据库上前滚;
 - 同时将数据库日志写入主数据库和备用数据库，称为日志镜像;
 - 使用具有磁盘镜像的暂挂 I/O 支持来定期获取主数据库的副本、分割镜像并将副本初始化为新的备用数据库服务器; 以及
 - 使用 DB2 高可用性灾难恢复 (HADR) 功能等可用性功能来保持主数据库和备用数据库同步。
1. 如果使用日志来同步主数据库和辅助或备用数据库，那么请配置 DB2 数据库以为您执行所需要日志管理。例如，如果希望 DB2 数据库为日志生成镜像，那么可将 `MIRRORLOGPATH` 配置参数设置为日志另一副本的保存位置。
 2. 如果使用 DB2 数据库暂挂 I/O 功能来分割主数据库的磁盘镜像，那么您必须执行以下操作:

- a. 初始化主数据库的磁盘镜像。
 - b. 当需要分割主数据库的镜像时，请按照“将分割镜像用作备用数据库”主题中的指示信息进行操作。
3. 如果使用 HADR 功能来管理主数据库和备用数据库的同步，那么请为 HADR 配置 DB2 数据库，并允许 DB2 数据库为您同步主数据库和备用数据库。

DB2 高可用性灾难恢复 (HADR) 复制的操作

DB2 高可用性灾难恢复 (HADR) 使用数据库日志将数据从主数据库复制到备用数据库。在备用数据库上重放日志时，某些活动可能会导致备用数据库落后于主数据库。某些活动要进行大量记录，它们生成的大量日志文件可能会导致存储问题。虽然使用日志将数据复制到备用数据库是可用性策略的核心，但记录本身可能会对解决方案的可用性产生负面影响。合理设计维护策略，配置系统以尽可能降低日志记录的负面影响，并允许日志记录保护您的事务数据。

在高可用性灾难恢复 (HADR) 中，将以下操作从主数据库复制到备用数据库：

- 数据定义语言 (DDL)
- 数据操作语言 (DML)
- 缓冲池操作
- 表空间操作
- 联机重组
- 脱机重组
- 存储过程和用户定义的函数 (UDF) 的元数据 (但不是相关对象或库文件)

联机重组过程中，详细记录所有操作。结果，HADR 可以复制操作，而不会使备用数据库比它在进行更多典型数据库更新时更加远远地落在后面。但是，由于生成大量日志记录，所以此行为可能对系统产生较大影响。

如果未如联机重组那样大量地记录脱机重组，通常按几百或几千个受影响的行来记录操作。这意味着备用数据库将落后，因为它等待每个日志记录，然后立刻重放许多更新。如果脱机重组是非集群的，那么在整个重组操作之后生成单一日志记录。此方式在最大程度上影响备用数据库与主数据库保持同步的功能。备用数据库从主数据库接收日志记录之后，将执行整个重组过程。

HADR 不复制存储过程、UDF 对象和库文件。必须在主数据库和备用数据库中相同路径上创建文件。如果备用数据库无法找到引用的对象或库文件，那么备用数据库上的存储过程或 UDF 调用将失败。

DB2 高可用性灾难恢复 (HADR) 不复制的操作

DB2 高可用性灾难恢复 (HADR) 使用数据库日志将数据从主数据库复制到备用数据库。主数据库允许不进行日志记录的操作，但不会将此类操作复制到备用数据库。如果要在备用数据库中反映未日志记录的操作 (例如，对历史记录文件的更新)，那么必须执行额外的步骤来实现此目的。

以下是一些情况示例，在这些情况下，不会将主数据库上的操作复制到备用数据库：

- 在指定了 NOT LOGGED INITIALLY 选项的情况下创建的表不会被复制。在 HADR 备用数据库接管主数据库后尝试访问这样的表将导致错误。

- 不能对大于 1GB 的 BLOB 和 CLOB 进行日志记录，因此不能复制它们。不复制未进行日志记录的 BLOB 和 CLOB。但是，在备用数据库上将它们分配空间。LOB 列数据将是二进制零。将复制所有已进行日志记录的 BLOB 和 CLOB。
- 不复制使用 UPDATE DATABASE CONFIGURATION 和 UPDATE DATABASE MANAGER CONFIGURATION 命令对数据库配置所作的更新。
- 不复制数据库配置参数和数据库管理器配置参数。
- 对于用户定义的函数（UDF）来说，不复制对数据库外部的对象（例如相关的对象和库文件）所作的更改。您需要通过其他方法在备用数据库上对它们进行设置。
- 不会自动地将恢复历史记录文件（db2rhist.asc）以及对其所作的更改从主数据库复制到备用数据库。

通过发出指定了 REPLACE HISTORY FILE 选项的 RESTORE DATABASE 命令，可以将历史记录文件的原始副本（从主数据库的备份映像中获取）放到备用数据库上：

```
RESTORE DB KELLY REPLACE HISTORY FILE
```

初始化 HADR 并接着对主数据库执行备份活动后，备用数据库上的历史记录文件就已过期。但是，每个备份映像中都存储了历史记录文件的一个副本。通过使用以下命令从备份映像中抽取历史记录文件，可以更新备用数据库上的历史记录文件：

```
RESTORE DB KELLY HISTORY FILE
```

请不要使用正规操作系统命令将数据库目录中的历史记录文件从主数据库复制到备用数据库。进行复制时，如果主数据库正在更新历史记录文件，那些文件就会损坏。

如果执行接管操作并且备用数据库有最新的历史记录文件，那么对新的主数据库执行的备份和复原操作将在历史记录文件中生成新记录，并且与原始主数据库上生成的记录完全混合。如果历史记录文件过期或者缺少条目，那么可能无法进行自动增量复原；而是，您将需要执行手动增量复原操作。

DB2 高可用性灾难恢复（HADR）备用数据库状态

无论何时，备用数据库都处于五个状态之一：本地同步复制、远程同步复制暂挂、远程同步复制、对等和断开对等。备用数据库所处的状态决定它能够执行的操作。可以使用 GET SNAPSHOT 命令来查看备用数据库所处的状态。

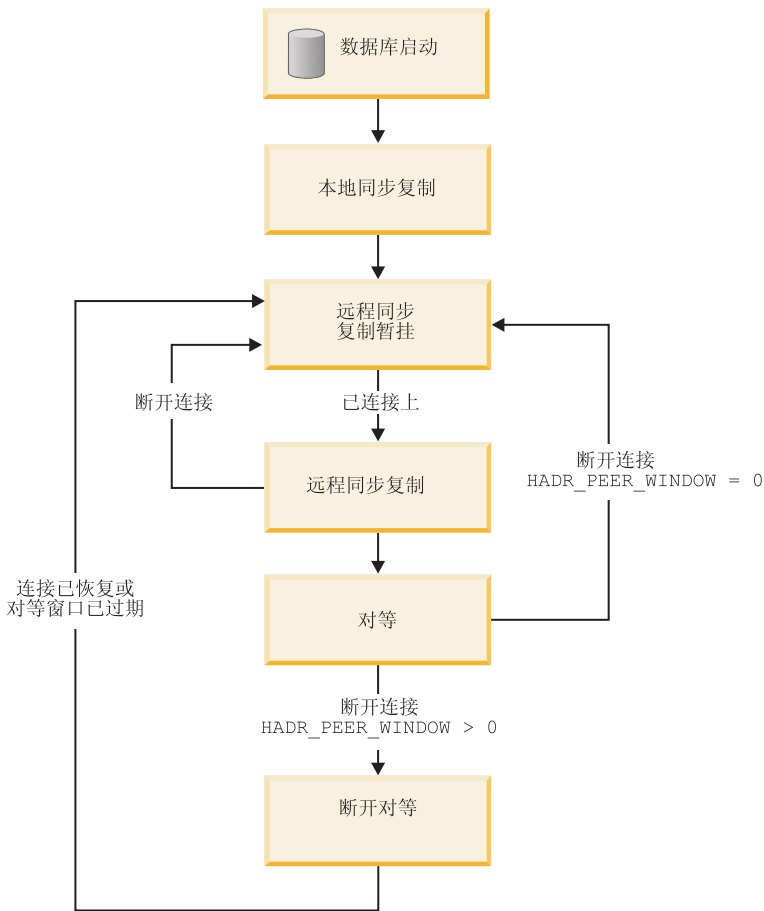


图 9. 备用数据库的状态

数据库启动、本地同步复制和远程同步复制暂挂

使用高可用性灾难恢复（HADR）功能，当启动备用数据库时，它进入本地同步复制状态，并尝试读取其本地日志路径中的日志文件。如果未在本地日志路径中找到日志文件，并且指定了日志归档方法，那么使用指定方法检索日志文件。读取日志文件之后，会在备用数据库上重放这些日志文件。在此期间，不需要连接到主数据库；但是，如果连接不存在，备用数据库会尝试与主数据库连接。当到达本地日志文件末尾时，备用数据库进入远程同步复制暂挂状态。

如果在备用数据库进入远程同步复制暂挂状态之后，本地日志文件变为可用，您可以关闭备用数据库并重新启动，以使它再次进入本地同步复制状态。如果从本地访问备用数据库上的这些日志文件比让 HADR 通过网络从主数据库复制这些文件效率更高，那么可以这样做。

远程同步复制暂挂、远程同步复制和对等

备用数据库保持远程同步复制暂挂状态，直到与主数据库建立连接为止，届时备用数据库进入远程同步复制状态。在此期间，主数据库从其日志路径中或者通过日志归档

方法读取日志数据，并将日志文件发送到备用数据库。备用数据库接收并重放日志数据。备用数据库接收到主数据库机器磁盘上的所有日志文件后，主数据库和备用数据库进入对等状态。

当处于对等状态时，只要主数据库将日志页刷新到磁盘中，日志页随即会装入备用数据库中。将日志页写入备用数据库上的本地日志文件，以确保主数据库和备用数据库拥有相同的日志文件顺序。然后可以在备用数据库上重放日志页。

如果在远程同步复制状态期间主数据库和备用数据库之间的连接断开，那么备用数据库将进入远程同步复制暂挂状态。如果在对等状态期间主数据库和备用数据库之间的连接断开，并且未设置 `HADR_PEER_WINDOW` 数据库配置参数（或设置为零），那么备用数据库将进入远程同步复制暂挂状态。但是，如果在对等状态期间主数据库与备用数据库之间的连接断开，并且 `HADR_PEER_WINDOW` 数据库配置参数设置为非零值，那么备用数据库将进入断开对等状态。

断开对等

如果将数据库配置参数 `HADR_PEER_WINDOW` 配置为大于零的时间值，那么当主数据库与备用数据库的连接断开时，在配置的时间段内，主数据库仍像主数据库与备用数据库处于对等状态那样继续运行。主数据库与备用数据库断开连接，但仍像处于对等状态那样运行的状态称为断开对等。主数据库在与备用数据库的连接断开后保持断开对等状态的时间段称为对等时间。当与备用数据库的连接已恢复或对等时间到期，备用数据库将脱离断开对等状态。

配置对等时间的优点是可以降低多重或级联失败期间丢失事务的风险。如果没有对等时间，当主数据库与备用数据库的连接断开时，主数据库将脱离对等状态。主数据库断开连接时，它将独立于备用数据库处理事务。如果主数据库在这种未处于对等状态的情况下出现失败，那么由于事务未被复制到备用数据库，这些事务可能会丢失。配置对等时间后，在从备用数据库接收到应答，确认已将日志写入备用数据库上的主存储器，或确认已将日志写入备用数据库上的日志文件（取决于 `HADR` 同步方式）之前，主数据库不会认为事务已落实。

配置对等时间的缺点是主数据库上的事务处理将要花较长的时间，甚至当主数据库在对等时间中等待恢复与备用数据库的连接或等待对等时间到期时发生超时。

可以使用 `GET SNAPSHOT` 命令或带 `-hadr` 参数的 `db2pd` 实用程序确定对等时间大小，即 `HADR_PEER_WINDOW` 数据库配置参数的值。

用于同步主数据库与备用数据库的备用数据库状态牵制和限制

一种用于同步主数据库和备用数据库的方法是将主数据库日志文件手动复制到备用数据库日志路径以进行同步复制。如果通过将主数据库日志手动复制到备用数据库日志路径来同步主数据库与备用数据库，必须在启动备用数据库之前复制主数据库的日志文件，原因如下：

1. 到达本地日志文件末尾后，备用数据库会进入远程同步复制暂挂状态，并且在重新启动备用数据库之前，不会再尝试访问本地日志文件。
2. 如果备用数据库进入远程同步复制状态，将日志文件复制到其日志路径会干扰备用数据库的本地日志文件写入操作。

使用 GET SNAPSHOT 命令来确定 HADR 备用数据库状态

通过发出带 DATABASE ON 选项的 GET SNAPSHOT 命令，可以确定 DB2 高可用性灾难恢复（HADR）备用数据库的状态。

要确定主/备用 HADR 数据库对中的 HADR 备用数据库的状态，可以从主数据库或备用数据库发出 GET SNAPSHOT 命令。

- 如果从备用数据库上发出 GET SNAPSHOT 命令，将在状态输出字段中返回备用数据库的状态。
- 如果从与备用数据库连接的主数据库发出 GET SNAPSHOT 命令，将在状态输出字段中返回备用数据库的状态。
- 如果从未与备用数据库连接的主数据库发出 GET SNAPSHOT 命令，将在状态输出字段中返回断开连接。

例如，如果拥有备用数据库 MUSIC，可以发出以下命令来查看其状态：

```
get snapshot for database on music
```

以下输出显示由 GET SNAPSHOT 命令返回的 HADR 状态段：

HADR 状态

角色	= 主数据库
状态	= 对等
同步方式	= 同步
连接状态	= 已连接, 11-03-2002 12:23:09.35092
丢失的脉动信号	= 0
本地主机	= host1.ibm.com
本地服务	= hadr_service
远程主机	= host2.ibm.com
远程服务	= hadr_service
远程实例	= dbinst2
超时（秒）	= 120
主日志位置（文件、页和 LSN）	= S0001234.LOG, 12, 0000000000BB800C
备用日志位置（文件、页和 LSN）	= S0001234.LOG, 12, 0000000000BB800C
日志间隔运行平均值（字节）	= 8723

在查看 GET SNAPSHOT 命令的输出内容时，您可能会注意到日志缺失。由于显式的日志截断操作或者由于停止并重新启动主数据库而导致日志文件截断时，主数据库将移至下一个日志文件开头，这样会产生日志缺失。但是，备用数据库停留在上一个日志文件末尾。只要主数据库写任何日志，该日志就会被复制，且备用数据库会更新它的日志位置。

DB2 高可用性灾难恢复（HADR）管理

对 DB2 高可用性灾难恢复的管理包括配置和维护 HADR 系统状态。

对 HADR 的管理包括下列任务：

- 第 26 页的『初始化高可用性灾难恢复（HADR）』
- 第 127 页的『停止 DB2 高可用性灾难恢复（HADR）』
- 第 147 页的『切换高可用性灾难恢复（HADR）中的数据库角色』
- 第 145 页的『执行 HADR 故障转移操作』
- 第 141 页的『监视高可用性灾难恢复（HADR）』
- 检查或改变与 HADR 相关的数据库配置参数。

- 编目 HADR 数据库（如果有需要）。

可以使用下列方法来管理 HADR:

- 命令行处理器
- 控制中心 GUI 工具
- DB2 管理 API

DB2 高可用性灾难恢复（HADR）命令

DB2 高可用性灾难恢复（HADR）功能为 DB2 高可用性数据库解决方案提供复杂的日志记录、故障转移和恢复功能。无论 HADR 提供的功能有多复杂，您只需要直接命令 HADR 执行下列几个操作：启动 HADR，停止 HADR 以及让备用数据库作为主数据库接管操作。

可以使用三个高可用性灾难恢复（HADR）命令来管理 HADR:

- 启动 HADR
- 停止 HADR
- 接管 HADR

要调用这些命令，请使用命令行处理器或管理 API。也可以使用控制中心的“管理高可用性灾难恢复”窗口中提供的 GUI 来调用这些命令。要从控制中心中打开“管理高可用性灾难恢复”窗口，请用鼠标右键单击数据库，然后单击“高可用性灾难恢复”——>“管理”。

如果发出带有 AS PRIMARY 或 AS STANDBY 选项的 START HADR 命令，并且数据库还没有指定的角色，就会将数据库角色更改为该角色。如果数据库尚未处于激活状态，此命令还将激活该数据库。

STOP HADR 命令将 HADR 数据库（主数据库或备用数据库）更改为标准数据库。任何与 HADR 相关的数据库配置参数都保持不变，因此，您可以方便地重新激活该数据库以使其成为 HADR 数据库。

只能对备用数据库发出的 TAKEOVER HADR 命令将备用数据库更改为主数据库。未指定 BY FORCE 选项时，主数据库和备用数据库将切换角色。指定了 BY FORCE 选项时，备用数据库将单向地切换为主数据库。在此情况下，备用数据库将尝试停止旧的主数据库上的事务处理。但是，不保证事务处理将会停止。只有在故障转移情况下，才使用 BY FORCE 选项来强制执行接管操作。在发出带有 BY FORCE 选项的 TAKEOVER HADR 命令前，您应该尽可能地确保当前主数据库确实已发生故障，也可以自己将其关闭。

HADR 数据库角色切换

可以动态地并且重复地在主角色与标准角色之间切换数据库。当数据库处于联机或脱机状态时，您既可以发出带有 AS PRIMARY 选项的 START HADR 命令，也可以发出 STOP HADR 选项的 START HADR 命令。

可以静态地在备用角色与标准角色之间切换数据库。仅当数据库处于前滚暂挂状态时，您才可以重复地执行此操作。当标准数据库处于脱机状态和前滚暂挂状态时，可发出带有 AS STANDBY 选项的 START HADR 命令将该数据库更改为备用数据库。当备用数据库处于脱机状态时，可使用 STOP HADR 命令将该数据库更改为标准数据库。发出 STOP HADR 命令后，数据库将处于前滚暂挂状态。发出带有 AS STANDBY

选项的后续 START HADR 命令将使该数据库重新成为备用数据库。在对备用数据库停止 HADR 后，如果发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令，那么无法使其重新成为备用数据库。由于该数据库已不再处于前滚暂挂状态，所以，您可以将其用作标准数据库。这称为获取备用数据库快照。在将现有备用数据库更改为标准数据库后，您应该考虑为实现高可用性而创建新的备用数据库。

要切换主数据库和备用数据库的角色，请在不使用 BY FORCE 选项的情况下执行接管操作。

要单向地将备用数据库更改为主数据库（而不将主数据库更改为备用数据库），请执行强制接管操作。以后，可能能够重新集成旧的主数据库以使其成为新的备用数据库。

HADR 角色是持久的。一旦确定了 HADR 角色，数据库就保持具有该角色，即使重复地停止并接着重新启动 DB2 实例，或者取消激活或激活 DB2 数据库亦如此。

启动备用数据库的操作是异步的

在发出带有 AS STANDBY 选项的 START HADR 命令时，一旦相关的引擎可分派单元 (EDU) 成功启动，该命令就会立即返回。此命令不会等待备用数据库连接至主数据库。相反，在主数据库连接到备用数据库之前，不认为主数据库已启动（对主数据库发出带有 BY FORCE 选项的 START HADR 命令时的情况例外）。如果备用数据库遇到错误，例如连接被主数据库拒绝，那么表示带有 AS STANDBY 选项的 START HADR 命令可能已成功返回。因此，HADR 无法将错误指示返回给任何用户提示。HADR 备用数据库会将一条消息写入 DB2 诊断日志并自行关闭。您应该监视 HADR 备用数据库的状态，以确保它成功地与 HADR 主数据库连接。

重放错误（备用数据库在重放日志记录时遇到的错误）也会导致备用数据库关闭。例如，如果没有足够的内存来创建缓冲池，或者在创建表空间时找不到路径，就会发生这些错误。您应该持续监视备用数据库的状态。

不要使用启用了客户机重新路由的数据库别名来从客户机运行 HADR 命令

如果已设置客户机自动重新路由，数据库服务器就有预定义的替代服务器，因此客户机应用程序可以在使用原始数据库服务器或使用替代服务器之间进行切换，并将工作中断降至最少。在这样的环境中，当客户机通过 TCP 连接至数据库时，实际连接可以是连接至原始数据库，也可以是连接至替代数据库。实现了 HADR 命令以通过正规客户机连接逻辑来标识目标数据库。因此，如果为目标数据库定义了替代数据库，就难以确定命令实际上作用于哪个数据库。虽然 SQL 客户机不需要知道它所连接的数据库，但 HADR 命令必须应用于特定的数据库。为了适应这种限制，应该在服务器本地发出 HADR 命令以绕过客户机重新路由（客户机重新路由只影响 TCP/IP 连接）。

HADR 命令必须在安装了有效许可证的服务器上运行

START HADR、STOP HADR 和 TAKEOVER HADR 命令要求在执行该命令的服务器上安装有效的 HADR 许可证。如果没有许可证，这些命令就会失败并返回特定于命令的错误代码（分别为 SQL1767N、SQL1769N 或 SQL1770N）以及原因码 98。要解决此问题，请使用 db2licm 来安装有效的 HADR 许可证，或者安装服务器分发包含有效 HADR 许可证的服务器版本。

第 6 章 在高可用性解决方案中检测和响应系统中断

实现高可用性解决方案并不能防止硬件或软件故障。但是，具备冗余系统和故障转移机制可以使您的解决方案能够检测和响应故障并重新路由工作负载，从而使用户应用程序仍然可以工作。

当出现故障时，数据库解决方案必须执行以下操作：

1. 检测故障。

故障转移软件可以使用脉动信号监视来确定系统组件的可用性。脉动信号监视器侦听来自所有系统组件的正常通信。如果脉动信号监视器无法侦听到来自某个组件的通信，它将向系统发出信号，指示该组件出现故障。

2. 响应故障：故障转移。

- a. 标识辅助组件、使之联机并初始化以接管故障组件的操作。
- b. 将工作负载重新路由至辅助组件。
- c. 从系统除去故障组件。

3. 恢复故障。

如果主数据库服务器出现故障，首先要做的是将客户机重定向至备用服务器或将故障转移至备用数据库，使客户机应用程序在尽可能少中断的情况下工作。一旦故障转移成功，您必须修复故障数据库服务器上的所有问题，使它能够重新集成到解决方案中。修复故障数据库服务器可能只需将它重新启动。

4. 恢复正常操作。

一旦修复故障数据库系统后，必须将它重新集成到数据库解决方案中。主数据库出现故障后，备用数据库接管其工作而成为新的主数据库，这时可以切换两者的数据库角色，从而重新集成原先的主数据库。也可以强制修复的数据库服务器重新作为主数据库服务器接管。

DB2 数据库可以为您执行下列步骤中的一些。例如：

- DB2 高可用性灾难恢复（HADR）脉动信号监视元素 `hadr_heartbeat` 可以检测主数据库何时出现故障。
- DB2 客户机重新路由可以将工作负载从故障数据库服务器转移到辅助数据库服务器上。
- DB2 故障监视器可以重新启动意外终止的数据库实例。

管理通知日志

管理通知日志（`instance_name.nfy`）是可从中获取有关大量数据库管理和维护活动的信息的存储库。数据库管理员可以使用这些信息来诊断问题、调整数据库或仅监视数据库。

DB2 数据库管理器会将以下类型的信息写入 UNIX 和 Linux 操作系统平台上的管理通知日志（在 Windows 操作系统平台上，事件日志用来记录管理通知事件）：

- DB2 实用程序（例如 REORG 和 BACKUP）的状态

- 客户机应用程序错误
- 服务类更改
- 许可证发放活动
- 文件路径
- 存储问题
- 监视活动
- 建立索引活动
- 表空间问题

还会使用标准化消息格式将管理通知日志消息记录到 `db2diag` 日志文件中。

通知消息提供了其他信息以补充提供的 `SQLCODE`。

管理通知日志文件可以下面两种不同的形式存在：

单个管理通知日志文件

一个活动的管理通知日志文件，名为 `instance_name.nfy`，其大小无限地增大。这是缺省形式，每当 `diagsize` 数据库管理器配置参数的值为 0（此参数的缺省值为 0）时，都存在这种形式。

旋转管理通知日志文件

单个活动日志文件（名为 `instance_name.N.nfy`，其中 `N` 是文件名下标，即从 0 开始不断增大的数字），尽管可以在 `diagpath` 配置参数定义的位置中找到一系列管理通知日志文件，但是每个这样的日志文件都在增大直到达到受限的大小，此时，该日志文件会被关闭，并且会创建并打开新的日志文件来进行记录，它带有递增的文件名下标（`instance_name.N+1.nfy`）。每当 `diagsize` 数据库管理器配置参数具有非零值时，都存在这种形式。

注：在 Windows 操作系统平台上，不提供单个管理通知日志文件和旋转管理通知日志文件。

通过相应地设置 `diagsize` 数据库管理器配置参数，可以选择系统上存在这两种形式中的哪一种。

配置

通过设置下列数据库管理器配置参数，可以在大小、位置和事件类型以及所记录详细信息级别这些方面对管理通知日志文件进行配置：

`diagsize`

`diagsize` 的值决定将采用哪种形式的管理通知日志文件。如果该值为 0，那么将采用单个管理通知日志文件。如果该值不为 0，那么将采用旋转管理通知日志文件，此非零值还指定所有旋转诊断日志文件以及所有旋转管理通知日志文件的总体大小。必须重新启动实例才能使 `diagsize` 参数的新值生效。请参阅“`diagsize` - 诊断日志文件大小配置参数”主题，以获取完整详细信息。

`diagpath`

诊断信息可指定写入到 `diagpath` 配置参数所定义位置处的管理通知日志文件中。请参阅“`diagpath` - 诊断数据目录路径配置参数”主题，以获取完整详细信息。

notifylevel

可以使用 **notifylevel** 配置参数来指定写入管理通知日志文件的事件类型和详细信息级别。请参阅“notifylevel - 通知级别配置参数”主题，以获取完整详细信息。

检测意外中断

在对组件故障作出响应之前，必须先检测到出现故障的组件。DB2 数据服务器有几个用于监视数据库运行状况或检测数据库是否失败的工具。可以配置这些工具，以便在检测到故障时通知您或执行预定义的操作。

可以使用下列工具来检测 DB2 数据库解决方案中的某个部件何时出现了故障：

DB2 故障监视器工具

DB2 故障监视器工具使 DB2 数据库实例正常运行。当 DB2 故障监视器连接的 DB2 数据库实例意外终止时，DB2 故障监视器将重新启动该实例。如果数据库解决方案是在集群中实现的，那么应该配置集群管理软件，以重新启动出现故障的数据库实例，而不是重新启动 DB2 故障监视器。

集群环境中的脉动信号监视

集群管理软件在集群中的节点之间使用脉动信号消息来监视节点的运行状况。当节点停止响应或发送任何消息时，集群管理器可检测到该节点出现故障。

监视 DB2 高可用性灾难恢复 (HADR) 数据库

HADR 功能具有它自己的脉动信号监视器。主数据库和备用数据库都期望定期接收到其他数据库发出的脉动信号消息。

监视高可用性灾难恢复 (HADR)

可以使用下列方法来监视 HADR 数据库状态。

db2pd 实用程序

此实用程序从 DB2 内存集中检索信息。例如，要查看数据库 MYDB 的高可用性灾难恢复信息，请发出以下命令：

```
db2pd -db mydb -hadr
```

GET SNAPSHOT FOR DATABASE 命令

此命令收集状态信息并对输出进行格式化。返回的信息表示发出此命令时的数据库管理器工作状态快照。在输出中，HADR 信息显示在标题 *HADR status* 下面。

db2GetSnapshot API

此 API 收集数据库管理器监视器信息并将其返回到用户分配的数据缓冲区。返回的信息表示调用此 API 时的数据库管理器工作状态快照。

HADR 配置参数不是动态的

如果在 HADR 数据库处于联机状态时更改参数，所作的更改将在您对该数据库发出 db2 get db cfg 时可视。但是，这些更改在您停止并重新启动该数据库前不会生效。要检索当前正在起作用的参数，请使用 GET SNAPSHOT 命令、db2pd 工具或快照监视器 API。

HADR 数据库角色

数据库的当前角色由数据库配置参数 `hadr_db_role` 指示。此配置参数的有效值是 PRIMARY、STANDBY 或 STANDARD（后者指示数据库不是 HADR 数据库）。

备用数据库的状态

当数据库具有备用角色时，它还处于前滚暂挂状态。因此，备用数据库配置将指示：

前滚暂挂	= DATABASE
复原暂挂	= YES

响应意外中断

如果数据库管理软件或集群管理软件检测到某个数据库服务器出现故障，那么数据库解决方案必须尽快而且尽可能平稳地响应该故障。数据库解决方案必须通过重新路由工作负载（如果可能）并将故障转移到辅助或备用数据库（如果有），尝试对用户应用程序屏蔽该故障。

如果数据库或集群管理软件检测到数据库服务器出现故障，那么您或您的数据库或集群管理软件必须执行以下操作：

1. 确定辅助数据库服务器、使之联机并初始化，以接管故障数据库服务器的操作。

如果是使用 DB2 高可用性灾难恢复（HADR）来管理主数据库服务器和备用数据库服务器，HADR 将使备用数据库与主数据库保持同步；而且 HADR 将管理备用数据库对主数据库的接管。

2. 将用户应用程序工作负载重新路由至辅助数据库服务器。

DB2 客户机重新路由可以将客户机应用程序从故障数据库服务器自动重新路由至先前为此目的标识和配置的辅助数据库服务器。

3. 从系统除去故障数据库服务器以将它修复。

一旦将用户应用程序重新路由至辅助或备用数据库，故障数据库服务器将不能处理任何客户机应用程序请求，直到重新启动或修复。例如，如果主数据库上出现故障的原因是数据库实例意外终止，那么 DB2 故障监视设施将自动将它重新启动。

客户机自动重新路由示例

DB2 数据服务器客户机重新路由可以将客户机应用程序从故障数据库服务器自动重新路由至先前为此目的标识和配置的辅助数据库服务器。您可以方便地创建客户机应用程序来测试和演示此 DB2 数据服务器功能。

以下是客户机应用程序的自动客户机路由示例（仅使用伪码显示）：

```
int checkpoint = 0;

    check_sqlca(unsigned char *str, struct sqlca *sqlca)
{
    if (sqlca->sqlcode == -30081)
    {
        // as communication is lost, terminate the application
        right away
        exit(1);
    }
    else
```



```

        // print out the error
        printf(...);
    if (sqlca->sqlcode == -30108)
    {
        // connection is re-established, re-execute the failed
        // transaction
        if (checkpoint == 0)
        {
            goto checkpt0;
        }
        else if (checkpoint == 1)
        {
            goto checkpt1;
        }
        else if (checkpoint == 2)
        {
            goto checkpt2;
        }
        ....
        exit;
    }
}
}

main( )
{
    connect to mydb;
    check_sqlca("connect failed", &sqlca);

    checkpt0:
    EXEC SQL set current schema XXX;
    check_sqlca("set current schema XXX failed", &sqlca);

    EXEC SQL create table t1...;
    check_sqlca("create table t1 failed", &sqlca);

    EXEC SQL commit;
    check_sqlca("commit failed", &sqlca);

    if (sqlca.sqlcode == 0)
    {
        checkpoint = 1;
    }

    checkpt1:
    EXEC SQL set current schema YYY;
    check_sqlca("set current schema YYY failed", &sqlca);

    EXEC SQL create table t2...;
    check_sqlca("create table t2 failed", &sqlca);

    EXEC SQL commit;
    check_sqlca("commit failed", &sqlca);

    if (sqlca.sqlcode == 0)
    {
        checkpoint = 2;
    }
    ...
}

```

在客户机上，对名为“mydb”的数据库编目，此数据库引用节点“hornet”，其中“hornet”也在节点目录（主机名为“hornet”，端口号为 456）予以编目。

示例 1（涉及非 HADR 数据库）

在服务器“hornet”（主机名为 hornet，并且带有端口号）上，创建数据库“mydb”。而且，还在备用服务器处创建数据库“mydb”（主机名为“montero”，端口号为 456）。您还需要在服务器“hornet”上更新数据库“mydb”的备用服务器，如下所示：

```
db2 update alternate server for database mydb using hostname montero port 456
```

在上述样本应用程序中，未设置客户机自动重新路由功能，如果 `create table t1` 语句中存在通信错误，那么该应用程序将终止。设置客户机自动重新路由功能之后，DB2 数据库管理器将尝试再次建立与主机“hornet”（端口为 456）的连接。如果仍然无法工作，那么 DB2 数据库管理器将尝试备用服务器位置（主机名为“montero”，端口为 456）。假定连接至备用服务器位置时没有出现通信错误，那么应用程序可以继续运行后面的语句（并重新运行发生故障的事务）。

示例 2（涉及 HADR 数据库）

在服务器“hornet”（主机名为 hornet，并且带有端口号），创建主数据库“mydb”。另外还在端口为 456 的主机“montero”上创建备用数据库。有关如何为主数据库和备用数据库设置 HADR 的信息可于《数据恢复及高可用性指南与参考》中找到。您还需要为数据库“mydb”更新备用服务器，如下所示：

```
db2 update alternate server for database mydb using hostname montero port 456
```

在上述样本应用程序中，未设置客户机自动重新路由功能，如果 `create table t1` 语句中存在通信错误，那么该应用程序将终止。设置客户机自动重新路由功能之后，DB2 数据库系统将尝试再次建立与主机“hornet”（端口为 456）的连接。如果仍然无法工作，那么 DB2 数据库系统将尝试备用服务器位置（主机名为“montero”，端口为 456）。假定连接至备用服务器位置时没有出现通信错误，那么应用程序可以继续运行后面的语句（并重新运行发生故障的事务）。

示例 3（涉及 SSL）

当同时将 SSL 用于连接时，还可以使用客户机重新路由功能。该设置与针对示例 2 显示的设置类似。

在客户机上，对引用节点“hornet_ssl”的数据库“mydb”的数据库别名“mydb_ssl”编目。在节点目录（主机名为“hornet”，SSL 端口号为 45678 且安全性参数设置为 SSL）中对“hornet_ssl”编目。

在备用服务器（主机名为“montero”，SSL 端口号为 45678 且安全性参数设置为 SSL）上，也对某个数据库别名编目。还需要在服务器“hornet”上更新别名“mydb_ssl”的备用服务器，如下所示：

```
db2 update alternate server for database mydb_ssl using hostname montero port 45678
```

在上述样本应用程序中，将连接语句更改为 `connect to mydb_ssl`。在未设置客户机自动重新路由功能的情况下，如果 `create table t1` 语句中存在通信错误，那么该应用程序将终止。在设置了客户机自动重新路由功能的情况下，DB2 数据库管理器将尝试使用 SSL 再次建立与主机“hornet”（端口为 45678）的连接。如果仍然无法工作，那么 DB2 数据库管理器将使用 SSL 尝试备用服务器位置（端口 45678 处的主机“montero”）。假定连接至备用服务器位置时没有出现通信错误，那么应用程序可以继续运行后面的语句（并重新运行发生故障的事务）。

执行 HADR 故障转移操作

因为当前主数据库不可用，如果要当前备用数据库成为新的主数据库，那么可以执行故障转移。

警告:

此过程可能导致数据丢失。执行此紧急过程之前，复查以下信息:

- 确保主数据库不再处理数据库事务。如果主数据库仍在运行，但无法与备用数据库通信，那么执行强制接管操作（发出指定了 **BY FORCE** 选项的 **TAKEOVER HADR** 命令）会产生两个主数据库。当有两个主数据库时，每个数据库将拥有不同的数据，并且不能再自动同步两个数据库。
 - 如有可能，请取消激活主数据库或停止其实例。（如果主系统已挂起、崩溃或不可访问，那么可能无法执行此操作。）执行接管操作之后，如果稍后重新启动了发生故障的数据库，它将不会自动地承担主数据库的角色。
- 事务损失的可能性和程度取决于特定配置和情况:
 - 如果主数据库在对等状态或断开对等状态下发生故障，且同步方式为同步（**SYNC**），那么备用数据库将不会丢失在主数据库发生故障之前已报告且已落实到应用程序的事务。
 - 如果主数据库在对等状态或断开对等状态下发生故障，且同步方式为接近同步（**NEARSYNC**），那么当主数据库和备用数据库同时发生故障时，备用数据库只会丢失由主数据库落实的事务。
 - 如果主数据库在对等状态或断开对等状态下发生故障，且同步方式为异步（**ASYNC**），那么在执行接管操作前备用数据库未接收事务的所有日志记录的情况下，此备用数据库可能丢失由主数据库落实的事务。如果主数据库和备用数据库同时发生故障，备用数据库也会丢失由主数据库落实的事务。
 - 如果主数据库在远程同步复制暂挂状态下发生故障，将丢失备用数据库尚未接收和处理的事务。

注： 数据库快照中显示的任何日志间隔将表明主数据库和备用数据库最后一次互相通信的间隔。从那时起，主数据库可能已处理了大量事务。

- 确保任何连接至新的主数据库（或者通过客户机重新路由功能来重新路由至新的主数据库）的应用程序已准备好处理下列情况:
 - 在故障转移期间丢失了数据。新的主数据库没有所有已在旧的主数据库上落实的事务。即使在 **HADR_SYNCMODE** 配置参数设置为 **SYNC** 时，也会发生这种情况。由于 **HADR** 备用数据库按顺序应用日志，所以您可以假定以下情况：如果在新的主数据库上落实 **SQL** 会话中的某个事务，那么已在这个新的主数据库上落实了同一会话中的所有先前事务。只能通过详细地分析日志流来确定跨多个会话的事务落实顺序。
 - 可能会发生以下情况：对原始主数据库发出了一个事务，在原始主数据库上已落实该事务并将其复制至新的主数据库（原始备用数据库），但由于原始主数据库在向客户机报告已落实事务之前崩溃，所以未将该事务报告为已落实。您编写的任何应用程序都应该能够处理以下情况：对原始主数据库发出了事务，但未报告已在原始主数据库上落实这些事务，将在新的主数据库（原始备用数据库）上落实这些事务。
 - 某些操作（例如对数据库配置和外部 **UDF** 对象所作的更改）不会被复制。
- 只能对备用数据库发出 **TAKEOVER HADR** 命令。

- HADR 无法与 DB2故障监视器 (db2fm) 配合使用, 此故障监视器可用于自动重新启动发生故障的数据库。如果启用故障监视器, 应该知道在可能发生故障的主数据库上可能执行的故障监视器操作。
- 仅当主数据库和备用数据库处于对等状态, 或备用数据库处于远程同步复制暂挂状态时, 接管操作才会发生。如果备用数据库处于任何其他状态, 将返回错误。

注: 通过将处于本地同步复制状态的备用数据库转换为标准数据库, 可以使其正常使用。要执行此操作, 通过发出 `DEACTIVATE DATABASE` 命令来关闭数据库, 然后发出 `STOP HADR` 命令。一旦停止了 HADR, 就必须在前述的备用数据库上完成前滚操作, 才能使用它。当数据库从备用数据库转换为标准数据库之后, 该数据库不能重新加入 HADR 数据库对。要在两台服务器上重新启动 HADR, 请执行初始化 HADR 的过程。

如果配置了对等时间, 请在该时间到期前关闭主数据库, 以避免在任何相关故障转移中丢失潜在事务。

在故障转移方案中, 可以通过命令行处理器 (CLP)、控制中心中的“管理高可用性灾难恢复”窗口或 `db2HADRTakeover` 应用程序编程接口 (API) 来执行接管操作。

以下过程说明如何使用 CLP 在主数据库或备用数据库上初始化故障转移:

1. 完全禁用发生故障的主数据库。当数据库遇到内部错误时, 正常关闭命令可能无法完全关闭该数据库。您可能需要使用操作系统命令来除去资源, 例如进程、共享内存或网络连接。
2. 在备用数据库上, 发出指定了 `BY FORCE` 选项的 `TAKEOVER HADR` 命令。在以下示例中, 故障转移发生在数据库 `LEAFS` 上:

```
TAKEOVER HADR ON DB LEAFS BY FORCE
```

需要使用 `BY FORCE` 选项将主数据库脱机。

如果未完全禁用主数据库, 备用数据库仍会与主数据库连接, 并且将消息发送到主数据库以请求其关闭。无论备用数据库是否接收到主数据库已关闭的确认信息, 它仍将切换为主数据库的角色。

要打开“接管 HADR”窗口:

1. 在控制中心中, 展开对象树, 直到找到要为其管理 HADR 的数据库为止。用鼠标右键单击此数据库, 然后在弹出菜单中单击“高可用性灾难恢复”→“管理”。这就打开了“管理高可用性灾难恢复”窗口。
2. 单击“接管 HADR”。这就打开了“接管 HADR”窗口。
3. 选择要执行故障转移操作。
4. 如果 HADR 数据库对中的两个数据库都是作为备用数据库启动的, 那么选择其中一个数据库来接管主数据库角色。
5. 单击“确定”。该窗口将关闭。可能会显示进度指示器以指示正在运行命令。该指示器完成后, 您将看到指示命令成功与否的通知。
6. 刷新“管理高可用性灾难恢复”窗口以确保备用数据库已切换为新的主数据库。
7. 如果未使用客户机自动重新路由功能, 那么可以将客户机应用程序重定向到新的主数据库。

在控制中心中通过联机帮助工具提供了详细信息。

切换高可用性灾难恢复（HADR）中的数据库角色

在高可用性灾难恢复（HADR）期间，使用 TAKEOVER HADR 命令来切换主数据库和备用数据库的角色。

- 只能对备用数据库发出 TAKEOVER HADR 命令。如果发出此命令时主数据库尚未与备用数据库连接，接管操作就会失败。
- 如果数据库处于对等状态，那么只能使用 TAKEOVER HADR 命令来切换主数据库和备用数据库的角色。如果备用数据库处于其他任何状态，就会返回错误消息。

可以使用命令行处理器（CLP）、控制中心中的“管理高可用性灾难恢复（HADR）”窗口或 db2HADRTakeover 应用程序编程接口（API）来切换 HADR 数据库角色。

要使用 CLP 来对备用数据库启动接管操作，请对备用数据库发出不带 BY FORCE 选项的 TAKEOVER HADR 命令。

在以下示例中，对备用数据库 LEAFS 执行接管操作：

```
TAKEOVER HADR ON DB LEAFS
```

在接管操作完成后，紧跟着发生日志满错误的可能性会略高。为了降低此类错误的可能性，在每次接管完成时都会自动启动异步缓冲池清空操作。随着异步缓冲池清空操作的进展，日志满错误的可能性也就随之降低。此外，如果配置提供了足够的活动日志空间量，就更不大可能会发生日志满错误。如果确实发生了日志满错误，当前事务将中止并回滚。

注：发出不带 BY FORCE 选项的 TAKEOVER HADR 命令将导致任何当前已连接至 HADR 主数据库的应用程序被强制断开。此操作设计成与客户机自动重新路由功能配合工作，以便于在角色切换后将客户机重新路由到新的 HADR 主数据库。但是，如果强制断开应用程序与主数据库的连接对环境有破坏性，您可以实现自己的过程来在执行角色切换前关闭此类应用程序，接着在角色切换完成后重新启动这些应用程序并使用新的 HADR 主数据库作为它们的目标。

要打开“接管 HADR”窗口：

1. 在控制中心中，展开对象树，直到找到要为其管理 HADR 的数据库为止。用鼠标右键单击此数据库，然后在弹出菜单中单击“高可用性灾难恢复”->“管理”。这就打开了“管理高可用性灾难恢复”窗口。
2. 确保数据库处于对等状态。
3. 单击“接管 HADR”。这就打开了“接管 HADR”窗口。
4. 选择要切换数据库角色。
5. 如果 HADR 数据库对中的两个数据库都是作为备用数据库启动的，那么选择其中一个数据库来接管主数据库角色。
6. 单击“确定”。该窗口将关闭。可能会显示进度指示器以指示正在运行命令。该指示器完成后，您将看到指示命令成功与否的通知。
7. 刷新“管理高可用性灾难恢复”窗口以确保数据库已切换角色。
8. 如果未使用客户机自动重新路由功能，那么可以将客户机应用程序重定向到新的主数据库。

控制中心中的上下文帮助工具提供了其他信息。

在执行接管操作之后重新集成数据库

如果因为主数据库故障而在高可用性灾难恢复（HADR）环境中执行了接管操作，那么可以将发生故障的数据库恢复联机，并将此数据库用作备用数据库或将其返回至作为主数据库时的状态。

要将发生故障的主数据库作为新的备用数据库重新集成到 HADR 数据库对中：

1. 修复原始主数据库所在的系统。这可能涉及修复发生故障的硬件或重新引导已崩溃的操作系统。
2. 将发生故障的主数据库作为备用数据库重新启动。在以下示例中，将数据库 LEAFS 作为备用数据库启动：

```
START HADR ON DB LEAFS AS STANDBY
```

注：如果数据库的两个副本的日志流不兼容，那么重新集成操作将失败。特别是，HADR 要求在原始主数据库上应用的操作不包括：在原始备用数据库作为新的主数据库接管前，从未反映在该数据库上的已进行日志记录的操作。如果发生此种情况，通过复原新主数据库的备份映像，或者通过初始化分割镜像，可以将原始主数据库作为备用数据库重新启动。

此命令成功返回并不意味着重新集成操作已成功；这仅仅表示已启动数据库。重新集成操作仍在进行中。以后，如果重新集成操作失败，该数据库将自行关闭。应该使用 `GET SNAPSHOT FOR DATABASE` 命令或 `db2pd` 工具来监视备用数据库状态，以确保备用数据库保持处于联机状态并进行正常的状态过渡。必要时，可以检查管理通知日志文件和 `db2diag` 日志文件以了解数据库状态。

原始主数据库作为备用数据库重新加入 HADR 数据库对之后，可以选择执行故障恢复操作来切换数据库角色，以再次将原始主数据库启用为主数据库。要执行此故障回退操作，在备用数据库上发出以下命令：

```
TAKEOVER HADR ON DB LEAFS
```

注：

1. 如果 HADR 数据库未处于对等状态或者在数据库对之间未建立连接，那么此命令将失败。
2. 强制关闭在主数据库上打开的会话，并回滚未完成的事务。
3. 当切换主数据库和备用数据库的角色时，无法指定 `TAKEOVER HADR` 命令的 `BY FORCE` 选项。

第 2 部分 数据恢复

恢复指的是在发生问题（例如介质或存储器故障、断电或者应用程序故障）后重建数据库或表空间。如果已经备份了数据库或各个表空间，那么可以在它们由于某种原因损坏时重建它们。

有三种类型的恢复操作：

- 崩溃恢复保护数据库在事务（也称为工作单元）意外中断后不会处于不一致或不可用的状态。
- 版本恢复指的是使用备份操作期间创建的映像来复原数据库的先前版本。
- 前滚恢复可以用来重新应用创建备份后落实的事务所作的更改。

断电后，DB2 数据库管理器会自动启动崩溃恢复以尝试恢复数据库。可以使用版本恢复或前滚恢复功能来恢复损坏的数据库。

第 7 章 开发备份和恢复策略

数据库可能会因为硬件和/或软件故障而不可用。您可能会不时遇到存储问题、断电或应用程序故障以及各种需要采取不同恢复措施的故障情况。如果已准备了一个进行过彻底演习的恢复策略，它可保护您的数据免被丢失。在开发恢复策略时，您应该回答的一些问题包括：

- 数据库是可恢复的吗？
- 恢复数据库可能花费多长时间？
- 在备份操作之间将花费多长时间？
- 可以为备份副本和归档日志分配多少存储空间？
- 表空间级的备份是否足够？或是否需要完整的数据库备份？
- 是否应该通过手动或高可用性灾难恢复（HADR）来配置备用系统？

数据库恢复策略应确保在数据库恢复操作需要时，所有信息都可用。它应包括一个进行数据库备份的固定时间表，在分区数据库环境中则包括缩放系统时（添加或删除数据库分区服务器或节点时）的备份。完整的策略还应包括恢复命令脚本、应用程序、用户定义的函数（UDF）、操作系统库中的存储过程代码以及装入副本的过程。

在以下几节中还讨论了不同的恢复方法，您将能发现最适用于您的业务环境的恢复方法。

数据库备份的概念与其他任何数据备份的概念一样：即，复制一份数据，然后将它存储在另一介质上，以防原始介质发生故障或毁坏。最简单的备份情况需要关闭数据库（以确保不发生更多的事务），然后简单地对其进行备份。以后，如果数据库被损坏或毁坏，就可以重新创建。

数据库的重新创建称为恢复。版本恢复指的是使用备份操作期间创建的映像来复原数据库的先前版本。前滚恢复是指复原了数据库或表空间备份映像后，重新应用记录在数据库日志文件中的事务。

崩溃恢复是指在完成并落实所有更改（这些更改是一个或多个工作单元（事务）的一部分）之前如果发生故障，会自动恢复数据库。这是通过回滚未完成的事务，并完成在发生崩溃时仍在内存中的已落实事务来实现的。

恢复日志文件和恢复历史记录文件是在创建数据库时自动创建的（第 152 页的图 10）。在需要恢复丢失或损坏的数据时，这些日志文件是很重要的。

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。

恢复历史记录文件包含当数据库的所有或部分必须恢复到给定时间点时，可用来确定恢复选项的备份信息的摘要。恢复历史记录文件用来跟踪其他操作中与恢复相关的事件，如备份与复原操作。此文件位于数据库目录中。

表空间更改历史记录文件（它也位于数据库目录中）包含可用来确定哪些日志文件对特定表空间的恢复是必需的信息。

不能直接修改恢复历史记录文件或表空间更改历史记录文件；但是可以使用 PRUNE HISTORY 命令来删除文件中的条目。还可以使用 *rec_his_retentn* 数据库配置参数来指定这些历史记录文件将保留的天数。

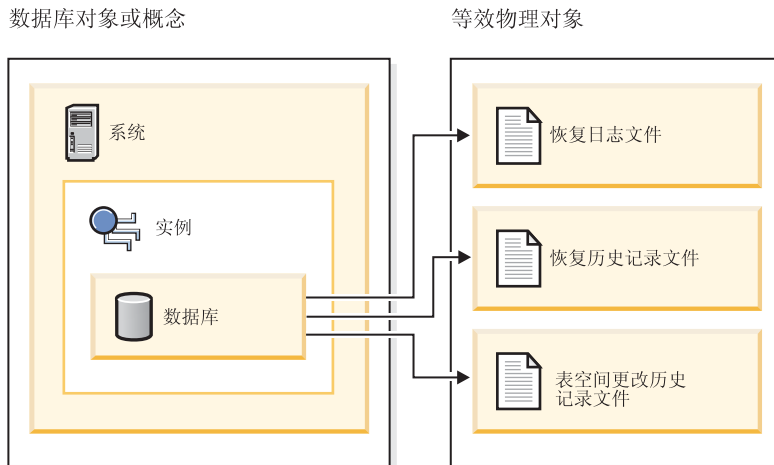


图 10. 数据库恢复文件

那些很容易重新创建的数据可存储在不可恢复数据库中。这些数据包括：用于只读应用程序的外部源中的数据以及不常进行更新的表；由于对它们进行的日志记录量较小，如果在复原操作之后还要进行复杂的日志文件管理工作和前滚操作，就不太合理了。如果 *logarchmeth1* 和 *logarchmeth2* 数据库配置参数都设置为“OFF”，那么该数据库是不可恢复的。这表明将只保存崩溃恢复所必需的日志。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。它被复原为生成备份映像时的状态且不支持前滚恢复。

那些不容易重新创建的数据应存储在可恢复的数据库中。这些数据包括在装入后它的源已被破坏的数据、手动输入到表中的数据以及在装入数据库后由应用程序或用户修改的数据。可恢复数据库将 *logarchmeth1* 或 *logarchmeth2* 数据库配置参数设置为“OFF”以外的值。活动日志仍可用于崩溃恢复，但您还有已归档日志，它包含已落实的事务数据。这样的数据库只能进行脱机复原。它被复原为创建备份映像时的状态。但对于前滚恢复，可通过使用活动日志和已归档日志将数据库前滚（即，越过创建备份映像的时间）至特定的时间点，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）。仅当数据库可恢复时，才支持联机表空间复原和前滚操作。如果数据库是不可恢复的，必须脱机执行数据库复原和前滚操作。联机备份操作期间，前滚恢复确保捕获了所有表更改，且在复原该备份时重新应用这些更改。

如果有一个可恢复数据库，那么可备份、复原并将各个表空间前滚，而不必对整个数据库操作。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，但不阻止用户访问其他表空间中的表。

自动备份操作

因为确定是否及何时运行维护活动（例如，备份操作）可能很费时间，所以可以使用“配置自动维护”向导来执行此操作。通过自动维护指定维护目标（包括自动维护可以运行的时间）。然后，DB2 使用这些目标来确定是否需要执行维护活动，然后在下一可用维护窗口期间（用户定义的、运行自动维护活动的时间段）仅运行所需的维护活动。

注：配置自动维护时，仍可执行手动备份操作。如果需要自动备份操作，DB2 将只执行这些自动备份操作。

确定备份频率

因为备份数据库时需要耗用时间和系统资源，所以恢复计划应该允许定期地安排备份操作。恢复计划可能既包括完整数据库备份操作也包括增量备份操作。

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）。要恢复数据库，可以使用包含所有表空间备份映像的完整数据库备份映像，也可以使用选择的表空间映像来重建数据库。表空间备份映像对于从被隔离的磁盘故障或应用程序错误进行恢复时很有用。在分区数据库环境中，只需要复原发生故障的数据库分区上的表空间，而不需要复原所有表空间或所有数据库分区。

现在，虽然因为可以根据表空间映像重建数据库而使数据库恢复不再需要完整数据库备份，但最好还是偶尔地创建完整数据库备份。

还应考虑不要覆盖备份映像和日志，应保存至少两个完整的数据库备份映像及其相关的日志，作为额外的预防措施。

如果在恢复和前滚非常活跃的数据库时，应用归档日志所需的时间是主要关注的问题，应考虑进行更频繁的数据库备份所需的成本。进行更频繁的数据库备份可减少前滚时需要应用的归档日志数目。

数据库在联机或脱机时都可启动备份操作。如果它是联机的，在运行备份操作的同时，其他应用程序或进程可以与该数据库连接并读取和修改数据。如果备份操作是脱机运行的，那么其他应用程序不能与数据库连接。

要缩短数据库处于不可用状态的时间，应考虑使用联机备份操作。仅当启用前滚恢复时，才支持联机备份操作。如果已启用前滚恢复，并且有一组完整的恢复日志，那么可以在需要时复原数据库。如果您的日志跨越了运行备份操作的时间段，那么只能使用联机备份映像进行恢复。

脱机备份操作比联机备份操作更快，因为不存在数据文件的争用。

Backup 实用程序使您可备份所选的表空间。如果使用 DMS 表空间，那么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。如果执行此操作并且磁盘发生故障，那么可能只影响其中一个表空间。复原或前滚其中一个表空间将比复原包含所有数据的单个表空间花费更少时间。

还可以通过在不同时间对不同表空间进行备份来节省时间，只要对它们的更改不相同。因此，如果长字段或 LOB 数据不像其他数据那样作频繁地更改，那么可以不那么

频繁地备份这些表空间。如果长字段和 LOB 数据不是恢复所必需的，那么可以考虑不备份包含该数据的表空间。如果可从单独的源复制 LOB 数据，那么当创建或改变一个表以包括 LOB 列时选择 NOT LOGGED 选项。

注：考虑以下情况，如果将长字段数据、LOB 数据和索引放在单独的表空间中，而备份时不把它们放在一起：如果备份的表空间未包含所有表数据，那么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。

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

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，如果允许数据库进行正向恢复，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要求。在每次测试之后，可能要提高创建备份的频率。如果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。

恢复的存储器注意事项

当确定要使用的恢复方法时，请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备份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备份副本、复原的数据库和归档的数据库日志。

如果一个表包含长型字段或大对象（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注意事项，并影响恢复计划。如果使用单独的表空间来存储长型字段和 LOB 数据，并知道备份长型字段和 LOB 数据所需的时间，那么可以决定使用仅以较低的频率保存此表空间备份的恢复计划。在创建或改变一个表以包括 LOB 列时，也可选择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。

要防止介质故障破坏数据库并因此导致无法复原它，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設備中。

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

- 使用 LOGARCHMETH1 或 LOGARCHMETH2 配置参数指定日志归档方法。
- 当某些日志不再出现在活动的日志集中后，以手动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。
- 使用用户出口程序将这些日志复制到您环境中的另一个存储设备中。

将相关数据保存在一起

在设计数据库的过程中，您将了解各个表之间存在的关系。可以在下列级别上表示这些关系：

- 应用程序级别（当事务更新多个表时）

- 数据库级别（当各个表之间存在引用完整性时，或者当一个表上的触发器影响另一个表时）。

当制定恢复计划时，应该考虑这些关系。您要将相关的数据集一起备份。可以在表空间级别或数据库级别上建立这样的集合。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。如果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。

不同操作系统和硬件平台之间的备份和复原操作

DB2 数据库系统支持在不同操作系统和硬件平台之间执行一些备份和复原操作。

可以将支持 DB2 备份和复原操作的平台分为以下三种系列：

- 大尾数法 Linux 和 UNIX
- 小尾数法 Linux 和 UNIX
- Windows

一种平台系列的数据库备份只能在相同平台系列的任何系统上复原。对于 Windows 操作系统，可以在 DB2 版本 9 数据库系统上复原在 DB2 通用数据库（UDB）V8 上创建的数据库。对于 Linux 和 UNIX 操作系统，只要备份和复原平台的字节存储次序（大尾数法或小尾数法）相同，就可以在 DB2 版本 9 上复原在 DB2 UDB V8 上产生的备份。

下表显示了 DB2 支持的每个 Linux 和 UNIX 平台，并指示了这些平台是采用大尾数法还是小尾数法：

表 9. DB2 支持的 Linux 和 UNIX 操作系统的字节存储次序

平台	字节存储次序
AIX	大尾数法
HP on IA64	大尾数法
Solaris x64	小尾数法
Solaris SPARC	大尾数法
Linux on zSeries®	大尾数法
Linux on pSeries®	大尾数法
Linux on IA-64	小尾数法
Linux on AMD64 和 Intel® EM64T	小尾数法
32 位 Linux on x86	小尾数法

目标系统必须与源系统具有相同（或比源系统更新的）DB2 数据库产品。不能将在一个数据库产品版本中创建的备份复原到运行较低版本的数据库产品的系统。例如，可以在 DB2 V9 数据库系统上复原 DB2 UDB V8 备份，但不能在 DB2 UDB V8 数据库系统上复原 DB2 V9 备份。

注：可以将数据库从在 32 位级别创建的备份映像复原为 64 位级别，但反之则不然。DB2 BACKUP 实用程序和 RESTORE 实用程序应该用来备份和复原数据库。建议不要将文件集从一台机器移至另一台机器，因为这可能破坏数据库的完整性。

在不允许某些备份和复原组合的情况下，可以使用其他方法在 DB2 数据库之间移动表：

- db2move 命令
- 使用 EXPORT 实用程序, 接着使用 IMPORT 或 LOAD 实用程序

第 8 章 恢复历史记录文件

恢复历史记录文件是与每个数据库一起创建的，且在发生下列情况时自动更新：

- 备份了数据库或表空间
- 复原了数据库或表空间
- 前滚了数据库或表空间
- 自动重建了数据库，并且复原了多个映像
- 创建了表空间
- 改变了表空间
- 停顿表空间
- 重命名表空间
- 删除表空间
- 装入表
- 删除表（启用已删除表恢复时）
- 重组表
- 调用按需进行的日志归档
- 写入新日志文件（使用可恢复日志记录时）
- 归档日志文件（使用可恢复日志记录时）
- 恢复数据库

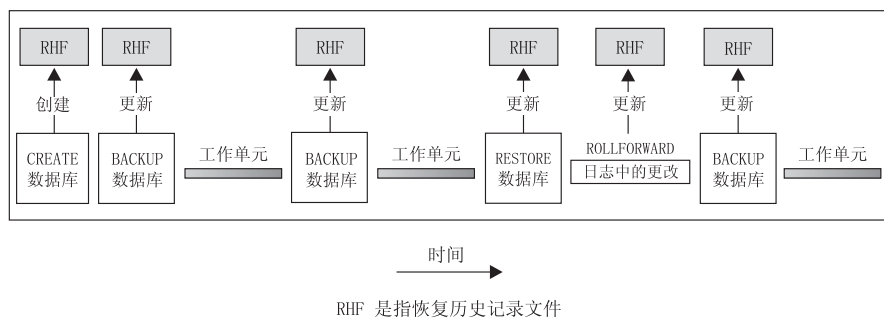


图 11. 创建和更新恢复历史记录文件

可以使用此文件中汇总的备份信息，将数据库的全部或部分恢复至给定的时间点。该文件中的信息包括：

- 唯一地标识每个条目的标识（ID）字段
- 已复制的部分数据库和复制方法
- 建立副本的时间
- 副本的位置（指示设备信息和访问副本的逻辑方式）
- 进行上次复原操作的时间

- 重命名表空间的时间，显示了该表空间的先前名称和当前名称
- 备份操作的状态：活动、不活动、到期的或删除的
- 数据库备份保存的或前滚恢复操作期间处理的最后一个日志序号。

要查看恢复历史记录文件中的条目，可使用 `LIST HISTORY` 命令。

每个备份操作（数据库备份、表空间备份或增量备份）都包括复制恢复历史记录文件。该恢复历史记录文件与数据库相关联。删除数据库会删除恢复历史记录文件。将数据库复原到新位置会复原该恢复历史记录文件。复原不会覆盖现有恢复历史记录文件，除非磁盘上的文件没有任何条目。如果是这种情况，将根据备份映像复原数据库历史记录。

如果当前数据库不能使用或不可用，且相关的恢复历史记录文件被损坏或被删除，那么 `RESTORE` 命令中的一个选项只允许复原恢复历史记录文件。这样，可以复查该恢复历史记录文件，以提供有关要用于复原该数据库的备份的信息。

该文件的大小由 `rec_his_retentn` 配置参数控制，该参数指定该文件中条目的保留期（以天计）。即使将此参数的值设置为零（0），仍会保留最新的完整数据库备份（加上它的复原集）。（除去此副本的唯一方法是使用带 `FORCE` 选项的 `FRUNE`。）保留期的缺省值是 366 天。可使用 `-1` 将保留期设置为不确定的天数。在这种情况下，需要显式地删除该文件。

恢复历史记录文件条目状态

数据库管理器在恢复历史记录文件中对备份操作、复原操作、表空间创建以及其他事件创建条目。恢复历史记录文件中的每个条目都有关联的状态：不活动、到期、已删除或 `do_not_delete`。

数据库管理器使用恢复历史记录文件条目的状态来确定是否需要与该条目关联的物理文件来恢复数据库。作为自动修剪的一部分，数据库管理器会更新恢复历史记录文件条目的状态。

活动数据库备份

活动的数据库备份是可使用当前日志进行复原和前滚，以恢复该数据库的当前状态的一种备份。

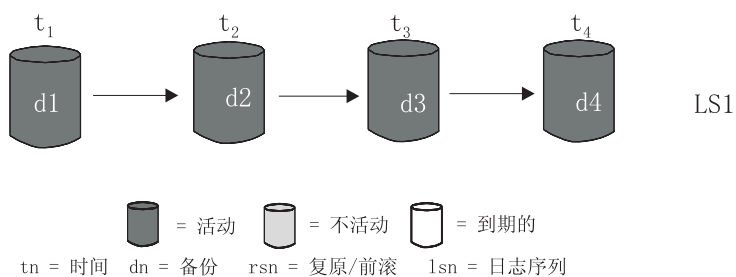


图 12. 活动数据库备份. `num_db_backups` 的值已设置为 4.

不活动数据库备份

而不活动的数据库备份是在复原时会将数据库移动回先前状态的一种备份。

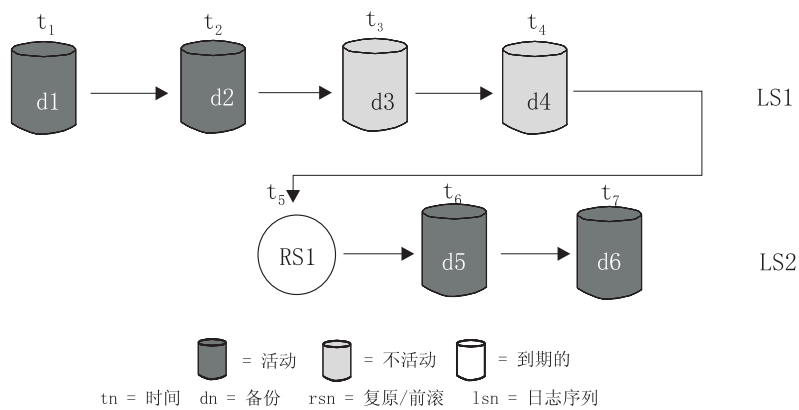


图 13. 不活动数据库备份

到期数据库备份

到期数据库备份映像是不再需要的映像，因为有更新的备份映像可用。

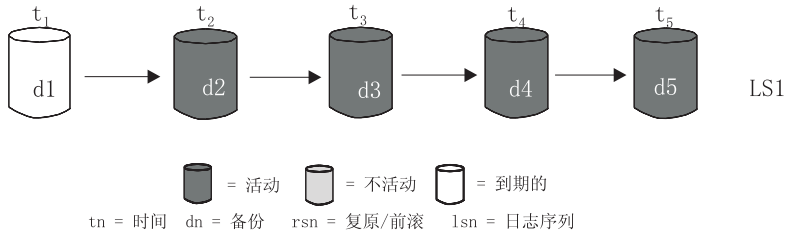


图 14. 到期数据库备份

标记为 `do_not_delete` 的条目

可以使用 `PRUNE HISTORY` 命令或 `db2Prune API` 除去（修剪）恢复历史记录文件条目。作为自动修剪的一部分，数据库管理器还会修剪恢复历史记录文件条目。

修剪标记为 `do_not_delete` 的条目只有三种方法：

- 带 `WITH FORCE` 选项调用 `PRUNE HISTORY` 命令
- 带 `PRUNE HISTORY` 和 `WITH FORCE` 选项调用 `ADMIN_CMD` 过程
- 带 `DB2_PRUNE_OPTION_FORCE` 选项调用 `db2Prune API`

除非执行这三种操作之一，否则永远不会从恢复历史记录文件中修剪标记为 `do_not_delete` 的条目。

数据库管理器不会将恢复历史记录文件条目的状态设置为 `do_not_delete`。您可以使用 `UPDATE HISTORY` 命令将恢复历史记录文件条目的状态设置为 `do_not_delete`。

以下是有关不同恢复历史记录文件条目的状态的更多示例：

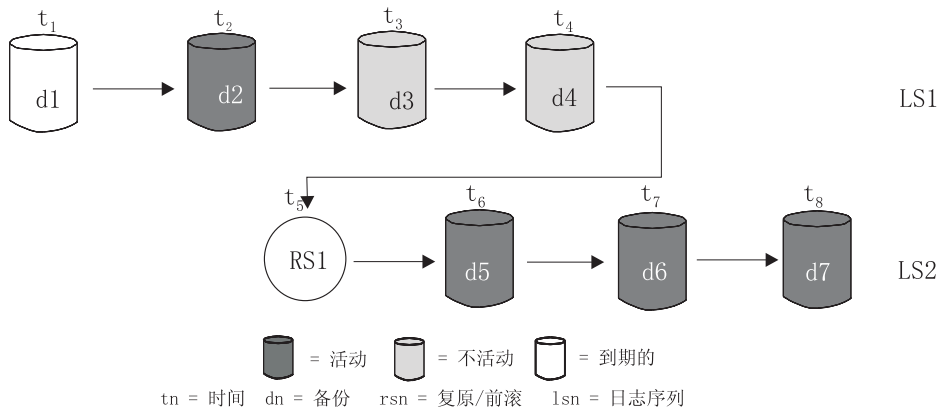


图 15. 混合活动的、不活动的和到期的数据库备份

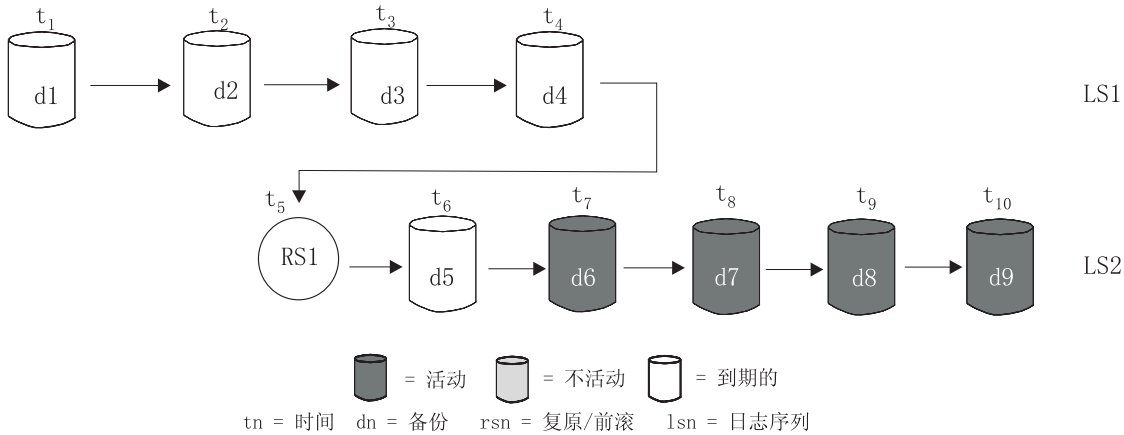


图 16. 到期日志序列

使用 DB_HISTORY 管理视图查看恢复历史记录文件条目

可以使用 DB_HISTORY() 管理视图来访问数据库历史记录文件的内容。此方法是使用 LIST HISTORY CLP 命令或 C 历史 API 的备用方法。

使用此函数需要数据库连接。

对数据库历史记录文件执行删除和更新操作只能通过 PRUNE 或 UPDATE HISTORY 命令进行。

此管理视图不可用于使用 DB2 通用数据库 版本 8.2 及更早版本创建的数据库。

要访问历史记录文件:

1. 确保已连接至数据库。
2. 在 SQL SELECT 语句中使用 DB_HISTORY() 管理视图来访问您连接至的数据库的数据库历史记录文件或 DB2NODE 环境指定的数据库分区上的数据库历史记录文件。例如, 要查看历史记录文件的内容, 可使用以下命令:

```
SELECT * FROM TABLE(DB_HISTORY()) AS LIST_HISTORY
```

要隐藏管理视图的语法, 可以按如下所示创建视图:

```
CREATE VIEW LIST_HISTORY AS
  SELECT * FROM TABLE(DB_HISTORY()) AS LIST_HISTORY
```

在创建了此视图之后, 可以对此视图运行查询。例如:

```
SELECT * FROM LIST_HISTORY
```

或者

```
SELECT dbpartitionnum FROM LIST_HISTORY
```

或者

```
SELECT dbpartitionnum, start_time, seqnum, tabname, sqlstate  
FROM LIST_HISTORY
```

表 10 列示了 LIST_HISTORY 表函数返回的列和列数据类型。

表 10. 历史表的内容

列名	数据类型
dbpartitionnum	smallint
EID	bigint
start_time	char(14)
seqnum	smallint
end_time	varchar(14)
firstlog	varchar(254)
lastlog	varchar(254)
backup_id	varchar(24)
tabschema	varchar(128)
tabname	varchar(128)
comment	varchar(254)
cmd_text	clob(2M)
num_tbsps	integer
tbspnames	clob(5M)
operation	char(1)
operationtype	char(1)
objecttype	varchar(255)
location	char(1)
devicetype	char(1)
entry_status	varchar(8)
sqlcaid	varchar(8)
sqlcab	integer
sqlcode	integer
sqlerrml	smallint
sqlerrmc	varchar(70)
sqlerrp	varchar(8)
sqlerrd1	integer
sqlerrd2	integer
sqlerrd3	integer
sqlerrd4	integer
sqlerrd5	integer
sqlwarn	varchar(11)
sqlstate	varchar(5)

修剪恢复历史记录文件

数据库管理器在恢复历史记录文件中对备份操作、复原操作、表空间创建以及其他事件创建条目。当由于不再需要关联的恢复对象来恢复数据库而导致恢复历史记录文件中的条目不再相关时，可能要从恢复历史记录文件中除去或修剪这些条目。

可以使用下列方法修剪恢复历史记录文件中的条目：

- 调用 PRUNE HISTORY 命令
- 调用 db2Prune API
- 带 PRUNE_HISTORY 参数调用 ADMIN_CMD 过程

当使用这些方法之一来修剪恢复历史记录文件时，数据库管理器会从恢复历史记录文件中除去（修剪）早于指定时间戳记的条目。

如果某个恢复历史记录文件条目与指定修剪的条件匹配，但恢复数据库仍然需要该条目，那么除非使用 WITH FORCE 参数或 DB2PRUNE_OPTION_FORCE 标志，否则数据库管理器不会修剪该条目。

如果使用 AND DELETE 参数或 DB2PRUNE_OPTION_DELETE 标志，那么与修剪的条目关联的日志文件也会被删除。

如果将 AUTO_DEL_REC_OBJ 数据库配置参数设置为 ON，并使用 AND DELETE 参数或 DB2PRUNE_OPTION_DELETE 标志，那么将删除与修剪的条目关联的日志文件、备份文件和装入副本映像。

使恢复历史记录文件修剪自动进行

可以将数据库管理器配置为自动修剪恢复历史记录文件条目并更新其状态。

可以使用 UPDATE HISTORY 命令、db2HistoryUpdate API 或带“UPDATE_HISTORY”参数的 ADMIN_CMD 过程来手动更新恢复历史记录文件条目的状态。可以使用 PRUNE HISTORY 命令、db2Prune API 或带“PRUNE_HISTORY”参数的 ADMIN_CMD 过程来从恢复历史记录文件中手动除去或修剪条目。但是，建议您配置数据库管理器来管理恢复历史记录文件，代替手动更新和修剪恢复历史记录文件。

数据库管理器会在下列情况下自动更新和修剪恢复历史记录文件条目：

- 在成功完成完整的（非增量）数据库备份操作或完整的（非增量）表空间操作后
- 在成功完成不需要前滚操作的数据库复原操作后
- 在成功完成数据库前滚操作后

在自动修剪期间，数据库管理器执行两项操作：

1. 更新恢复历史记录文件条目的状态
2. 修剪到期的恢复历史记录文件条目

数据库管理器通过以下方式更新恢复历史记录文件条目：

- 所有不再需要的活动数据库备份映像都标记为“已到期”。
- 所有标记为“不活动”并且在已到期数据库备份之前建立的数据库备份映像，也标记为“已到期”。所有相关的“不活动”表空间备份映像和装入副本也标记为“已到期”。

- 如果复原了一个活动的数据库备份映像，但它不是历史记录文件中记录的最新数据库备份，那么属于同一个日志序列的任何后续数据库备份映像将被标记为“不活动的”。
- 如果复原了一个不活动的数据库备份映像，那么属于当前日志序列的任何不活动的数据库备份都被再次标记为“活动的”。不再在当前日志序列中的所有数据库备份映像都标记为“不活动的”。
- 与当前日志序列（也称当前日志链）不对应的任何数据库或表空间备份映像都标记为“不活动”。

当前日志序列由已复原的数据库备份映像和已处理的日志文件确定。一旦数据库备份映像复原后，所有后继的数据库备份映像都变得“不活动”，因为已复原的映像将开始一个新的日志链。（在复原备份映像时没有前滚操作的情况下就是这样。如果已发生了前滚操作，那么日志链中在中断后建立的所有数据库备份都将标记为“不活动”。可以想像，由于已对包含已损坏的当前备份映像的整个日志序列执行了 Rollforward 实用程序，所以必须复原旧的数据库备份映像。）

- 如果在表空间级的备份映像复原后应用当前日志序列无法达到数据库当前状态，该映像变为“不活动”。
- 不修剪状态为 `do_not_delete` 的任何条目，也不删除其关联的日志文件、备份映像和装入副本映像。
- 升级数据库时，历史记录文件中的所有联机数据库备份条目和所有联机或脱机表空间备份条目都标记为“已到期”，以便自动重建操作不会选择这些条目作为重建所需的映像。由于装入副本映像和日志归档条目不能用于恢复目的，所以也会将这些条目标记为“已到期”。

下列数据库配置参数控制数据库管理器修剪哪些条目：

num_db_backups

指定为数据库保留的数据库备份的数目

rec_his_retentn

指定将保留关于备份的历史记录信息的天数

auto_del_rec_obj

指定数据库管理器是否应删除与修剪的恢复历史记录文件条目关联的日志文件、备份映像和装入副本映像

要将数据库管理器配置为自动管理恢复历史记录文件，请设置下列配置参数：

- `num_db_backups`
- `rec_his_retentn`
- `auto_del_rec_obj`

如果 `auto_del_rec_obj` 设置为 ON，那么一旦有比 `num_db_backups` 配置参数更多的成功数据库备份条目，数据库管理器将自动修剪比 `rec_his_retentn` 旧的恢复历史记录文件条目。

防止恢复历史记录文件条目被修剪

通过将恢复历史记录文件条目的状态设置为 `do_not_delete`，可以防止重要恢复历史记录文件条目被修剪和关联的恢复对象被删除。

可以使用 PRUNE HISTORY 命令、带 PRUNE_HISTORY 的 ADMIN_CMD 过程或 db2Prune API 除去（修剪）恢复历史记录文件条目。如果将 AND DELETE 参数与 PRUNE HISTORY 一起使用，或者将 DB2PRUNE_OPTION_DELETE 标志与 db2Prune 一起使用，而且 AUTO_DEL_REC_OBJ 数据库配置参数设置为 ON，那么关联的恢复对象也将被物理地删除。

作为自动修剪的一部分，数据库管理器还会修剪恢复历史记录文件条目。如果 AUTO_DEL_REC_OBJ 数据库配置参数设置为 ON，那么数据库管理器将删除与被修剪的任何条目关联的恢复对象。

要保护重要恢复历史记录文件条目和关联的恢复对象：

使用 UPDATE HISTORY 命令、db2HistoryUpdate API 或带“UPDATE_HISTORY”的 ADMIN_CMD 过程将重要恢复文件条目的状态设置为 do_no_delete 修剪标记为 do_not_delete 的条目有三种方法：

- 带 WITH FORCE 选项调用 PRUNE HISTORY 命令
- 带 PRUNE HISTORY 和 WITH FORCE 选项调用 ADMIN_CMD 过程
- 带 DB2_PRUNE_OPTION_FORCE 选项调用 db2Prune API

除非执行这三个过程之一，否则永远不会从恢复历史记录文件中修剪标记为 do_not_delete 的条目。

限制：

- 只有备份映像、装入副本映像和日志文件的状态可以设置为 do_not_delete。
- 备份条目的状态不会传播到与该备份操作相关的装入副本映像、非增量备份或日志文件。如果想要保存特定数据库备份条目及其相关日志文件条目，必须设置数据库备份条目和每个相关日志文件条目的状态。

第 9 章 管理恢复对象

随着您定期备份数据库，可能会累积大量数据库备份映像以及许多数据库日志和装入副本映像。IBM 数据服务器数据库管理器可简化这些恢复对象的管理。

存储恢复对象会占用大量存储空间。一旦运行了后续备份操作，便可以删除旧恢复对象，因为不再需要它们来复原数据库。但是，除去旧恢复对象很费时间。同时，当您删除旧恢复对象时，还可能会意外地损坏仍然需要的恢复对象。

使用数据库管理器来删除复原数据库时不再需要的恢复对象有两种方法：

- 可以带 `AND DELETE` 参数调用 `PRUNE HISTORY` 命令，或者带 `DB2PRUNE_OPTION_DELETE` 标志调用 `db2Prune` API。
- 可以将数据库管理器配置为自动删除不需要的恢复对象。

使用 `PRUNE HISTORY` 命令或 `db2Prune` API 来删除数据库恢复对象

可以使用 `AUTO_DEL_REC_OBJ` 数据库配置参数和 `PRUNE HISTORY` 命令或 `db2Prune` API 来删除恢复对象。

当调用 `PRUNE HISTORY` 命令或调用 `db2Prune` API 时，IBM 数据服务器数据库管理器会执行以下操作：

- 从恢复历史记录文件中修剪状态不是 `DB2HISTORY_STATUS_DO_NOT_DEL` 的条目

当带 `AND DELETE` 参数调用 `PRUNE HISTORY` 命令、或者带 `DB2PRUNE_OPTION_DELETE` 标志调用 `db2Prune` API 时，数据库管理器会执行以下操作：

- 从恢复历史记录文件中修剪早于指定时间戳记且状态不是 `DB2HISTORY_STATUS_DO_NOT_DEL` 的条目
- 删除与修剪的条目关联的物理日志文件

如果将 `AUTO_DEL_REC_OBJ` 数据库配置参数设置为 `ON`，那么当您带 `AND DELETE` 参数调用 `PRUNE HISTORY` 命令、或者带 `DB2PRUNE_OPTION_DELETE` 标志调用 `db2Prune` API 时，数据库管理器将执行以下操作：

- 从恢复历史记录文件中修剪状态不是 `DB2HISTORY_STATUS_DO_NOT_DEL` 的条目
- 删除与修剪的条目关联的物理日志文件
- 删除与修剪的条目关联的备份映像
- 删除与修剪的条目关联的装入副本映像

要删除不需要的恢复对象：

1. 将 `AUTO_DEL_REC_OBJ` 数据库配置参数设置为 `ON`。
2. 带 `AND DELETE` 参数调用 `PRUNE HISTORY` 命令，或者带 `DB2PRUNE_OPTION_DELETE` 标志调用 `db2Prune` API。

自动管理数据库恢复对象

可以使用 `AUTO_DEL_REC_OBJ` 数据库配置参数和自动恢复历史记录文件修剪来配置 IBM 数据服务器数据库管理器，使它在每次完整数据库备份操作之后自动删除不需要的恢复对象。

每次完整（非增量）数据库备份操作成功后，数据库管理器将根据 `num_db_backup` 和 `rec_his_retentn` 配置参数的值修剪恢复历史记录文件：

- 如果恢复历史记录文件中的数据库备份条目比 `num_db_backup` 配置参数的值多，那么数据库管理器将从恢复历史记录文件中修剪早于 `rec_his_retentn` 配置参数的值且状态不是 `DB2HISTORY_STATUS_DO_NOT_DEL` 的条目。

如果将 `AUTO_DEL_REC_OBJ` 数据库配置参数设置为 `ON`，那么除了从恢复历史记录文件中修剪条目以外，数据库管理器还会执行以下操作：

- 删除与修剪的条目关联的物理日志文件
- 删除与修剪的条目关联的备份映像
- 删除与修剪的条目关联的装入副本映像

如果当前恢复历史记录中没有完整数据库备份映像可用（可能从未备份映像），那么将删除比 `REC_HIS_RETENTN` 指定的时间范围更早的映像。

如果数据库管理器因文件已不在恢复历史记录文件列出的位置而无法将其删除，那么数据库管理器将修剪该历史记录条目。

如果数据库管理器因数据库管理器与存储器管理器或设备之间的通信错误而无法删除文件，那么数据库管理器将不修剪该历史记录文件条目。错误解决后，该文件可以在下一自动修剪期间删除。

要将数据库管理器配置为自动删除不需要的恢复对象：

1. 将 `AUTO_DEL_REC_OBJ` 数据库配置参数设置为 `ON`
2. 设置 `rec_his_retentn` 和 `num_db_backups` 配置参数以启用自动恢复历史记录文件修剪。

防止恢复对象被删除

自动恢复对象管理可节省管理时间和存储空间。但是，您可能需要防止特定恢复对象被自动删除。通过将关联的恢复历史记录文件条目的状态设置为 `do_not_delete`，可以防止重要恢复对象被删除。

关于此任务

如果将 `AUTO_DEL_REC_OBJ` 数据库配置参数设置为 `ON`，那么修剪关联的恢复历史记录文件条目时，将删除恢复对象。在下列其中一种情况下，将修剪恢复历史记录文件条目：

- 带 `AND DELETE` 参数调用 `PRUNE HISTORY` 命令
- 带 `DB2PRUNE_OPTION_DELETE` 标志调用 `db2Prune API`
- 在每次成功完整备份表空间或数据库后，数据库管理器会自动修剪恢复历史记录文件。

无论是调用 PRUNE HISTORY 命令、调用 db2Prune API 还是将数据库管理器配置为自动修剪恢复历史记录文件中的条目，都不会修剪标记为 do_not_delete 的条目，也不会删除关联的恢复对象。

过程

使用 UPDATE HISTORY 命令将关联恢复文件条目的状态设置为 do_no_delete。

限制:

- 只有备份映像、装入副本映像和日志文件的状态可以设置为 do_not_delete。
- 备份条目的状态不会传播到与该备份操作相关的日志文件、装入副本映像或非增量备份。如果想要保存特定数据库备份条目及其相关日志文件条目，必须设置数据库备份条目和每个相关日志文件条目的状态。

管理快照备份对象

必须使用 db2acsutil 命令来管理快照备份对象。不要使用文件系统实用程序来移动或删除快照备份对象。

开始前

要执行快照备份与复原操作，需要适用于存储设备的 DB2 ACS API 驱动程序。集成到 IBM 数据服务器的是适用于下列存储硬件的 DB2 ACS API 驱动程序:

- IBM TotalStorage[®] SAN Volume Controller
- IBM System Storage[™] DS6000[™]
- IBM 系统存储器 DS8000[™]
- IBM 系统存储器 N 系列
- NetApp V 系列
- NetApp FAS 系列

在可以管理快照备份对象之前，必须启用 DB2 高级副本服务 (ACS)。请参阅: 第 257 页的『启用 DB2 高级副本服务 (ACS)』。

限制

db2acsutil 命令当前仅在 AIX 和 Linux 上受支持。

过程

1. 要列示可用的快照备份对象，请使用 **QUERY** 参数。

例如，要为名为 dbminst1 的数据库管理器实例列示可用的快照备份对象，请使用下列语法:

```
db2acsutil query instance dbminst1
```

2. 要检查给定快照备份操作的进度，请使用 **STATUS** 参数。

例如，要了解可能当前在名为 database1 的数据库上运行的快照备份操作的进度，请使用下列语法:

```
db2acsutil query status db database1
```

3. 要删除特定快照备份对象，请使用 **DELETE** 参数。

例如，要为名为 database1 的数据库删除已经存在 10 多天的所有快照备份对象，请使用下列语法：

```
db2acsutil delete older than 10 days ago db database1
```

第 10 章 监视备份、复原和恢复操作的进度

可以使用 LIST UTILITIES 命令来监视数据库上的备份、复原和前滚操作。

要对备份、复原和恢复操作使用进度监视：

发出 LIST UTILITIES 命令并指定 SHOW DETAIL 选项。

```
list utilities show detail
```

以下是用于监视脱机数据库备份操作性能的输出的示例：

```
LIST UTILITIES SHOW DETAIL
```

```
标识                = 2
类型                = 备份
数据库名称          = 样本
描述                = 脱机数据库
开始时间            = 10/30/2003 12:55:31.786115
正在调速:
优先级              = 最低
进度监视:
估计完成百分比      = 41
全部工作单元        = 20232453 个字节
已完成的工作单元    = 230637 个字节
开始时间            = 10/30/2003 12:55:31.786115
```

对于备份操作，将指定要处理的最初估计字节数。随备份操作进度更新要处理的字节数。显示的字节与映像的大小不相等，不应该用作备份映像大小的估计值。视映像是增量备份，还是压缩备份而定，实际映像可能小很多。

对于复原操作，将不给定任何最初的估计值。而是指定 UNKNOWN。因为从映像中读取每个缓冲区，所以将会更新实际读取的字节量。对于其中可能复原多个映像的自动增量复原操作，将使用阶段来跟踪进度。每个阶段提供一个从增量链中复原的映像。最初，只有显示一个阶段。复原第一个映像之后，将显示阶段的总数。在复原每个映像时，将根据已处理的字节数来更新完成的阶段数。

对于崩溃恢复和前滚恢复，将有两个进度监视阶段：FORWARD 和 BACKWARD。FORWARD 阶段期间中，读取日志文件并将日志记录应用于数据库。对于崩溃恢复，通过使用起始日志序号至最后一个日志文件的结尾来估计工作总量。对于前滚恢复，当此阶段开始时，将工作总量估计值指定为 UNKNOWN。按字节计的已处理工作量将随进程的继续而更新。

在 BACKWARD 阶段中，回滚 FORWARD 阶段期间中应用任何未落实的更改。将提供按字节计的需处理日志数据量的估计值。按字节计的已处理工作量将随进程的继续而更新。

表空间状态

表空间的当前状态由它的状态反映。与恢复相关的最常见表空间状态是：

- 备份暂挂。在前滚操作的某个时间点后，或不带有复制选项的装入操作后，表空间将置于些状态。在可使用该表空间之前必须对其备份。（如果未进行备份就不能更新表空间，但允许只读操作。）

- **复原暂挂。** 如果取消了对表空间的前滚操作，或对表空间的前滚操作遇到了不可恢复错误（此时必须再次复原并前滚表空间），会将表空间置于此状态。在复原操作期间，如果无法复原表空间，该表空间也会处于此状态。
- **正在前滚。** 表空间在对它进行的前滚操作正在进行中时，被置于此状态。一旦前滚操作成功完成，表空间就不再处于“正在前滚”状态。如果取消了对表空间的前滚操作，表空间也会结束此状态。
- **前滚暂挂。** 表空间在复原后可发生了输入/输出（I/O）错误后被置于此状态。复原后，表空间可前滚到日志的末尾或某时间点。发生了 I/O 错误后，表空间必须前滚到日志的末尾。

第 11 章 backup 概述

DB2 BACKUP DATABASE 命令的最简单格式仅要求您指定要备份的数据库的别名。
例如:

```
db2 backup db sample
```

如果成功完成命令，您将获取一个新的备份映像，它位于发出命令的路径或目录中。它位于此目录中的原因是，此示例中的命令不会显式地指定备份映像的目标位置。由 DB2 版本 9.5 和更高版本创建的备份映像是使用文件方式 600 生成的，这意味着在 UNIX 上仅实例所有者具有读写特权；在 Windows 上仅 DB2ADMNS（和 Administrators）组的成员对备份映像具有访问权。

注：如果 DB2 客户机和服务器不在同一系统上，DB2 将确定客户机上的当前工作目录，并在服务器上将其用作备份目标目录。因此，建议为备份映像指定目标目录。

在调用 BACKUP 实用程序时指定的目标位置创建备份映像。位置可以是:

- 目录（对于备份到磁盘或软盘）
- 设备（对于备份到磁带）
- Tivoli Storage Manager (TSM) 服务器
- 另一供应商服务器

每当调用数据库备份操作时，都会使用摘要信息自动更新恢复历史记录文件。此文件在数据库配置文件所在的目录中创建。

当不再需要旧备份映像时，如果要将它们删除，可以除去文件（如果那些备份是以文件形式存储的）。然后，如果运行指定了 BACKUP 选项的 LIST HISTORY 命令，还将返回有关已删除的备份映像的信息。必须使用 PRUNE 命令来从恢复历史记录文件中除去那些条目。

如果恢复对象是使用 Tivoli Storage Manager (TSM) 保存的，那么可以使用 db2adutil 实用程序来查询、抽取、验证和删除恢复对象。在 Linux 和 UNIX 上，此实用程序位于 sqllib/adsm 目录中；在 Windows 操作系统上，它位于 sqllib\bin 中。

在所有操作系统上，在磁盘上创建的备份映像文件名由几个元素并置组成，这些元素由句点分隔:

```
DB_alias.Type.Inst_name.NODEnnnn.CATNnnnn.timestamp.Seq_num
```

例如:

```
STAFF.0.DB201.NODE0000.CATN0000.19950922120112.001
```

注：在 Windows 操作系统上，DB2 通用数据库 V8.2.2 和更早版本在存储备份映像时使用 4 层子目录树。

```
DB_alias.Type\Inst_name\NODEnnnn\CATNnnnn\yyyymmdd\hhmmss.Seq_num
```

例如:

```
SAMPLE.0\DB2\NODE0000\CATN0000\20010320\122644.001
```

数据库别名

在调用 BACKUP 实用程序时指定的，由 1 至 8 个字符组成的数据库别名。

类型 备份操作的类型，其中：0 表示完整的数据库级别备份、3 表示表空间级别的备份而 4 表示由 LOAD...COPY TO 命令生成的备份映像。

实例名 从 DB2INSTANCE 环境变量中提取的、由 1 到 8 个字符组成的当前实例名。

节点号 数据库分区号。在单一分区数据库环境中，分区号始终是 NODE0000。在分区数据库环境中，它是 NODExxxx，其中 xxxx 是 db2nodes.cfg 文件中对数据库分区指定的编号。

目录分区号

数据库目录分区的数据库分区号。在单一分区数据库环境中，分区号始终是 CATN0000。在分区数据库环境中，它是 CATNxxxx，其中 xxxx 是 db2nodes.cfg 文件中对数据库分区指定的编号。

时间戳记

执行备份操作时的日期与时间的 14 个字符表示法。该时间戳记的格式为 *yyyymmddhhnss*，其中：

- *yyyy* 表示年度（1995 至 9999）
- *mm* 表示月份（01 到 12）
- *dd* 表示某月中的某一天（01 到 31）
- *hh* 表示小时（00 到 23）
- *mn* 表示分钟（00 到 59）
- *ss* 表示秒（00 到 59）

序号 用作文件扩展名的一个 3 位的数字。

将备份映像写到磁带时：

- 不创建文件名，但以上描述的信息会存储在备份头中以在验证时使用。
- 必须有一个磁带设备可通过标准操作系统接口来使用。但是，在大型分区数据库环境中，每个数据库分区服务器都有一个专用的磁带设备是不大可能的。可以将这些磁带设备与一个或多个 TSM 服务器连接，以便将对这些磁带设备的访问权提供给每个数据库分区服务器。
- 在分区数据库环境中，还可以使用提供虚拟磁带设备功能的产品，例如，REELlibrarian 4.2 或 CLIO/S。可以使用这些产品来访问通过伪磁带设备连接至其他节点（数据库分区服务器）的磁带设备。对远程磁带设备的访问权是透明地提供的，而伪磁带设备可以通过标准操作系统接口来访问。

不能备份处于不可用状态的数据库，除非该数据库处于备份暂挂状态。如果有任何表空间处于异常状态，那么不能备份该数据库或该表空间，除非它处于备份暂挂状态。

不允许对同一表空间进行并发备份操作。一旦对表空间启动了备份操作，任何后续尝试都将失败（SQL2048）。

如果数据库或表空间由于在执行复原操作期间发生了系统崩溃而处于部分复原状态，必须成功地复原该数据库或表空间才能对它们进行备份。

如果要备份的表空间的列表包含了临时表空间的名称，备份操作会失败。

BACKUP 实用程序可对建立不同数据库的备份副本的多个进程提供并行控制。此并行控制可使备份目标设备保持打开，直到所有备份操作结束。如果在备份操作期间发生错误，有一个打开的容器无法关闭，那么其他面向同一目标驱动器的备份操作就可能接收到访问错误。要校正这类访问错误，必须终止导致了该错误的备份操作，并与目标设备断开连接。如果正在对并发备份到磁带的操作使用 BACKUP 实用程序，应确保这些进程的目标驱动器不是同一磁带。

显示备份信息

可通过 db2ckbkp 显示关于现有备份映像的信息。此实用程序允许您：

- 测试备份映像的完整性并确定是否可对其进行复原。
- 显示存储在备份头中的信息。
- 显示有关备份映像中对象和日志文件头的信息。

使用备份

使用 BACKUP DATABASE 命令来复制数据库数据并将其存储在另一介质上，以防原始介质发生故障或损坏。可以备份整个数据库，可以备份数据库分区，也可以只备份选择的表空间。

开始前

不必连接到将要备份的数据库：备份数据库实用程序自动建立与指定数据库的连接，而此连接会在备份操作完成时终止。如果已连接到要备份的数据库，当发出 BACKUP DATABASE 命令时将断开连接，备份操作将继续进行。

数据库可以是本地数据库或远程数据库。备份映像保留在数据库服务器上，除非您使用的是存储管理产品，如 Tivoli Storage Manager (TSM) 或 DB2 高级副本服务 (ACS)。

如果要执行脱机备份并且已使用 ACTIVATE DATABASE 语句激活数据库，那么在运行脱机备份之前必须取消激活该数据库。如果存在与该数据库的活动连接，为了成功取消激活该数据库，具有 SYSADM 权限的用户必须连接至该数据库并发出下列命令：

```
CONNECT TO database-alias
QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS;
UNQUIESCE DATABASE;
TERMINATE;
DEACTIVATE DATABASE database-alias
```

在分区数据库环境中，可以使用 BACKUP DATABASE 命令来逐个备份数据库分区，使用 ON DBPARTITIONNUM 命令参数来一次性备份多个数据库分区或使用 ALL DBPARTITIONNUMS 参数来同时备份所有数据库分区。可以使用 LIST NODES 命令来标识具有想要备份的用户表的数据库分区。

除非您在使用单一系统视图 (SSV) 备份，否则，如果您要在分区数据库环境中执行脱机备份，那么应将目录分区独立于所有其他数据库分区进行备份。例如，可以先备份目录分区，然后备份所有其他数据库分区。因为备份操作可能需要在目录分区上进行独占数据库连接（在此期间不能连接其他数据库分区），所以必须执行上述操作。如果要执行联机备份，那么可以同时或以任何顺序备份所有数据库分区（包括目录分区）。

也可以使用“命令编辑器”来备份数据库分区。因为此方法不支持前滚恢复，所以应定期备份这些节点上的数据库。还应与您建立的任何备份副本一起，保留一份 `db2nodes.cfg` 文件，以保护可能对此文件造成的损坏。

在分布式请求系统上，备份操作适用于分布式请求数据库和存储在数据库目录（包装器、服务器和昵称等等）中的元数据。不备份数据源对象（表和视图），除非它们也存储在分布式请求数据库中。

如果某个数据库是使用数据库管理器的前发行版创建的且尚未升级，那么必须先升级该数据库才能对其进行备份。

关于此任务

以下限制适用于 BACKUP 实用程序：

- 表空间备份操作和表空间复原操作不能同时运行，即使涉及的是不同的表空间。
- 如果想要能够在分区数据库环境中执行前滚恢复，那么必须在节点列表上定期备份数据库且必须具有系统中余下节点的至少一个备份映像（即使不包含该数据库的用户数据）。下列两种情况都需要在不包含数据库的用户数据的数据库分区服务器中存在数据库分区的备份映像：
 - 在建立上一个备份之后已将一个数据库分区服务器添加到数据库系统，因此需要在此数据库分区服务器上执行正向恢复。
 - 使用时间点恢复，它要求系统中的所有数据库分区都处于前滚暂挂状态。
- DMS 表空间的联机备份操作与下列操作不兼容：
 - 装入
 - 重组（联机和脱机）
 - 删除表空间
 - 截断表
 - 创建索引
 - 最初未记录任何内容（与 CREATE TABLE 和 ALTER TABLE 语句配合使用）
- 如果尝试对当前活动的数据库执行脱机备份，那么将接收到错误。在运行脱机备份之前，可以通过发出 DEACTIVATE DATABASE 命令来确保数据库未处于活动状态。

通过命令行处理器（CLP）、控制中心的“备份数据库”向导、运行带 BACKUP DATABASE 参数的 ADMIN_CMD 过程或 `db2Backup` 应用程序编程接口（API），可以调用备份实用程序。

以下是通过 CLP 发出的 BACKUP DATABASE 命令的示例：

```
db2 backup database sample to c:\DB2Backups
```

过程

要打开“备份数据库”向导：

1. 从“控制中心”中，展开对象树，直到找到要备份的数据库或表空间对象为止。
2. 右键单击该对象，然后从弹出菜单中选择“备份”。这就打开了“备份数据库”向导。

下一步任务

在“控制中心”中通过上下文帮助工具提供详细信息。

如果执行了脱机备份，那么在备份完成后，必须重新激活该数据库：

```
ACTIVATE DATABASE sample
```

执行快照备份

快照备份操作使用存储设备的快速复制技术来执行备份的数据复制部分。

开始前

要执行快照备份与复原操作，需要适用于存储设备的 DB2 ACS API 驱动程序。集成到 IBM 数据服务器的是适用于下列存储硬件的 DB2 ACS API 驱动程序：

- IBM TotalStorage SAN Volume Controller
- IBM 系统存储器 DS6000
- IBM 系统存储器 DS8000
- IBM 系统存储器 N 系列
- NetApp V 系列
- NetApp FAS 系列

在可以执行快照备份之前，必须启用 DB2 高级副本服务（ACS）。请参阅：第 257 页的『启用 DB2 高级副本服务（ACS）』。

过程

您可以使用带有 **USE SNAPSHOT** 参数的 **BACKUP DATABASE** 命令、带有 **SQLU_SNAPSHOT_MEDIA** 介质类型的 **db2Backup API** 或带有 **BACKUP DATABASE** 和 **USE SNAPSHOT** 参数的 **ADMIN_CMD** 过程来执行快照备份：

BACKUP DATABASE 命令：

```
db2 backup db sample use snapshot
```

带有 **BACKUP DATABASE** 参数的 **ADMIN_CMD** 过程：

```
CALL SYSPROC.ADMIN_CMD  
('backup db sample use snapshot')
```

db2Backup API

```
int sampleBackupFunction( char dbAlias[],  
                          char user[],  
                          char pswd[],  
                          char workingPath[] )  
{  
    db2MediaListStruct mediaListStruct = { 0 };  
  
    mediaListStruct.locations = &workingPath;  
    mediaListStruct.numLocations = 1;  
    mediaListStruct.locationType = SQLU_SNAPSHOT_MEDIA;  
  
    db2BackupStruct backupStruct = { 0 };  
  
    backupStruct.piDBAlias = dbAlias;  
    backupStruct.piUsername = user;  
    backupStruct.piPassword = pswd;  
    backupStruct.piVendorOptions = NULL;  
    backupStruct.piMediaList = &mediaListStruct;
```

```
db2Backup(db2Version950, &backupStruct, &sqlca);  
  
    return 0;  
}
```

结果

在执行快照备份之后，您可以从快照备份进行复原。

将分割镜像用作备份映像

使用以下过程来创建主数据库的分割镜像，以用作备份映像。此过程可用于替代在主数据库上执行备份数据库操作。

要将分割镜像用作“备份映像”，遵循下列步骤：

1. 暂挂主数据库上的 I/O ：

```
db2 set write suspend for database
```

数据库暂挂期间，您不应该运行其他实用程序或工具。您只应生成数据库副本。

2. 使用适当的操作系统级别的命令来从主数据库中分割镜像。

注： 确保复制整个数据库目录（包括卷目录）。还必须复制日志目录和存在于数据库目录之外的任何容器目录。要收集此信息，请参阅 DBPATHS 管理视图，该视图显示了需要分割的数据库的所有文件和目录。

3. 恢复主数据库上的 I/O:

```
db2 set write resume for database
```

4. 主系统上发生故障，需要通过备份复原。

5. 停止主数据库实例:

```
db2stop
```

6. 使用操作系统级命令来复制主系统上的分割数据。不要复制分割日志文件，因为前滚恢复需要主日志。

7. 启动主数据库实例:

```
db2start
```

8. 初始化主数据库:

```
db2inidb database_alias as mirror
```

9. 将主数据库前滚至日志末尾或时间点，然后停止。

备份到磁带

备份数据库或表空间时，必须正确地设置块大小和缓冲区大小，如果要使用可变块大小（例如，在 AIX 上，如果块大小已设置为零），那么尤其要这样。

对备份时可使用的固定块大小的数量有限制。因为 DB2 将备份映像头写为 4 KB 的块，所以会存在此限制。DB2 支持的固定块大小只有 512、1024、2048 和 4096 个字节。如果使用固定块大小，那么可以指定任何备份缓冲区大小。但是您可能会发现，如果固定块大小不是 DB2 所支持的大小之一，那么备份操作无法成功完成。

如果数据库很大，使用固定块大小意味着您的备份操作可能花费很长时间才能完成。您可能要考虑使用可变块大小。

注：当前不支持使用可变块大小。如果必须使用此选项，应确保您有经过良好测试的过程，以使您（使用按可变块大小创建的备份映像）进行成功的恢复。

使用可变块大小时，必须指定备份缓冲区大小小于或等于正在使用的磁带设备的最大限制。要获得优化的性能，缓冲区大小必须等于正在使用的设备的最大块大小限制。

必须先发出以下命令，然后才能在 Windows 操作系统上使用磁带设备：

```
db2 initialize tape on <device> using <blksize>
```

其中：

<device>

是有效的磁带设备名。在 Windows 操作系统上，缺省值为 \\.\TAPE0。

<blksize>

是磁带的分块因子。它必须是 4096 的因子或倍数。缺省值是该设备的缺省块大小。

使用可变块大小从备份映像进行复原时可能会返回错误。如果发生这种情况，那么可能需要使用适当的块大小重写该映像。以下是 AIX 上的一个示例：

```
tctl -b 0 -Bn -f /dev/rmt0 read > backup_filename.file
dd if=backup_filename.file of=/dev/rmt0 obs=4096 conv=sync
```

备份映像转储到名为 backup_filename.file 的文件中。dd 命令使用块大小 4096，将映像转储到磁带上。

如果映像太大，不能转储到文件中，此方法可能会出现。有一种可能的解决方案是：使用 dd 命令将该映像从一个磁带设备转储至另一磁带设备。只要映像在一卷磁带上放得下，就可以使用这种方法。使用两个磁带设备时，dd 命令是：

```
dd if=/dev/rmt1 of=/dev/rmt0 obs=4096
```

如果不可能使用两台磁带设备，那么可以使用 dd 命令将映像转储至原始设备，然后再将该映像从原始设备转储至磁带。使用此方法的问题在于 dd 命令必须跟踪转储至原始设备的块数。此块数必须在将映像移动回磁带时指定。如果使用 dd 命令将映像从原始设备转储至磁带，该命令会将原始设备的整个内容转储至磁带。dd 实用程序不能确定使用了多少原始设备空间来保留映像。

当使用 BACKUP 实用程序时，您需要知道磁带设备的最大块大小限制。以下是一些示例：

设备	连接	块大小限制	DB2 缓冲区大小限制 (以 4-KB 页计)
8 毫米	scsi	131,072	32
3420	s370	65,536	16
3480	s370	61 440	15
3490	s370	61 440	15
3490E	s370	65,536	16
7332 (4 毫米) ¹	scsi	262,144	64
3490e	scsi	262,144	64
3590 ²	scsi	2,097,152	512

设备	连接	块大小限制	DB2 缓冲区大小限制 (以 4-KB 页计)
3570 (magstar MP)		262,144	64

注:

1. 7332 没有块大小的限制。256 KB 只是一个建议的值。块大小限制由父适配器确定。
2. 在 3590 支持 2 MB 的块大小时, 如果性能完全可以满足您的需要, 就可以尝试使用较低的值 (例如 256 KB)。
3. 关于设备限制的信息, 请查看设备文档或向设备供应商咨询。

验证磁带设备的兼容性

仅在 UNIX、Linux 和 AIX 操作系统上, 要确定您的磁带设备是否支持备份 DB2 数据库, 请执行以下过程:

作为数据库管理器实例所有者运行操作系统命令 `dd`, 以便对磁带设备执行读写操作。如果 `dd` 命令成功, 那么可以使用磁带设备来备份 DB2 数据库。

备份到命名管道

现在已支持将数据库备份到基于 UNIX 的系统上的本地命名管道, 或从基于 UNIX 的系统上的本地命名管道复原数据库。

命名管道的写程序与阅读器必须是同一台机器。管道必须存在并位于本地文件系统上。因为将命名管道视为本地设备, 所以不需要指定目标是命名管道。

以下是一个 AIX 示例:

1. 创建命名管道:

```
mkfifo /u/dmcinnis/mypipe
```

2. 如果打算由 RESTORE 实用程序来使用备份映像, 那么复原操作必须在备份操作之前调用, 才不会丢失任何数据:

```
db2 restore db sample into mynewdb from /u/dmcinnis/mypipe
```

3. 使用此管道作为数据库备份操作的目标:

```
db2 backup db sample to /u/dmcinnis/mypipe
```

备份分区数据库

在分区数据库环境中备份数据库可能会带来以下困难: 跟踪每个数据库分区的备份是否成功; 管理多个日志文件和备份映像; 以及确保所有数据库分区的日志文件和备份映像需要最少的恢复时间来复原数据库。使用单一系统视图 (SSV) 备份是备份分区数据库最简单的方法。

在分区数据库环境中备份数据库的方法有三种:

- 使用 `BACKUP DATABASE` 命令、带 `ADMIN_CMD` 过程的 `BACKUP DATABASE` 命令或 `db2Backup API` 函数一次备份一个数据库分区
- 将 `BACKUP DATABASE` 命令和 `db2_all` 命令一起使用来备份指定的所有数据库分区

- 运行单一系统视图 (SSV) 备份来同时备份部分或全部数据库分区

每次备份一个数据库分区既耗时又容易出错。使用 `db2_all` 命令备份所有分区比逐一备份数据库分区要简单，因为通常只需进行一次命令调用。但是，当使用 `db2_all` 来备份分区数据库时，有时仍然需要多次调用 `db2_all`，因为不能同时备份包含目录的数据库分区和非目录数据库分区。无论是一次备份一个数据库还是使用 `db2_all` 进行备份，管理使用这两种方法创建的备份映像都会很难，因为每个数据库分区的备份映像的时间戳记不同，而且在数据库分区的备份映像之间协调最低恢复时间也很难。

由于上述原因，在分区数据库环境中备份数据库的推荐方法为使用 SSV 备份。

要使用 SSV 备份来同时备份分区数据库的部分或全部数据库分区：

1. 可选：允许数据库保持联机，或使数据库脱机。

数据库联机或脱机时都可以进行分区数据库备份。如果数据库联机，那么 `BACKUP` 实用程序将获取至其他数据库分区的共享连接，因此用户应用程序能够连接到正在备份的数据库分区。

2. 在包含数据库目录的数据库分区上，使用适当的参数对分区数据库进行备份。

-

可以带 `ON DBPARTITIONNUMS` 参数调用 `BACKUP DATABASE` 命令。

-

您可使用 `ADMIN_CMD` 过程带 `ON DBPARTITIONNUMS` 参数调用 `BACKUP DATABASE` 命令。

-

可以带 `iAllNodeFlag` 参数调用 `db2Backup API`。

3. 可选：在备份映像中包含恢复所需的日志文件。

缺省情况下，如果您执行 SSV 备份（即，您指定 `ON DBPARTITIONNUM` 参数），那么备份映像将包含日志文件。如果不希望备份映像包含日志文件，那么请在运行备份时使用 `EXCLUDE LOGS` 命令参数。对于非 SSV 备份，缺省情况下备份映像中将不包含日志文件。

有关更多信息，请参阅：第 125 页的『使用备份映像包括日志文件』。

4. 可选：删除以前的备份映像。用于删除旧备份映像的方法取决于存储备份映像的方式。例如，如果将备份映像存储在磁盘中，那么可以删除这些文件；如果使用 Tivoli Storage Manager 来存储备份映像，那么可以使用 `db2adutl` 实用程序来删除备份映像。如果使用的是 DB2 高级副本服务 (ACS)，那么可以使用 `db2acsutil` 来删除快照备份对象。

使用 IBM Tivoli Space Manager 分层存储管理来备份分区表

Tivoli Space Manager 分层存储管理器 (HSM) 客户机程序自动将合格的文件迁移到辅助存储器，以保持本地文件系统上的可用空间达到特定水平。

借助表分区功能，表数据分布在多个称为数据分区的存储对象中。HSM 支持将各个数据分区备份到辅助存储器。

在使用 SMS 表空间时，每个数据分区范围都表示为相应目录中的一个文件。因此，可以非常方便地将各个数据范围（数据分区）迁移到辅助存储器。

当使用 DMS 表空间时，每个容器都表示为文件。在此情况下，应该将不频繁访问的范围存储在它们自己的表空间中。在发出使用了 EVERY 子句的 CREATE TABLE 语句时，使用 NO CYCLE 子句来确保表级别 IN 子句中列示的表空间数与正在创建的数据分区数匹配。以下示例演示了这种情况：

示例 1

```
CREATE TABLE t1 (c INT) IN tbsp1, tbsp2, tbsp3 NO CYCLE
PARTITION BY RANGE(c)
(STARTING FROM 2 ENDING ( 6 EVERY 2);
```

启用自动备份

数据库可能会由于各种硬件或软件故障而变得不可用。确保有数据库的最新完整备份，这对于规划和实现系统灾难恢复策略而言至关重要。通过在灾难恢复策略中使用数据库自动备份，DB2 就能够正确并且定期地备份数据库。

可以使用图形用户界面工具、命令行界面或 AUTOMAINT_SET_POLICY 系统存储过程来配置自动备份。

- 要使用图形用户界面工具来配置自动备份：
 1. 通过下面两种方式来打开“配置自动维护”向导：一种方法是在控制中心中右键单击数据库对象；另一种方法是在运行状况中心中右键单击要配置自动备份的数据库实例。从弹出窗口中选择**配置自动维护**。
 2. 在此向导中，可以启用自动备份和指定用于执行 BACKUP 实用程序的维护窗口。
- 要使用命令行界面来配置自动备份，请将下列每个配置参数都设置为 ON：
 - AUTO_MAINT
 - AUTO_DB_BACKUP
- 要使用 AUTOMAINT_SET_POLICY 系统存储过程来配置自动备份：
 - 1.

创建指定备份媒体、是联机还是脱机备份以及备份频率等详细信息的配置 XML 输入。

可以复制 `SQLLIB/samples/automaintcfg` 目录中名为 `DB2DefaultAutoBackupPolicy.xml` 的样本文件的内容并修改该 XML，以满足您的配置要求。

2. 可选：创建包含配置 XML 输入的 XML。
3. 带下列参数调用 AUTOMAINT_SET_POLICY：
 - 维护类型：AutoBackup
 - 配置 XML 输入：要么为包含配置 XML 输入文本的 BLOB；要么为包含配置 XML 输入的文件名称。

有关使用 AUTOMAINT_SET_POLICY 系统存储过程的更多信息，请参阅第 46 页的『使用 `SYSPROC.AUTOMAINT_SET_POLICY` 或 `SYSPROC.AUTOMAINT_SET_POLICYFILE` 来配置自动维护策略』主题。

自动备份数据库

由于大量发生硬件或软件故障，数据库可能会变得不可用。自动备份数据库功能减轻了 DBA 的数据库备份管理任务，它始终会确保在需要时对数据库执行最新的完整备份。它根据下面的一种或多种情况来确定是否需要执行备份操作：

- 您从未完成完整的数据库备份
- 自从上一次执行完整备份以来所经过的时间超过了指定的小时数
- 自从上一次执行备份以来所消耗的事务日志空间超过了指定的 4 KB 页数（仅适用于归档日志记录方式）。

通过为系统规划和实现灾难恢复策略来保护数据。如果能够满足您的需要，那么可以将自动备份数据库功能作为您的备份和恢复策略的一部分。

如果对数据库启用了前滚恢复（归档日志记录），那么可以对联机备份或脱机备份启用自动备份数据库功能。否则，只能进行脱机备份。自动备份数据库支持磁盘、磁带、Tivoli Storage Manager (TSM) 和供应商 DLL 介质类型。

通过“控制中心”或“运行状况中心”中的“配置自动维护”向导，您可以配置：

- 请求备份之间的时间或日志页数
- 备份媒体
- 它将是联机备份还是脱机备份。

如果选择了备份至磁盘，那么自动备份功能部件将定期从“配置自动维护”向导中指定的目录中删除备份映像。只能保证在任何给定时间都可以提供最新的备份映像。建议将此目录专用于自动备份功能部件，而不用来存储其他备份映像。

可以使用 `auto_db_backup` 和 `auto_maint` 数据库配置参数来启用或禁用自动备份数据库功能。在分区数据库环境中，如果对每个数据库分区都启用了数据库配置参数，那么每个数据库分区上都将运行自动备份数据库。

也可以使用名为 `AUTOMAINT_SET_POLICY` 和 `AUTOMAINT_SET_POLICYFILE` 的系统存储过程之一来配置自动备份。

优化备份性能

执行备份操作时，DB2 将自动为缓冲区个数、缓冲区大小和并行性设置选择最佳值。这些值根据可用实用程序堆内存的数量、可用处理器数和数据库配置而定。因此，根据系统上可用的存储量，应考虑通过增大 `UTIL_HEAP_SZ` 配置参数来分配更多内存。目的是最大程度上减少完成备份操作所需的时间。除非显式地输入以下 `BACKUP DATABASE` 命令参数的值，否则 DB2 将为它们选择一个值：

- `WITH num-buffers BUFFERS`
- `PARALLELISM n`
- `BUFFER buffer-size`

如果未指定缓冲区数和缓冲区大小而导致 DB2 设置这些值，那么对大型数据库的影响应该最低。但是，对于小型数据库来说，会导致备份映像大幅增大。即使写入磁盘的最后一个数据缓冲区只包含很少数据，也会将整个缓冲区写入映像。在小型数据库中，这表示相当一部分的映像可能为空。

还可以选择执行以下任何操作来缩短完成一次备份操作所需的时间:

- 指定表空间备份。

使用指定了TABLESPACE 选项的 BACKUP DATABASE 命令, 可以备份(继而恢复)部分数据库。这样便于对表数据、索引和单独表空间中的长字段或大对象(LOB)数据进行管理。

- 增大 BACKUP DATABASE 命令上 PARALLELISM 参数的值, 以使它反映正在备份的表空间数。

PARALLELISM 参数定义在压缩备份操作期间从数据库读取数据和压缩数据时, 已启动的进程或线程数。将每个进程或线程分配给特定表空间, 因此, 为 PARALLELISM 参数指定的值大于要备份的表空间数并无益处。备份完此表空间后, 它会请求另一个表空间。但是应注意: 每个进程或线程都需要内存和 CPU 开销。

- 增加备份缓冲区大小。

理想的备份缓冲区大小是表空间扩展数据块大小的倍数加一页。如果有多个扩展数据块大小不同的表空间, 那么将值指定为扩展数据块大小的公倍数加一页。

- 增加缓冲区的数量。

使用的缓冲区至少是备份目标(或会话)的两倍, 以确保备份目标设备无需等待数据。

- 使用多个目标设备。

使用 backup 所需的特权、权限和授权

特权使用户能够创建或访问数据库资源。权限级别提供了对特权、较高级别数据库管理器维护和实用程序操作进行分组的方法。这两者一起用于控制对数据库管理器及其数据库对象的访问。用户只能访问那些他们具有相应授权(即必需的特权或权限)的对象。

必须具有 SYSADM、SYSCTRL 或 SYSMOINT 权限才能使用 BACKUP 实用程序。

联机备份与其他实用程序的兼容性

某些实用程序可与联机备份实用程序同时运行, 但是另一些实用程序却不行。

下列实用程序与联机备份实用程序兼容:

- EXPORT
- ONLINE INSPECT

下列实用程序仅在某些情况下与联机备份实用程序兼容:

- ONLINE CREATE INDEX

在 SMS 方式下, 由于 ALTER TABLE 锁, 联机索引创建实用程序与联机备份实用程序不兼容。联机索引创建实用程序以互斥方式获取该锁, 而联机备份实用程序以共享方式获取该锁。

在 DMS 方式下，大多数情况下可以并发运行联机索引创建实用程序和联机备份实用程序。如果有大量索引，联机索引创建实用程序就有可能以内部方式获取联机备份锁，这将与任何并发联机备份实用程序发生冲突。

- **ONLINE INDEX REORG**

与联机索引创建实用程序相同，在 SMS 方式下，由于 ALTER TABLE 锁的缘故，联机索引重组实用程序与联机备份实用程序不兼容。联机索引重组实用程序以互斥方式获取该锁，而联机备份实用程序以共享方式获取该锁。另外，联机索引重组操作在切换阶段前使表停顿并获取 Z 锁定，这将导致联机备份实用程序无法运行。但是，在获取 Z 表锁之前，ALTER TABLE 锁应防止同时运行联机备份实用程序。

在 DMS 方式下，可以并发运行联机索引重组实用程序和联机备份实用程序。

另外，联机索引重组实用程序在切换阶段前将使表停顿并获取 Z 锁定，这将导致联机备份实用程序无法运行。

- **REBALANCE**

当并发运行联机备份实用程序和重新平衡器时，联机备份实用程序将使重新平衡器暂停，并且不会等待它完成。

- **IMPORT**

IMPORT 实用程序与联机备份实用程序兼容，但以下情况除外：在发出指定了 REPLACE 选项的 IMPORT 命令时；在此情况下，IMPORT 实用程序将获取表的 Z 锁定并导致联机备份实用程序无法并发运行。

- **ALLOW READ ACCESS LOAD**

当发出指定了 COPY NO 选项的 LOAD 命令时，ALLOW READ ACCESS LOAD 操作与联机备份不兼容。在此方式下，两个实用程序都修改表空间状态，从而导致其中一个实用程序报告错误。

当发出指定了 COPY YES 选项的 LOAD 命令时，虽然仍可能有一些兼容性问题，但 ALLOW READ ACCESS LOAD 操作与联机备份兼容。在 SMS 方式下，这两个实用程序可以并发执行，但它们将会采用不兼容的表锁定方式，从而可能会遇到要等待锁定表情况。在 DMS 方式下，这两个实用程序都采用不兼容的“内部 B”（OLB）锁定方式，并可能遇到要等待该锁定的情况。如果这两个实用程序在同一个表空间上并发执行，那么 LOAD 实用程序可能必须等待 BACKUP 实用程序完成表空间处理之后才能继续运行。

- **ONLINE TABLE REORG**

当联机备份实用程序运行时，无法启动联机表重组实用程序的清除阶段。根据需要，可以暂停表重组以允许联机备份实用程序完成，然后再继续运行联机表重组实用程序。

对于 DMS 表空间来说，可以在对同一表空间中的表执行联机重组时启动 DMS 表空间联机备份。在截断阶段，可能会出现与重组操作相关联的锁定等待情况。

对于 SMS 表空间来说，不能在对同一表空间中的表执行联机重组时启动 SMS 表空间联机备份。这两个操作都需要获取互斥锁定。

- 需要 Z 锁定的 DDL（例如 ALTER TABLE、DROP TABLE 和 DROP INDEX）

联机 DMS 表空间备份与需要 Z 锁定的 DDL 兼容。

联机 SMS 表空间备份必须等待 Z 锁定被释放。

- RUNSTATS (允许读写)

Runstats 与联机备份兼容, 但系统目录表空间为 SMS 表空间时除外。这是因为, 如果 SYSIBM.SYSTABLES 位于 SMS 表空间中, 那么 runstats 和联机备份将挂起对表的不兼容表锁定, 从而导致等待锁定。

下列实用程序与联机备份实用程序不兼容:

- REORG TABLE
- RESTORE
- ROLLFORWARD
- ONLINE BACKUP
- ALLOW NO ACCESS LOAD
- SET WRITE

备份示例

示例 1

在以下示例中, 数据库 SAMPLE 被备份至使用 2 个并发 TSM 客户机会话的 TSM 服务器上。BACKUP 实用程序将计算最佳的缓冲区数。根据内存量和可用的目标设备数自动计算出最佳缓冲区大小 (以 4 KB 页计)。也可以根据可用处理器数和要备份的表空间数自动计算并行性设置。

```
db2 backup database sample use tsm open 2 sessions with 4 buffers
```

```
db2 backup database payroll tablespace (syscatspace, userspace1) to  
/dev/rmt0, /dev/rmt1 with 8 buffers without prompting
```

示例 2

以下是对可恢复数据库每周进行的增量备份策略的样本。包括一个每周进行的完整数据库备份操作、一个每天进行的非累积 (差异) 备份操作以及一个在每周中期进行的累积 (增量) 备份操作:

```
(Sun) db2 backup db kdr use tsm  
(Mon) db2 backup db kdr online incremental delta use tsm  
(Tue) db2 backup db kdr online incremental delta use tsm  
(Wed) db2 backup db kdr online incremental use tsm  
(Thu) db2 backup db kdr online incremental delta use tsm  
(Fri) db2 backup db kdr online incremental delta use tsm  
(Sat) db2 backup db kdr online incremental use tsm
```

示例 3

要在 Windows 环境中对磁带设备启动备份操作, 发出:

```
db2 backup database sample to \\.\tape0
```

第 12 章 recover 概述

RECOVER 实用程序执行必需的复原和前滚操作，以根据恢复历史记录文件中找到的信息将数据库恢复到指定时间。当使用此实用程序时，指定将数据库恢复至时间点或日志文件的结尾。然后，实用程序将选择最适合的备份映像并执行恢复操作。

RECOVER 实用程序不支持以下 RESTORE DATABASE 命令选项：

- TABLESPACE tablespace-name。不支持表空间复原操作。
- INCREMENTAL。不支持增量复原操作。
- OPEN num-sessions SESSIONS。无法指示将与 TSM 或其他供应商产品一起使用的 I/O 会话数。
- BUFFER buffer-size。无法设置用于复原操作的缓冲区大小。
- DLREPORT filename。无法指定用于报告文件变成取消链接的文件名。
- WITHOUT ROLLING FORWARD。无法指定在成功复原操作后，不要将该数据库置于前滚暂挂状态。
- PARALLELISM n。无法指示复原操作的并行度。
- WITHOUT PROMPTING。无法指定复原操作以无人照管方式运行。

此外，RECOVER 实用程序不允许指定任何 REBUILD 选项。但是，如果 RECOVER 实用程序无法根据恢复历史记录文件中的信息来找到任何数据库备份映像，它就会自动使用适当的 REBUILD 选项。

恢复数据

通过 RECOVER DATABASE 命令，可以使用恢复历史记录文件中的信息将数据库恢复到指定的时间。

对于在前滚阶段结束的未完成恢复操作，如果随后发出 RECOVER DATABASE 命令，RECOVER 实用程序将尝试继续执行上一次恢复操作，而不重新执行复原阶段。如果要强制 RECOVER 实用程序重新完成复原阶段，请发出指定了 RESTART 选项的 RECOVER DATABASE 命令，以强制 RECOVER 实用程序忽略先前任何未能完成的恢复操作。如果使用的是应用程序编程接口（API），请在 iRecoverAction 字段中指定调用程序操作 DB2RECOVER_RESTART 以强制 RECOVER 实用程序重新执行复原阶段。

如果 RECOVER DATABASE 命令在复原阶段被中断，它就无法继续。您需要重新发出 RECOVER DATABASE 命令。

不应连接到要恢复的数据库：恢复数据库实用程序会自动建立与指定数据库的连接，并且此连接将在恢复操作完成时终止。

数据库可以是本地数据库或远程数据库。

可以通过命令行处理器（CLP）或 db2Recover 应用程序编程接口（API）来调用 RECOVER 实用程序。

以下示例显示如何通过 CLP 使用 RECOVER DATABASE 命令:

```
db2 recover db sample
```

注: 在分区数据库环境中, 必须从数据库的目录分区调用 RECOVER 实用程序。

使用 db2adutl 来恢复数据

下列示例说明如何使用 db2adutl 命令以及 *logarchopt1* 和 *vendoropt* 数据库配置参数来执行跨节点恢复。

在下列示例中, 计算机 1 名为 bar, 它正在运行 AIX。此机器的所有者是 roecken。bar 上的数据库名为 sample。计算机 2 名为 dps。此机器也在运行 AIX 并由 regress9 所拥有。

PASSWORDACCESS = generate

计算机 1

1. 设置数据库以便将日志归档至 TSM。更新 sample 数据库的数据库配置参数 *logarchmeth1*:

```
bar:/home/roecken> db2 update db cfg for sample using LOGARCHMETH1 tsm
```

将返回以下信息:

```
成功完成 DB20000I UPDATE DATABASE CONFIGURATION 命令。
```

注: 在更新数据库配置之前, 可能需要对数据库进行脱机备份。

2. 强制应用程序断开连接:

```
db2 force applications all
```

3. 验证是否已强制所有应用程序断开连接:

```
db2 list applications
```

您应该接收到一条消息, 说明未返回任何数据。

注: 在分区数据库环境中, 必须对所有数据库分区都执行此步骤。

4. 备份数据库:

```
db2 backup db sample use tsm
```

将返回类似以下的信息:

```
备份成功。此备份的时间戳记为: 20040216151025
```

注: 在分区数据库环境中, 必须对所有数据库分区都执行此步骤。根据您正在执行联机备份还是脱机备份, 在数据库分区上执行此步骤的顺序有所不同。有关更多信息, 请参阅第 175 页的『使用备份』。

5. 连接至 sample 数据库, 然后在该数据库中创建一个表。
6. 将数据装入新表中。在此示例中, 表名为 a, 从定界 ASCII 码文件中装入的数据为 mr。指定了 COPY YES 选项来生成装入的数据的副本, 并且 USE TSM 选项指定数据的副本存储在 Tivoli Storage Manager 上。

注: 仅当数据库启用了前滚恢复功能时才能指定 COPY YES 选项; 即, 必须将 *logarchmeth1* 数据库配置参数设置为 USEREXIT 或 LOGRETAIN。

```
bar:/home/roecken> db2 load from mr of del modified by noheader replace
into a copy yes use tsm
```

实用程序返回了一系列消息来指示它的进度:

```
SQL3109N 实用程序正开始从"/home/roecken/mr"文件中装入数据。
SQL3500W 实用程序在"02/16/2004 15:12:13.392633"时进入"装入"阶段。

SQL3519W 开始装入一致点。输入记录数 ="0"。

SQL3520W 装入一致点成功。
SQL3110N 实用程序已完成处理。从输入文件读取了"1"行。
```

```
SQL3519W 开始装入一致点。输入记录数 ="1"。    SQL3520W 装入一致点成功。
SQL3515W 实用程序在"02/16/2004 15:12:13.445718"时已完成"装入"阶段。
读取的行数           = 1
跳过的行数           = 0
装入的行数           = 1
拒绝的行数           = 0
删除的行数           = 0
落实的行数           = 1
```

现在, 在 TSM 上应该具有一个备份映像、一个装入副本映像和一个日志文件。可以按如下所示对 zample 数据库运行查询:

```
bar:/home/roecken/sql1lib/adsm> db2adutl query db zample
```

将返回以下信息:

```
正在检索 FULL DATABASE BACKUP 信息。
  1 时间: 20040216151025 最早的日志: S0000000.LOG 数据库分区号: 0
    会话数: 1
正在检索 INCREMENTAL DATABASE BACKUP 信息。
找不到 ZAMPLE 的 INCREMENTAL DATABASE BACKUP 映像
```

```
正在检索 DELTA DATABASE BACKUP 信息。
找不到 ZAMPLE 的 DELTA DATABASE BACKUP 映像
```

```
正在检索 TABLESPACE BACKUP 信息。
找不到 ZAMPLE 的 TABLESPACE BACKUP 映像
```

```
正在检索 INCREMENTAL TABLESPACE BACKUP 信息。
找不到 ZAMPLE 的 INCREMENTAL TABLESPACE BACKUP 映像
```

```
正在检索 DELTA TABLESPACE BACKUP 信息。
找不到 ZAMPLE 的 DELTA TABLESPACE BACKUP 映像
```

```
正在检索 LOAD COPY 信息。
  1 时间: 20040216151213
```

```
正在检索 LOG ARCHIVE 信息。
  日志文件: S0000000.LOG, 链号: 0, 数据库分区号: 0
  生成时间: 2004-02-16-15.10.38
```

7. 要启用跨节点恢复, 必须为另一个节点和帐户提供对 bar 计算机上的对象的访问权。在此示例中, 对 dps 节点和 regress9 用户提供了访问权。

```
bar:/home/roecken/sql1lib/adsm> db2adutl grant user regress9
on nodename dps for db zample
```

将返回以下信息:

成功添加了 regress9 访问 dps 节点上的 ZAMPLE 的许可权。

要查询 db2adutl grant 操作的结果, 发出以下命令:

```
bar:/home/roecken/sql1lib/adsm> db2adutl queryaccess
```

将返回以下信息:

节点	用户名	数据库名称	类型
DPS	regress9	ZAMPLE	A

访问类型: B - 备份映像 L - 日志 A - 同时使用这两种访问类型

PASSWORDACCESS = 生成环境

计算机 2

尚未设置第 2 台计算机 dps。在 dps 上对 zample 数据库的 db2adutl 查询返回了下列结果:

```
dps:/home/regress9/sql1lib/adsm> db2adutl query db zample
--- 数据库目录是空的 ---
警告: 在 ADSM 服务器上 DB2 没有创建文件空间
警告: 在 ADSM 中找不到任何别名的 DB2 备份映像。

dps:/home/regress9/sql1lib/adsm> db2adutl query db zample nodename
bar owner roecken
--- 数据库目录是空的 ---
```

对 ZAMPLE 数据库的查询

```
正在检索 FULL DATABASE BACKUP 信息。
1 时间: 20040216151025 最早的日志: S0000000.LOG 数据库分区号: 0
会话数: 1
正在检索 INCREMENTAL DATABASE BACKUP 信息。
找不到 ZAMPLE 的 INCREMENTAL DATABASE BACKUP 映像
```

```
正在检索 DELTA DATABASE BACKUP 信息。
找不到 ZAMPLE 的 DELTA DATABASE BACKUP 映像
```

```
正在检索 TABLESPACE BACKUP 信息。
找不到 ZAMPLE 的 TABLESPACE BACKUP 映像
```

```
正在检索 INCREMENTAL TABLESPACE BACKUP 信息。
找不到 ZAMPLE 的 INCREMENTAL TABLESPACE BACKUP 映像
```

```
正在检索 DELTA TABLESPACE BACKUP 信息。
找不到 ZAMPLE 的 DELTA TABLESPACE BACKUP 映像
```

```
正在检索 LOAD COPY 信息。
1 时间: 20040216151213
```

```
正在检索 LOG ARCHIVE 信息。
日志文件: S0000000.LOG, 链号: 0, 数据库分区号: 0
生成时间: 2004-02-16-15.10.38
```


在 dps 计算机上尚不存在 `zample` 数据库。

1. 将 `zample` 数据库复原到 dps 计算机:

```
dps:/home/regress9> db2 restore db zample use tsm options
''-fromnode=bar -fromowner=roecken'' without prompting
```

将返回以下信息:

```
DB20000I 已成功完成 RESTORE DATABASE 命令。
```

注: 如果 dps 上已经存在 `zample` 数据库, 那么将省略 `OPTIONS` 参数, 并且将使用数据库配置参数 `vendoropt`。此配置参数会覆盖备份或复原操作的 `OPTIONS` 参数。

对 `zample` 数据库的前滚操作将失败, 原因是 `Rollforward` 实用程序找不到日志文件。前滚操作如下所示:

```
dps:/home/regress9> db2 rollforward db zample to end of logs and stop
```

将返回以下错误:

```
SQL4970N 数据库"ZAMPLE"上的前滚恢复不能达到指定的停止点(日志结束或时间点), 原因是在节点"0"上丢失了日志文件。
```

2. 要强制 `Rollforward` 实用程序查找另一台机器上的日志文件, 必须配置正确的 `logarchopt` 值, 在此情况下, 为 `logarchopt1` 数据库配置参数:

```
dps:/home/regress9> db2 update db cfg for zample using logarchopt1
''-fromnode=bar -fromowner=roecken''
```

3. 要使 `Rollforward` 实用程序能够使用装入副本映像, 还必须设置 `vendoropt` 数据库配置参数:

```
dps:/home/regress9> db2 update db cfg for zample using VENDOROPT
''-fromnode=bar -fromowner=roecken''
```

4. 现在, 可以对 `zample` 数据库进行前滚了:

```
dps:/home/regress9> db2 rollforward db zample to end of logs and stop
```

将返回以下信息:

```
前滚状态
输入数据库别名          = zample
返回了状态的节点数      = 1
节点号                   = 0
前滚状态                 = 未暂挂
要读取的下一个日志文件 =
已处理的日志文件        = S0000000.LOG - S0000000.LOG
上次落实的事务           = 2004-02-16-20.10.38.000000 UTC
```

```
DB20000I 已成功完成 ROLLFORWARD 命令。
```

PASSWORDACCESS = 提示环境

在“提示”环境中, 需要额外的信息, 尤其是创建了对象的机器的 TSM 节点名和密码。

对于 `db2adutl`, 更新 `dsm.sys` 文件 (在基于 Windows 的平台上称为 `dsm.opt` 文件) 并将 `NODENAME bar` (因为 `bar` 是源计算机的名称) 添加至服务器子句:

```
dps:/home/regress9/sql1ib/adsm> db2adutl query db zample nodename bar
owner roecken password *****
```

将返回以下信息:

对 ZAMPLE 数据库的查询

正在检索 FULL DATABASE BACKUP 信息。

1 时间: 20040216151025 最早的日志: S0000000.LOG 数据库分区号: 0
会话数: 1

正在检索 INCREMENTAL DATABASE BACKUP 信息。

找不到 ZAMPLE 的 INCREMENTAL DATABASE BACKUP 映像

正在检索 DELTA DATABASE BACKUP 信息。

找不到 ZAMPLE 的 DELTA DATABASE BACKUP 映像

正在检索 TABLESPACE BACKUP 信息。

找不到 ZAMPLE 的 TABLESPACE BACKUP 映像

正在检索 INCREMENTAL TABLESPACE BACKUP 信息。

找不到 ZAMPLE 的 INCREMENTAL TABLESPACE BACKUP 映像

正在检索 DELTA TABLESPACE BACKUP 信息。

找不到 ZAMPLE 的 DELTA TABLESPACE BACKUP 映像

正在检索 LOAD COPY 信息。

1 时间: 20040216151213

正在检索 LOG ARCHIVE 信息。

日志文件: S0000000.LOG, 链号: 0, 数据库分区号: 0
生成时间: 2004-02-16-15.10.38

1. 如果该数据库不存在, 那么请创建空的 `zample` 数据库。如果 `zample` 数据库已存在, 那么可以跳过此步骤和用来更新数据库配置的下面两个步骤。

```
dps:/home/regress9> db2 create db zample
```

2. 更新 `zample` 数据库的数据库配置参数 `tsm_nodename`:

```
dps:/home/regress9> db2 update db cfg for zample using tsm_nodename bar
```

3. 更新 `zample` 数据库的数据库配置参数 `tsm_password`:

```
dps:/home/regress9> db2 update db cfg for zample using  
tsm_password *****
```

4. 复原 `zample` 数据库:

```
dps:/home/regress9> db2 restore db zample use tsm options  
"-fromnode=bar -fromowner=roecken" without prompting
```

成功完成复原操作, 但是发出了一条警告:

SQL2540W 复原成功, 但是在以"无中断"方式进行处理时, 在"数据库复原"期间遇到了警告"2523"。

此时, `Rollforward` 实用程序又找不到正确的日志文件:

```
dps:/home/regress9> db2 rollforward db zample to end of logs and stop
```

将返回以下错误消息:

SQL1268N 由于检索节点"0"上的数据库"ZAMPLE"的日志文件"S0000000.LOG"时发生错误"-2112880618", 前滚恢复已停止。

5. 因为数据库复原操作替换了数据库配置文件，所以必须将 TSM 数据库配置值设置为正确的值。首先，必须复位 `tsm_nodename` 配置参数：

```
dps:/home/regress9> db2 update db cfg for zample using tsm_nodename bar
```

6. 必须复位 `tsm_password` 数据库配置参数：

```
dps:/home/regress9> db2 update db cfg for zample using tsm_password *****
```

7. 必须复位 `logarchopt1` 数据库配置参数，以便 Rollforward 实用程序可以找到正确的日志文件：

```
dps:/home/regress9> db2 update db cfg for zample using logarchopt1  
"-fromnode=bar -fromowner=roecken"
```

8. 还必须复位 `vendoropt` 数据库配置参数，以便还可以使用装入恢复文件：

```
dps:/home/regress9> db2 update db cfg for zample using VENDOROPT  
"-fromnode=bar -fromowner=roecken"
```

9. 当设置数据库配置参数时，可以对数据库进行前滚：

```
dps:/home/regress9> db2 rollforward db zample to end of logs and stop
```

对 `zample` 数据库运行 `ROLLFORWARD QUERY STATUS` 命令将显示以下结果：

```
前滚状态  
  
输入数据库别名           = zample  
返回了状态的节点数       = 1  
  
节点号                     = 0  
前滚状态                   = 未暂挂  
要读取的下一个日志文件   =  
已处理的日志文件         = S0000000.LOG - S0000000.LOG  
上次落实的事务           = 2004-02-16-20.10.38.000000 UTC
```

```
DB20000I 已成功完成 ROLLFORWARD 命令。
```

恢复已删除的表

您可能会偶然删除包含仍需要的数据的表。如果是这样，您应该考虑在删除表操作后使关键的表成为可复原的。可通过调用数据库复原操作恢复表数据，后跟一个前滚到删除表前的某时间点的数据库前滚操作。如果数据库很大，就可能会花很多时间，并使数据在恢复期间不可用。已删除的表的恢复功能使您可使用表空间级的复原和前滚操作来恢复已删除的表数据。这样可比数据库级别的恢复要快，且您的数据库将对用户保持可用。

要使已删除的表可以复原，必须对该表所在的表空间启用 `DROPPED TABLE RECOVERY` 选项。这可以在表空间创建期间或通过调用 `ALTER TABLESPACE` 语句来完成。`DROPPED TABLE RECOVERY` 选项是表空间特定的，并限于对常规表空间使用。要确定是否对表空间启用了已删除的表的恢复功能，可以查询 `SYSCAT.TABLESPACES` 目录表中的 `DROP_RECOVERY` 列。

创建表空间时，缺省情况下已删除的表的恢复选项已打开。如果不要启用表空间的已删除的表的恢复功能，那么可以在发出 `CREATE TABLESPACE` 语句时显式将 `DROPPED TABLE RECOVERY` 选项设置为 `OFF`，或者可以使用 `ALTER TABLESPACE` 语句来对现有表空间禁用已删除的表的恢复功能。如果有许多删除表操作要恢复或者如果历史记录文件很大，那么正向恢复时已删除的表的恢复功能可能会影响性能。

对表（对该表的表空间启用了已删除的表的恢复功能）运行 `DROP TABLE` 语句时，将在日志文件中建立另一条目（标识已删除的表）。还会在恢复历史记录文件中建立一个条目，包含可用于重新创建表的信息。

对于分区表来说，即使对一个或多个表空间中的未分区表关闭已删除的表的恢复功能，已删除的表的恢复功能也始终处于打开状态。对于分区表来说，只写一个已删除的表日志记录。此日志记录对于恢复该表的所有数据分区来说已足够了。

对可从已删除的表中复原的数据类型有一些限制。不可能复原：

- `DROPPED TABLE RECOVERY` 选项不能用于临时表。
- 与行类型相关的元数据。（数据已复原，但不是元数据。）将复原有类型表的层次结构表中的数据。此数据包含的信息可能比已删除的类型表中的信息多。
- XML 数据。如果尝试恢复包含 XML 数据的已删除的表，那么相应的列数据将为空。

如果在删除表时它处于重组暂挂状态，那么历史记录文件中的 `CREATE TABLE DDL` 与导入文件的 `CREATE TABLE DDL` 不完全匹配。在执行重组建议的第一个 `ALTER` 操作之前，导入文件将采用该表的格式，但历史记录文件中的 `CREATE TABLE` 语句将与包含任何 `ALTER TABLE` 语句结果的表的状态相匹配。

如果正在恢复 `GRAPHIC` 或 `VARGRAPHIC` 数据类型的数据，那么该数据可能会包括多个代码页。为了恢复此数据，需要指定 `IMPORT` 或 `LOAD` 命令的 `usegraphiccodepage` 文件类型修饰符。在此情况下，使用 `LOAD` 命令来恢复数据将会提高恢复操作的性能。

一次只能复原一个已删除的表。可执行下列操作来复原已删除的表：

1. 通过调用 `LIST HISTORY DROPPED TABLE` 命令来识别已删除的表。已删除的表标识列示在“备份标识”列中。
2. 复原在删除该表前所建立的数据库级别或表空间级别备份映像。
3. 创建包含表数据的文件将写至的导出目录。此目录必须可供所有数据库分区访问，或者在每个数据库分区上都存在。此导出目录下的子目录是由每个数据库分区自动创建的。这些子目录的名称是 `NODEnnnn`，其中 `nnnn` 代表数据库分区或节点号。包含已废弃的表数据的数据文件（就如它存在于每个数据库分区上那样）将导出到称为 `data` 的较低子目录中。例如，
`\export_directory\NODE0000\data`。
4. 在删除表之后前滚至某时间点，对 `ROLLFORWARD DATABASE` 命令使用 `RECOVER DROPPED TABLE` 选项。也可前滚至日志末尾，以使对表空间或数据库中的其他表进行的更新不会丢失。
5. 使用 `CREATE TABLE` 语句从恢复历史记录文件重新创建表。
6. 将在前滚操作期间导出的表数据导入表中。如果进行废弃时表处于重组暂挂状态，那么需要更改 `CREATE TABLE DDL` 的内容以与数据文件的内容相匹配。

崩溃恢复

对数据库执行的事务（也称工作单元）可能被意外中断。如果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，那么该数据库就会处于不一致和不可用的状态。崩溃恢复是将数据库移动回一致并可用状态的进程。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（第 195 页的图 17）。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。

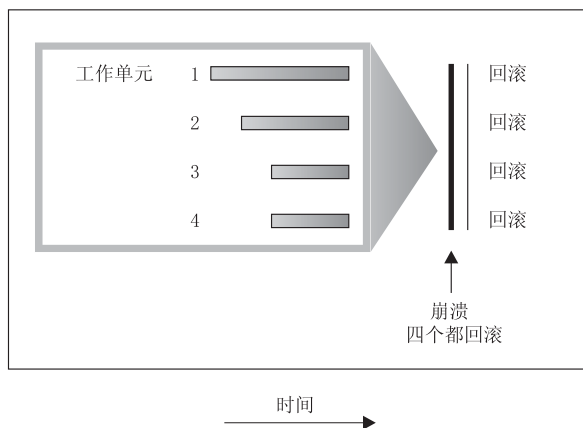


图 17. 回滚工作单元（崩溃恢复）

事务处理失败是由于出现了严重错误或导致数据库或数据库管理器异常结束的情况。部分完成的工作单元或发生故障时未清空至磁盘中的 UOW 使数据库处于不一致状态。在事务处理故障之后必须恢复数据库。导致事务处理故障的情况有：

- 机器上的断电故障，它会导致使用该机器的数据库管理器和数据库分区崩溃
- 硬件故障，例如内存毁坏、磁盘、CPU 或网络故障。
- 导致 DB2 崩溃的严重操作系统错误
- 应用程序异常终止。

如果您希望不完整工作单元的回滚是由数据库管理器自动完成的，那么应将 *autorestart* 数据库配置参数设置为 ON，以启用该自动重新启动参数。（这是缺省值。）如果不想要重新启动行为，那么将 *autorestart* 数据库配置参数设置为 OFF。这样，将需要在数据库故障发生时发出 **RESTART DATABASE** 命令。如果数据库 I/O 在发生崩溃之前已处于暂挂状态，那么必须指定 **RESTART DATABASE** 命令的 **WRITE RESUME** 选项才能使崩溃恢复继续进行。管理通知日志记录数据库重新启动操作开始的时间。

如果对用于正向恢复的数据库应用崩溃恢复（即，未将 *logarchmeth1* 配置参数设置为 OFF），且在崩溃恢复期间因个别表空间而发生错误，那么会让该表空间脱机，直到修复后才能对其进行访问。崩溃恢复继续进行。在崩溃恢复完成时，该数据库中的其他表空间将是可访问的，并且可与该数据库建立连接。但是，如果脱机的表空间包含系统目录，那么必须先修复它才允许进行所有连接。

恢复已损坏的表空间

损坏的表空间带有一个或多个不能访问的容器。这通常是由介质问题引起的：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，脱机磁盘或卸下的文件系统）。

如果损坏的表空间是系统目录表空间，那么不能重新启动数据库。如果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：

- 复原数据库
- 复原目录表空间。

注:

1. 因为必须前滚数据库, 所以表空间复原只对可恢复数据库有效。
2. 如果复原目录表空间, 那么必须执行前滚操作至日志末尾。

如果损坏的表空间不是系统目录表空间, 那么 DB2 会尝试使尽可能多的数据库部分可用。

如果损坏的表空间只是临时的表空间, 应在可进行与数据库的连接后, 尽快创建新的临时表空间。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。如果希望, 可删除脱机的临时表空间。使用系统临时表空间的表重组有特殊的注意事项:

- 如果数据库或数据库管理器配置参数 *indexrec* 设置为 RESTART, 在数据库激活期间必须重建所有无效的索引; 这包括构建阶段期间崩溃的重组中的索引。
- 如果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。

恢复可恢复数据库中的表空间

需要进行崩溃恢复时, 使已损坏的表空间处于脱机状态, 并且不可对它进行访问。将它处于前滚暂挂状态。如果没有其他问题, 即使在数据库带有已损坏的表空间的情况下, 重新启动操作也可以成功地使该数据库联机。一旦处于联机状态, 已损坏的表空间就不可用, 但其余数据库可用。要修复已损坏的表空间并使它可用, 请遵循下面的过程。

要使损坏的表空间可用, 使用以下过程之一:

- 方法 1
 1. 修复损坏但没有丢失原始数据的容器。
 2. 完成表空间前滚操作直至日志的结尾。

注: 前滚操作将首先尝试使表空间由脱机状态变为正常状态。

- 方法 2
 1. 修复损坏且可能丢失原始数据的容器。
 2. 执行表空间复原操作。
 3. 完成表空间前滚操作直至日志的结尾或时间点。

恢复不可恢复数据库中的表空间

在需要进行崩溃恢复时, 如果有已损坏的表空间, 那么只有先删除已损坏的表空间才能成功重新启动数据库。在不可恢复的数据库中, 不保留恢复已损坏的表空间所需的日志。因此, 对这类表空间可以采取的唯一有效操作是删除它们。

要重新启动表空间已损坏的数据库:

1. 调用不合格的重新启动数据库操作。如果没有损坏的表空间, 它将成功。如果失败 (SQL0290N), 那么可以查看管理通知日志文件以获取当前已损坏的表空间的完整列表。
2. 如果您希望删除所有已损坏的表空间, 那么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。如果损坏的表空

间包括在 DROP PENDING TABLESPACES 列表中，那么表空间将进入删除暂挂状态，且必须在恢复操作完成之后删除表空间。

重新启动操作继续，无需恢复指定的表空间。如果损坏的表空间未包括在 DROP PENDING TABLESPACES 列表中，那么重新启动数据库操作将失败，并返回 SQL0290N。

注：在 DROP PENDING TABLESPACES 列表中包括表空间名称并不表示该表空间将处于删除暂挂状态。仅当重新启动操作期间发现表空间损坏时才会处于此种状态。

3. 如果重新启动数据库操作成功，那么调用 LIST TABLESPACES 命令来找出哪些表空间处于删除暂挂状态。
4. 发出 DROP TABLESPACE 语句来删除处于删除暂挂状态的每个表空间。一旦执行了此操作，您就能够收回所损坏表空间使用的空间或重新创建这些表空间。
5. 如果不想删除表空间和丢失损坏的表空间中的数据，您可以：
 - 修复损坏的容器（没有丢失原始数据）。
 - 重新发出 RESTART DATABASE 命令。
 - 执行数据库复原操作。

减少介质故障的影响

要降低介质故障的可能性，并简化从此类型的故障的恢复：

- 镜像或复制保存重要数据库的数据和日志的磁盘。
- 使用“独立磁盘冗余阵列”(RAID) 配置，例如 RAID 级别 5。
- 在分区数据库环境中，要设置一个严格的过程，来处理该目录分区上的数据和日志。因为此数据库分区是维护数据库的关键：
 - 确保它位于可靠的磁盘上
 - 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

如果您担心可能会因为磁盘崩溃而使数据或日志损坏，那么请使用某种形式的磁盘故障容错。通常，这是通过使用磁盘阵列（一组磁盘）来完成的。

磁盘阵列有时只是指 RAID（独立磁盘冗余阵列）。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输入/输出（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，由操作系统或应用程序完成。

硬件磁盘阵列

在硬件磁盘阵列中，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。

有几种类型的 RAID 体系结构，在功能和性能上有所不同，但只有 RAID 级别 1 和级别 5 是现在常用的。

RAID 级别 1 也称为磁盘镜像和双工技术。磁盘镜像技术使用单个磁盘控制器将数据（一个完整的文件）从一个磁盘复制到另一个磁盘。磁盘双工技术类似于磁盘镜像技术，但磁盘是与另一个磁盘控制器连接的（与两个 SCSI 适配器相似）。因此可较好地保护数据：任何一个磁盘发生故障时，仍可从另一个磁盘访问数据。使用磁盘双工技术时虽然磁盘控制器可能会发生故障，但不会影响可提供的数据保护。它的性能很好，但实现它需要的磁盘数是通常磁盘数的两倍。

RAID 级别 5 涉及到跨所有磁盘，按扇区进行的数据和带奇偶校验的条带分割。奇偶性校验与数据交错，而不是存储在专用的驱动器上。数据保护性能很好：如果有任何磁盘发生故障，仍然可以使用其他磁盘上的信息以及已分割的奇偶性校验信息来访问数据。读性能良好，但写性能较差。RAID 级别 5 配置需要最少三个相同的磁盘。开销所需的磁盘空间量随阵列中磁盘的数量而变化。如果 RAID 级别 5 配置了 5 个磁盘，空间开销将是百分之二十。

使用 RAID（但不是 RAID 级别 0）磁盘阵列时，发生故障的磁盘不会影响您访问阵列上的数据。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。使用 RAID 级别 5 时，如果有两个磁盘同时发生故障，所有数据都会丢失（但同时发生磁盘故障的可能性非常小）。

您可能会考虑对日志使用 RAID 级别 1 硬件磁盘阵列或软件磁盘阵列，因为这可以为故障点提供可恢复性并提供良好的写性能（这对日志来说很重要）。要达到此目的，可使用 *mirrorlogpath* 配置参数来指定 RAID 级别 1 文件系统上的镜像日志路径。如果（由于不能将时间花在磁盘故障后的数据恢复上）可靠性很关键，而写性能不是，那么可以考虑使用 RAID 级别 5 硬件磁盘阵列。另外，如果写性能很重要，而不太关心所需的附加磁盘空间量，那么可以考虑对您的数据以及日志使用 RAID 级别 1 硬件磁盘阵列。

有关可用的 RAID 级别的详细信息，访问以下 Web 站点：http://www.acnc.com/04_01_00.html

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，但是磁盘流量是由操作系统或在服务器上运行的应用程序来管理的。软件阵列像其他程序一样，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 负载和容量。

典型的软件磁盘阵列提供磁盘镜像。虽然冗余磁盘是必需的，但是由于不需要高成本的磁盘控制器，所以软件磁盘阵列的实现相对便宜。

注意：

如果操作系统引导驱动器在磁盘阵列中，那么当该驱动器发生故障时，系统将无法启动。如果在磁盘阵列运行之前该驱动器发生故障，那么磁盘阵列就不允许访问该驱动器。引导驱动器应与磁盘阵列分开。

减少事务故障的影响

要降低事务故障的影响，应尽量确保：

- 每个 DB2 服务器配备不间断电源
- 在所有数据库分区上有足够的磁盘空间来存储数据库日志
- 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
- 分区数据库环境中的系统时钟的同步。

从分区数据库环境中的事务故障进行恢复

如果事务处理失败发生在分区数据库环境中，通常需要对发生了故障的数据库分区服务器和参与了该事务的任何其他数据库分区服务器都进行数据库恢复：

- 对发生了故障的数据库分区服务器的崩溃恢复发生在更正了故障情况后。
- 对其他（仍活动的）数据库分区服务器的数据库分区故障恢复紧接在检测到故障后发生。

在分区数据库环境中，提交事务的数据库分区服务器是协调程序分区，而处理该事务的第一个代理程序是协调代理程序。协调代理程序负责将工作分布至其他数据库分区服务器上，并跟踪那些参与了该事务的服务器。当应用程序对一个事务发出 `COMMIT` 语句时，该协调代理程序使用两阶段落实协议来落实该事务。在第一阶段期间，协调程序分区将 `PREPARE` 请求分布至所有其他参与该事务的数据库分区服务器。然后，这些服务器用以下其中一项应答：

READ-ONLY

在此服务器中未发生任何数据更改

YES 在此服务器中发生了数据更改

NO 由于错误，服务器未准备落实

如果其中一个服务器应答 `NO`，那么回滚该事务。否则，协调程序分区开始第二阶段。

在第二阶段，协调程序分区写入一条 `COMMIT` 日志记录，然后将 `COMMIT` 请求分布至所有应答了 `YES` 的服务器。在所有其他数据库分区服务器都已落实后，它们会将 `COMMIT` 的应答发送至协调程序分区。当协调代理程序从所有参与的服务器接收到所有 `COMMIT` 应答时，该事务完成。在此时间点，协调代理程序会写入一条 `FORGET` 日志记录。

活动数据库分区服务器上的事务故障恢复

如果任何数据库分区服务器检测到另一个服务器当机，那么与该发生故障的数据库分区服务器相关的所有工作都会停止：

- 如果仍处于活动状态的数据库分区服务器是某个应用程序的协调程序分区，且该应用程序在发生故障的数据库分区服务器上运行（尚未准备 `COMMIT`），那么会中断该协调代理程序，以便执行故障恢复。如果该协调代理程序处于 `COMMIT` 处理的第二个阶段，会将 `SQL0279N` 返回给应用程序，应用程序随之会丢失它的数据库连接。否则，协调代理程序将一个 `ROLLBACK` 请求分布至所有其他参与该事务的服务器，并将 `SQL1229N` 返回至该应用程序。
- 如果发生故障的数据库分区服务器是该应用程序的协调程序分区，那么仍在活动服务器上为该应用程序工作的代理程序会被中断，以便执行故障恢复。在事务未处于就绪状态的每个数据库分区上本地回滚事务。在事务处于就绪状态的那些数据库分

区上，事务变得不确定。由于协调程序数据库分区不可用，所以协调程序数据库分区不知道事务在某些数据库分区上处于不确定状态。

- 如果该应用程序与发生故障的数据库分区服务器连接（在它发生故障之前），但是本地数据库分区服务器和发生故障的数据库分区服务器都不是协调程序分区，那么会中断为此应用程序工作的代理程序。协调程序分区将向其他数据库分区服务器发送 ROLLBACK 或 DISCONNECT 消息。如果协调程序分区返回 SQL0279，那么事务将仅在仍然活动的数据库分区服务器上处于不确定状态。

试图向该发生故障的服务器发送请求的任何进程（如，代理程序或死锁检测器）都会得到通知：它不能发送该请求。

发生故障的数据库分区服务器上的事务故障恢复

如果事务失败导致数据库管理器异常结束，那么可以发出指定了 RESTART 选项的 db2start 命令，以便在重新启动数据库分区后立即重新启动数据库管理器。如果无法重新启动数据库分区，那么可以发出 db2start，以便在另一数据库分区上重新启动数据库管理器。

如果数据库管理器异常结束，那么服务器上的数据库分区可能会处于不一致状态。要使用它们可用，可以在数据库分区服务器上触发崩溃恢复：

- 通过 RESTART DATABASE 命令显式地触发
- 在 autorestart 数据库配置参数已设置为 ON 后，通过 CONNECT 请求隐式触发

崩溃恢复将重新应用活动日志文件中的日志记录，以确保所有已完成的事务的结果都在数据库中。重新应用了这些更改后，除不确定事务外的所有未落实的事务都将本地回滚。分区数据库环境中有两种类型的不确定事务：

- 在不是协调程序分区的数据库分区服务器上，已就绪但未落实的事务就是不确定的。
- 在协调程序分区上，已落实但还未被记录为完成（即，还未写入 FORGET 记录）的事务是不确定的。当协调代理程序未从为该应用程序工作的所有服务器接收到全部 COMMIT 应答时，会发生这种情况。

崩溃恢复试图通过以下其中一项操作解决所有不确定事务。要执行的操作取决于数据库分区服务器是否为应用程序的协调程序分区：

- 如果重新启动的服务器不是该应用程序的协调程序分区，它会将一个查询消息发送至该协调代理程序，以发现该事务的结果。
- 如果重新启动的服务器是该应用程序的协调程序分区，它会将一个消息发送至协调代理程序仍在等待它们的 COMMIT 应答的所有其他代理程序（下级代理）。

崩溃恢复可能并不能解决所有不确定事务。例如，某些数据库分区服务器可能会不可用。如果协调程序分区在参与事务的其他数据库分区之前完成崩溃恢复，那么崩溃恢复将不能解决不确定事务。因为崩溃恢复由每个数据库分区独立执行，所以上述情况是意料之中的事情。在这种情况下，会返回 SQL 警告消息 SQL1061W。由于不确定事务占用了资源（例如锁定和活动日志空间），有可能导致不能对数据库进行任何更改，因为不确定事务占用了活动日志空间。因此，应确定在崩溃恢复之后是否还有不确定事务，并尽快恢复解决这些不确定事务所需的所有数据库分区服务器。

注：在分区数据库环境中，会对每个节点运行 RESTART 数据库命令。为确保对所有节点重新启动该数据库，请使用以下建议命令：

```
db2_all "db2 restart database <database_name>"
```

如果解决不确定事务所需的一个或多个服务器不能及时恢复，且需要访问其他服务器上的数据库分区，可以通过作出启发式决策来手动解决这些不确定事务。可以使用 LIST INDOUBT TRANSACTIONS 命令来查询、落实和回滚服务器上的不确定事务。

注：LIST INDOUBT TRANSACTIONS 命令还用于分布式事务环境中。为了区分这两种类型的不确定事务，LIST INDOUBT TRANSACTIONS 命令返回的输出中的 *originator* 字段显示以下其中一项：

- DB2 企业服务器版，指示该事务始发于分区数据库环境。
- XA，它指示该事务始发于分布式环境中。

标识发生故障的数据库分区服务器

当一个数据库分区服务器发生故障时，应用程序通常会接收到下列其中一个 SQLCODE。检测哪个数据库管理器发生故障的方法取决于接收到的 SQLCODE：

SQL0279N

当在 COMMIT 处理期间终止的事务中涉及了数据库分区服务器时，会接收到此 SQLCODE。

SQL1224N

当发生故障的数据库分区服务器是该事务的协调程序分区时，会接收到此 SQLCODE。

SQL1229N

当发生故障的数据库分区服务器不是该事务的协调程序分区时，会接收到此 SQLCODE。

确定哪个发生了故障的数据库分区服务器是一个两阶段进程。

1. 通过检查 SQLCA 来查找已检测到故障的分区服务器。与 SQLCODE SQL1229N 相关的 SQLCA 在 *sqlerrd* 字段的第六个数组位置包含检测到错误的服务器的节点号。（为服务器写入的节点号与 *db2nodes.cfg* 文件中的节点号对应。）
2. 针对发生了故障的服务器的节点号，检查在步骤一中找到的有关服务器的管理通知日志。

注：如果正在一个处理器上使用多逻辑节点，那么一个逻辑节点发生故障会导致同一个处理器上的其他逻辑节点发生故障。

从数据库分区服务器的故障恢复

要从数据库分区服务器的故障中恢复，请执行下列步骤。

1. 校正导致该故障的问题。
2. 通过从任何数据库分区服务器发出 *db2start* 命令，重新启动数据库管理器。
3. 通过在发生故障的一个或多个数据库分区服务器上发出 *RESTART DATABASE* 命令，重新启动数据库。

恢复大型机或中型服务器上的不确定事务

当 DB2 Connect 配置了 DB2 同步点管理器时恢复主机上的不确定事务

如果在某个事务期间您的应用程序访问了主机或 System i 数据库服务器，那么恢复不确定事务的方法会有所不同。要访问主机或 System i 数据库服务器，使用 DB2 Connect。如果 DB2 Connect 配置了 DB2 同步点管理器，那么恢复的步骤就会不同。

主机或 System i 服务器上不确定事务的恢复通常由“事务管理器”（TM）和 DB2 同步点管理器（SPM）自动执行。主机或 System i 服务器上的不确定事务不占用本地 DB2 位置的任何资源，但只要该事务在主机或 System i 这一位置上是不确定的，就肯定会占用该位置的资源。如果主机或 System i 服务器的管理员确定必须作出启发性决策，那么管理员可能会与本地 DB2 数据库管理员联系（例如，通过电话）来确定是落实还是回滚主机或 System i 服务器上的事务。如果发生这种情况，那么可以使用 LIST DRDA® INDOUBT TRANSACTIONS 命令来确定 DB2 Connect 实例中事务的状态。可以使用下列步骤作为涉及到 SNA 通信环境的大多数情况的一个参考。

1. 与 SPM 连接，如下所示：

```
db2 => connect to db2spm

Database Connection Information

Database product      = SPM0500
SQL authorization ID = CRUS
Local database alias = DB2SPM
```

2. 发出 LIST DRDA INDOUBT TRANSACTIONS 命令，以显示 SPM 已知的不确定事务。以下示例显示一个 SPM 已知的不确定事务。db_name 是主机或 System i 服务器的本地别名。partner_lu 是主机或 System i 服务器的标准 LU 名称。这为主机或 System i 服务器提供最佳标识，并且应该由主机或 System i 服务器的调用程序提供。luwid 为事务提供唯一标识，且在所有主机和 System i 服务器上都可用。如果显示不确定的事务，而且如果 uow_status 字段的值是 C（落实）或 R（回滚），可以使用该字段来确定事务的结果。如果发出带有 WITH PROMPTING 参数的 LIST DRDA INDOUBT TRANSACTIONS 命令，那么可以用交互式方式落实、回滚或忽略该事务。

```
db2 => list drda indoubt transactions
DRDA Indoubt Transactions:
1.db_name: DBAS3    db_alias: DBAS3    role: AR
   uow_status: C partner_status: I partner_lu: USIBMSY.SY12DQA
corr_tok: USIBMST.STB3327L
   luwid: USIBMST.STB3327.305DFDA5DC00.0001
   xid: 53514C2000000017 00000000544D4442 0000000000305DFD A63055E962000000
      00035F
```

3. 如果未显示 partner_lu 和 luwid 的不确定事务，或者如果 LIST DRDA INDOUBT TRANSACTIONS 命令返回下列内容：

```
db2 => list drda indoubt transactions
SQL1251W No data returned for heuristic query.
```

说明该事务已回滚。

另一种情况发生的可能性不大，但不排除有发生的可能性。如果显示了某个不确定事务，其 partner_lu 对应的 luwid 是合适的，但 uow_status 值为“T”，SPM 就不能确定

应该落实还是回滚该事务。在此情况下，应该使用 WITH PROMPTING 参数在 DB2 Connect 工作站上落实或回滚该事务。然后，允许 DB2 Connect 根据启发式决策与主机或 System i 服务器重新同步。

当 DB2 Connect 不使用 DB2 同步点管理器时恢复主机上的不确定事务

如果在某个事务期间您的应用程序访问了主机或 System i 数据库服务器，那么恢复不确定事务的方法会有所不同。要访问主机或 System i 数据库服务器，请使用 DB2 Connect。如果 DB2 Connect 配置了 DB2 同步点管理器，那么恢复步骤将有所不同。

当在 DB2 Connect 个人版或 DB2 Connect 企业版中的多站点更新中使用 TCP/IP 连接来更新 DB2 z/OS 版但未使用 DB2 同步点管理器时，使用本节中的信息。这种情况中的不确定事务的恢复不同于使用 DB2 同步点管理器的不确定事务的恢复。当在此环境中出现不确定事务时，会在客户机、数据库服务器和/或“事务管理器”(TM) 数据库上生成一个警告条目，这取决于谁检测到该问题。该警告条目被置于 db2alert.log 文件中。

一旦 TM 和参与数据库及其连接再次全部可用，就会自动执行任何不确定事务的再同步。您应该允许在数据库服务器中自动执行再同步，而不是试探性地强制决定。但是，如果您必须这样做，可参考下列步骤。

注：因为未涉及到 DB2 同步点管理器，所以不能使用 LIST DRDA INDOUBT TRANSACTIONS 命令。

1. 在 z/OS 主机上，发出命令 DISPLAY THREAD TYPE(INDOUBT)。

从此列表中，标识您要试探性完成的事务。有关 DISPLAY 命令的详细信息，请参阅 *DB2 for z/OS Command Reference*。显示的 LUWID 可与“事务管理器数据库”中的同一个 luwid 匹配。

2. 根据您要执行的操作，发出 RECOVER THREAD(<LUWID>) ACTION (ABORTICOMMIT) 命令。

有关 RECOVER THREAD 命令的详细信息，请参阅 *DB2 for z/OS Command Reference*。

灾难恢复

术语灾难恢复用于描述在发生火灾、地震、恶意破坏或其他大灾害时复原数据库所需要执行的活动。灾难恢复计划可以包括以下其中一项或多项：

- 在紧急情况下使用的场所
- 用于恢复数据库的另一台机器
- 以非现场方式存储数据库备份和/或表空间备份以及归档日志。

如果灾难恢复计划是在另一台机器上复原整个数据库，那么至少需要完整数据库备份和该数据库的所有归档日志。当您有数据库中每个表空间的完整表空间备份时，虽然也可以重建数据库，但这种方法需要使用许多备份映像，因此，与使用完整数据库备份来进行恢复相比，这种方法更为耗时。

您可以选择将归档日志应用于数据库来使备用数据库保持最新。也可以选择将数据库备份或表空间备份以及日志归档保存在备用场所，而只在发生灾难后才执行复原和前

滚操作。（在后一种情况下，最好使用最近的备份映像。）但是，发生灾难后，通常并不能将所有事务都恢复至发生灾难前的状态。

表空间备份对于灾难恢复的有用性取决于发生故障的范围。通常，如果复原整个数据库，灾难恢复就会比较简单并且不需要很长时间；因此，应该在备用场所保存一份完整数据库备份。如果该灾难是磁盘损坏，那么可以使用该磁盘上每个表空间的表空间备份来进行恢复。如果由于磁盘故障（或任何其他原因）而无法访问某个容器，那么可以将该容器复原到另一位置。

保护数据完整性或防止整个站点故障的另一种方法是实现 DB2 高可用性灾难恢复（HADR）功能。设置此功能后，HADR 通过将数据更改从源数据库（称为主数据库）复制到目标数据库（称为备用数据库）来防止数据丢失。

也可以使用复制功能来保护数据，以防部分站点故障或整个站点故障。复制功能允许定期地将数据复制到多个远程数据库。DB2 数据库提供了许多复制工具，这些工具允许您指定应该复制的数据、应该将数据复制到的数据库表以及复制更新内容的频率。

也可以使用诸如对等远程复制（PPPC）之类的存储器镜像功能来保护数据。PPPC 提供了卷或磁盘同步复制功能来预防灾难。PPPC 提供了卷或磁盘同步复制功能来预防灾难。

在进行灾难恢复规划时，DB2 提供了若干选项。根据业务需要，您可能决定使用表空间或完整数据库备份来防止数据丢失，也可能认为您的环境更适合于使用诸如 HADR 之类的解决方案。不论作出何种选择，都应该在测试环境中测试恢复过程，然后在生产环境中实现这些过程。

版本恢复

版本恢复指的是使用备份操作期间创建的映像来复原数据库的先前版本。对不可恢复数据库（即，该数据库没有归档日志）使用此方法。还可通过对 `RESTORE DATABASE` 命令使用 `WITHOUT ROLLING FORWARD` 选项，来对可恢复数据库使用此方法。数据库复原操作将使用先前创建的备份映像来复原整个数据库。数据库备份使您可以将数据库复原到与进行备份时完全相同的状态。但是，从备份时间到故障时间之间的每个工作单元都将丢失（请参阅第 205 页的图 18）。

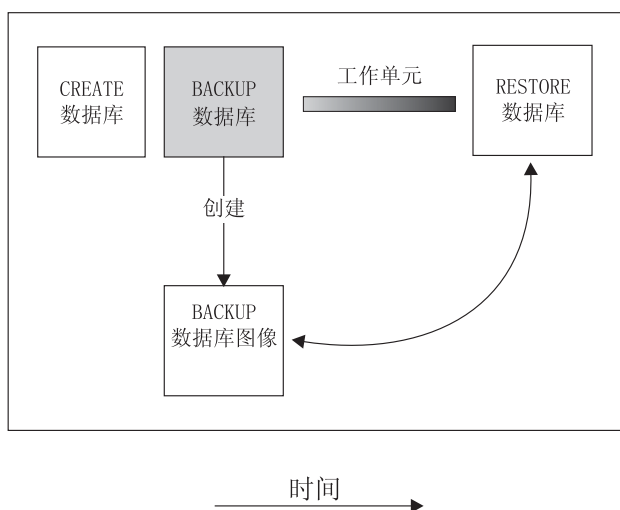


图 18. 版本恢复. 显示从进行备份开始到发生故障时为止工作单元将丢失。

使用版本恢复方法，必须定期安排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。必须复原所有数据库分区，而且用于复原数据库操作的所有备份映像必须是同时创建的。（将备份每个数据库分区并单独复原。）同时建立的每个数据库分区的备份称为版本备份。

前滚恢复

要使用前滚恢复方法，必须已建立数据库的备份，并且已归档日志（方法是将 *logarchmeth1* 和 *logarchmeth2* 配置参数设置为 OFF 之外的值）。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像时的状态。如果复原数据库，但没有对复原数据库操作指定 **WITHOUT ROLLING FORWARD** 选项，那么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。

注：在下列情况下，不能使用 **WITHOUT ROLLING FORWARD** 选项：

- 正在从联机备份映像复原
- 正在发出表空间级复原

要考虑的两种恢复类型是：

- **数据库前滚恢复。**在此类型的前滚恢复中，记录在数据库日志中的事务应用到以下数据库复原操作中（请参阅第 206 页的图 19）。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）。

在分区数据库环境中，数据库分布在许多数据库分区中，因此必须对数据库目录表所在的数据库分区（目录分区）发出 **ROLLFORWARD DATABASE** 命令。如果正在执行时间点前滚恢复，那么必须前滚所有数据库分区，以确保所有数据库分区处于同一级别。如果需要复原单个数据库分区，那么可以将前滚恢复执行至日志末尾，将其复原到数据库中其他数据库分区所处的级别。如果前滚一个数据库分区，那

么只可使用对日志末尾的恢复。时间点恢复适用于所有数据库分区。

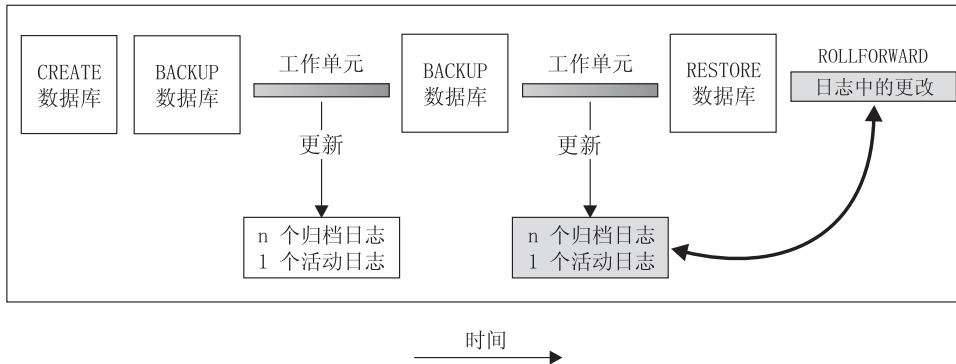


图 19. 数据库前滚恢复. 在运行时间较长的事务中, 可以有多个活动日志。

- 表空间前滚恢复。如果启用了某个数据库对它进行正向恢复, 也可对它进行备份、复原并前滚表空间 (请参阅第 207 页的图 20)。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) 或一个或多个单独表空间的备份映像。还需要影响要恢复的表空间的日志记录。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)。

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` 命令将日志应用于表空间以使其前滚至某个时间点或日志末尾。
- 如果一个或多个表空间在崩溃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涉及复原数据库操作。例如, 掉电可能导致表空间处于前滚暂挂状态。在此情况下, 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, 使其恢复至日志末尾。如果在进行崩溃恢复之前解决此问题, 那么崩溃恢复操作足以使数据库恢复到一致并且可用的状态。

注: 如果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。

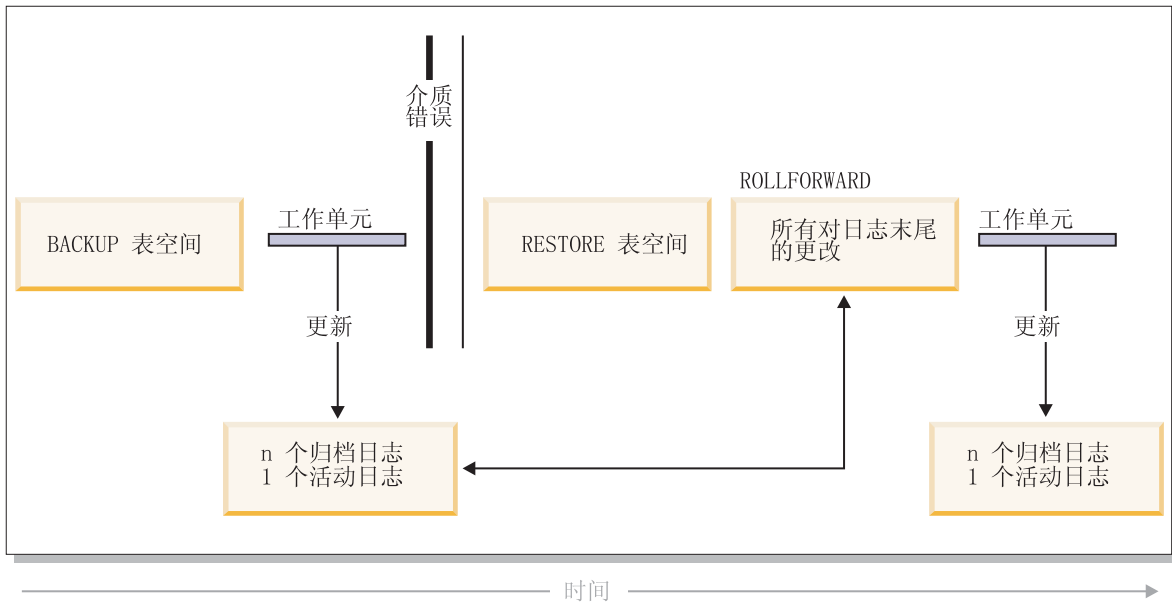


图 20. 表空间前滚恢复. 在运行时间较长的事务中，可以有多个活动日志。

在分区数据库环境中，如果要将表空间前滚到某个时间点，那么不必提供该表空间所在的数据库分区列表。DB2 将前滚请求提交到所有数据库分区。这意味着必须在表空间驻所在的所有数据库分区上复原表空间。

在分区数据库环境中，如果要将表空间前滚至日志末尾，而又不希望在所有数据库分区上前滚表空间，那么必须提供数据库分区列表。如果要将（所有数据库分区上）所有处于前滚暂挂状态的表空间前滚至日志末尾，那么不必提供数据库分区列表。缺省情况下，会将数据库前滚请求发送至所有数据库分区。

如果将包含任何分区表部分的表空间前滚到某个时间点，那么还必须将所有其他包含该表的表空间前滚到同一时间点。但是，可以将包含部分分区表的单个表空间前滚至日志末尾。

注：如果分区表带有任何已连接、已拆离或已删除数据分区，那么时间点前滚还必须包括这些数据分区的所有表空间。要确定某个分区表是否带有任何已连接、已拆离或已删除数据分区，请查询 `SYSDATAPARTITIONS` 目录表。

增量备份与恢复

由于数据库，尤其是仓库的大小持续扩展到太字节和百万兆字节范围，备份和恢复这些数据库所需的时间及硬件资源也有了大幅的增长。因为对大数据库进行多个备份所需的存储量是巨大的，所以完整的数据库和表空间备份并不总是处理大数据库的最佳途径。请考虑以下问题：

- 仓库中只有一小部分数据更改时，应不需要备份整个数据库。
- 将表空间追加到现有数据库然后只备份表空间是危险的作法，原因是无法保证在表空间备份之间，已备份的表空间外没有任何更改。

为了解决这些问题，DB2 提供了增量备份和恢复。增量备份是一个备份映像，它只包含自上次进行备份以来有过更新的页。除更新的数据和索引页之外，每个增量备份映像还包含通常存储在完整备份映像中的所有初始数据库元数据（例如，数据库配置、表空间定义和数据库历史记录等等）。

注：

1. 如果表空间包含长型字段或大对象数据，并执行增量备份，并且如果自从上次备份之后已修改该表空间中的任何页，那么将把所有长型字段和大对象数据复制到备份映像中。
2. 如果对包含脏页（即，此页包含的数据已被更改但尚未写入磁盘）的表空间执行增量备份，就会备份所有大对象数据。对于正常数据来说，仅当这些数据已更改时，才会对它们进行备份。

支持两种类型的增量备份：

- **增量备份** 增量备份映像是自从上次最新的、成功的完整备份操作以来，更改过的所有数据库数据的副本。也称为累积备份映像，因为进行的一系列增量备份中的每一个都会有上次增量备份映像的内容。增量备份映像的前身通常是同一对象最新的、成功的完整备份。
- **差异备份** 差异备份映像或增量差异备份映像是自从上次相关表空间的成功备份（包括完整、增量或差异备份）以来，已更改过的所有数据库数据的副本。也称为差异备份映像或非累积备份映像。差异备份映像的前身是最新的成功备份，包括差异备份映像中每个表空间的备份。

增量备份映像和差异备份映像的关键差别在于：对连续不断更改的对象进行持续备份时，它们的行为不同。每个连续增量映像都包含了前一个增量映像的完整内容以及自上一次生成完整备份后更改过的或新增的任何数据。差异备份映像只包含自上次生成任何类型的映像后更改过的页。

允许以联机和脱机操作方式组合数据库和表空间增量备份。计划备份策略时应格外小心，因为将数据库增量备份和表空间增量备份结合在一起即指示了数据库备份（或多个表空间的表空间备份）的前身不再需要是一个单独的映像，它可以是在不同时间进行的先前数据库和表空间备份的唯一集合。

要将数据库或表空间复原到一致状态，恢复过程必须以要复原的整个对象（数据库或表空间）的一致映像开始，然后必须按以下描述的顺序应用每个相应的增量备份映像。

为启用对数据库更新的跟踪，DB2 支持新的数据库配置参数 *trackmod*，它可以是以下两个可接受的值中的一个：

- **NO**。增量备份不允许使用此配置。不会以任何方式跟踪或记录数据库页更新。这是缺省值。
- **YES**。增量备份允许使用此配置。启用了更新跟踪后，更改会对与数据库的第一个成功连接生效。必须选对该表空间进行完整备份，才能对特定表空间进行增量备份。

对于 SMS 和 DMS 表空间，此跟踪的详细程度为表空间级。在表空间级的跟踪中，每个表空间的标志指示该表空间中是否存在需要备份的页。如果表空间不存在任何需要备份的页，那么备份操作可以完全跳过该表空间。

对数据库的更新跟踪会对更新或插入数据的事务的运行性能产生较小影响。

从增量备份映像复原

- 根据增量备份映像进行的复原操作通常包括下列步骤:

1. 标识增量目标映像。

确定要复原的最终映像，并从 DB2 RESTORE 实用程序请求增量复原操作。此映像也称为增量复原的目标映像，因为它将成为要复原的最后映像。增量目标映像是使用 RESTORE DATABASE 命令中的 TAKEN AT 参数指定的。

2. 复原最新的完整数据库或表空间映像以建立一个基线，以根据它来应用每个后继的增量备份映像。
3. 按产生的顺序，在“步骤 2”中复原的基线映像的顶部，复原需要的各个完整增量备份映像或表空间增量备份映像。
4. 重复“步骤 3”直到“步骤 1”中的目标映像读了两次。在整个增量复原操作期间会访问两次目标映像。在第一次访问期间，只从映像读取初始数据，而不读取任何用户数据。只在第二次访问期间才读取并处理完整的映像。

必须访问两次增量复原操作的目标映像，以确保数据库最初是使用正确的历史记录、数据库配置以及将在复原操作期间创建的数据库表空间定义来配置的。如果自从进行了最初的完整数据库备份映像以来已删除了表空间，将从备份映像中读取该映像的表空间数据，但在增量复原处理期间会忽略该数据。

- 有两种方法来复原增量备份映像。
 - 对于自动增量复原，但在指定要使用的目标映像时发出 RESTORE 命令一次。然后，DB2 使用数据库历史记录来确定余下的必需备份映像并复原它们。
 - 对于手动增量复原，必须对需要复原的每个备份映像发出一次 RESTORE 命令（如上面的步骤中所述）。

- **自动增量复原示例**

要复原一组增量备份映像，使用自动增量备份，对 RESTORE DATABASE 命令指定“TAKEN AT 时间戳记”选项。使用想要复原的上一个映像的时间戳记。例如：

```
db2 restore db sample incremental automatic taken at 20031228152133
```

这样就可以使 DB2 RESTORE 实用程序自动执行本节开头所述的每个步骤。在处理的最初阶段，将读取时间戳记为 20001228152133 的备份映像，RESTORE 实用程序将验证数据库、其历史记录和表空间定义是否存在并有效。

在处理的第二阶段，将查询数据库历史记录来构建执行请求的复原操作所需的备份映像链。如果由于某些原因上述操作不可能实现，那么 DB2 无法构建所需映像的完整链，复原操作会终止并返回错误消息。此时，不能进行自动复原，应发出指定了 INCREMENTAL ABORT 选项的 RESTORE DATABASE 命令。这将清除所有余下的资源，以便可以继续手动增量复原。

注意：强烈建议不要使用 PRUNE HISTORY 命令的 FORCE 选项。此命令的缺省操作可防止您删除历史记录条目（从最新的完整数据库备份映像恢复时可能会需要它们），但如果使用 FORCE 选项，就有可能删除自动复原操作所需的条目。

在处理的第三个阶段，DB2 将复原生成链中余下的每个备份映像。如果在此阶段期间发生了错误，那么将必须发出指定了 INCREMENTAL ABORT 选项的 RESTORE

DATABASE 命令来清除所有余下的资源。然后，必须确定是否能解决错误，才能重新发出 RESTORE 命令或再次尝试手动增量复原。

- **手动增量复原示例**

要复原一组增量备份映像，使用手动增量复原，并通过使用 RESTORE DATABASE 命令的 TAKEN AT timestamp 选项来指定目标映像，并遵循上面所述的步骤。例如：

1.

```
db2 restore database sample incremental taken at <ts>
```

其中：<ts> 指向要复原的最后一个增量备份映像（目标映像）。

2.

```
db2 restore database sample incremental taken at <ts1>
```

其中：<ts1> 指向初始完整数据库（或表空间）映像。

3.

```
db2 restore database sample incremental taken at <tsX>
```

其中：<tsX> 指向按创建顺序排列的每个增量备份映像。

4. 重复“步骤 3”，复原每个增量备份映像，直到包括映像 <ts>。

如果正在执行数据库复原操作，并已生成表空间备份映像，那么必须以表空间映像备份时间戳记的年代顺序复原这些表空间映像。

可以使用 db2ckrst 实用程序来查询数据库历史记录，并生成增量复原所需的备份映像时间戳记列表。同时还生成用于手动增量复原的简化了的复原语法。建议保留备份的完整记录，并仅将此实用程序用作指南。

自动增量复原的局限性

1. 如果在执行要复原的备份操作后已更改了表空间名，并且在发发表空间级别复原操作时使用了新名称，那么将无法根据数据库历史记录正确生成所需的备份映像链，并且会发生错误（SQL2571N）。

示例：

```
db2 backup db sample -> <ts1>
db2 backup db sample incremental -> <ts2>
db2 rename tablespace from userspace1 to t1
db2 restore db sample tablespace ('t1') incremental automatic taken
at <ts2>
SQL2571N 自动增量复原未能继续。原因码： "3"。
```

建议的变通方法：使用手动增量复原。

2. 如果删除数据库，数据库历史记录也会删除。如果复原已删除的数据库，数据库历史记录将复原到它在复原备份时的状态，而自那以后的所有历史记录条目都将丢失。如果稍后尝试执行需要使用其中任何一个已丢失的历史记录条目的自动增量复原，那么 RESTORE 实用程序将尝试复原不正确的备份链，并返回“顺序不对”错误（SQL2572N）。

示例：

```
db2 backup db sample -> <ts1>
db2 backup db sample incremental -> <ts2>
db2 backup db sample incremental delta -> <ts3>
db2 backup db sample incremental delta -> <ts4>
db2 drop db sample
db2 restore db sample incremental automatic taken at <ts2>
db2 restore db sample incremental automatic taken at <ts4>
```

建议的变通方法:

- 使用手动增量复原。
 - 在发出自动增量复原命令前，首先从映像 <ts4> 复原历史记录文件。
3. 如果将备份映像从一个数据库复原到另一个数据库，然后建立增量备份（差异备份），那么不能再使用自动增量复原原来复原此备份映像。

示例:

```
db2 create db a
db2 create db b

db2 update db cfg for a using trackmod on

db2 backup db a -> ts1
db2 restore db a taken at ts1 into b

db2 backup db b incremental -> ts2

db2 restore db b incremental automatic taken at ts2
```

QL2542N 根据提供的源数据库别名"B"和时间戳记"ts1"，找不到数据库映像文件的匹配项。

建议的变通方法:

- 使用如下所示的手动增量复原:

```
db2 restore db b incremental taken at ts2
db2 restore db a incremental taken at ts1 into b
db2 restore db b incremental taken at ts2
```
- 在对数据库 B 执行手动复原操作之后，发出完整数据库备份以启动新的增量链

优化恢复性能

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调页的磁盘，而该磁盘没有足够的实内存，这样会破坏调整。

通过为缓冲区数目、缓冲区大小和并行性设置选择最佳值，DB2 自动尝试将完成备份或复原操作所需的时间降至最小。此值根据可用实用程序堆内存量、可用处理器数和数据库配置而定。

- 要缩短完成一次复原操作所需的时间，请使用多个源设备。
- 如果一个表包含大量的长型字段和 LOB 数据，那么复原它可能会占用非常多的时间。如果允许数据库进行前滚恢复，那么 RESTORE 命令能够复原选择的表空间。如果该

长型字段和 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。通过将长型字段和 LOB 数据存储在单独的表空间中，不选择复原包含该长型字段和 LOB 数据的表空间，可以减少完成复原操作所需的时间。如果可从单独的源复制 LOB 数据，那么当创建或改变一个表以包括 LOB 列时选择 NOT LOGGED 选项。如果选择不复原包含长型字段和 LOB 数据的表空间，但需要复原包含该表的表空间，那么必须前滚至日志末尾，以便所有包含表数据的表空间都是一致的。

注：如果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，那么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：
 - 应该使用多个设备。
 - 不要使 I/O 设备控制器带宽超负荷。
- DB2 使用多个代理程序来执行崩溃恢复和数据库前滚恢复。您可能会发现执行这些操作期间的性能更佳（尤其是在对称多处理器（SMP）机器上）；在数据库恢复期间使用多个代理程序可利用 SMP 机器上可用的多余 CPU。

并行恢复引入的代理程序类型是 db2agnsc。DB2 根据机器上 CPU 的数量选择将用于数据库恢复的代理程序的数量。

DB2 将日志记录分发到这些代理程序以使它们可在适当的时候并发地重新应用。例如，可通过此方法使那些与插入、删除、更新、添加键和删除键操作相关的日志记录的处理并行化。因为日志记录是在页级并行化的（相同数据页上的日志记录由相同的代理程序处理），所以即使所有工作都是对一个表完成的，性能还是会增强。

- 执行恢复操作时，DB2 将自动为缓冲区个数、缓冲区大小和并行性设置选择最佳值。此值根据可用实用程序堆内存的数量、可用处理器数和数据库配置而定。因此，根据系统上可用的存储量，应考虑通过增大 UTIL_HEAP_SZ 配置参数来分配更多内存。

使用 **recover** 所需的特权、权限和授权

特权使用户能够创建或访问数据库资源。权限级别提供了对特权、较高级别数据库管理器维护和实用程序操作进行分组的方法。这两者一起用于控制对数据库管理器及其数据库对象的访问。用户只能访问那些他们具有相应授权（即必需的特权或权限）的对象。

必须具有 SYSADM、SYSCTRL 或 SYSMAINT 权限才能使用 RECOVER 实用程序。

第 13 章 restore 概述

DB2 RESTORE DATABASE 命令的最简单的格式只需要您指定要复原的数据库的别名。
例如:

```
db2 restore db sample
```

在此示例中，因为 SAMPLE 数据库存在且在发出 RESTORE DATABASE 命令时将被替换，所以返回以下消息:

```
SQL2539W 警告！正在复原到与备份映像数据库相同的现有数据库。数据库文件将被删除。  
要继续吗？(y/n)
```

如果指定 y，复原操作将成功完成。

数据库复原操作需要一个互斥连接：即，启动该任务后，RESTORE 实用程序会防止其他应用程序访问数据库，直到复原操作成功完成，所以不能再对该数据库运行任何应用程序。但表空间复原操作可以联机完成。

直到复原操作（可能后跟前滚恢复）成功完成，表空间才可用。

如果有跨越多个表空间的表，那么应该一起备份并复原这个表空间集合。

执行部分或子集复原操作时，可以使用表空间级别的备份映像或完整数据库级别的备份映像，并从该映像中选择一个或多个表空间。从建立备份映像时开始的所有与这些表空间相关的日志文件必须存在。

可以将数据库从在 32 位级别创建的备份映像复原为 64 位级别，但反之则不然。

DB2 BACKUP 实用程序和 RESTORE 实用程序应该用来备份和复原数据库。建议不要将文件集从一台机器移至另一台机器，因为这可能破坏数据库的完整性。

使用复原

在发生问题（例如介质或存储器故障，或者应用程序故障）后使用 RESTORE DATABASE 命令来恢复数据库或表空间。如果已经备份了数据库或各个表空间，那么可以在它们由于某种原因损坏时对其进行重建。

复原到现有数据库时，不应连接至要复原的数据库：RESTORE 实用程序自动建立与特定数据库的连接，并且此连接在复原操作完成时终止。复原到新数据库时，需要实例连接才能创建数据库。复原到新远程数据库时，必须首先连接至新数据库所在的实例。然后创建新数据库，指定服务器的代码页和地域。复原将使用备份映像的代码页来覆盖目标数据库的代码页。

数据库可以是本地数据库或远程数据库。

RESTORE 实用程序有以下限制:

- 仅当先前已使用 DB2 BACKUP 实用程序备份了数据库时，才能使用 RESTORE 实用程序。

- 如果非实例所有者用户（在 UNIX 上）或者 DB2ADMNS 或 Administrators 组的成员（在 Windows 上）尝试恢复备份映像，那么他们将接收到错误（SQL2061N）。如果其他用户需要访问备份映像，那么需要在生成备份后更改文件许可权。
- 数据库复原操作在正在运行前滚进程时不能启动。
- 要将一个表空间复原到现有数据库中，该表空间当前必须已存在，并且是同一个表空间；“同一表空间”表示在备份与复原操作之间未删除并重新创建该表空间。磁盘上的数据库与备份映像中的数据库必须相同。
- 对于表空间级备份来说，不能对其执行表空间级复原以将其复原到新数据库。
- 不能执行涉及系统目录表的联机表空间级复原操作。
- 对于在单一数据库分区环境中创建的备份来说，不能将其复原到现有分区数据库环境中。而是，必须将该备份复原到单一数据库分区环境，然后根据需要添加数据库分区。
- 将使用一个代码页的备份映像复原成使用另一个代码页的系统时，备份映像的代码页将覆盖系统代码页。
- 不能使用 RESTORE DATABASE 命令将未启用自动存储器的表空间转换为自动存储器启用的表空间。

可以通过命令行处理器（CLP）、控制中心的“复原数据库”笔记本或向导或者 db2Restore 应用程序编程接口（API）来调用 RESTORE 实用程序。

以下是通过 CLP 发出的 RESTORE DATABASE 命令的示例：

```
db2 restore db sample from D:\DB2Backups taken at 20010320122644
```

要打开“复原”向导：

1. 从“控制中心”中，展开对象树，直到找到要复原的数据库或表空间对象为止。
2. 右键单击该对象，然后从弹出菜单中选择“复原”。这就打开了“复原”向导。

在“控制中心”中通过上下文帮助工具提供详细信息。

从快照备份映像复原

从快照备份复原使用存储设备的快速复制技术来执行复原的数据复制部分。

开始前

要执行快照备份与复原操作，需要适用于存储设备的 DB2 ACS API 驱动程序。集成到 IBM 数据服务器的是适用于下列存储硬件的 DB2 ACS API 驱动程序：

- IBM TotalStorage SAN Volume Controller
- IBM 系统存储器 DS6000
- IBM 系统存储器 DS8000
- IBM 系统存储器 N 系列
- NetApp V 系列
- NetApp FAS 系列

必须执行快照备份，然后才可以从快照备份复原。请参阅：第 177 页的『执行快照备份』。

过程

可以使用带 **USE SNAPSHOT** 参数的 **RESTORE DATABASE** 命令或带有 **SQLU_SNAPSHOT_MEDIA** 介质类型的 **db2Restore** API 来从快照备份复原:

RESTORE DATABASE 命令:

```
db2 restore db sample use snapshot
```

db2Restore API

```
int sampleRestoreFunction( char dbAlias[],
                          char restoredDbAlias[],
                          char user[],
                          char pswd[],
                          char workingPath[] )
{
    db2MediaListStruct mediaListStruct = { 0 };

    rmediaListStruct.locations = &workingPath;
    rmediaListStruct.numLocations = 1;
    rmediaListStruct.locationType = SQLU_SNAPSHOT_MEDIA;

    db2RestoreStruct restoreStruct = { 0 };

    restoreStruct.piSourceDBAlias = dbAlias;
    restoreStruct.piTargetDBAlias = restoredDbAlias;
    restoreStruct.piMediaList = &mediaListStruct;
    restoreStruct.piUsername = user;
    restoreStruct.piPassword = pswd;
    restoreStruct.iCallerAction = DB2RESTORE_STORDEF_NOINTERRUPT;

    struct sqlca sqlca = { 0 };

    db2Restore(db2Version900, &restoreStruct, &sqlca);

    return 0;
}
```

复原到现有的数据库

可以将任何数据库或表空间备份映像复原到现有数据库。对于数据库级别复原来讲，备份映像可能在别名、数据库名称或数据库种子值等方面与现有数据库有所不同。对于表空间级复原来讲，要将一个表空间复原到现有数据库中，该表空间当前必须已存在并且是同一表空间；“同一表空间”表示在备份与复原操作之间未删除并重新创建该表空间。磁盘上的数据库与备份映像中的数据库必须相同。

数据库种子值是数据库的唯一标识，它在该数据库的整个生命期永不更改。种子值是在创建数据库时由数据库管理器指定的。DB2 始终使用备份映像中的种子值。

复原到现有的数据库时，**RESTORE** 实用程序会:

- 删除现有数据库中的表、索引和长型字段数据，并用备份映像中的数据将它们替换。
- 替换表示要复原的每个表空间的表条目。
- 保留恢复历史记录文件，除非它已损坏或者没有任何条目。如果恢复历史记录文件已损坏或不包含任何条目，数据库管理器会从备份映像中复制文件。如果要替换恢复历史记录文件，可以发出指定了 **REPLACE HISTORY FILE** 选项的 **RESTORE** 命令。
- 保留现有数据库的认证类型。

- 保留现有数据库的数据库目录。该目录定义了数据库的所在位置与编目方式。
- 比较数据库种子值。如果种子值不同：
 - 删除与现有的数据库相关的日志。
 - 从备份映像中复制数据库配置文件。
 - 如果在 `RESTORE DATABASE` 命令中指定了 `NEWLOGPATH`，应将 `NEWLOGPATH` 设置为 `logpath` 数据库配置参数的值。

如果数据库种子值相同：

- 如果映像是不可恢复数据库，那么删除日志。
- 保留当前数据库配置文件。
- 如果在 `RESTORE DATABASE` 命令中指定了 `NEWLOGPATH`，应将 `NEWLOGPATH` 设置为 `logpath` 数据库配置参数的值；否则应将当前日志路径复制到数据库配置文件。验证日志路径：如果该路径不能由数据库使用，应更改数据库配置以使用缺省日志路径。

复原到新的数据库

可以创建一个新数据库并向它复原完整的数据库备份映像。如果未创建新的数据库，那么 `RESTORE` 实用程序将创建一个数据库。

复原到新数据库时，`RESTORE` 实用程序：

- 使用通过目标数据库别名参数指定的数据库别名创建新数据库。（如果未指定目标数据库别名，`RESTORE` 实用程序会使用通过源数据库别名参数指定的相同别名来创建数据库。）
- 从备份映像复原数据库配置文件。
- 如果在 `RESTORE DATABASE` 命令中指定了 `NEWLOGPATH`，应将 `NEWLOGPATH` 设置为 `logpath` 数据库配置参数的值。验证日志路径：如果该路径不能由数据库使用，应更改数据库配置以使用缺省日志路径。
- 从备份映像复原认证类型。
- 从备份映像中的数据库目录复原注释。
- 复原数据库的恢复历史记录文件。
- 使用备份映像的代码页来覆盖数据库代码页。

在测试和生产环境中使用增量复原

一旦对生产数据库启用增量备份与恢复，就可以使用增量或差异备份映像来创建或刷新测试数据库。可以通过使用手动或自动增量复原来完成此任务。要将备份映像从生产数据库复原到测试数据库，在 `RESTORE DATABASE` 命令上使用 `INTO target-database-alias` 选项。例如，在具有下列备份映像的生产数据库中：

```
backup db prod
备份成功。此备份的时间戳记为: <ts1>
```

```
backup db prod incremental
备份成功。此备份的时间戳记为: <ts2>
```

手动增量复原的一个示例将是：

```
restore db prod incremental taken at <ts2> into test without
prompting
DB20000I 已成功完成 RESTORE DATABASE 命令。
```

```
restore db prod incremental taken at <ts1> into test without
prompting
DB20000I 已成功完成 RESTORE DATABASE 命令。
restore db prod incremental taken at <ts2> into test without
prompting
DB20000I 已成功完成 RESTORE DATABASE 命令。
```

如果数据库 TEST 已存在，那么复原操作将覆盖已在其中的任何数据。如果数据库 TEST 不存在，那么 RESTORE 实用程序将创建它，并使用备份映像的数据对它进行填充。

因为自动增量复原操作依赖于数据库历史记录，所以根据测试数据库是否存在，复原步骤将稍微有所变化。要对数据库 TEST 执行自动增量复原，它的历史记录必须包含数据库 PROD 的备份映像历史记录。该备份映像的数据库历史记录将替换对于数据库 TEST 已存在的所有数据库历史记录，条件是：

- 当发出 RESTORE DATABASE 命令时数据库 TEST 不存在，或
- 当发出 RESTORE DATABASE 命令时，数据库 TEST 存在且数据库 TEST 历史记录不包含任何记录。

以下示例显示对不存在的数据库 TEST 的自动增量复原：

```
restore db prod incremental automatic taken at <ts2> into test without
prompting
DB20000I 已成功完成 RESTORE DATABASE 命令。
```

RESTORE 实用程序将创建 TEST 数据库，然后将它填充。

如果数据库 TEST 确实存在且数据库历史记录不为空，那么必须删除该数据库，才能执行自动增量复原操作，如下所示：

```
drop db test
DB20000I 已成功完成 DROP DATABASE 命令。

restore db prod incremental automatic taken at <ts2> into test without
prompting
DB20000I 已成功完成 RESTORE DATABASE 命令。
```

如果不想删除该数据库，那么在发出 RESTORE DATABASE 命令之前可以通过使用指向将来的时间戳记和 WITH FORCE OPTION 参数来发出 PRUNE HISTORY 命令：

```
connect to test
Database Connection Information

数据库服务器           = <server id>
SQL 授权标识           = <id>
本地数据库别名       = TEST

prune history 9999 with force option
DB20000I 已成功完成 PRUNE 命令。

connect reset
DB20000I 已成功完成 SQL 命令。
restore db prod incremental automatic taken at <ts2> into test without
prompting
QL2540W 复原成功，但在以"无任何中断"方式处理时在"数据库复原"期间遇到了
警告"2539"。
```

这种情况下，RESTORE DATABASE COMMAND 将以与数据库 TEST 不存在时的同一方式运行。

如果数据库 TEST 确定存在且数据库历史记录为空，不必删除数据库 TEST 就可以执行自动增量复原操作：

```
restore db prod incremental automatic taken at <ts2> into test without
prompting
QL2540W 复原成功，但在以"无任何中断"方式处理时在"数据库复原"期间遇到了
警告"2539"。
```

可以继续建立测试数据库的增量或差异备份而不用先建立完整数据库备份。但是，如果需要复原其中一个增量映像或差异映像，那么将必须执行手动增量复原。这是因为自动增量复原操作要求自动增量复原期间复原的每个备份映像是根据同一个数据库别名创建的。

如果在使用生产备份映像完成复原操作之后建立测试数据库的完整备份映像，那么可以建立增量或差异备份并通过使用手动或自动方式对它们进行复原。

执行重定向复原操作

在下列其中一种情况下，将执行重定向复原操作：

- 要将备份映像复原到不同于源机器的目标机器上
- 要将表空间容器复原到另一个物理位置中
- 由于至少一个容器不可访问，所以复原操作失败

注：不能使用重定向复原操作将数据从一个操作系统移至另一个操作系统。

在重定向复原操作期间，如果目录和文件容器不存在，将自动创建。数据库管理器不会自动创建设备容器。

DB2 只支持添加、更改或删除 DMS 表空间的表空间容器。对于 SMS 表空间，重定向复原操作是唯一可用来修改表空间容器配置的方法。

可以通过调用 RESTORE DATABASE 命令并指定 REDIRECT 参数，或者通过使用控制中心中的“复原数据库”向导来重新定义表空间容器。增量备份映像重定向复原操作的调用过程与非增量备份映像重定向复原操作的调用过程相似。请发出指定了 REDIRECT 选项的 RESTORE DATABASE 命令，并指定应该用于数据库增量复原操作的备份映像。此外，可以根据备份映像生成重定向复原脚本，然后就可以在需要时修改该脚本。请参阅第 221 页的『使用自动生成的脚本来执行重定向复原』。

容器重定向操作为管理表空间容器提供了相当大的灵活性。例如，即使不支持向 SMS 表空间添加容器，您也可以通过在调用重定向复原操作时指定其他容器来达到此目的。

示例

重定向复原操作包括一个由两个步骤组成的数据库复原过程和一个干扰表空间容器定义步骤：

1. 发出带有 REDIRECT 选项的 RESTORE DATABASE 命令。
2. 使用 SET TABLESPACE CONTAINERS 命令来定义已复原数据库的表空间容器（指定目标系统上的表空间位置）
3. 再次发出 RESTORE DATABASE 命令，这次指定 CONTINUE 选项。

以下示例显示如何对数据库 SAMPLE 执行重定向复原操作：

```

db2 restore db sample redirect without prompting
SQL1277W 正在执行重定向复原操作。
现在可以查看表空间配置，但不能查看表空间
使用自动存储器可以重新配置它们的容器。
DB20000I 已成功完成 RESTORE DATABASE 命令。
db2 set tablespace containers for 2 using (path 'userspace1.0', path
'userspace1.1')
DB20000I 已成功完成 SET TABLESPACE CONTAINERS 命令。
db2 restore db sample continue
DB20000I 已成功完成 RESTORE DATABASE 命令。

```

通过使用自动生成的脚本复原数据库来重新定义表空间容器

复原数据库时，RESTORE 实用程序假定物理容器布局与备份数据库时的物理容器布局相同。如果需要更改任何物理容器的位置或大小，那么必须发出指定了 REDIRECT 选项的 RESTORE DATABASE 命令。使用此选项时，需要指定存储在备份映像中的物理容器的位置，并且需要提供每个将要改变的非自动表空间的全部容器。可以在备份时捕获容器信息，但是这样做会很繁琐。

为了便于执行重定向复原，RESTORE 实用程序允许根据现有备份映像生成重定向复原脚本，方法是发出指定了 REDIRECT 和 GENERATE SCRIPT 选项的 RESTORE DATABASE 命令。RESTORE 实用程序将检查备份映像、从备份映像中提取容器信息并生成 CLP 脚本，该脚本包含容器的所有详细信息。接着，可以在脚本中修改任何路径或容器大小，然后运行该 CLP 脚本以使用一组新容器来重新创建数据库。即使只有备份映像，而不知道容器布局，也可以使用生成的脚本来复原数据库。该脚本是在客户机上创建的。通过使用该脚本作为基础，可以确定复原后数据库的日志文件和容器所需空间的位置，并且可以相应地更改日志文件和容器路径。

生成的脚本由四节组成：

初始化 第一节设置命令选项并指定要运行命令的数据库分区。以下是第一节的示例：

```

UPDATE COMMAND OPTIONS USING S ON Z ON SAMPLE_NODE0000.out V ON;
SET CLIENT ATTACH_DBPARTITIONNUM 0;
SET CLIENT CONNECT_DBPARTITIONNUM 0;

```

其中：

- S ON 指定命令出错时应该停止执行该命令
- Z ON SAMPLE_NODE0000.out 指定应该将输出定向到名为 <dbalias>_NODE<dbpartitionnum>.out 的文件
- V ON 指定应该将当前命令打印到标准输出。

在分区数据库环境中运行脚本时，需要指定要运行脚本命令的数据库分区，这一点尤其重要。

指定了 REDIRECT 选项的 RESTORE 命令

第二节启动 RESTORE 命令并使用 REDIRECT 选项。此节可以使用所有 RESTORE 命令选项，不能与 REDIRECT 选项配合使用的命令选项除外。以下是第二节的示例：

```

RESTORE DATABASE SAMPLE
-- USER '<username>'
-- USING '<password>'
FROM '/home/jseifert/backups'
TAKEN AT 20050906194027
-- DBPATH ON '<target-directory>'
INTO SAMPLE

```

```

-- NEWLOGPATH '/home/jseifert/jseifert/NODE0000/SQL00001/SQLLOGDIR/'
-- WITH <num-buff> BUFFERS
-- BUFFER <buffer-size>
-- REPLACE HISTORY FILE
-- REPLACE EXISTING
REDIRECT
-- PARALLELISM <n>
-- WITHOUT ROLLING FORWARD
-- WITHOUT PROMPTING
;

```

表空间定义

此节包含备份映像中包含的或命令行上指定的每个表空间的表空间定义。每个表空间都有一节，每一节都包含注释块，该注释块包含有关表空间名称、类型和大小的信息。该信息是使用表空间快照的格式提供的。可以使用提供的信息来确定所需的表空间大小。查看使用自动存储器创建的表空间的输出时，您将看不到 SET TABLESPACE CONTAINERS 子句。以下是表空间定义节的示例：

```

-- *****
-- ** 表空间名称 = SYSCATSPACE
-- ** 表空间标识 = 0
-- ** 表空间类型 = 系统管理空间
-- ** 表空间内容类型 = 任意数据
-- ** 表空间页大小 (字节) = 4096
-- ** 表空间扩展数据块大小 (页) = 32
-- ** 使用自动存储器 = No
-- ** 总页数 = 5572
-- *****
SET TABLESPACE CONTAINERS FOR 0
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
      USING (
    PATH 'SQLT0000.0'
  );
-- *****
-- ** 表空间名称 = TEMPSPACE1
-- ** 表空间标识 = 1
-- ** 表空间类型 = 系统管理空间
-- ** 表空间内容类型 = 系统临时数据
-- ** 表空间页大小 (字节) = 4096
-- ** 表空间扩展数据块大小 (页) = 32
-- ** 使用自动存储器 = No
-- ** 总页数 = 0
-- *****
SET TABLESPACE CONTAINERS FOR 1
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
      USING (
    PATH 'SQLT0001.0'
  );
-- *****
-- ** 表空间名称 = DMS
-- ** 表空间标识 = 2
-- ** 表空间类型 = 数据库管理空间
-- ** 表空间内容类型 = 任意数据
-- ** 表空间页大小 (字节) = 4096
-- ** 表空间扩展数据块大小 (页) = 32
-- ** 使用自动存储器 = No
-- ** 启用自动调整大小 = No
-- ** 总页数 = 2000
-- ** 可用页数 = 1960
-- ** 高水位标记 (页) = 96
-- *****
SET TABLESPACE CONTAINERS FOR 2
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS

```

```

                                USING (
      FILE  '/tmp/dms1'          1000
    , FILE  '/tmp/dms2'          1000
    );

```

指定了 **CONTINUE** 选项的 **RESTORE** 命令

最后一节发出指定了 **CONTINUE** 选项的 **RESTORE** 命令以完成重定向复原。
以下是最后一节的示例:

```
RESTORE DATABASE SAMPLE CONTINUE;
```

使用自动生成的脚本来执行重定向复原

执行重定向复原操作时，需要指定存储在备份映像中的物理容器的位置，并且需要提供每个将要改变的表空间的全部容器。使用以下过程来根据现有备份映像生成重定向复原脚本，修改生成的脚本，然后运行该脚本以执行重定向复原。

仅当先前已使用 **DB2 BACKUP** 实用程序备份了数据库时，才能执行重定向复原。

- 如果数据库存在，您必须能够连接至该数据库才能生成脚本。因此，如果数据库需要升级或崩溃恢复，那么必须在尝试生成重定向复原脚本之前进行此操作。
- 如果正在分区数据库环境中工作，并且目标数据库不存在，那么不能运行该命令来在所有数据库分区上同时生成重定向复原脚本。相反，一次只能在一个数据库分区上运行用于生成重定向复原脚本的命令，从目录分区开始。

或者，可以先创建一个与目标数据库具有相同名称的哑数据库。在创建了哑数据库之后，可以在所有数据库分区上同时生成重定向复原脚本。

- 在发出 **RESTORE** 命令以生成脚本时，即使指定了 **REPLACE EXISTING** 选项，脚本中出现的 **REPLACE EXISTING** 选项也会被注释掉。
- 为了提高安全性，密码不会出现在生成的脚本中。您需要手动填写密码。
- 不能使用控制中心中的“复原”向导来生成重定向复原脚本。

要使用脚本来执行重定向复原:

1. 使用 **RESTORE** 实用程序来生成重定向复原脚本。可以通过命令行处理器 (CLP) 或 **db2Restore** 应用程序编程接口 (API) 来调用 **RESTORE** 实用程序。以下是指定了 **REDIRECT** 选项和 **GENERATE SCRIPT** 选项的 **RESTORE DATABASE** 命令示例:

```
db2 restore db test from /home/jseifert/backups taken at 20050304090733
      redirect generate script test_node0000.clp
```

此命令将在客户机上创建名为 **test_node0000.clp** 的重定向复原脚本。

2. 在文本编辑器中打开重定向复原脚本以进行所需的修改。可以修改:
 - 复原选项
 - 自动存储路径
 - 容器布局和路径
3. 运行修改后的重定向复原脚本。例如:

```
db2 -tvf test_node0000.clp
```

数据库重建

重建数据库是使用一组复原操作来复原数据库或它的一部分表空间的过程。数据库重建提供的功能使 DB2 更健壮且用途更广泛，并为您提供一个更加完整的恢复解决方案。

允许从表空间备份映像重建数据库意味着您不必再进行那么多完整数据库备份。随着数据库大小的不断增长，执行完整数据库备份的机会越来越有限。由于还可以采用表空间备份，您不必再那么频繁地进行完整数据库备份。相反，您可以更多地进行表空间备份，并计划在出现灾难时使用它们和日志文件。

在恢复情况下，如果需要使一部分表空间比其他表空间更快地处于联机状态，那么可以使用重建来完成此任务。仅使一部分表空间处于联机状态的功能在测试和生产环境中特别有用。

重建数据库涉及一系列潜在的复原操作。重建操作可以使用数据库映像和/或表空间映像。可以使用完整备份和/或增量备份。初始复原操作复原目标映像，该映像定义可以复原的数据库的结构（例如，表空间集和数据库配置）。对于可恢复数据库，重建允许您构建可连接的数据库并且使该数据库包含需要处于联机状态的一部分表空间，而让稍后才恢复的表空间处于脱机状态。

用来重建数据库的方法取决于数据库是可恢复的还是不可恢复的。

- 如果数据库是可恢复的，使用下列其中一种方法：
 - 使用完整数据库备份映像、增量数据库备份映像或表空间备份映像作为目标映像，通过仅使用 `REBUILD` 选项从目标映像复原 `SYSCATSPACE` 及任何其他表空间来重建数据库。然后，可以将数据库前滚至某个时间点。
 - 使用完整数据库备份映像、增量数据库备份映像或表空间备份映像作为目标映像，通过指定使用 `REBUILD` 选项复原目标映像时在数据库中定义的表空间集来重建数据库。`SYSCATSPACE` 必须是此表空间集的一部分。此操作将复原指定的并且是在目标映像中定义的那些表空间，然后使用恢复历史记录文件来自动查找并复原不在目标映像中的其余表空间所需的任何其他备份映像。复原完成之后，立即将数据库前滚至某个时间点。
- 如果数据库是不可恢复的：
 - 使用完整或增量数据库备份映像作为目标映像，通过使用适当的 `REBUILD` 语法从目标映像复原 `SYSCATSPACE` 和任何其他表空间来重建数据库。复原完成后，可以连接至该数据库。

指定目标映像

要对数据库执行重建，一开始请发出 `RESTORE` 命令并指定用作复原操作目标的最新备份映像。此映像称为重建操作的目标映像，因为它定义要复原的数据库的结构，包括可以复原的表空间、数据库配置和日志序列。重建目标映像使用 `RESTORE DATABASE` 命令中的 `TAKEN AT` 参数指定的。目标映像可以是任何类型的备份（完整备份、表空间备份、增量备份、联机备份或脱机备份）。创建目标映像时在数据库中定义的表空间将是可用于重建数据库的表空间。

必须使用下列其中一种方法指定想要复原的表空间：

- 指定想要复原数据库中定义的所有表空间并提供异常列表（如果想要排除一些表空间）

- 指定想要复原目标映像中具有用户数据的所有表空间并提供异常列表（如果想要排除一些表空间）
- 指定想要复原的在数据库中定义的表空间列表

一旦知道了想要重建数据库包含的表空间，就发出带有适当 `REBUILD` 选项的 `RESTORE` 命令并指定要使用的目标映像。

重建阶段

在发出带有适当 `REBUILD` 选项的 `RESTORE` 命令并成功地复原了目标映像之后，就认为数据库处于重建阶段。在复原目标映像之后，进行的任何其他表空间复原将会使数据复原到现有表空间中，如已重建的数据库中定义的那样。然后，这些表空间将在重建操作完成时与数据库一起前滚。

如果发出带有适当 `REBUILD` 选项的 `RESTORE` 命令，但数据库不存在，那么将根据目标映像中的属性创建一个新数据库。如果数据库存在，您将接收到一条警告消息，通知您重建阶段即将开始。将会询问您是否想要继续重建操作。

重建操作从目标映像复原所有初始元数据。这包括属于数据库但不属于表空间数据或日志文件的所有数据。初始元数据的示例有：

- 表空间定义
- 历史记录文件，它是记录管理操作的数据库文件

重建操作还复原数据库配置。目标映像设置日志链，它确定在重建阶段可以用于其余复原的映像。只能使用同一日志链上的映像。

如果磁盘上已经存在数据库并且想要从目标映像生成历史记录文件，那么应该指定 `REPLACE HISTORY FILE` 选项。自动逻辑使用此时磁盘上的历史记录文件来查找重建数据库所需的其余映像。

复原了目标映像之后：

- 如果数据库是可恢复的，那么数据库将处于前滚暂挂状态，并且复原的所有表空间也处于前滚暂挂状态。数据库中定义的但未复原的所有表空间都处于复原暂挂状态。
- 如果数据库是不可恢复的，那么复原的数据库和数据库表空间将进入正常状态。未复原的所有表空间将处于删除暂挂状态，因为不能再复原它们。对于这种类型的数据库来说，重建阶段完成了。

对于可恢复的数据库，当发出第一个 `ROLLFORWARD DATABASE` 命令并且 `Rollforward` 实用程序开始处理日志记录时，重建阶段结束。如果前滚操作在开始处理日志记录后失败，并且接着发出了复原操作，那么不认为该复原是重建阶段的一部分。应该将这种复原看作是正常表空间复原，它们不是重建阶段的一部分。

自动处理

在复原目标映像之后，`RESTORE` 实用程序确定是否有其余表空间需要复原。如果有，那么使用对带有 `REBUILD` 选项的 `RESTORE DATABASE` 命令使用的那个连接进行复原。该实用程序使用磁盘上的历史记录文件来查找在目标映像之前的最新备份映像，该备份映像包含需要复原的其余每个表空间。`RESTORE` 实用程序使用历史记录文

件中存储的备份映像位置数据来自动复原每个映像。后续这些复原表空间级复原，只能以脱机方式执行。如果选择的映像不属于当前日志链，那么会返回错误。从该映像复原的每个表空间将处于前滚暂挂状态。

RESTORE 实用程序尝试自动复原所有必需表空间。在某些情况下，可能由于历史记录文件具有的问题而无法复原一些表空间，或者在复原其中一个必需映像时发生错误。在此情况下，可以手动完成重建，或者更正问题然后重新发出重建。

如果自动重建无法成功完成，那么 **RESTORE** 实用程序会将它收集的用于其余复原步骤的所有信息都写入诊断日志（db2diag 日志文件）中。可以使用这些信息手动完成重建。

如果正在重建某个数据库，那么仅获取属于重建过程的表空间的容器。

如果需要通过重定向复原来重新定义任何容器，那么将需要为其余复原和后续前滚操作设置新容器的新路径和大小。

如果从其余这些映像中的一个映像复原的表空间的数据不符合新容器的定义，那么该表空间将处于复原暂挂状态，并且在复原结束时会返回一则警告消息。可以在诊断日志中找到更多有关该问题的信息。

完成重建阶段

在已经复原了所有要复原的表空间后，您可以根据数据库的配置作出两种选择。如果数据库是不可恢复的，那么数据库将是可连接的，并且复原的所有表空间将处于联机状态。不能再复原处于删除暂挂状态的任何表空间，如果将来要在数据库上执行备份，那么应该删除这些表空间。

如果数据库是可恢复的，那么可以发出前滚命令来使已复原的表空间处于联机状态。如果 SYSCATSPACE 尚未复原，那么前滚将失败，并且必须在前滚操作开始之前复原此表空间。这表示必须在重建阶段复原 SYSCATSPACE。

注：在分区数据库环境中，SYSCATSPACE 不存在于非目录分区中，因此不能在那里将它重建。但是，在目录分区上，SYSCATSPACE 必须是重建的表空间，否则前滚操作将不会成功。

前滚数据库使数据库脱离前滚暂挂状态，并使处于前滚暂挂状态的任何表空间向前滚动。不会对任何处于复原暂挂状态的表空间运行 Rollforward 实用程序。

前滚操作的停止时间必须比在重建阶段复原最新备份映像的结束时间要晚。如果给出任何其他时间，那么会发生错误。如果前滚操作无法到达复原的最早映像的备份时间，那么 Rollforward 实用程序将无法使数据库到达一个一致的时间，前滚将失败。

在最早备份映像和最新备份映像的时间范围内，必须使所有日志文件可供 Rollforward 实用程序使用。所需的日志是从最早备份映像到目标备份映像都遵循日志链的那些日志，如目标映像中的截断数组所定义的那样，否则前滚操作将失败。如果在重建阶段复原了比目标映像更新的任何备份映像，那么需要从目标映像到复原的最新备份映像的附加日志。如果日志不可用，那么前滚操作将使日志无法访问的那些表空间处于复原暂挂状态。可以发出 LIST HISTORY 命令来显示复原重建条目以及前滚所需的日志范围。

正确的日志文件必须可用。如果依靠 Rollforward 实用程序来检索日志，那么必须确保配置了“DB2 日志管理器”以指示检索日志文件的位置。如果日志路径或归档路径更改了，那么需要使用 ROLLFORWARD DATABASE 命令的 OVERFLOW LOG PATH 选项。

当成功完成前滚命令时，使用 ROLLFORWARD DATABASE 命令的 AND STOP 选项来使数据库可用。此时，数据库不再处于前滚暂挂状态。如果前滚操作开始了，但它未成功完成就出现错误，那么前滚操作将在故障点处停止并返回错误。数据库仍然处于前滚暂挂状态。必须执行步骤以更正该问题（例如，修正日志文件），然后发出另一个前滚操作来继续处理。

如果无法修正该错误，那么可以通过发出 ROLLFORWARD STOP 命令使数据库前进到故障点。一旦使用了 STOP 选项，日志中超出该故障点的所有日志数据将不再可用。数据库处于该故障点处，并且已恢复的所有表空间都处于联机状态。尚未恢复的表空间处于复原暂挂状态。数据库处于正常状态。

您必须决定恢复处于复原暂挂状态的其余表空间的最好方法。该方法可以是进行新的复原并前滚表空间，也可以是重新发出整个重建操作。这将取决于遇到的问题类型。如果 SYSCATSPACE 是其中一个处于复原暂挂状态的表空间，那么数据库将不可连接。

重建和表空间容器

重建期间，只有参与重建过程的那些表空间才获取它们的容器。将在从映像复原表空间用户数据时获取属于每个表空间的容器。

复原目标映像时，数据库在备份时知道的每个表空间只复原它们的定义。这表示通过重建创建的数据库将认识在备份时知道的那些表空间。对于同样应该从目标映像复原其用户数据的那些表空间，也会在此时获取它们的容器。

通过中间表空间复原来复原的任何其余表空间将在复原包含表空间数据的映像时获取它们的容器。

使用重定向复原重建

在重定向复原的情况下，必须在复原目标映像期间定义所有表空间容器。如果指定 REDIRECT 选项，您将重新获得重新定义表空间容器的控制权。如果使用 SET TABLESPACE CONTAINERS 命令重新定义了表空间容器，那么将在那时获取那些新容器。将在正常时间获取尚未重新定义的任何表空间容器，即，在从映像复原表空间用户数据时。

如果复原的表空间的数据不符合新容器的定义，那么该表空间将处于复原暂挂状态，并且在复原结束时会返回一个警告（SQL2563W）。在 DB2 诊断日志中将有一条消息详细说明该问题。

重建和临时表空间

DB2 备份映像通常由下列组件构成：

- 初始数据库元数据，例如，表空间定义、数据库配置文件和历史记录文件。
- 对 BACKUP 实用程序指定的非临时表空间的数据
- 最终数据库元数据，如日志文件标题

- 日志文件（如果指定了 INCLUDE LOGS 选项）

在每个备份映像中，无论是数据库备份还是表空间备份，或者完整备份还是增量备份（差异备份），始终可以找到这些核心组件。

数据库备份映像将包含上述所有组件，以及备份时在数据库中定义的每个表空间的数据。

表空间备份映像将始终包含上面列示的数据库元数据，但它仅包含对 BACKUP 实用程序指定的那些表空间的数据。

对待临时表空间的方式与对待非临时表空间的方式不同。从不备份临时表空间数据，但这些数据的存在对于数据库的框架来说很重要。虽然从不备份临时表空间数据，但将临时表空间视为数据库的一部分，因此，在使用备份映像存储的元数据中对其进行特别标记。这使得它们看起来与在备份映像中一样。此外，表空间定义还保存关于任何临时表空间是否存在的信息。

虽然备份映像不曾包含临时表空间的数据，但在数据库重建操作过程中，在复原目标映像时（无论映像是什么类型），同样会复原临时表空间，这样是只为了获取和分配它们的容器。获取和分配容器是作为重建过程的一部分自动完成的。因此，重建数据库时，不能排除临时表空间。

为数据库重建选择目标映像

重建目标映像应该是想要用作复原操作的起始点的最新备份映像。此映像称为重建操作的目标映像，因为它定义要复原的数据库的结构，包括可以复原的表空间、数据库配置和日志序列。它可以是任何类型的备份（完整备份、表空间备份、增量备份、联机备份或脱机备份）。

目标映像设置日志序列（或日志链），它确定在重建阶段可以用于其余复原的映像。只能使用同一日志链上的映像。

下列示例说明如何选择应用作重建操作的目标映像的映像。

假设有一个称为 SAMPLE 的数据库，该数据库中具有下列表空间：

- SYSCATSPACE（系统目录）
- USERSP1（用户数据表空间）
- USERSP2（用户数据表空间）
- USERSP3（用户数据表空间）

第 227 页的图 21 按时间顺序显示已经进行的数据库级别备份和表空间级别备份：

1. 完整数据库备份 DB1
2. 完整表空间备份 TS1
3. 完整表空间备份 TS2
4. 完整表空间备份 TS3
5. 数据库复原并前滚至 TS1 与 TS2 之间的一个时间点
6. 完整表空间备份 TS4
7. 完整表空间备份 TS5

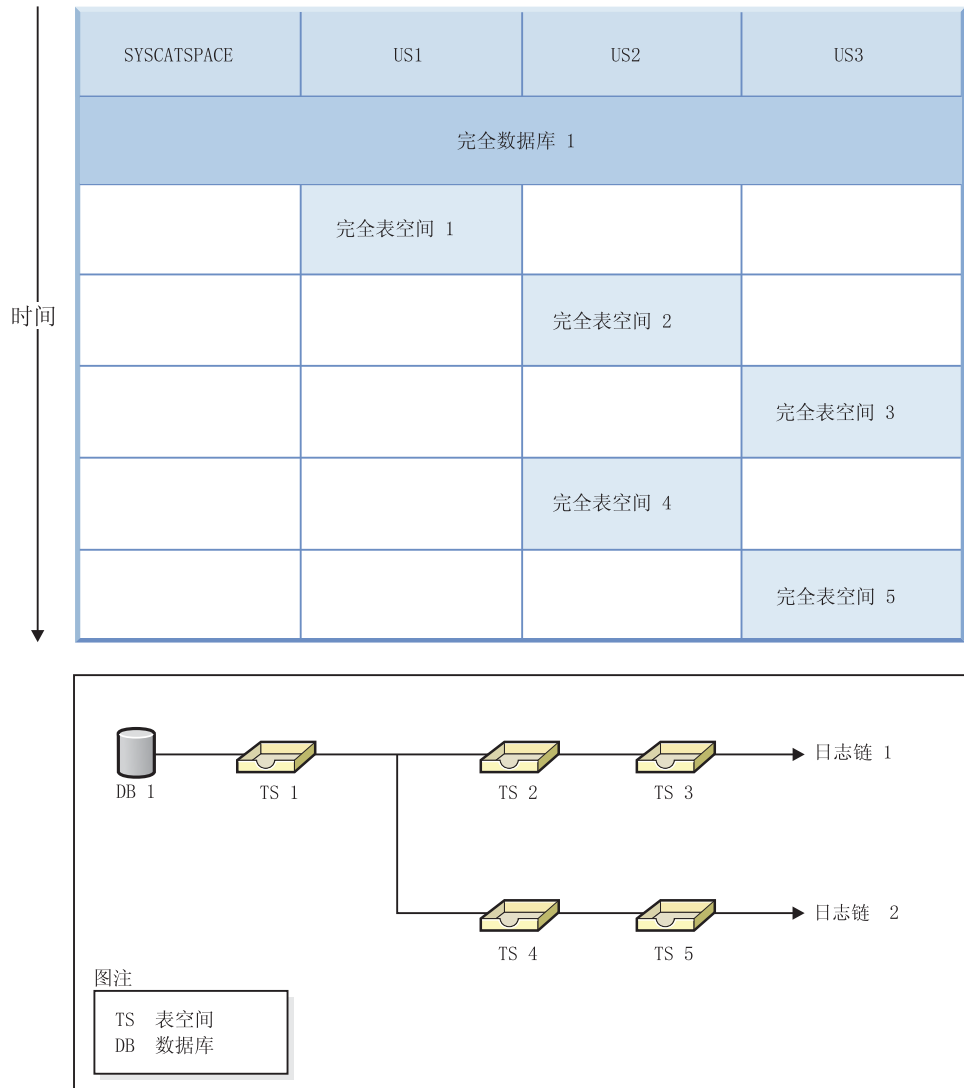


图 21. 数据库 SAMPLE 的数据库级别备份和表空间级别备份

示例 1

以下示例演示了将数据库 SAMPLE 重建至当前时间点需要发出的 CLP 命令。首先需要选择想要重建的表空间。由于目标是将数据库重建至当前时间点，因此需要选择最新备份映像作为目标映像。最新备份映像映像 TS5，它在日志链 2 上：

```
db2 restore db sample rebuild with all tablespaces in database taken at
      TS5 without prompting
db2 rollforward db sample to end of logs
db2 rollforward db sample stop
```

这将自动复原备份映像 TS5、TS4、TS1 和 DB1，然后将数据库前滚至日志链 2 的末尾。

注：所有属于日志链 2 的日志必须可以访问，这样前滚操作才能完成。

示例 2

第二个示例演示了将数据库 SAMPLE 重建至日志链 1 的末尾需要发出的 CLP 命令。选择的目標映像应该是日志链 1 上的最新备份映像，它是 TS3：

```
db2 restore db sample rebuild with all tablespaces in database
      taken at TS3 without prompting
db2 rollforward db sample to end of logs
db2 rollforward db sample stop
```

这将自动复原备份映像 TS3、TS2、TS1 和 DB1，然后将数据库前滚至日志链 1 的末尾。

注：所有属于日志链 1 的日志必须可以访问，这样前滚操作才能完成。

选择错误的目标映像进行重建

假设有一个称为 SAMPLE2 的数据库，该数据库中具有下列表空间：

- SYSCATSPACE（系统目录）
- USERSP1（用户数据表空间）
- USERSP2（用户数据表空间）

图 22 显示了 SAMPLE2 的备份日志链，它由以下备份组成：

1. BK1 是完整数据库备份，它包括所有表空间
2. BK2 是 USERSP1 的完整表空间备份
3. BK3 是 USERSP2 的完整表空间备份

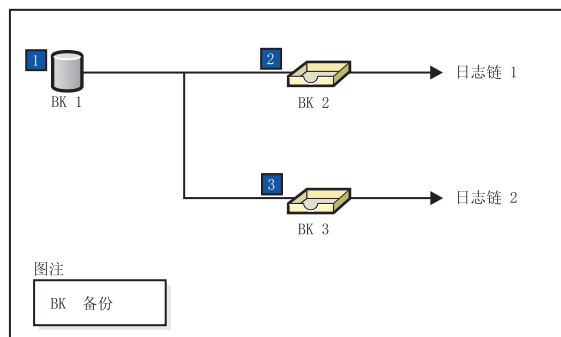


图 22. 数据库 SAMPLE2 的备份日志链

以下示例演示了使用表空间 SYSCATSPACE 和 USERSP2 从 BK3 重建数据库需要发出的 CLP 命令：

```
db2 restore db sample2 rebuild with tablespace (SYSCATSPACE,
      USERSP2) taken at BK3 without prompting
```

现在假设在此复原完成之后，您决定还要复原 USERSP1，因此发出以下命令：

```
db2 restore db sample2 tablespace (USERSP1) taken at BK2
```

此复原失败并提供一条消息，说明 BK2 来自错误的日志链（SQL2154N）。如第 228 页的图 22 中所示，唯一可用来复原 USERSP1 的映像是 BK1。因此，需要输入以下命令：

```
db2 restore db sample2 tablespace (USERSP1) taken at BK1
```

复原成功，因此数据库可以相应地前滚。

重建所选表空间

重建数据库允许您重建包含原始数据库中的一部分表空间的数据库。在下列情况下，仅重建数据库中的一部分表空间非常有用：

- 在只想对一部分表空间进行工作的测试和开发环境中。
- 在恢复情况下，如果需要使较关键的表空间比其他表空间更快地处于联机状态，那么可以先复原一部分表空间，稍后再复原其他表空间。

要重建包含原始数据库中的一部分表空间的数据库，请考虑以下示例。

在此示例中，有一个称为 SAMPLE 的数据库，它具有下列表空间：

- SYSCATSPACE（系统目录）
- USERSP1（用户数据表空间）
- USERSP2（用户数据表空间）
- USERSP3（用户数据表空间）

进行了下列备份：

- BK1 是 SYSCATSPACE 和 USERSP1 的备份
- BK2 是 USERSP2 和 USERSP3 的备份
- BK3 是 USERSP3 的备份

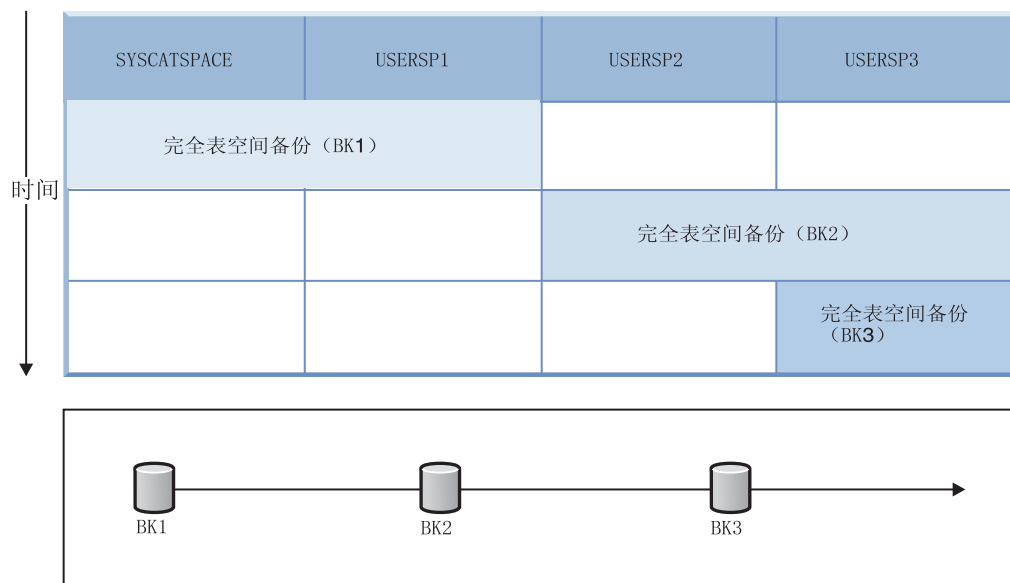


图 23. 为 SAMPLE 数据库提供的备份映像

下列过程演示了使用 CLP 发出的 RESTORE DATABASE 和 ROLLFORWARD DATABASE 命令仅将 SYSCATSPACE 和 USERSP1 重建至日志末尾:

```
db2 restore db mydb rebuild with all tablespaces in image
      taken at BK1 without prompting
db2 rollforward db mydb to end of logs
db2 rollforward db mydb stop
```

此时, 该数据库是可连接的, 但只有 SYSCATSPACE 和 USERSP1 处于 NORMAL 状态。USERSP2 和 USERSP3 处于复原暂挂状态。以后仍可以复原 USERSP2 和 USERSP3。

重建和增量备份映像

可以使用增量映像重建数据库。缺省情况下, RESTORE 实用程序将尝试对所有增量映像使用自动增量复原。这表示如果不使用 RESTORE DATABASE 命令的 INCREMENTAL 选项, 但目标映像又是增量备份映像, 那么 RESTORE 实用程序将使用自动增量复原发出重建操作。如果目标映像不是增量映像, 但另一个需要的映像是增量映像, 那么 RESTORE 实用程序将确保使用自动增量复原来复原这些增量映像。无论是否指定带有 AUTOMATIC 选项的 INCREMENTAL 选项, RESTORE 实用程序的工作方式都相同。

如果指定了 INCREMENTAL 选项, 但未指定 AUTOMATIC 选项, 那么需要手动执行整个重建过程。RESTORE 实用程序将只从目标映像复原初始元数据, 就像是正常手动增量复原一样。然后, 需要使用必需的增量复原链来完成目标映像的复原。接下来, 需要复原其余映像以重建数据库。

建议您使用自动增量复原来重建数据库。仅在复原失败的情况下才应尝试使用手动方法重建数据库。

重建分区数据库

要重建分区数据库，分别重建每个数据库分区。对于每个数据库分区，从目录分区开始，首先复原需要的所有表空间。未复原的所有表空间都处于复原暂挂状态。复原了所有数据库分区之后，在目录分区上发出 `ROLLFORWARD DATABASE` 命令以前滚所有数据库分区。

注：如果在将来的某一天您需要复原最初未包括在重建阶段中的任何表空间，那么需要确保在后来前滚表空间时 `Rollforward` 实用程序使数据库分区上的所有数据保持同步。如果在原始复原和前滚操作中丢失了某个表空间，那么直到尝试访问数据时才会检测到这种情况，将出现数据访问错误。需要复原和前滚丢失的表空间，以使它恢复与其余分区的同步。

要使用表空间级备份映像来重建分区数据库，请考虑以下示例。

在此示例中，有一个称为 `SAMPLE` 的可恢复数据库，它具有三个数据库分区：

- 数据库分区 1 包含表空间 `SYSCATSPACE`、`USERSP1` 和 `USERSP2`，它是目录分区
- 数据库分区 2 包含表空间 `USERSP1` 和 `USERSP3`
- 数据库分区 3 包含表空间 `USERSP1`、`USERSP2` 和 `USERSP3`

进行了下列备份，其中 `BKxy` 表示分区 `y` 上的备份编号 `x`：

- `BK11` 是 `SYSCATSPACE`、`USERSP1` 和 `USERSP2` 的备份
- `BK12` 是 `USERSP2` 和 `USERSP3` 的备份
- `BK13` 是 `USERSP1`、`USERSP2` 和 `USERSP3` 的备份
- `BK21` 是 `USERSP1` 的备份
- `BK22` 是 `USERSP1` 的备份
- `BK23` 是 `USERSP1` 的备份
- `BK31` 是 `USERSP2` 的备份
- `BK33` 是 `USERSP2` 的备份
- `BK42` 是 `USERSP3` 的备份
- `BK43` 是 `USERSP3` 的备份

下列过程演示了使用 `CLP` 发出的 `RESTORE DATABASE` 和 `ROLLFORWARD DATABASE` 命令将整个数据库重建至日志末尾。

1. 在数据库分区 1 上，发出带有 `REBUILD` 选项的 `RESTORE DATABASE` 命令：

```
db2 restore db sample rebuild with all tablespaces in database
      taken at BK31 without prompting
```

2. 在数据库分区 2 上，发出带有 `REBUILD` 选项的 `RESTORE DATABASE` 命令：

```
db2 restore db sample rebuild with tablespaces in database
      taken at BK42 without prompting
```

3. 在数据库分区 3 上，发出带有 `REBUILD` 选项的 `RESTORE DATABASE` 命令：

```
db2 restore db sample rebuild with all tablespaces in database
      taken at BK43 without prompting
```

4. 在目录分区上，发出带有 `TO END OF LOGS` 选项的 `ROLLFORWARD DATABASE` 命令：

```
db2 rollforward db sample to end of logs
```

5. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db sample stop
```

此时, 该数据库在所有数据库分区上都是可连接的, 并且所有表空间都处于 NORMAL 状态。

数据库重建的限制

以下列表对数据库重建的限制进行了概述:

- 重建的其中一个表空间必须是目录分区上的 SYSCATSPACE。
- 不能使用控制中心 GUI 工具来执行重建操作。必须使用命令行处理器 (CLP) 来发出命令, 或者使用相应的应用程序编程接口 (API)。
- 除非目标映像是脱机数据库备份, 否则不能对版本 9.1 之前的目标映像使用 REBUILD 选项。如果目标映像是脱机数据库备份, 那么只能将此映像中的表空间用于重建。重建操作成功完成之后, 需要迁移数据库。尝试使用版本 9.1 之前的任何其他类型的目标映像进行重建将导致错误。
- 除非目标映像是完整数据库备份, 否则当目标映像所在的操作系统不同于正在进行复原的操作系统时, 不能对该目标映像发出 REBUILD 选项。如果目标映像是完整数据库备份, 那么只能将此映像中的表空间用于重建。如果目标映像所在的操作系统不同于正在进行复原的操作系统, 那么使用任何其他类型的这种映像进行重建都将导致错误。

优化复原性能

执行复原操作时, DB2 将自动为缓冲区个数、缓冲区大小和并行性设置选择最佳值。此值根据可用实用程序堆内存的数量、可用处理器数和数据库配置而定。因此, 根据系统上可用的存储量, 应考虑通过增大 UTIL_HEAP_SZ 配置参数来分配更多内存。目的是将完成复原操作所用的时间降至最少。除非显式地输入以下 RESTORE DATABASE 命令参数的值, 否则 DB2 将为它们选择一个值:

- WITH num-buffers BUFFERS
- PARALLELISM n
- BUFFER buffer-size

对于复原操作, 始终使用备份操作所使用的缓冲区大小的倍数。在发出 RESTORE DATABASE 命令时可以指定缓冲区大小, 但需要确保其大小是备份缓冲区大小的倍数。

还可以选择执行以下任何操作来缩短完成一次复原操作所需的时间:

- 增加复原缓冲区大小。

复原缓冲区大小必须是在备份操作期间指定的备份缓冲区大小的正整数倍数。如果指定了不正确的缓冲区, 那么分配的缓冲区将是最小的可接受大小。

- 增加缓冲区的数量。

指定的值必须是为备份缓冲区指定的页数的倍数。最小页数为 8。

- 增加 PARALLELISM 参数的值。

这将增加将用来在复原操作期间写至数据库的缓冲区操纵程序 (BM) 的数目。

- 增加实用程序堆大小

这将增大可由其他实用程序同时使用的内存。

使用 restore 所需的特权、权限和授权

特权使用户能够创建或访问数据库资源。权限级别提供了对特权、较高级别数据库管理器维护和实用程序操作进行分组的方法。这两者一起用于控制对数据库管理器及其数据库对象的访问。用户只能访问那些他们具有相应授权（即必需的特权或权限）的对象。

要从完整的数据库备份复原到现有数据库，必须具有 SYSADM、SYSCTRL 或 SYSMANT 权限。要复原到新数据库，必须具有 SYSADM 或 SYSCTRL 权限。

复原示例

重定向复原会话 - CLP 示例

示例 1

以下是对别名为 MYDB 的数据库的典型非增量重定向复原方案：

1. 发出 RESTORE DATABASE 命令，使用 REDIRECT 选项。

```
db2 restore db mydb replace existing redirect
```

2. 对想要重新定义其容器的每个表空间发出 SET TABLESPACE CONTAINERS 命令。
例如，在 Windows 环境中：

```
db2 set tablespace containers for 5 using  
      (file 'f:\ts3con1'20000, file 'f:\ts3con2'20000)
```

要验证复原数据库的容器是否是在此步骤中指定的那些容器，对正重新定义其容器位置的每个表空间发出 LIST TABLESPACE CONTAINERS 命令。

3. 在成功完成了步骤 1 和 2 之后，发出：

```
db2 restore db mydb continue
```

这是“重定向复原”操作的最后一步。

4. 如果步骤 3 失败，或者如果中止了复原操作，那么可从步骤 1 开始重新启动重定向的复原。

注：

1. 成功完成步骤 1 之后且在完成步骤 3 之前，通过发出以下命令来中止复原操作：

```
db2 restore db mydb abort
```

2. 如果步骤 3 失败，或者如果中止了复原操作，那么可从步骤 1 开始重新启动重定向的复原。

示例 2

以下是对别名为 MYDB 且具有下列备份映像的数据库的典型手动增量重定向复原方案：

```
backup db mydb  
备份成功。此备份的时间戳记为: <ts1>
```

```
backup db mydb incremental  
备份成功。此备份的时间戳记为: <ts2>
```

1. 发出指定了 INCREMENTAL 和 REDIRECT 选项的 RESTORE DATABASE 命令。
2. 对必须重新定义容器的每个表空间发出 SET TABLESPACE CONTAINERS 命令。
例如, 在 Windows 环境中:

```
db2 restore db mydb incremental taken at <ts2> replace existing redirect
      (file 'f:\ts3con1'20000, file 'f:\ts3con2'20000)
```

要验证复原数据库的容器是否是在此步骤中指定的那些容器, 可发出 LIST TABLESPACE CONTAINERS 命令。

3. 在成功完成了步骤 1 和 2 之后, 发出:

```
db2 restore db mydb continue
```

4. 现在可以发出余下的增量复原命令, 如下所示:

```
db2 restore db mydb incremental taken at <ts1>
db2 restore db mydb incremental taken at <ts2>
```

这是“重定向复原”操作的最后一步。

注:

1. 成功完成步骤 1 之后且在完成步骤 3 之前, 通过发出以下命令来中止复原操作:

```
db2 restore db mydb abort
```
2. 成功完成步骤 3 之后且在发出步骤 4 中所有必需的命令之前, 可以通过发出以下命令来中止复原操作:

```
db2 restore db mydb incremental abort
```
3. 如果步骤 3 失败, 或者如果中止了复原操作, 那么可从步骤 1 开始重新启动重定向的复原。
4. 如果步骤 4 中的复原命令失败, 那么可以重新发出失败的命令以继续复原过程。

示例 3

以下是对同一个数据库的典型自动增量重定向复原方案:

1. 发出指定了 INCREMENTAL AUTOMATIC 和 REDIRECT 选项的 RESTORE DATABASE 命令。

```
db2 restore db mydb incremental automatic taken at <ts2>
      replace existing redirect
```

2. 对必须重新定义容器的每个表空间发出 SET TABLESPACE CONTAINERS 命令。
例如, 在 Windows 环境中:

```
db2 set tablespace containers for 5 using
      (file 'f:\ts3con1'20000, file 'f:\ts3con2'20000)
```

要验证复原数据库的容器是否是在此步骤中指定的那些容器, 可发出 LIST TABLESPACE CONTAINERS 命令。

3. 在成功完成了步骤 1 和 2 之后, 发出:

```
db2 restore db mydb continue
```

这是“重定向复原”操作的最后一步。

注:

1. 成功完成步骤 1 之后且在完成步骤 3 之前, 通过发出以下命令来中止复原操作:

```
db2 restore db mydb abort
```

2. 如果步骤 3 失败，或者如果复原操作已中止，那么可以在发出以下命令之后从步骤 1 开始重新启动重定向复原：

```
db2 restore db mydb incremental abort
```

重建会话 - CLP 示例

情况 1

在下列示例中，有一个称为 MYDB 的可恢复数据库，该数据库中存在下列表空间：

- SYSCATSPACE（系统目录）
- USERSP1（用户数据表空间）
- USERSP2（用户数据表空间）
- USERSP3（用户数据表空间）

进行了下列备份：

- BK1 是 SYSCATSPACE 和 USERSP1 的备份
- BK2 是 USERSP2 和 USERSP3 的备份
- BK3 是 USERSP3 的备份

示例 1

下列命令将整个数据库重建至最新时间点：

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令：

```
db2 restore db mydb rebuild with all tablespaces in database  
taken at BK3 without prompting
```

2. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令（假定所有日志都已保存并且可以访问）：

```
db2 rollforward db mydb to end of logs
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令：

```
db2 rollforward db mydb stop
```

此时，该数据库是可连接的并且所有表空间都处于 NORMAL 状态。

示例 2

下列命令仅将 SYSCATSPACE 和 USERSP2 重建至某个时间点（BK3 的末尾比该时间点要早，而该时间点又比日志末尾要迟）：

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令并指定想要包括的表空间。

```
db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP2)  
taken at BK2 without prompting
```

2. 发出带有 TO PIT 选项的 ROLLFORWARD DATABASE 命令（假定所有日志都已保存并且可以访问）：

```
db2 rollforward db mydb to PIT
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令：

```
db2 rollforward db mydb stop
```

此时，该数据库是可连接的，但只有 SYSCATSPACE 和 USERSP2 处于 NORMAL 状态。USERSP1 和 USERSP3 处于 RESTORE_PENDING 状态。

要在以后使用正常表空间复原（不带 REBUILD 选项）复原 USERSP1 和 USERSP3:

1. 发出不带 REBUILD 选项的 RESTORE DATABASE 命令并指定想要复原的表空间。首先复原 USERSP1:

```
db2 restore db mydb tablespace (USERSP1) taken at BK1 without prompting
```

2. 然后复原 USERSP3:

```
db2 restore db mydb tablespace taken at BK3 without prompting
```

3. 发出带有 END OF LOGS 选项的 ROLLFORWARD DATABASE 命令并指定要复原的表空间（假定所有日志都已保存并且可以访问）:

```
db2 rollforward db mydb to end of logs tablespace (USERSP1, USERSP3)
```

前滚将所有日志最多重放至 PIT，然后对这两个表空间停止，因为自第一次前滚以来尚未对它们执行任何操作。

4. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db mydb stop
```

示例 3

下列命令只将 SYSCATSPACE 和 USERSP1 重建至日志末尾:

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb rebuild with all tablespaces in image  
taken at BK1 without prompting
```

2. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令（假定所有日志都已保存并且可以访问）:

```
db2 rollforward db mydb to end of logs
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db mydb stop
```

此时，该数据库是可连接的，但只有 SYSCATSPACE 和 USERSP1 处于 NORMAL 状态。USERSP2 和 USERSP3 处于 RESTORE_PENDING 状态。

示例 4

在以下示例中，备份 BK1 和 BK2 不再位于历史记录文件中所述的那个位置，但发出重建时并不知道这一点。

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令，并指定想要将整个数据库重建至最新时间点:

```
db2 restore db mydb rebuild with all tablespaces in database  
taken at BK3 without prompting
```

此时成功地复原了目标映像，但从 RESTORE 实用程序返回一个错误，说明无法找到所需的映像。

2. 现在必须手动完成重建。因为数据库处于重建阶段，所以可以按如下所示完成重建:

- a. 发出 RESTORE DATABASE 命令并指定 BK1 备份映像的位置:

```
db2 restore db mydb tablespace taken at BK1 from <location>
without prompting
```

- b. 发出 RESTORE DATABASE 命令并指定 BK2 备份映像的位置:

```
db2 restore db mydb tablespace (USERSP2) taken at BK2 from
<location> without prompting
```

- c. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令 (假定所有日志都已保存并且可以访问):

```
db2 rollforward db mydb to end of logs
```

- d. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db mydb stop
```

此时, 该数据库是可连接的并且所有表空间都处于 NORMAL 状态。

示例 5

在本例中, USERSP3 表空间包含生成特定报告需要的独立数据, 但是您并不希望报告的生成妨碍原始数据库。为了访问数据而不影响原始数据库, 可以使用 REBUILD 生成只有这个表空间和 SYSCATSPACE 的新数据库。为了使数据库在进行复原和前滚操作后仍然可以连接, 所以同时也需要 SYSCATSPACE。

使用 SYSCATSPACE 和 USERSP3 中最新的数据构建新的数据库:

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令, 并指定将 SYSCATSPACE 和 USERSP3 表空间复原到一个新的数据库 NEWDB 中:

```
db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP3)
taken at BK3 into newdb without prompting
```

2. 在 NEWDB 上发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令 (这项命令假定所有的日志已被保存并可访问):

```
db2 rollforward db newdb to end of logs
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db newdb stop
```

此时, 新数据库是可连接的, 并且只有 SYSCATSPACE 和 USERSP3 处于 NORMAL 状态。USERSP1 和 USERSP2 处于 RESTORE_PENDING 状态。

注: 如果现有数据库和新数据库之间的容器路径有问题 (例如, 如果因为文件系统不存在而需更改原始数据库的容器, 或容器正被原始数据库使用), 那么需要执行重定向恢复。以上示例假设对表空间使用了缺省的自动存储数据库路径。

情况 2

在以下示例中, 有一个称为 MYDB 的可恢复数据库, 该数据库具有 SYSCATSPACE 和 1000 个名为 Txxxx 的用户表空间, 其中 x 表示表空间号 (例如 T0001)。有一个完整数据库备份映像 (BK1)

示例 6

下列命令复原除 T0999 和 T1000 之外的所有表空间:

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb rebuild with all tablespaces in image except
tablespace (T0999, T1000) taken at BK1 without prompting
```

2. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令（假定所有日志都已保存并且可以访问）：

```
db2 rollforward db mydb to end of logs
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令：

```
db2 rollforward db mydb stop
```

此时，该数据库是可连接的，并且除 T0999 和 T1000 之外的所有表空间都处于 NORMAL 状态。T0999 和 T1000 处于 RESTORE_PENDING 状态。

情况 3

这种情况中的示例演示了如何使用增量备份来重建可恢复数据库。在下列示例中，有一个称为 MYDB 的数据库，该数据库中存在下列表空间：

- SYSCATSPACE（系统目录）
- USERSP1（数据表空间）
- USERSP2（用户数据表空间）
- USERSP3（用户数据表空间）

进行了下列备份：

- FULL1 是 SYSCATSPACE、USERSP1、USERSP2 和 USERSP3 的完整备份
- DELTA1 是 SYSCATSPACE 和 USERSP1 的差异备份
- INCR1 是 USERSP2 和 USERSP3 的增量备份
- DELTA2 是 SYSCATSPACE、USERSP1、USERSP2 和 USERSP3 的差异备份
- DELTA3 是 USERSP2 的差异备份
- FULL2 是 USERSP1 的完整备份

示例 7

下列命令使用增量自动复原仅将 SYSCATSPACE 和 USERSP2 重建至最新时间点。

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令。INCREMENTAL AUTO 选项是可选的。RESTORE 实用程序将检测映像的详细程度，并在需要时使用自动增量复原。

```
db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP2)
      incremental auto taken at DELTA3 without prompting
```

2. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令（假定所有日志都已保存并且可以访问）：

```
db2 rollforward db mydb to end of logs
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令：

```
db2 rollforward db mydb stop
```

此时，该数据库是可连接的，但只有 SYSCATSPACE 和 USERSP2 处于 NORMAL 状态。USERSP1 和 USERSP3 处于 RESTORE_PENDING 状态。

示例 8

下列命令使用增量自动复原将整个数据库重建至最新时间点。

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令。INCREMENTAL AUTO 选项是可选的。RESTORE 实用程序将检测映像的详细程度，并在需要时使用自动增量复原。

```
db2 restore db mydb rebuild with all tablespaces in database
incremental auto taken at DELTA3 without prompting
```

2. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令（假定所有日志都已保存并且可以访问）：

```
db2 rollforward db mydb to end of logs
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令：

```
db2 rollforward db mydb stop
```

此时，该数据库是可连接的并且所有表空间都处于 NORMAL 状态。

示例 9

下列命令将整个数据库（USERSP3 除外）重建至最新时间点。

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令。虽然目标映像是非增量映像，但 RESTORE 实用程序还是检测必需的重建链是否包括增量映像，并且将以增量方式自动复原这些映像。

```
db2 restore db mydb rebuild with all tablespaces in database except
tablespace (USERSP3) taken at FULL2 without prompting
```

2. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令（假定所有日志都已保存并且可以访问）：

```
db2 rollforward db mydb to end of logs
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令：

```
db2 rollforward db mydb stop
```

情况 4

这种情况中的示例演示了如何使用包含日志文件的备份映像来重建可恢复数据库。在下列示例中，有一个称为 MYDB 的数据库，该数据库中存在下列表空间：

- SYSCATSPACE（系统目录）
- USERSP1（用户数据表空间）
- USERSP2（用户数据表空间）

示例 10

下列命令仅将数据库中的 SYSCATSPACE 和 USERSP2 重建至最新时间点。有一个完整联机数据库备份映像（BK1），它包含日志文件。

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令：

```
db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP2)
taken at BK1 logtarget /logs without prompting
```

2. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令（假定 BK1 末尾之后的所有日志都已保存并且可以访问）：

```
db2 rollforward db mydb to end of logs overflow log path (/logs)
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令：

```
db2 rollforward db mydb stop
```

此时，该数据库是可连接的，但只有 SYSCATSPACE 和 USERSP2 处于 NORMAL 状态。USERSP1 处于 RESTORE_PENDING 状态。

示例 11

下列命令将数据库重建至最新时间点。有两个包含日志文件的完整联机表空间备份映像：

- BK1 是 SYSCATSPACE 的备份，使用日志文件 10-45
 - BK2 是 USERSP1 和 USERSP2 的备份，使用日志文件 64-80
1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令：

```
db2 restore db mydb rebuild with all tablespaces in database
      taken at BK2 logtarget /logs without prompting
```

前滚操作将从日志文件 10 开始，如果该文件不在主日志文件路径中，那么它总是位于溢出日志路径中。由于日志范围 46-63 未包含在任何备份映像中，所以需要使它们可用于前滚操作。

2. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令，并使用日志文件 64-80 的溢出日志路径：

```
db2 rollforward db mydb to end of logs overflow log path (/logs)
```

3. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令：

```
db2 rollforward db mydb stop
```

此时，该数据库是可连接的并且所有表空间都处于 NORMAL 状态。

情况 5

在下列示例中，有一个称为 MYDB 的可恢复数据库，该数据库中存在下列表空间：

- SYSCATSPACE (0)，SMS 系统目录（相对容器）
- USERSP1 (1) SMS 用户数据表空间（相对容器）
- USERSP2 (2) DMS 用户数据表空间（绝对容器 /usersp2）
- USERSP3 (3) DMS 用户数据表空间（绝对容器 /usersp3）

进行了下列备份：

- BK1 是 SYSCATSPACE 和 USERSP1 的备份
- BK2 是 USERSP2 和 USERSP3 的备份
- BK3 是 USERSP3 的备份

示例 12

下列命令使用重定向复原将整个数据库重建至最新时间。

1. 发出带有 REBUILD 选项的 RESTORE DATABASE 命令：

```
db2 restore db mydb rebuild with all tablespaces in database
      taken at BK3 redirect without prompting
```

2. 对想要重新定义其容器的每个表空间发出 SET TABLESPACE CONTAINERS 命令。
例如：

```
db2 set tablespace containers for 3 using (file '/newusersp2' 10000)
```

- 3.

```
db2 set tablespace containers for 4 using (file '/newusersp3' 15000)
```

4. 发出带有 CONTINUE 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb continue
```

5. 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令 (假定所有日志都已保存并且可以访问):

```
db2 rollforward db mydb to end of logs
```

6. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db mydb stop
```

此时, 该数据库是可连接的并且所有表空间都处于 NORMAL 状态。

情况 6

在下列示例中, 有一个称为 MYDB 的数据库, 它具有三个数据库分区:

- 数据库分区 1 包含表空间 SYSCATSPACE、USERSP1 和 USERSP2, 它是目录分区
- 数据库分区 2 包含表空间 USERSP1 和 USERSP3
- 数据库分区 3 包含表空间 USERSP1、USERSP2 和 USERSP3

进行了下列备份, 其中 BKxy 表示分区 y 上的备份编号 x:

- BK11 是 SYSCATSPACE、USERSP1 和 USERSP2 的备份
- BK12 是 USERSP2 和 USERSP3 的备份
- BK13 是 USERSP1、USERSP2 和 USERSP3 的备份
- BK21 是 USERSP1 的备份
- BK22 是 USERSP1 的备份
- BK23 是 USERSP1 的备份
- BK31 是 USERSP2 的备份
- BK33 是 USERSP2 的备份
- BK42 是 USERSP3 的备份
- BK43 是 USERSP3 的备份

示例 13

下列命令将整个数据库重建至日志末尾。

1. 在数据库分区 1 上, 发出带有 REBUILD 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb rebuild with all tablespaces in database  
taken at BK31 without prompting
```

2. 在数据库分区 2 上, 发出带有 REBUILD 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb rebuild with tablespaces in database taken at  
BK42 without prompting
```

3. 在数据库分区 3 上, 发出带有 REBUILD 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb rebuild with all tablespaces in database  
taken at BK43 without prompting
```

4. 在目录分区上, 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令 (假定在所有数据库分区上, 所有日志都已保存并且可以访问):

```
db2 rollforward db mydb to end of logs
```

5. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db mydb stop
```

此时, 该数据库在所有数据库分区上都是可连接的, 并且所有表空间都处于 NORMAL 状态。

示例 14

下列示例将 SYSCATSPACE、USERSP1 和 USERSP2 重建至最新时间点。

1. 在数据库分区 1 上, 发出带有 REBUILD 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb rebuild with all tablespaces in database  
taken at BK31 without prompting
```

2. 在数据库分区 2 上, 发出带有 REBUILD 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb rebuild with all tablespaces in image taken at  
BK22 without prompting
```

3. 在数据库分区 3 上, 发出带有 REBUILD 选项的 RESTORE DATABASE 命令:

```
db2 restore db mydb rebuild with all tablespaces in image taken at  
BK33 without prompting
```

注意: 此命令省略了 USERSP1, 它是完成重建操作所必需的。

4. 在目录分区上, 发出带有 TO END OF LOGS 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db mydb to end of logs
```

5. 发出带有 STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db mydb stop
```

前滚成功, 并且该数据库在所有数据库分区上都是可连接的。所有表空间都处于 NORMAL 状态, 但 USERSP3 和 USERSP1 除外 (USERSP3 在它所在的所有数据库分区上处于 RESTORE PENDING 状态, 而 USERSP1 在数据库分区 3 上处于 RESTORE PENDING 状态)。

尝试访问数据库分区 3 上 USERSP1 中的数据时, 将发生数据访问错误。要修正此错误, 需要恢复 USERSP1:

- a. 在数据库分区 3 上, 发出 RESTORE DATABASE 命令并指定包含 USERSP1 的备份映像:

```
db2 restore db mydb tablespace taken at BK23 without prompting
```

- b. 在目录分区上, 发出带有 TO END OF LOGS 选项和 AND STOP 选项的 ROLLFORWARD DATABASE 命令:

```
db2 rollforward db mydb to end of logs on dbpartitionnum (3) and stop
```

此时, 可以访问数据库分区 3 上 USERSP1 中的数据, 因为 USERSP1 处于 NORMAL 状态。

情况 7

在下列示例中, 有一个称为 MYDB 的不可恢复数据库, 它具有下列表空间:

- SYSCATSPACE (0), SMS 系统目录
- USERSP1 (1) SMS 用户数据表空间

- USERSP2 (2) DMS 用户数据表空间
- USERSP3 (3) DMS 用户数据表空间

只有该数据库的一个备份 BK1:

例 15

下列命令演示了如何对不可恢复数据库使用重建。

仅使用 SYSCATSPACE 和 USERSP1 重建数据库:

```
db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP1)
      taken at BK1 without prompting
```

复原之后，该数据库是可连接的。如果发出 LIST TABLESPACES 命令，您将看到 SYSCATSPACE 和 USERSP1 处于 NORMAL 状态，而 USERSP2 和 USERSP3 处于 DELETE_PENDING/OFFLINE 状态。现在就可以使用处于 NORMAL 状态的那两个表空间了。

如果想要备份数据库，那么首先需要使用 DROP TABLESPACE 命令删除 USERSP2 和 USERSP3，否则备份将失败。

要在以后复原 USERSP2 和 USERSP3，需要从 BK1 重新发出数据库复原操作。

第 14 章 rollforward 概述

DB2 ROLLFORWARD DATABASE 命令的最简单的格式只需要您指定想要前滚恢复的数据库的别名。例如:

```
db2 rollforward db sample to end of logs and stop
```

在此示例中, 命令返回:

```
                                前滚状态
输入数据库别名                    = sample
返回了状态的节点数                = 1
节点号                              = 0
前滚状态                            = 未暂挂
要读取的下一个日志文件            =
已处理的日志文件数                  = -
上次落实事务时间                    = 2001-03-11-02.39.48.000000
```

DB20000I 已成功完成 ROLLFORWARD 命令。

以下是一种可以用来执行前滚恢复操作的方法:

1. 在不指定 STOP 选项的情况下调用 Rollforward 实用程序。
2. 在指定 QUERY STATUS 选项的情况下调用 Rollforward 实用程序

如果指定恢复至日志末尾, 而返回的时间点又早于您预期的时间点, QUERY STATUS 选项会指示丢失了一个或多个日志文件。

如果指定时间点恢复, QUERY STATUS 选项将有助于您确保前滚操作已在正确的时间点完成。

3. 在指定 STOP 选项的情况下调用 Rollforward 实用程序。在操作停止后, 不可能前滚其他更改。

另一种可以用来执行前滚恢复操作的方法是:

1. 在指定 AND STOP 选项的情况下调用 Rollforward 实用程序。
2. 是否需要执行其他步骤取决于前滚操作的输出:
 - 如果前滚操作成功, 前滚操作就会完成, 并且数据库将是可连接并可使用的。此时, 不可能前滚其他更改。
 - 如果返回了任何错误, 那么需要执行操作以解决问题 (例如, 如果缺少日志文件, 就需要查找该日志文件, 或者, 如果发生了检索错误, 那么应确保日志归档功能正常运行)。然后, 在指定 AND STOP 选项的情况下重新运行 Rollforward 实用程序。

必须先使用 RESTORE 实用程序成功地复原数据库, 然后才能前滚它, 但表空间就不一样。表空间可以暂时处于前滚暂挂状态, 但不需要执行复原操作以撤销该状态 (例如在断电后执行该操作)。

调用 Rollforward 实用程序时:

- 如果数据库处于前滚暂挂状态，将前滚该数据库。如果表空间也处于前滚暂挂状态，那么必须在数据库前滚操作完成后，再次调用 `Rollforward` 实用程序来前滚该表空间。
- 如果数据库不处于前滚暂挂状态，但该数据库中的表空间处于前滚暂挂状态：
 - 如果指定表空间列表，将只前滚那些表空间。
 - 如果不指定表空间列表，将前滚处于前滚暂挂状态的所有表空间。

数据库前滚操作是脱机运行的。直到前滚操作成功完成数据库才可用，除非在调用实用程序时指定了 `STOP` 选项，否则该操作不能完成。

表空间前滚操作可脱机运行。直到前滚操作成功完成数据库才可用。当达至日志末尾时或在调用实用程序时指定了 `STOP` 选项，会发生这种情况。

只要表空间中未包括 `SYSCATSPACE`，就可对表空间执行联机前滚操作。对表空间执行联机前滚操作时，该表空间不可用，但数据库中的其他表空间是可用的。

首次创建数据库时，只对它启用了循环日志记录。这意味着日志是可复用的，而不是被保存或归档。使用循环日志记录，是不能进行前滚恢复的：只能进行崩溃恢复或版本恢复。归档日志将建立备份后对数据库进行的更改记录成文档。通过将 `logarchmeth1` 数据库配置参数设置为其缺省值 `OFF` 以外的值来启用日志归档（和前滚恢复）。将 `logarchmeth1` 设置为 `OFF` 以外的值时，数据库将处于备份暂挂状态，而且必须脱机备份该数据库才能再次使用。

注：在恢复历史记录文件中为前滚操作中使用的每个日志文件生成条目。

使用前滚

使用 `ROLLFORWARD DATABASE` 命令将数据库日志文件中记录的事务应用于复原的数据库备份映像或表空间备份映像。

不应连接到要进行前滚恢复的数据库：`Rollforward` 实用程序会自动建立与特定数据库的连接，且此连接在前滚操作完成时终止。

如果未取消正在进行中的前滚操作，就不要复原表空间；否则，在表空间集中可能有些表空间处于“前滚进行中”状态，而另一些表空间则处于“前滚暂挂”状态。正在进行中的前滚操作将只在处于“前滚进行中”状态的表空间上执行。

数据库可以是本地数据库或远程数据库。

`Rollforward` 实用程序有以下限制：

- 每次只能调用一个前滚操作。如果有多个要复原的表空间，可在同一操作中指定所有表空间。
- 如果在最近的备份操作后有重命名的表空间，应确保在前滚表空间时使用新名称。先前表空间名将不被识别。
- 不能取消正在运行的前滚操作。只能取消已完成的前滚操作，对那些尚未指定 `STOP` 选项或在完成前已失败的前滚操作则不能取消。
- 指定的时间戳记小于前一时间戳记时，不能继续将表空间前滚至某时间点。如果未指定时间点，将使用前一时间点。可以只指定 `STOP` 来发出一个在指定时间点前结束的前滚操作，但只有在涉及的表空间都是从同一脱机备份映像复原时才能这样做。

在这种情况下，不需要日志处理。如果在正在进行的前滚操作完成或取消前使用另一表空间列表启动另一前滚操作，会返回一条错误消息（SQL4908）。对所有数据库分区调用 LIST TABLESPACES 命令，以确定当前正在前滚（“前滚进行中”状态）的表空间，以及已准备好进行前滚（前滚暂挂状态）的表空间。有 3 种选择：

- 完成所有表空间上正在进行的前滚操作。
 - 完成表空间子集上正在进行的前滚操作。（如果前滚操作将继续到特定时间点，从而需要涉及到所有数据库分区，那么可能无法完成表空间子集上正在进行的前滚操作。）
 - 取消正在进行的前滚操作。
- 在分区数据库环境中，必须从数据库的目录分区调用 Rollforward 实用程序。
 - 仅从 DB2 版本 9 客户机提供表空间的时间点前滚。要将表空间前滚至某个时间点，应将运行较低版本的数据库产品的任何客户机升级至版本 9。
 - 无法从前发行版前滚日志。

可以通过命令行处理器（CLP）、控制中心的“复原”向导或者 db2Rollforward 应用程序编程接口（API）来调用 Rollforward 实用程序。

以下是通过 CLP 发出的 ROLLFORWARD DATABASE 命令的示例：

```
db2 rollforward db sample to end of logs and stop
```

要打开“复原”向导：

1. 从“控制中心”中，展开对象树，直到找到要复原的数据库或表空间对象为止。
2. 右键单击该对象，然后从弹出菜单中选择“前滚”。这就打开了“前滚”向导。

在“控制中心”中通过上下文帮助工具提供详细信息。

前滚表空间中的更改

如果数据库启用了正向恢复，您可以选择备份、复原和前滚表空间，而不必前滚整个数据库。可能需要对各个表空间实现恢复策略，因为这可节省时间：复原数据库的一部分所需的时间比恢复整个数据库所需的时间少。例如，如果一个磁盘损坏且它只包含一个表空间，那么可以复原并前滚该表空间，而不必恢复整个数据库，也不会影响用户访问该数据库的其余部分，除非损坏的表空间包含了系统目录表；在这种情况下，您不能连接至数据库。（如果包含系统目录表空间的表空间级备份映像是可用的，那么可以独立复原系统目录表空间。）表空间级的备份还允许您较为频繁地备份数据库的关键部分，因此所需的时间比备份整个数据库的时间少。

复原表空间后，它始终处于前滚暂挂状态。要使该表空间可以使用，必须对它执行前滚恢复。在大多数情况中，可以选择前滚至日志末尾或前滚至特定的时间点。但是不能将包含系统目录表的表空间前滚至某时间点。这些表空间必须前滚至日志末尾，以确保数据库中的所有表空间保持一致。

当前滚表空间时，DB2 将处理所有日志文件，即使它们未包含影响该表空间的日志记录。要跳过已知不包含影响表空间的任何日志记录的日志文件，将 DB2_COLLECT_TS_REC_INFO 注册表变量设置为 ON。这是缺省值。为了确保收集跳过日志文件所需的信息，在创建和使用日志文件之前，必须设置注册表变量。

数据库目录中的表空间更改历史记录文件（DB2TSCHG.HIS）记录应对每个表空间处理的那些日志。通过使用 db2logsForRfwd 实用程序可以查看此文件的内容，通过使用

PRUNE HISTORY 命令可以从中删除条目。在数据库复原操作期间，从备份映像复原 DB2TSCHG.HIS，然后在数据库前滚操作期间对它进行更新。如果没有任何信息可用于日志文件，那么就好像每个表空间的恢复都需要该日志文件一样来对其进行处理。

因为在日志不活动之后每个日志文件的信息都会清空至磁盘，所以崩溃可能会导致此信息丢失。为弥补这一点，如果恢复操作从日志文件的中间开始，那么就好像该日志包含对系统中每个表空间的修改来处理整个日志。在这之后，将处理活动日志并重建它们的信息。如果在崩溃情况下丢失较旧的或归档的日志文件的信息，且它们在数据文件中没有任何信息，那么就好像它们包含表空间恢复操作期间对每个表空间的修改一样来处理这些日志文件。

在前滚表空间之前，调用 LIST TABLESPACES SHOW DETAIL 命令。此命令将返回最小恢复时间，这是表空间可前滚至的最早时间点。对表空间或表空间中的表运行数据定义语言 (DDL) 语句时，将更新此最小恢复时间。必须将该表空间至少前滚至最小恢复时间，以便与系统目录表中的信息同步。如果恢复多个表空间，表空间必须至少前滚至要恢复的所有表空间中的最高“最小恢复时间”。在分区数据库环境中，对所有数据库分区发出 LIST TABLESPACES SHOW DETAIL 命令。表空间必须前滚至所有数据库分区上的所有表空间中的最高“最小恢复时间”。

如果要将表空间前滚到某时间点，而且有一个表包含在多个表空间中，那么必须同时前滚所有这些表空间。例如，如果该表数据包含在一个表空间中，而该表的索引包含在另一个表空间中，那么必须同时将这两个表空间前滚至相同的时间点。

如果表中的数据对象位于各自的表空间中，并且已将该对象数据重组，那么必须同时复原并前滚数据和对象的表空间。在重组该表之后，应该为受影响的表空间建立备份。

如果要将一个表空间前滚到某时间点，那么表空间中的表是：

- 另一个表空间中的具体化查询表或登台表的基础表
- 另一个表空间中的表的具体化查询表或登台表

应该将两个表空间前滚到同一时间点。否则，在前滚操作结束时，具体化查询表或登台表将处于设置完整性暂挂状态。具体化查询表将需要完全刷新，并且登台表将被标记为不完整。

如果要将一个表空间前滚至一个时间点，而该表空间中的一个表与另一个表空间所包含的另一个表存在引用完整性关系，那么应同时将这两个表空间前滚至相同的时间点。否则，在前滚操作结束时，引用完整性关系中的子表将处于设置完整性暂挂状态。以后检查子表是否存在约束违例时，需要对整个表进行检查。如果下列其中任何一个表存在，那么它们将与子表一起处于设置完整性暂挂状态：

- 子表的任何派生具体化查询表
- 子表的任何后代登台表
- 子表的任何后代外键表

这些表将需要完全完整性处理，以使它们脱离设置完整性暂挂状态。如果同时前滚这两个表空间，那么该约束将在该时间点前滚操作结束时仍然有效。

应确保时间点表空间前滚操作不会导致在某些表空间中回滚一个事务，而在另一些空间中落实该事务。这种情况可能在下列时候发生：

- 对一个事务已更新的表空间的子集执行时间点前滚操作，且该时间点在此事务落实的时间之前。
- 要前滚至某个时间点的表空间中所包含的任何表有一个相关的触发器，或者这个表由一个触发器来更新，该触发器影响的表空间不是要前滚的表空间。

解决方案是：找到一个可阻止这种情况发生的适当的时间点。

可以发出 `QUIESCE TABLESPACES FOR TABLE` 命令来创建事务一致的时间点，以将表空间前滚。停顿请求（是共享的，用于更新或互斥方式）等待（通过锁定）对那些表空间运行的所有事务完成，并阻拦新请求。准许停顿请求时，表空间处于一致状态。要确定停止前滚操作的适当时间，可查看恢复历史记录文件以找出抑制点，并检查它是否是在最小复原时间后发生的。

表空间时间点前滚操作完成后，表空间将置于备份暂挂状态。必须建立该表空间的备份，因为在前滚到的时间点与当前时间之间对该表空间所作的更新都被除去。再也不能从先前的数据库级别或表空间级别备份映像将该表空间前滚到当前时间。以下示例显示表空间级的备份映像为什么是必需的，以及如何使用。（要使表空间可用，可以备份整个数据库、处于备份暂挂状态的表空间，或包括处于备份暂挂状态的表空间的表空间集合。）

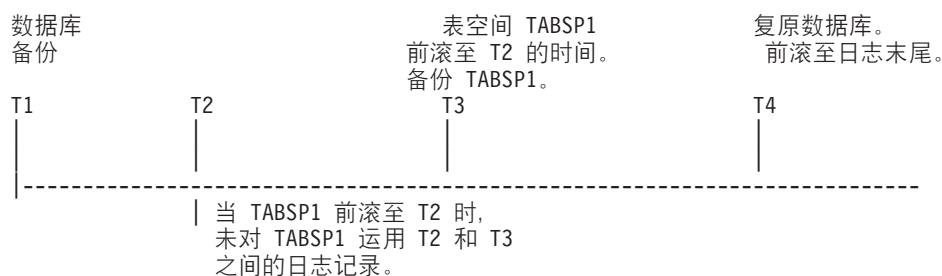


图 24. 表空间备份要求

在上述示例中，在 T1 时备份数据库。然后在 T3 时，表空间 TABSP1 前滚至某特定的时间点（T2），表空间在时间 T3 后备份。因为表空间处于备份暂挂状态，所以备份操作是必需的。表空间备份映像的时间戳记在 T3 后，但表空间是在 T2。T2 和 T3 之间的日志记录没有应用于 TABSP1。在时间 T4 处，将使用在 T1 处创建的备份映像复原数据库，并前滚至日志末尾。表空间 TABSP1 将在时间 T3 处置于复原暂挂状态，因为数据库管理器假设在 T3 和 T4 之间，对 TABSP1 执行操作，而无需对表空间应用了 T2 和 T3 之间的日志更改。如果这些日志更改是作为对数据库的前滚操作的一部分应用的，该假设将不成立。必须在表空间前滚到某时间点后建立的表空间级的备份，使您可将那个表空间前滚至前一时间点前滚操作之前（在上例中为 T3）。

假定要将表空间 TABSP1 复原到 T4，应该从 T3 之后建立的备份映像（必需的备份或稍后的一个）复原该表空间，然后将 TABSP1 前滚至日志末尾。

在前面的示例中，将数据库复原到时间 T4 最有效的方法应是按下列顺序执行必需的步骤：

1. 复原数据库。
2. 复原表空间。
3. 前滚数据库。
4. 前滚表空间。

因为您在前滚数据库之前复原表空间，所以在前滚数据库时将不使用资源向表空间应用日志记录。

如果找不到时间 T3 后的 TABSP1 备份映像，或者想将 TABSP1 复原到 T3（或更早），可以：

- 将表空间前滚至 T3。无需再次复原表空间，因为它已根据数据库备份映像复原。
- 再次使用在时间 T1 建立的数据库备份来复原表空间，然后将该表空间前滚至时间 T3 之前的一个时间。
- 删除表空间。

在分区数据库环境中：

- 必须同时将一个表空间的所有部分前滚至同一时间点。这可确保该表空间在各数据库分区中是一致的。
- 如果某些数据库分区处于前滚暂挂状态，而且在其他数据库分区上，某些表空间处于前滚暂挂状态（但数据库分区不是），您必须首先前滚数据库分区，然后前滚表空间。
- 如果打算将表空间前滚至日志末尾，不必在每个数据库分区上复原：只需在需要恢复的数据库分区上复原即可。但如果打算将表空间前滚到某时间点，那么必须在每个数据库分区上将其复原。

在包含分区表的数据库中：

- 如果将包含任何分区表部分的表空间前滚到某个时间点，那么还必须将所有其他包含该表的表空间前滚到同一时间点。但是，允许将包含部分分区表的单个表空间前滚至日志末尾。如果分区表带有任何已连接、已拆离或已删除数据分区，那么时间点前滚还必须包括这些数据分区的所有表空间。要确定某个分区表是否带有任何已连接、已拆离或已删除的数据分区，请查询 SYSCAT.DATAPARTITIONS 目录视图。

rollforward 所需的授权

特权使用户能够创建或访问数据库资源。权限级别提供了对特权、较高级别数据库管理器维护和实用程序操作进行分组的方法。这两者一起用于控制对数据库管理器及其数据库对象的访问。用户只能访问那些他们具有相应授权（即必需的特权或权限）的对象。

必须具有 SYSADM、SYSCTRL 或 SYSMAINT 权限才能使用 Rollforward 实用程序。

前滚会话 - CLP 示例

示例 1

ROLLFORWARD DATABASE 命令允许每次指定多个操作，各个操作由关键字 AND 隔开。例如，要前滚至日志末尾，然后完成，可将下列独立的命令：

```
db2 rollforward db sample to end of logs
db2 rollforward db sample complete
```

组合为：

```
db2 rollforward db sample to end of logs and complete
```

虽然这两种方法是等效的，但建议您分两个步骤来完成此类操作。在停止前滚操作前验证它是否具有预期的进度，以免丢失任何日志，这一点很重要。

如果前滚命令遇到错误，前滚操作就无法完成。在此情况下，将返回该错误，这样，您就能够修正该错误并重新发出以上命令。但是，如果无法修正该错误，那么可以通过发出以下命令强制前滚完成：

```
db2 rollforward db sample complete
```

此命令使数据库联机并复原到发生故障前日志点。

示例 2

将数据库前滚至日志末尾（已复原了两个表空间）：

```
db2 rollforward db sample to end of logs
db2 rollforward db sample to end of logs and stop
```

这两个语句是等效的。不需要 AND STOP 或 AND COMPLETE 表空间就可以前滚恢复至日志末尾。不需要表空间名称。如果未指定，将包括所有需要前滚恢复的表空间。如果将只前滚这些表空间的一个子集，那么必须指定它们的名称。

示例 3

复原了 3 个表空间后，将其中一个前滚至日志末尾，另两个前滚到某时间点，所有操作都要联机执行：

```
db2 rollforward db sample to end of logs tablespace(TBS1) online
db2 rollforward db sample to 1998-04-03-14.21.56.245378 and stop
tablespace(TBS2, TBS3) online
```

应注意，两个前滚操作不能并发运行。只有在成功地完成了第一个前滚操作后，才能调用第二个命令。

示例 4

复原数据库后，前滚到某时间点，使用 OVERFLOW LOG PATH 来指定用户出口用来保存已归档日志的目录：

```
db2 rollforward db sample to 1998-04-03-14.21.56.245378 and stop
overflow log path (/logs)
```

示例 5

在以下示例中，有一个称为 sample 的数据库。备份该数据库，在备份映像中包含恢复日志；复原该数据库；然后，将该数据库前滚到备份时间戳记末尾。

备份该数据库，在备份映像中包含恢复日志：

```
db2 backup db sample online include logs
```

使用备份映像复原该数据库：

```
db2 restore db sample
```

将该数据库前滚到备份时间戳记末尾：

```
db2 rollforward db sample to end of backup
```

示例 6 (分区数据库环境)

有三个数据库分区: 0、1 和 2。在所有数据库分区上定义表空间 TBS1, 在数据库分区 0 和 2 上定义表空间 TBS2。在数据库分区 1 上复原了数据库, 并在数据库分区 0 和 2 上复原了 TBS1 之后, 在数据库分区 1 上前滚数据库:

```
db2 rollforward db sample to end of logs and stop
```

这会返回警告 SQL1271 (已恢复数据库, 但数据库分区 0 和 2 上的一个或多个表空间处于脱机状态)。

```
db2 rollforward db sample to end of logs
```

此命令在数据库分区 0 和 2 上前滚 TBS1。在此情况下, 子句 TABLESPACE(TBS1) 是可选的。

示例 7 (分区数据库环境)

在以下示例中, 有一个称为 sample 的分区数据库。使用单系统视图备份备份所有数据库分区; 在所有数据库分区上复原该数据库; 将该数据库前滚到备份时间戳记末尾。

执行单系统视图 (SSV) 备份:

```
db2 backup db sample on all nodes online include logs
```

在所有数据库分区上复原该数据库:

```
db2_all "db2 restore db sample taken at 1998-04-03-14.21.56.245378"
```

将该数据库前滚到备份时间戳记末尾:

```
db2 rollforward db sample to end of backup on all nodes
```

示例 8 (分区数据库环境)

在以下示例中, 有一个称为 sample 的分区数据库。使用 db2_all, 通过一个命令备份所有数据库分区; 在所有数据库分区上复原该数据库; 并将该数据库前滚到备份时间戳记末尾。

使用 db2_all, 通过一个命令备份所有数据库分区:

```
db2_all "db2 backup db sample include logs to /shared/dir/"
```

在所有数据库分区上复原该数据库:

```
db2_all "db2 restore db sample from /shared/dir/"
```

将该数据库前滚到备份时间戳记末尾:

```
db2 rollforward db sample to end of backup on all nodes
```

示例 9 (分区数据库环境)

只在数据库分区 0 和 2 上复原表空间 TBS1 之后, 在数据库分区 0 和 2 上前滚 TBS1:

```
db2 rollforward db sample to end of logs
```

忽略数据库分区 1。

```
db2 rollforward db sample to end of logs tablespace(TBS1)
```

此命令失败，因为 TBS1 未对在数据库分区 1 上进行前滚恢复作好准备。报告 SQL4906N。

```
db2 rollforward db sample to end of logs on
dbpartitionnums (0, 2) tablespace(TBS1)
```

成功完成。

```
db2 rollforward db sample to 1998-04-03-14.21.56.245378 and stop
tablespace(TBS1)
```

此命令失败，因为 TBS1 未对在数据库分区 1 上进行前滚恢复做好准备；必须将所有段一起前滚。

注：在将表空间前滚到某时间点之后，将不接受 dbpartitionnum 子句。前滚操作必须在表空间所在的所有数据库分区上进行。

在数据库分区 1 上复原 TBS1 后：

```
db2 rollforward db sample to 1998-04-03-14.21.56.245378 and stop
tablespace(TBS1)
```

成功完成。

示例 10（分区数据库环境）

在所有数据库分区上复原表空间后前滚至 PIT2，但不指定 AND STOP。前滚操作仍在进行中。取消并前滚至 PIT1：

```
db2 rollforward db sample to pit2 tablespace(TBS1)
db2 rollforward db sample cancel tablespace(TBS1)

** restore TBS1 on all dbpartitionnums **

db2 rollforward db sample to pit1 tablespace(TBS1)
db2 rollforward db sample stop tablespace(TBS1)
```

示例 11（分区数据库环境）

前滚恢复 db2nodes.cfg 文件中列示的 8 个数据库分区（3 至 10）上的表空间：

```
db2 rollforward database dwtest to end of logs tablespace (tssprodt)
```

前滚恢复至日志末尾（而不是时间点）的操作成功完成。不必指定表空间所在的数据库分区。实用程序缺省到 db2nodes.cfg 文件。

示例 12（分区数据库环境）

前滚恢复单一数据库分区数据库分区组（在数据库分区 6 上）上的 6 个小表空间：

```
db2 rollforward database dwtest to end of logs on dbpartitionnum (6)
tablespace(tsstore, tssbuyer, tsstime, tsswhse, tsslscat, tssvendor)
```

前滚恢复至日志末尾（而不是时间点）的操作成功完成。

示例 13（分区表 - 在所有数据分区上前滚至日志末尾）

使用表空间 tbsp1、tbsp2 和 tbsp3 创建了分区表，索引在 tbsp0 中。后来，用户对 tbsp4 中的表添加了数据分区，并从 tbsp5 中的表连接了数据分区。可以将所有表空间前滚至日志末尾。

```
db2 rollforward db PBARDB to END OF LOGS and stop
    tablespace(tbsp0, tbsp1, tbsp2, tbsp3, tbsp4, tbsp5)
```

成功完成。

示例 14 (分区表 - 在一个表空间上前滚至日志末尾)

最初使用表空间 `tbsp1`、`tbsp2` 和 `tbsp3` 创建了分区表，索引在 `tbsp0` 中。后来，用户对 `tbsp4` 中的表添加了数据分区，并从 `tbsp5` 中的表连接了数据分区。表空间 `tbsp4` 损坏并要求复原和前滚至日志末尾。

```
db2 rollforward db PBARDB to END OF LOGS and stop tablespace(tbsp4)
```

成功完成。

示例 15 (分区表 - 在所有数据分区上前滚到时间点，这些数据分区包括那些已添加、已连接、已拆离或带索引的数据分区)

使用表空间 `tbsp1`、`tbsp2` 和 `tbsp3` 创建了分区表，索引在 `tbsp0` 中。后来，用户对 `tbsp4` 中的表添加了数据分区，从 `tbsp5` 中的表连接了数据分区，并从 `tbsp1` 拆离了数据分区。用户对分区表使用的所有表空间（包括 `INDEX IN` 子句中指定的那些表空间）执行前滚到 PIT 操作。

```
db2 rollforward db PBARDB to 2005-08-05-05.58.53.000000 and stop
    tablespace(tbsp0, tbsp1, tbsp2, tbsp3, tbsp4, tbsp5)
```

成功完成。

示例 16 (分区表 - 在一小部分表空间上前滚到 PIT)

使用三个表空间（`tbsp1`、`tbsp2` 和 `tbsp3`）创建了分区表。后来，用户从 `tbsp3` 中拆离了所有数据分区。只允许在 `tbsp1` 和 `tbsp2` 上前滚到 PIT。

```
db2 rollforward db PBARDB to 2005-08-05-06.02.42.000000 and stop
    tablespace( tbsp1, tbsp2)
```

成功完成。

第 15 章 使用 IBM Tivoli Storage Manager (TSM) 进行数据恢复

当调用 BACKUP DATABASE 或 RESTORE DATABASE 命令时，可以指定要使用 IBM Tivoli Storage Manager (TSM) 产品来管理数据库或表空间备份或复原操作。除了在下列平台上之外，需要的 TSM 客户机 API 的最低级别为 V4.2.0:

- 64 位 Solaris 系统，它需要 TSM 客户机 API V4.2.1。
- 64 位 Windows 操作系统，它需要 TSM 客户机 API V5.1。
- 所有 Windows X64 系统，它需要 TSM 客户机 API V5.3.2。
- 32 位 Linux for iSeries[®] 和 Linux for pSeries，它们需要 TSM 客户机 API V5.1.5 或更新版本。
- 64 位 Linux for System i 和 Linux for pSeries，它们需要 TSM 客户机 API V5.2.2 或更新版本。
- 64 位 Linux on AMD Opteron 系统，它们需要 TSM 客户机 API V5.2.0 或更新版本。
- Linux for zSeries，它们需要 TSM 客户机 API V5.2.2 或更新版本。

配置 Tivoli Storage Manager 客户机

可能需要执行下列步骤以配置 TSM 环境，这样数据库管理器才能使用 TSM 选项:

1. 必须安装和配置可用的 TSM 客户机和服务器。另外，必须在每台 DB2 服务器上安装 TSM 客户机 API。
2. 设置 TSM 客户机 API 使用的环境变量:

DSMI_DIR

标识 API 可信代理程序文件 (dsmtca) 所在的用户定义目录路径。

DSMI_CONFIG

标识 dsm.opt 文件 (它包含 TSM 用户选项) 的用户定义目录路径。与另外两个变量不同，此变量应包含标准路径和文件名。

DSMI_LOG

标识将在其中创建错误日志 (dsierror.log) 的用户定义目录路径。

注: 在多分区数据库环境中，这些设置必须在 sqllib/userprofile 目录中指定。

3. 如果对这些环境变量进行了任何更改，并且数据库管理器正在运行，那么应该:
 - 使用 db2stop 命令停止数据库管理器。
 - 使用 db2start 命令启动数据库管理器。
4. 根据服务器配置的不同，Tivoli 客户机可能需要密码才能与 TSM 服务器交互。如果将 TSM 环境配置为使用 PASSWORDACCESS=generate，那么 Tivoli 客户机需要建立它的密码。

将可执行文件 dsmapipw 安装在实例所有者的 sqllib/adsm 目录中。此可执行文件允许您建立和重设 TSM 密码。

要执行 `dsmapiw` 命令，必须作为本地管理员或“root”用户登录。当执行此命令时，将提示您输入下列信息：

- *旧密码*，它是 TSM 服务器识别的该 TSM 节点的当前密码。第一次执行此命令时，此密码是 TSM 管理员在 TSM 服务器上注册节点时所提供的密码。
- *新密码*，它是 TSM 节点的新密码，存储在 TSM 服务器上。（将提示您输入新密码两次，以检查是否有输入错误。）

注：调用 `BACKUP DATABASE` 或 `RESTORE DATABASE` 命令的用户不需要知道此密码。只需要在为初始连接建立密码时，以及 TSM 服务器上的密码重新设置后，才需要运行 `dsmapiw` 命令。

使用 Tivoli Storage Manager 时的注意事项

要使用 TSM 中的特定功能部件，可能需要指定使用该功能部件的对象的标准路径名。记住在 Windows 操作系统上，将使用 `\` 来代替 `/`。以下是不同对象的标准路径名：

- 完整数据库恢复对象为：`/<database>/NODEnnnn/FULL_BACKUP.timestamp.seq_no`
- 增量数据库恢复对象为：`/<database>/NODEnnnn/DB_INCR_BACKUP.timestamp.seq_no`
- 增量差异数据库恢复对象为：`/<database>/NODEnnnn/DB_DELTA_BACKUP.timestamp.seq_no`
- 完整表空间恢复对象为：`/<database>/NODEnnnn/TSP_BACKUP.timestamp.seq_no`
- 增量表空间恢复对象为：`/<database>/NODEnnnn/TSP_INCR_BACKUP.timestamp.seq_no`
- 增量差异表空间恢复对象为：`/<database>/NODEnnnn/TSP_DELTA_BACKUP.timestamp.seq_no`

其中 `<database>` 为数据库别名，而 `NODEnnnn` 为节点号。显示的大写名称必须输入为大写。

- 对于多个备份映像使用同一个数据库别名的情况，时间戳记和序号就成为标准名称中的专有部分。需要查询 TSM 以确定要使用的备份版本。
- 如果执行联机备份操作并指定了 `USE TSM` 选项和 `INCLUDE LOGS` 选项，那么当两个进程尝试同时写同一台磁带机时，就会发生死锁。如果正在将磁带机用作日志和备份映像的存储设备，那么需要为 TSM 定义两个独立的磁带池，一个用于存储备份映像，另一个用于存储归档日志。

第 16 章 DB2 高级副本服务 (ACS)

DB2 高级副本服务 (ACS) 使您能够使用存储设备的快速复制技术来执行备份和复原操作的数据复制部分。

在传统的备份或复原操作中，数据库管理器使用操作系统调用将数据复制到磁盘或者从磁盘或存储设备复制数据。能够使用存储设备来执行数据复制可以使备份和复原操作更为快速。使用 DB2 ACS 的备份操作被称为快照备份。

要执行快照备份与复原操作，需要适用于存储设备的 DB2 ACS API 驱动程序。集成到 IBM 数据服务器的是适用于下列存储硬件的 DB2 ACS API 驱动程序：

- IBM TotalStorage SAN Volume Controller
- IBM 系统存储器 DS6000
- IBM 系统存储器 DS8000
- IBM 系统存储器 N 系列
- NetApp V 系列
- NetApp FAS 系列

有关设置和使用 DB2 高级副本服务 (ACS) 的详细指示信息，请参阅高级副本服务的 Tivoli 文档，网址为：<http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html>

启用 DB2 高级副本服务 (ACS)

要使用 DB2 高级副本服务 (ACS) 或执行快照备份操作，您必须安装、激活和配置 DB2 ACS。

开始前

DB2 ACS 是 IBM DB2 高可用性 (HA) 功能部件的一部分。要使用 DB2 ACS，您必须具有 DB2 HA 功能部件的许可证。

要执行快照备份与复原操作，需要适用于存储设备的 DB2 ACS API 驱动程序。集成到 IBM 数据服务器的是适用于下列存储硬件的 DB2 ACS API 驱动程序：

- IBM TotalStorage SAN Volume Controller
- IBM 系统存储器 DS6000
- IBM 系统存储器 DS8000
- IBM 系统存储器 N 系列
- NetApp V 系列
- NetApp FAS 系列

过程

1. 安装 DB2 ACS。请参阅：第 258 页的『安装 DB2 高级副本服务 (ACS)』。
2. 创建您将 DB2 ACS 与之配合使用的数据库管理器实例。

当创建新的数据库管理器实例时，会在新的实例 `sqllib` 目录中创建名为 `acs` 的目录。因为每个数据库管理器实例均具有 `acs` 目录，所以您可以采用不同方式配置每个数据库管理器实例。

3. 对于您将 DB2 ACS 与之配合使用的每个数据库管理器实例，请执行下列步骤：
 - a. 激活 DB2 ACS。请参阅：第 259 页的『激活 DB2 高级副本服务（ACS）』。
 - b. 配置 DB2 ACS。请参阅：第 259 页的『配置 DB2 高级副本服务（ACS）』。

结果

在启用 DB2 ACS 之后，可以执行快照备份操作。

有关设置和使用 DB2 高级副本服务（ACS）的详细指示信息，请参阅高级副本服务的 Tivoli 文档，网址为：<http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html>

安装 DB2 高级副本服务（ACS）

DB2 高级副本服务（ACS）所需的文件和库由 IBM 数据服务器安装程序安装。

限制

DB2 ACS 支持 IBM 数据服务器支持的一部分硬件和操作系统。有关 DB2 ACS 支持的硬件和操作系统的列表，请参阅：第 302 页的『DB2 高级副本服务（ACS）支持的操作系统和硬件』。

开始前

安装 ACS 之前，必须已安装了下列库：

在 AIX 上：

- `ln -s /opt/freeware/lib/powerpc-ibm-aix5.3.0/libgcc_s.a /usr/lib/libgcc_s.a`

在 Red Hat Enterprise Linux 上：

- `ln -s libssl.so.0.9.7xxx libssl.so.0.9.7`
- `ln -s libcrypto.so.0.9.7xxx libcrypto.so.0.9.7`
- `ln -s libssl.so.0.9.7xxx libssl.so`
- `ln -s libssl.so.0.9.7xxx libssl.so.0`

过程

1. 安装 IBM 数据服务器。
2. 在 TCP/IP 服务文件中为 DB2 ACS 代理程序添加端口。例如：

```
db2acs 5400/tcp # DB2 ACS service port
```

下一步任务

在安装 DB2 ACS 之后，必须激活 DB2 ACS 和配置 DB2 ACS，然后才可以执行快照备份操作。

有关设置和使用 DB2 高级副本服务 (ACS) 的详细指示信息, 请参阅高级副本服务的 Tivoli 文档, 网址为: <http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html>

激活 DB2 高级副本服务 (ACS)

在可以使用 DB2 高级副本服务 (ACS) 来为给定数据库管理器实例执行快照备份之前, 必须对该实例激活 DB2 ACS 功能。通过运行脚本来激活 DB2 ACS。

开始前

必须执行下列任务, 然后才可以激活 DB2 ACS:

1. 安装 DB2 ACS
2. 创建您将 DB2 ACS 与之配合使用的数据库管理器实例。

关于此任务

在创建数据库管理器实例期间以及升级 IBM 数据服务器时, 数据库管理器自动调用 `setup.sh` 来激活 DB2 ACS 功能。

还可以手动激活 DB2 ACS。

过程

要手动激活 DB2 ACS, 请作为具有 root 用户权限的用户运行带有适当参数的 `setup.sh` 来激活 DB2 ACS。

有关 `setup.sh` 的更多信息, 请参阅: 第 261 页的『DB2 高级副本服务 (ACS) 设置脚本 `setup.sh`』。

结果

运行 `setup.sh` 脚本的一个重要结果就是验证了 `sqllib/acs` 目录中的 DB2 ACS 可执行文件的所有权和许可权。

下一步任务

在激活 DB2 ACS 之后, 必须配置 DB2 ACS, 然后才可以执行快照备份操作。

有关设置和使用 DB2 高级副本服务 (ACS) 的详细指示信息, 请参阅高级副本服务的 Tivoli 文档, 网址为: <http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html>

配置 DB2 高级副本服务 (ACS)

在可以使用 DB2 高级副本服务 (ACS) 来执行快照备份之前, 必须配置 DB2 ACS。可以使用配置文件来配置 DB2 ACS。

开始前

必须先执行下列任务才能配置 DB2 ACS:

1. 安装 DB2 ACS
2. 创建您将 DB2 ACS 与之配合使用的数据库管理器实例。

3. 激活 DB2 ACS

过程

在不带任何参数的情况下运行 `sqllib/acs` 目录中的 `setup.sh` 脚本。这将引导您通过基于文本的交互式向导来配置 DB2 ACS。该向导将在机器上创建一个配置概要文件并修改 `/etc/initab`，以触发 DB2 ACS 守护程序的启动。

以下是 `setup.sh` 向导的样本输出：

```
./setup.sh
Do you have a full TSM license to enable all features of TSM for ACS ?[y/n]

n

***** Profile parameters for section GLOBAL: *****
ACS_DIR [/home/krodger/sqllib/acs ]
ACSD [localhost 57328 ]
TRACE [NO ]

***** Profile parameters for section ACSD: *****
ACS_REPOSITORY *mandatory parameter* /home/krodger/acsrepository

***** Profile parameters for section CLIENT: *****
MAX_VERSIONS [ADAPTIVE ] 2
LVM_FREEZE_THAW [YES ]
DEVICE_CLASS [STANDARD ]

***** Profile parameters for section STANDARD: *****
COPYSERVICES_HARDWARE_TYPE *mandatory parameter*
NAS_NSERIES COPYSERVICES_PRIMARY_SERVERNAME *mandatory parameter* fas960a
COPYSERVICES_USERNAME [superuser_] root

=====

The profile has been successfully created.
Do you want to continue by specifying passwords for the defined devices? [y/n]

y

Please specify the passwords for the following profile sections:
STANDARD
master

Creating password file at /home/krodger/sqllib/acs/shared/pwd.acsd.
A copy of this file needs to be available to all components that connect to acsd.

BK11555I: Profile successfully created. Performing additional checks.
Make sure to restart all ACS components to reload the profile.
```

结果

在配置 DB2 ACS 之后，可以执行快照备份操作。

有关设置和使用 DB2 高级副本服务（ACS）的详细指示信息，请参阅高级副本服务的 Tivoli 文档，网址为：<http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html>

配置 DB2 高级副本服务（ACS）目录

当创建新的数据库管理器实例时，会在新的实例 `sqllib` 目录中创建名为 `acs` 的目录。DB2 高级副本服务（ACS）使用此 `acs` 目录来存储配置文件，例如目标卷控制文件和恢复对象的共享库。目前对改变或配置此 `acs` 目录的方法有一些限制。

1. 在任何 DB2 ACS 或快照备份操作中，均不得涉及 acs 目录。
2. 在使用 IBM Tivoli Storage Manager (TSM) 的所有数据库分区和快照备份的备份系统上，acs 目录可以是 NFS 导出的和 NFS 共享的。

DB2 高级副本服务 (ACS) 设置脚本 setup.sh

setup.sh 脚本激活并配置 DB2 高级副本服务 (ASC)。

位置

脚本 setup.sh 位于 sqllib/acs 目录中。

语法

以下是 setup.sh 的语法：

```
usage: setup.sh -a <action>
             -d <DB2_Instance_Directory>
             -u <Instance_user_ID_name>
             -g <Instance_primary_group_name>
```

其中 action 可以是下列其中一项：

- 启动
- 停止
- 查询
- 启用
- 禁用

用法

在创建数据库管理器实例期间以及升级 IBM 数据服务器时，数据库管理器自动调用 setup.sh 来激活 DB2 ACS 功能。

另外，您也可以手动调用 setup.sh 脚本：

激活 DB2 ACS

您可以作为具有 root 用户权限的用户运行带有上述参数的 setup.sh 来激活 DB2 ACS。

配置 DB2 ACS

您可以通过运行不带任何参数的 setup.sh 来配置 DB2 ACS。如果运行不带任何参数的 setup.sh，那么将会有向导引导您完成 DB2 ACS 的配置过程。

运行 setup.sh 脚本的一个重要结果就是验证了 sqllib/acs 目录中的 DB2 ACS 可执行文件的所有权和许可权。

DB2 高级副本服务 (ACS) API

DB2 高级副本服务 (ACS) 应用程序编程接口 (API) 定义了一组功能，数据库管理器可使用它们来与存储硬件通信，从而执行快照备份操作。

要执行快照备份与复原操作，需要适用于存储设备的 DB2 ACS API 驱动程序。集成到 IBM 数据服务器的是适用于下列存储硬件的 DB2 ACS API 驱动程序：

- IBM TotalStorage SAN Volume Controller
- IBM 系统存储器 DS6000
- IBM 系统存储器 DS8000
- IBM 系统存储器 N 系列
- NetApp V 系列
- NetApp FAS 系列

DB2 高级副本服务 (ACS) API 函数

数据库管理器通过 DB2 ACS API 函数来向存储硬件通知 DB2 ACS 请求。

db2ACSQueryApiVersion - 返回 DB2 高级副本服务 (ACS) API 的当前版本

返回 DB2 高级副本服务 (ACS) API 的当前版本。

API 包含文件

db2ACSApi.h

API 和数据结构语法

```
db2ACS_Version db2ACSQueryApiVersion();
```

参数

无。

使用说明

可能的返回值:

- DB2ACS_API_VERSION1
- DB2ACS_API_VERSION_UNKNOWN

db2ACSInitialize - 初始化 DB2 高级副本服务 (ACS) 会话

初始化新的 DB2 高级副本服务 (ACS) 会话。此调用在数据库管理器的 DB2 ACS 库和存储硬件的 DB2 ACS API 驱动程序之间建立通信。

包含文件

db2ACSApi.h

语法和数据结构

```
/* =====
 * Session Initialization
 * ===== */
db2ACS_RC db2ACSInitialize(
    db2ACS_CB          * pControlBlock,
    db2ACS_ReturnCode * pRC );
```

参数

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在调用 db2ACSInitialize() 之前，数据库管理器填充下列字段：

```
pControlBlock->session
pControlBlock->options
```

在返回之前，DB2 ACS API 驱动程序填充下列字段：

```
pControlBlock->handle
pControlBlock->vendorInfo
```

PRC 数据类型：db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息，其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前，DB2 ACS API 驱动程序填充 **PRC** 的字段。

返回码

表 11. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INIT_FAILED	数据库管理器尝试初始化 DB2 ACS 会话，但初始化失败。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_COMM_ERROR	存在与诸如磁带机之类的存储设备的通信错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_NO_DEV_AVAIL	当前不存在诸如磁带机之类的存储设备可用。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBEGINQUERY() 的调用成功了，那么数据库管理器可以调用 db2ACSENDQUERY()
- 如果先前对 db2ACSBEGINOPERATION() 的调用成功了，那么数据库管理器可以调用 db2ACSENDOPERATION()
- 如果先前对 db2ACSINITIALIZE() 的调用成功了，那么数据库管理器可以调用 db2ACSTERMEDIATE()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

在数据库管理器可以进行任何 DB2 ACS API 调用（除了调用 db2ACSQUERYAPIVERSION() 之外）之前，数据库管理器必须调用 db2ACSINITIALIZE()。一旦数据库管理器通过调用 db2ACSINITIALIZE() 建立了 DB2 ACS 会话，该数据库管理器就可以执行 DB2 ACS 查询、读取、写入或删除操作的任何组合。通过调用 db2ACSTERMEDIATE()，数据库管理器可以终止 DB2 ACS 会话。

db2ACSTERMEDIATE - 终止 DB2 高级副本服务 (ACS) 会话

终止 DB2 高级副本服务 (ACS) 会话。

包含文件

db2ACSAPI.H

语法和数据结构

```
/* =====
 * Session Termination
 * ===== */
db2ACS_RC db2ACSTERMEDIATE(
                db2ACS_CB          * pControlBlock,
                db2ACS_ReturnCode * pRC );
```

参数

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSINITIALIZE() 之前，数据库管理器为此参数分配内存。在调用 db2ACSTERMEDIATE() 之后，数据库管理器负责释放此内存。

在调用 db2ACSTERMEDIATE() 之前，数据库管理器填充下列字段：

pControlBlock->options

DB2 ACS API 驱动程序可能在 pControlBlock->vendorInfo.vendorCB 中使内存无效或释放内存。

pRC

数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息，其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前，DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 12. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	释放为此会话分配的所有内存并将其终止。
DB2ACS_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

DB2 ACS API 驱动程序应该在 db2ACSTerminate() 中释放驱动程序分配给 DB2 ACS 会话的所有内存。

不论 db2ACSTerminate() 是否成功完成，在事先未调用 db2ACSInitialize() 的情况下，数据库管理器均无法再次对此 DB2 ACS 会话调用 DB2 ACS 函数。

db2ACSPrepare - 准备执行快照备份操作。

当执行快照备份时，数据库管理器会暂挂数据库。db2ACSPrepare() 执行所有步骤来准备进行快照备份操作，直至（但不包括）数据库管理器暂挂数据库这一刻。

包含文件

db2ACSApi.h

语法和数据结构

```

/* =====
 * Prepare
 * ===== */
db2ACS_RC db2ACSPrepare(
    db2ACS_GroupList * pGroupList,
    db2ACS_CB * pControlBlock,
    db2ACS_ReturnCode * pRC );

```

参数

pGroupList

数据类型: db2ACS_GroupList *

db2ACS_GroupList 包含在快照备份操作中要包含的组的列表。

如果 **pGroupList** 为 NULL, 那么快照备份操作中将包含所有组 (路径)。

如果 **pGroupList** 不是 NULL:

- **pGroupList** 包含快照备份操作中要包括的组 (路径) 的列表。
- 数据库管理器负责分配和释放 **pGroupList** 的内存。
- 数据库管理器填充下列字段, 然后将 **pGroupList** 传递到 db2ACSPrepare():

```
pGroupList->numGroupID  
pGroupList->id
```

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSPrepare() 之前, 数据库管理器填充下列字段:

```
pControlBlock->handle  
pControlBlock->vendorInfo  
pControlBlock->options
```

pRC

数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息, 其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存, 并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前, DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 13. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

表 13. 返回码 (续)

返回码	描述	注意
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

如果 db2ACSPrepare() 成功执行，那么数据库管理器将暂挂数据库，然后才调用 db2ACSSnapshot()。

db2ACSBeginOperation - 开始 DB2 高级副本服务 (ACS) 操作。

开始 DB2 高级副本服务 (ACS) 操作。

包含文件

db2ACSApi.h

语法和数据结构

```

/* =====
 * Operation Begin
 *
 * A valid ACS operation is specified by passing an ObjectType OR'd with one of
 * the following Operations, such as:
 *
 *      (DB2ACS_OP_CREATE | DB2ACS_OBJTYPE_SNAPSHOT)
 * ===== */
db2ACS_RC db2ACSBeginOperation(
    db2ACS_Operation    operation,
    db2ACS_CB          * pControlBlock,
    db2ACS_ReturnCode  * pRC );

```

参数

operation

数据类型: db2ACS_Operation。

operation 是位屏蔽，指示要开始的 DB2 ACS 操作和所涉及对象的类型。

操作类型:

DB2ACS_OP_CREATE
DB2ACS_OP_READ
DB2ACS_OP_DELETE

对象类型:

DB2ACS_OBJTYPE_BACKUP
DB2ACS_OBJTYPE_LOG
DB2ACS_OBJTYPE_LOADCOPY
DB2ACS_OBJTYPE_SNAPSHOT

例如: (DB2ACS_OP_CREATE | DB2ACS_OBJTYPE_SNAPSHOT) 或 (DB2ACS_OP_DELETE | DB2ACS_OBJTYPE_LOADCOPY)。

数据库管理器将 **operation** 传递到 db2ACSBeginOperation() 函数调用。

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSBeginOperation() 之前, 数据库管理器填充下列字段:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

如果 **operation** 是 DB2ACS_OP_CREATE 或 DB2ACS_OP_READ, 那么数据库管理器也会填充下列字段:

pControlBlock->operation

pControlBlock->operation 中包含的信息仅在特定的 DB2 ACS 操作的上下文中才有效。pControlBlock->operation 将在 db2ACSBeginOperation() 期间进行设置, 并且在 db2ACSEndOperation() 返回之前将保持不变。数据库管理器或 DB2 ACS API 驱动程序均不应该引用 DB2 ACS 操作范围外的 pControlBlock->operation。

pRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息, 其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存, 并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前, DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 14. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_OPTIONS	数据库管理器指定了无效选项。	

表 14. 返回码 (续)

返回码	描述	注意
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

无。

db2ACSEndOperation – 结束 DB2 高级副本服务 (ACS) 操作。

结束 DB2 高级副本服务 (ACS) 操作。

包含文件

db2ACSApi.h

语法和数据结构

```

/* =====
 * Operation End
 * ===== */
db2ACS_RC db2ACSEndOperation(
    db2ACS_EndAction    endAction,
    db2ACS_CB           * pControlBlock,
    db2ACS_ReturnCode   * pRC );

```

参数

endAction

数据类型：db2ACS_EndAction。

endAction 是位屏蔽，指示 DB2 ACS API 驱动程序应如何结束 DB2 ACS 操作。

值：

DB2ACS_END_COMMIT
DB2ACS_END_ABORT

数据库管理器将 **endAction** 传递到 db2ACSEndOperation() 函数调用。

pControlBlock

数据类型: db2ACS_CB

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSEndOperation() 之前, 数据库管理器填充下列字段:

```
pControlBlock->handle  
pControlBlock->vendorInfo  
pControlBlock->options
```

pRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息, 其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存, 并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前, DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 15. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_COMMIT_FAILED	DB2 ACS API 驱动程序无法落实事务。	
DB2ACS_RC_ABORT_FAILED	数据库管理器尝试中止 DB2 ACS 操作, 但中止尝试失败。	

如果 DB2 ACS API 驱动程序遇到错误, 那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作:

- 如果先前对 db2ACSBeginQuery() 的调用成功了, 那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了, 那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了, 那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息, 请参阅主题: 第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

如果数据库管理器将 DB2ACS_END_ABORT 作为 **endAction** 参数来传递, 那么结果应该是删除快照备份对象。

db2ACSBEGINQUERY - 开始有关快照备份对象的查询

开始有关可用于复原操作的快照备份对象的 DB2 高级副本服务 (ACS) 查询操作。

包含文件

db2ACSApi.h

语法和数据结构

```
db2ACS_RC db2ACSBEGINQUERY(  
    db2ACS_QueryInput * pQueryInput,  
    db2ACS_CB * pControlBlock,  
    db2ACS_ReturnCode * pRC );
```

参数

pQueryInput

数据类型: db2ACS_QueryInput *

db2ACS_QueryInput 具有与 db2ACS_ObjectInfo 相同的字段。db2ACS_ObjectInfo 包含有关使用 DB2 高级副本服务 (ACS) API 创建的对象的信息。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在调用 db2ACSBEGINQUERY() 之前，数据库管理器填充 pQueryInput 的字段。

DB2 ACS API 驱动程序必须支持在查询中使用下列通配符:

- 字符串字段中的 DB2ACS_WILDCARD
- 用于数据库分区字段的 DB2ACS_ANY_PARTITIONNUM
- 用于 32 位无符号的整数 (Uint32) 字段的 DB2ACS_ANY_UINT32

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSBEGINQUERY() 之前，数据库管理器填充下列字段:

```
pControlBlock->handle  
pControlBlock->vendorInfo  
pControlBlock->options
```

pRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息，其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前，DB2 ACS API 驱动程序填充 pRC 的字段。

返回码

表 16. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	

表 16. 返回码 (续)

返回码	描述	注意
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

db2ACSBeginQuery() 不会返回任何查询数据。

db2ACSGetNextObject - 列示下一个可用于复原的快照备份对象

返回可用于复原操作的快照备份对象列表中的下一项。

包含文件

db2ACSApi.h

语法和数据结构

```
db2ACS_RC db2ACSGetNextObject(
    db2ACS_QueryOutput * pQueryOutput,
    db2ACS_CB          * pControlBlock,
    db2ACS_ReturnCode * pRC );
```

参数

pQueryOutput

数据类型: db2ACS_QueryOutput *

db2ACS_QueryOutput 包含有关快照备份对象的查询结果信息。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前，DB2 ACS API 驱动程序填充 **pQueryOutput** 的字段。

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSGetNextObject() 之前，数据库管理器填充下列字段:

```
pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options
```

pRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息，其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前，DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 17. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_OBJ_NOT_FOUND	DB2 ACS API 驱动程序找不到数据库管理器指定的快照备份对象。	函数调用并未失败，但不存在与传递到 db2ACSBeginQuery() 的条件相符的快照备份对象。
DB2ACS_RC_END_OF_DATA	DB2 ACS API 驱动程序找不到更多的快照备份对象。	函数调用并未失败，但不存在更多与传递到 db2ACSBeginQuery() 的条件相符的快照备份对象。

表 17. 返回码 (续)

返回码	描述	注意
DB2ACS_RC_MORE_DATA	有更多的数据要从存储位置传送到数据库管理器。	将返回有关与传递到 db2ACSBeginQuery() 的条件相符的快照备份对象的信息，并且存在更多与传递到 db2ACSBeginQuery() 的条件相符的快照备份对象。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

数据库管理器必须先调用 db2ACSBeginQuery()，然后才调用 db2ACSGetNextObject()。数据库管理器在传递到 db2ACSBeginQuery() 的 db2ACS_QueryInput 参数中指定搜索条件。

db2ACSGetNextObject() 返回有关与传递到 db2ACSBeginQuery() 的搜索条件相符的一个快照备份对象的信息。如果 db2ACSGetNextObject() 返回 DB2ACS_RC_MORE_DATA，那么数据库管理器可以再次调用 db2ACSGetNextObject()，以接收有关与搜索条件相符的另一快照备份对象的信息。如果 db2ACSGetNextObject() 返回 DB2ACS_RC_END_OF_DATA，那么不存在更多与搜索条件相符的快照备份对象。

db2ACSEndQuery - 结束有关快照备份对象的查询

数据库管理器使用 DB2 高级副本服务 (ACS) API 函数 db2ACSBeginQuery() 和 db2ACSGetNextObject() 来查询有关可用于复原操作的快照备份对象的信息。db2ACSEndQuery() 终止该 DB2 ACS 查询会话。

包含文件

db2ACSApi.h

语法和数据结构

```
db2ACS_RC db2ACSEndQuery(
    db2ACS_CB          * pControlBlock,
    db2ACS_ReturnCode * pRC );
```

参数

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSEndQuery() 之前，数据库管理器填充下列字段：

```
pControlBlock->handle  
pControlBlock->vendorInfo  
pControlBlock->options
```

pRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息，其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前，DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 18. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

在没有再次先调用 `db2ACSBeginQuery()` 的情况下，数据库管理器无法再次对此 DB2 ACS 会话调用 `db2ACSGetNextObject()`。

db2ACSSnapshot - 执行 DB2 高级副本服务 (ACS) 操作

执行 DB2 高级副本服务 (ACS) 操作。

包含文件

`db2ACSApi.h`

语法和数据结构

```
typedef union db2ACS_ReadList
{
    db2ACS_GroupList      group;
} db2ACS_ReadList;

db2ACS_RC db2ACSSnapshot(
    db2ACS_Action          action,
    db2ACS_ObjectID       objectID,
    db2ACS_ReadList       * pReadList,
    db2ACS_CB             * pControlBlock,
    db2ACS_ReturnCode     * pRC );
```

参数

action 数据类型: `db2ACS_Action`

要执行的 DB2 ACS 操作的类型。值:

```
DB2ACS_ACTION_WRITE
DB2ACS_ACTION_READ_BY_OBJECT
DB2ACS_ACTION_READ_BY_GROUP
```

数据库管理器将 **action** 传递到 `db2ACSSnapshot()`。

objectID

数据类型: `db2ACS_ObjectID`

`db2ACS_ObjectID` 是由查询返回到存储库的每个存储对象的唯一标识。`db2ACS_ObjectID` 被保证是唯一的且仅在单一 DB2 ACS 会话时间框架中是持久存在的。

如果数据库管理器在调用 `db2ACSBeginOperation()` 时将 `DB2ACS_OP_READ` 或 `DB2ACS_OP_DELETE` 指定为 **operation**，那么数据库管理器会将 **objectID** 的值传递到 `db2ACSSnapshot()`。

pReadList

数据类型: `db2ACS_ReadList *`

`db2ACS_ReadList` 包含组列表。

仅在 **action** 为 `DB2ACS_ACTION_READ_BY_GROUP` 时，才使用 **pReadList**。

如果 **action** 是 `DB2ACS_ACTION_READ_BY_GROUP`，那么数据库管理器负责分配内存以及填充 **pReadList** 的字段，然后调用 `db2ACSSnapshot()` 和释放 **pReadList** 的内存。

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSSnapshot() 之前, 数据库管理器填充下列字段:

```
pControlBlock->handle  
pControlBlock->vendorInfo  
pControlBlock->options
```

pRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息, 其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存, 并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前, DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 19. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误, 那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作:

- 如果先前对 db2ACSBeginQuery() 的调用成功了, 那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了, 那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了, 那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息, 请参阅主题: 第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

数据库管理器调用 `db2ACSBEGINoperation()`，然后调用 `db2ACSPartition()`、`db2ACSPrepare()` 和 `db2ACSSnapshot()`。数据库管理器在对 `db2ACSBEGINoperation()` 的调用中的 **operation** 参数中指定 DB2 ACS API 驱动程序应执行的 DB2 ACS 操作的类型。

db2ACSPartition - 将数据库分区的目标数据分组在一起

将组标识与数据库管理器列示的每个路径相关联，使这些路径从属于数据库分区。

包含文件

`db2ACSApi.h`

语法和数据结构

```
/* =====  
 * Partition  
 * ===== */  
db2ACS_RC db2ACSPartition(  
    db2ACS_PathList      * pPathList,  
    db2ACS_CreateObjectInfo * pCreateObjInfo,  
    db2ACS_CB            * pControlBlock,  
    db2ACS_ReturnCode    * pRC );
```

参数

pPathList

数据类型: `db2ACS_PathList`

`db2ACS_PathList` 包含数据库路径列表，包括有关特定于 DB2 ACS 操作的每个这些路径的一些额外信息。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

`db2ACS_PathList` 结构的 **entry** 字段是一组类型为 `db2ACS_PathEntry` 的元素。`db2ACS_PathEntry` 包含有关数据库路径的信息。

在调用 `db2ACSPartition` 之前，数据库管理器填充 **pPathList** 中每个 `db2ACS_PathEntry` 条目的下列字段：

- **path**
- **type**
- **toBeExcluded**

DB2 ACS API 驱动程序为属于该数据库分区的数据管理器标识的每个路径提供了组标识。在返回之前，DB2 ACS API 驱动程序填充 **pPathList** 中每个 `db2ACS_PathEntry` 的 **groupID** 字段。

pCreateObjInfo

数据类型: `db2ACS_CreateObjectInfo`

`db2ACS_CreateObjectInfo` 包含有关 DB2 ACS 备份对象创建的信息。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在调用 `db2ACSPartition` 之前，数据库管理器填充 **pCreateObjInfo** 的字段。

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSPartition() 之前, 数据库管理器填充下列字段:

```
pControlBlock->handle  
pControlBlock->vendorInfo  
pControlBlock->options
```

PRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息, 其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存, 并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前, DB2 ACS API 驱动程序填充 **PRC** 的字段。

返回码

表 20. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INIT_FAILED	数据库管理器尝试初始化 DB2 ACS 会话, 但初始化失败。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_OBJ_OUT_OF_SCOPE	数据库管理器尝试对 DB2 ACS API 驱动程序未管理的恢复对象执行 DB2 ACS 操作。	

如果 DB2 ACS API 驱动程序遇到错误, 那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作:

- 如果先前对 db2ACSBeginQuery() 的调用成功了, 那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了, 那么数据库管理器可以调用 db2ACSEndOperation()

- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

DB2 高级副本服务自动处理单一数据库分区上的数据。即，一个数据库分区的数据将一起备份或复原，并且独立于其他数据库分区 - 即使当该操作是涉及多个数据库分区的数据的一部分时也是如此。db2ACSPartition 将单一数据库分区的数据路径信息分组在一起。

数据库管理器调用 db2ACSPartition，然后调用 db2ACSSnapshot。数据库管理器通过 pPathList 参数来列示与此数据库分区相关联的所有路径。通过在 pReadList 参数中指定传递到 db2ACSSnapshot 的部分路径，数据库管理器可以对这一部分列示在 pPathList 中的路径执行 DB2 ACS 操作。

db2ACSVerify - 验证 DB2 高级副本服务 (ACS) 操作是否已成功完成

验证 DB2 高级副本服务 (ACS) 操作是否成功

包含文件

db2ACSApi.h

语法和数据结构

```
/* =====
 * Verify
 * ===== */
db2ACS_RC db2ACSVerify(
    db2ACS_PostObjectInfo * pPostObjInfo,
    db2ACS_CB * pControlBlock,
    db2ACS_ReturnCode * pRC );
```

参数

pPostObjInfo

数据类型: db2ACS_PostObjectInfo

db2ACS_DB2ID 是一组数据，在快照备份对象创建时无法知道，但必须在对象存储库中对其进行维护。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在调用 db2ACSVerify 之前，数据库管理器填充 pPostObjInfo 的字段。在 DB2 ACS 操作之后，pPostObjInfo 包含相关的信息。例如，在成功的快照备份之后，pPostObjInfo 可能包含第一个活动日志文件。如果在 DB2 ACS 操作后没有相关数据，那么数据库管理器会将 pPostObjInfo 设置为 NULL。

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSVerify() 之前，数据库管理器填充下列字段:

pControlBlock->handle
 pControlBlock->vendorInfo
 pControlBlock->options

pRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息，其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前，DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 21. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

如果 db2ACSVerify 返回“快照备份操作成功”，这表示快照备份生成的恢复对象可用于复原操作。

db2ACSDelete - 删除使用 DB2 高级副本服务 (ACS) 创建的恢复对象

删除使用 DB2 高级副本服务 (ACS) 创建的恢复对象

包含文件

db2ACSApi.h

语法和数据结构

```
/* =====  
 * Delete  
 * ===== */  
db2ACS_RC db2ACSDelete(  
    db2ACS_ObjectID      objectID,  
    db2ACS_CB            * pControlBlock,  
    db2ACS_ReturnCode    * pRC );
```

参数

objectID

数据类型: db2ACS_ObjectID

db2ACS_ObjectID 是由查询返回到存储库的每个存储对象的唯一标识。db2ACS_ObjectID 被保证是唯一的且仅在单一 DB2 ACS 会话时间框架中是持久存在的。

数据库管理器可以使用 db2ACSQuery() 来获取有效的 **objectID** 以传递到 db2ACSDelete()。

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSDelete() 之前, 数据库管理器填充下列字段:

```
pControlBlock->handle  
pControlBlock->vendorInfo  
pControlBlock->options
```

pRC 数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息, 其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存, 并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前, DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 22. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	删除了指定的对象。无法对该对象执行其他 DB2 ACS 操作。

表 22. 返回码 (续)

返回码	描述	注意
DB2ACS_RC_DELETE_FAILED	DB2 ACS API 驱动程序无法成功地删除数据库管理器指定的快照备份对象。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_OBJ_NOT_FOUND	DB2 ACS API 驱动程序找不到数据库管理器指定的快照备份对象。	

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

当数据库管理器调用 db2ACSDelete 时，DB2 ACS API 驱动程序删除由 **objectID** 标识的恢复对象。

当用户调用带 DELETE 参数的 db2acsutil 时，数据库管理器将调用 db2ACSDelete。

db2ACSStoreMetaData - 存储使用 DB2 高级副本服务 (ACS) 生成的恢复对象的元数据

存储有关使用 DB2 高级副本服务 (ACS) 创建的恢复对象的元数据

包含文件

db2ACSApi.h

语法和数据结构

```
db2ACS_RC db2ACSStoreMetaData(
    db2ACS_MetaData          * pMetaData,
    db2ACS_CB                * pControlBlock,
    db2ACS_ReturnCode        * pRC );
```

参数

pMetaData

数据类型: db2ACS_MetaData

db2ACS_MetaData 存储快照备份元数据。

数据库管理器为此参数分配内存, 并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

存储在 **pMetaData** 的 **data** 字段中的元数据是数据库管理器固有的, 可能会随着时间的推移而发生变化, 因此 DB2 ACS API 驱动程序只需将此数据视为二进制流。

pControlBlock

数据类型: db2ACS_CB *

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 db2ACSStoreMetaData() 之前, 数据库管理器填充下列字段:

```
pControlBlock->handle  
pControlBlock->vendorInfo  
pControlBlock->options
```

pRC

数据类型: db2ACS_ReturnCode *

db2ACS_ReturnCode 包含诊断信息, 其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存, 并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前, DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 23. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作：

- 如果先前对 db2ACSBeginQuery() 的调用成功了，那么数据库管理器可以调用 db2ACSEndQuery()
- 如果先前对 db2ACSBeginOperation() 的调用成功了，那么数据库管理器可以调用 db2ACSEndOperation()
- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务（ACS）API 返回码』。

使用说明

快照备份操作由若干 DB2 ACS API 函数调用组成，例如：db2ACSInitialize、db2ACSBeginOperation、db2ACSPrepare 和 db2ACSSnapshot。db2ACSStoreMetaData 也是整体操作的一部分。为了使快照备份操作成功，包括 db2ACSStoreMetaData 在内的所有这些 API 调用均必须成功。如果 db2ACSStoreMetaData 失败，那么由 DB2 ACS 备份操作生成的恢复对象不可使用。

db2ACSRetrieveMetaData - 检索有关使用 DB2 高级副本服务（ACS）生成的恢复对象的元数据

检索有关使用 DB2 高级副本服务（ACS）创建的恢复对象的元数据

包含文件

db2ACSApi.h

语法和数据结构

```
db2ACS_RC db2ACSRetrieveMetaData(  
    db2ACS_MetaData          * pMetaData,  
    db2ACS_ObjectID         objectID,  
    db2ACS_CB               * pControlBlock,  
    db2ACS_ReturnCode       * pRC );
```

参数

pMetaData

数据类型：db2ACS_MetaData

db2ACS_MetaData 存储快照备份元数据。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

存储在 **pMetaData** 的 **data** 字段中的元数据是数据库管理器固有的，可能会随着时间的推移而发生变化，因此 DB2 ACS API 驱动程序只需将此数据视为二进制流。

objectID

数据类型：db2ACS_ObjectID

db2ACS_ObjectID 是由查询返回到存储库的每个存储对象的唯一标识。db2ACS_ObjectID 被保证是唯一的且仅在单一 DB2 ACS 会话时间框架中是持久存在的。

数据库管理器可以使用 `db2ACSQuery()` 来获取有效的 **objectID**，以便传递到 `db2ACSRetrieveMetaData()`。

pControlBlock

数据类型: `db2ACS_CB *`

`db2ACS_CB` 包含初始化和终止 DB2 ACS 会话所需的基本信息。

在调用 `db2ACSRetrieveMetaData()` 之前，数据库管理器填充下列字段:

```
pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options
```

pRC 数据类型: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` 包含诊断信息，其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 `db2ACS_ReturnCode` 参数的内容将记录在数据库管理器诊断日志中。

数据库管理器为此参数分配内存，并将指向例示对象的指针传递到该函数。数据库管理器负责释放此内存。

在返回之前，DB2 ACS API 驱动程序填充 **pRC** 的字段。

返回码

表 24. 返回码

返回码	描述	注意
DB2ACS_RC_OK	操作已成功。	
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。
DB2ACS_RC_OBJ_NOT_FOUND	DB2 ACS API 驱动程序找不到数据库管理器指定的快照备份对象。	DB2 ACS API 驱动程序遇到错误。数据库管理器无法使用 DB2 ACS API 会话。

如果 DB2 ACS API 驱动程序遇到错误，那么该驱动程序可能会中止 DB2 ACS 操作。DB2 ACS 会话不能用于除以下各项之外的任何操作:

- 如果先前对 `db2ACSBeginQuery()` 的调用成功了，那么数据库管理器可以调用 `db2ACSEndQuery()`
- 如果先前对 `db2ACSBeginOperation()` 的调用成功了，那么数据库管理器可以调用 `db2ACSEndOperation()`

- 如果先前对 db2ACSInitialize() 的调用成功了，那么数据库管理器可以调用 db2ACSTerminate()

有关 DB2 ACS API 返回码的更多信息，请参阅主题：第 301 页的『DB2 高级副本服务 (ACS) API 返回码』。

使用说明

无。

DB2 高级副本服务 (ACS) API 数据结构

要调用 DB2 高级副本服务 (ACS) API 函数，必须使用 DB2 ACS API 数据结构。

db2ACS_BackupDetails DB2 高级副本服务 (ACS) API 数据结构

db2ACS_BackupDetails 包含有关快照备份操作的信息。

```
/* ----- */
typedef struct db2ACS_BackupDetails
{
    /* A traditional DB2 backup can consist of multiple objects (logical tapes),
     * where each object is uniquely numbered with a non-zero natural number.
     * ----- */
    db2UInt32          sequenceNum;

    char               imageTimestamp[SQLU_TIME_STAMP_LEN + 1];
} db2ACS_BackupDetails;
```

sequenceNum

数据类型: db2UInt32。

通过其唯一的编号来标识备份对象。

imageTimestamp

数据类型: char[]。

长度 SQLU_TIME_STAMP_LEN + 1 的字符串。

db2ACS_CB DB2 高级副本服务 (ACS) API 数据结构

db2ACS_CB 包含初始化和终止 DB2 ACS 会话所需的基本信息。

```
/* =====
 * DB2 Backup Adapter Control Block
 * ===== */
typedef struct db2ACS_CB
{
    /* Output: Handle value for this session.
     * ----- */
    db2UInt32          handle;
    db2ACS_VendorInfo vendorInfo;

    /* Input fields and parameters.
     * ----- */
    db2ACS_SessionInfo session;
    db2ACS_Options      options;

    /* Operation info is optional, possibly NULL, and is only ever valid
     * within the context of an operation (from call to BeginOperation() until
     * the EndOperation() call returns).
     *
     * The operation info will be present during creation or read operations
     */
}
```

```

    * of snapshot and backup objects.
    * ----- */
    db2ACS_OperationInfo    * operation;
} db2ACS_CB;

```

handle

数据类型: db2UInt32。

引用 DB2 ACS 会话的句柄。

vendorInfo

数据类型: db2ACS_VendorInfo。

db2ACS_VendorInfo 包含有关 DB2 ACS API 驱动程序的信息。

session

数据类型: db2ACS_SessionInfo。

db2ACS_SessionInfo 包含有关 DB2 ACS 会话的所有信息。

options

数据类型: db2ACS_Options。

db2ACS_Options 指定要用于 DB2 ACS 操作的选项。此字符串的内容特定于 DB2 ACS API 驱动程序。

operation

数据类型: db2ACS_OperationInfo *。

db2ACS_OperationInfo 包含有关快照备份操作的信息。

db2ACS_CreateObjectInfo DB2 高级副本服务 (ACS) API 数据结构

db2ACS_CreateObjectInfo 包含有关 DB2 ACS 备份对象创建的信息。

```

/* =====
 * Object Creation Parameters.
 * ===== */
typedef struct db2ACS_CreateObjectInfo
{
    db2ACS_ObjectInfo    object;
    db2ACS_DB2ID        db2ID;

    /* -----
     * The following fields are optional information for the database manager
     * to use as it sees fit.
     * ----- */

    /* Historically both the size estimate and management
     * class parameters have been used by the TSM client API for traditional
     * backup objects, log archives, and load copies, but not for snapshot
     * backups.
     * ----- */
    db2UInt64            sizeEstimate;
    char                 mgmtClass[DB2ACS_MAX_MGMTCLASS_SZ + 1];

    /* The appOptions is a copy of the iOptions field of flags passed to DB2's
     * db2Backup() API when this execution was initiated. This field will
     * only contain valid data when creating a backup or snapshot object.
     * ----- */
    db2UInt32            appOptions;
} db2ACS_CreateObjectInfo;

```

object 数据类型: db2ACS_ObjectInfo

db2ACS_ObjectInfo 包含有关使用 DB2 高级副本服务 (ACS) API 创建的对象的信息。

db2ID 数据类型: db2ACS_DB2ID

db2ACS_DB2ID 标识了 IBM 数据服务器。

sizeEstimate

数据类型: db2UInt64。

正在创建的备份对象大小的估计值。此估计值不适用于日志归档、装入副本或快照备份对象。

mgmtClass

数据类型: db2ACS_MgmtClass。

长度 db2ACS_MAX_MGMTCLASS_SZ + 1 的字符串。

这不适用于快照备份对象。

appOptions

数据类型: db2UInt32。

将备份选项的副本传递至初始化快照备份的备份命令。

db2ACS_DB2ID DB2 高级副本服务 (ACS) API 数据结构

db2ACS_DB2ID 标识了 IBM 数据服务器。

```
/* =====  
 * DB2 Data Server Identifier  
 * ===== */  
typedef struct db2ACS_DB2ID  
{  
    db2UInt32          version;  
    db2UInt32          release;  
    db2UInt32          level;  
    char               signature[DB2ACS_SIGNATURE_SZ + 1];  
} db2ACS_DB2ID;
```

version

数据类型: db2UInt32。

IBM 数据服务器的版本。例如: 9

release

数据类型: db2UInt32。

IBM 数据服务器的发行版级别。例如: 5

level 数据类型: db2UInt32。

IBM 数据服务器的级别标识。例如: 0

signature

数据类型: char[]。

长度 DB2ACS_SIGNATURE_SZ + 1 的字符串。例如: “SQL09050”

db2ACS_GroupList DB2 高级副本服务 (ACS) API 数据结构

db2ACS_GroupList 包含在快照备份操作中要包含的组的列表。

```
/* =====  
 * Snapshot Group List  
 *
```

```

* This is an array of size 'numGroupIDs', indicating the set of groups that
* are to be included in the snapshot operation.
* ===== */
typedef struct db2ACS_GroupList
{
    db2UInt32          numGroupIDs;
    db2UInt32          * id;
} db2ACS_GroupList;

```

numGroupIDs

数据类型: db2UInt32。

数组 **id** 中组的编号。

id 数据类型: db2UInt32 *。

组标识的数组。已标识的组是快照备份操作中要包含的组（或路径列表）。

db2ACS_LoadcopyDetails DB2 高级副本服务 (ACS) API 数据结构

db2ACS_LoadcopyDetails 包含有关装入副本操作的信息。

```

/* ----- */
typedef struct db2ACS_LoadcopyDetails
{
    /* Just like the BackupDetails, a DB2 load copy can consist of multiple
    * objects (logical tapes), where each object is uniquely numbered with a
    * non-zero natural number.
    * ----- */
    db2UInt32          sequenceNum;

    char               imageTimestamp[SQLU_TIME_STAMP_LEN + 1];
} db2ACS_LoadcopyDetails;

```

sequenceNum

数据类型: db2UInt32。

通过其唯一的编号来标识备份对象。

imageTimestamp

数据类型: char[]。

长度 SQLU_TIME_STAMP_LEN + 1 的字符串

db2ACS_LogDetails DB2 高级副本服务 (ACS) API 数据结构

db2ACS_LogDetails 包含用于标识特定数据库日志文件的信息。

```

/* ----- */
typedef struct db2ACS_LogDetails
{
    db2UInt32          fileID;
    db2UInt32          chainID;
} db2ACS_LogDetails;

```

fileID 数据类型: db2UInt32。

一个编号，它是数据库日志文件的文件名。

chainID

数据类型: db2UInt32。

一个编号，它标识了数据库日志文件 **fileID** 所属的数据库日志文件链。

db2ACS_ObjectInfo DB2 高级副本服务 (ACS) API 数据结构

db2ACS_ObjectInfo 包含有关使用 DB2 高级副本服务 (ACS) API 创建的对象的信息。

```
/* =====  
 * Object Description and Associated Information.  
 *  
 * This structure is used for both input and output, and its contents define  
 * the minimum information that must be recorded about any object created  
 * through this interface.  
 * ===== */  
typedef struct db2ACS_ObjectInfo  
{  
    db2ACS_ObjectType      type;  
    SQL_PDB_NODE_TYPE     dbPartitionNum;  
  
    char                   db[SQL_DBNAME_SZ + 1];  
    char                   instance[DB2ACS_MAX_OWNER_SZ + 1];  
    char                   host[SQL_HOSTNAME_SZ + 1];  
    char                   owner[DB2ACS_MAX_OWNER_SZ + 1];  
  
    union  
    {  
        db2ACS_BackupDetails  backup;  
        db2ACS_LogDetails     log;  
        db2ACS_LoadcopyDetails loadcopy;  
        db2ACS_SnapshotDetails snapshot;  
    } details;  
} db2ACS_ObjectInfo;
```

type 数据类型: db2ACS_ObjectType。

指定快照备份对象类型。值:

DB2ACS_OBJTYPE_ALL
DB2ACS_OBJTYPE_BACKUP
DB2ACS_OBJTYPE_LOG
DB2ACS_OBJTYPE_LOADCOPY
DB2ACS_OBJTYPE_SNAPSHOT

DB2ACS_OBJTYPE_ALL 只能用作查询的过滤器。不存在类型为 0 的对象。

dbPartitionNum

数据类型: SQL_PDB_NODE_TYPE。

此数据库分区的标识。

db 数据类型: char[]。

长度 SQL_DBNAME_SZ + 1 的字符串。

instance

数据类型: char[]。

长度 DB2ACS_MAX_OWNER_SZ + 1 的字符串。

host 数据类型: char[]。

长度 SQL_HOSTNAME_SZ + 1 的字符串。

owner 数据类型: char[]。

长度 DB2ACS_MAX_OWNER_SZ + 1 的字符串。

details

backup

数据类型: db2ACS_BackupDetails

db2ACS_BackupDetails 包含有关快照备份操作的信息。

log

数据类型: db2ACS_LogDetails

db2ACS_LogDetails 包含用于标识特定数据库日志文件的信息。

loadcopy

数据类型: db2ACS_LoadcopyDetails

db2ACS_LoadcopyDetails 包含有关装入副本操作的信息。

snapshot

数据类型: db2ACS_SnapshotDetails

db2ACS_SnapshotDetails 包含有关快照备份操作的信息。

db2ACS_ObjectStatus DB2 高级副本服务 (ACS) API 数据结构

db2ACS_ObjectStatus 包含有关快照备份操作的状态或进度的信息, 或包含有关快照备份对象的状态或可用性的信息。

```
typedef struct db2ACS_ObjectStatus
{
    /* The total and completed bytes refer only to the ACS snapshot backup
    * itself, not to the progress of any offloaded tape backup.
    *
    * A bytesTotal of 0 indicates that the progress could not be determined.
    * ----- */
    db2UInt64          bytesCompleted;
    db2UInt64          bytesTotal;
    db2ACS_ProgressState progressState;
    db2ACS_UsabilityState usabilityState;
} db2ACS_ObjectStatus;
```

bytesCompleted

数据类型: db2UInt64。

已完成的快照备份的数量 (以字节计)。

bytesTotal

数据类型: db2UInt64。

已完成的快照备份的大小 (以字节计)。

progressState

数据类型: db2ACS_ProgressState。

快照备份操作的状态。值:

```
DB2ACS_PSTATE_UNKNOWN
DB2ACS_PSTATE_IN_PROGRESS
DB2ACS_PSTATE_SUCCESSFUL
DB2ACS_PSTATE_FAILED
```

usabilityState

数据类型: db2ACS_UsabilityState。

快照备份对象的状态以及如何可以使用快照备份对象。值:

```

DB2ACS_USTATE_UNKNOWN
DB2ACS_USTATE_LOCALLY_MOUNTABLE
DB2ACS_USTATE_REMOTELY_MOUNTABLE
DB2ACS_USTATE_REPETITIVELY_RESTORABLE
DB2ACS_USTATE_DESTRUCTIVELY_RESTORABLE
DB2ACS_USTATE_SWAP_RESTORABLE
DB2ACS_USTATE_PHYSICAL_PROTECTION
DB2ACS_USTATE_FULL_COPY
DB2ACS_USTATE_DELETED
DB2ACS_USTATE_FORCED_MOUNT
DB2ACS_USTATE_BACKGROUND_MONITOR_PENDING
DB2ACS_USTATE_TAPE_BACKUP_PENDING
DB2ACS_USTATE_TAPE_BACKUP_IN_PROGRESS
DB2ACS_USTATE_TAPE_BACKUP_COMPLETE

```

db2ACS_OperationInfo DB2 高级副本服务 (ACS) API 数据结构

db2ACS_OperationInfo 包含有关快照备份操作的信息。

```

/* =====
 * Operation Info
 *
 * The information contained within this structure is only valid within the
 * context of a particular operation. It will be valid at the time
 * BeginOperation() is called, and will remain unchanged until EndOperation()
 * returns, but must not be referenced outside the scope of an operation.
 * ===== */
typedef struct db2ACS_OperationInfo
{
    db2ACS_SyncMode          syncMode;

    /* List of database and backup operation partitions.
     *
     * For details, refer to the db2ACS_PartitionList definition.
     * ----- */
    db2ACS_PartitionList    * dbPartitionList;
} db2ACS_OperationInfo;

```

syncMode

数据类型: db2ACS_SyncMode.

单独的数据库分区上备份操作之间的同步级别。

值:

DB2ACS_SYNC_NONE

在多个数据库分区上的相关操作之间没有同步。在操作期间, 如果没有利用多个数据库分区之间的任何同步, 那么使用该值。

DB2ACS_SYNC_SERIAL

当在多个数据库分区上执行并发快照备份操作时使用该值。当快照备份操作发出后, 每个数据库分区将暂挂其输入和输出 (IO), 然后以串行方式而不是并发方式恢复数据库分区上的 I/O。

SYNC_PARALLEL

对多个分区并发执行快照操作。一旦快照备份操作涉及的所有数据库分区完成快照备份操作准备, 输入和输出 (IO) 将在所有数据库分区上暂挂。剩余快照备份步骤将在所有涉及的数据数据库分区上并发执行。

dbPartitionList

数据类型: db2ACS_PartitionList *。

db2ACS_PartitionList 包含有关数据库中的数据库分区以及 DB2 ACS 操作中涉及的数据库分区的信息。

db2ACS_Options DB2 高级副本服务 (ACS) API 数据结构

db2ACS_Options 指定要用于 DB2 ACS 操作的选项。此字符串的内容特定于 DB2 ACS API 驱动程序。

```
/* =====  
 * DB2 Backup Adapter User Options  
 * ===== */  
typedef struct db2ACS_Options  
{  
    db2UInt32          size;  
    void              * data;  
} db2ACS_Options;
```

size 数据类型: db2UInt32。

data 的大小 (以字节计)。

data 数据类型: void *。

指向包含选项的内存块的指针。

db2ACS_PartitionEntry DB2 高级副本服务 (ACS) API 数据结构

db2ACS_PartitionEntry 是 db2ACS_PartitionList 的元素。

```
typedef struct db2ACS_PartitionEntry  
{  
    SQL_PDB_NODE_TYPE    num;  
    char                 host[SQL_HOSTNAME_SZ + 1];  
} db2ACS_PartitionEntry;
```

num 数据类型: SQL_PDB_NODE_TYPE。

此数据库分区条目的标识。

host 数据类型: char[]。

长度 SQL_HOSTNAME_SZ + 1 的字符串。

db2ACS_PartitionList DB2 高级副本服务 (ACS) API 数据结构

db2ACS_PartitionList 包含有关数据库中的数据库分区以及 DB2 ACS 操作中涉及的数据库分区的信息。

```
typedef struct db2ACS_PartitionList  
{  
    db2UInt64          numPartsInDB;  
    db2UInt64          numPartsInOperation;  
    db2ACS_PartitionEntry * partition;  
} db2ACS_PartitionList;
```

numPartsInDB

数据类型: db2UInt64。

数据库中的数据库分区数目。

numPartsInOperation

数据类型: db2UInt64。

DB2 ACS 操作中涉及的数据库分区的数目。

partition

数据类型: db2ACS_PartitionEntry *。

db2ACS_PartitionEntry 是 db2ACS_PartitionList 的元素。

db2ACS_PathEntry DB2 高级副本服务 (ACS) API 数据结构

db2ACS_PathEntry 包含有关数据库路径的信息。

```
typedef struct db2ACS_PathEntry
{
    /* INPUT: The path and type will be provided by the database server, as well
     *        as a flag indicating if the path is to be excluded from the backup.
     * ----- */
    char                path[DB2ACS_MAX_PATH_SZ + 1];
    db2ACS_PathType    type;
    db2Uuint32         toBeExcluded;

    /* OUTPUT: The group ID is to be provided by the backup adapter for use by
     *        the DB2 server. The group ID will be used during with snapshot
     *        operations as an indication of which paths are dependent and must
     *        be included together in any snapshot operation. Unique group IDs
     *        indicate that the paths in those groups are independent for the
     *        purposes of snapshot operations.
     * ----- */
    db2Uuint32         groupID;
} db2ACS_PathEntry;
```

path 数据类型: char[]。

长度 DB2ACS_MAX_PATH_SZ + 1 的字符串。

type 数据类型: db2ACS_PathType。

路径类型。值:

```
DB2ACS_PATH_TYPE_UNKNOWN
DB2ACS_PATH_TYPE_LOCAL_DB_DIRECTORY
DB2ACS_PATH_TYPE_DBPATH
DB2ACS_PATH_TYPE_DB_STORAGE_PATH
DB2ACS_PATH_TYPE_TBSP_CONTAINER
DB2ACS_PATH_TYPE_TBSP_DIRECTORY
DB2ACS_PATH_TYPE_TBSP_DEVICE
DB2ACS_PATH_TYPE_LOGPATH
DB2ACS_PATH_TYPE_MIRRORLOGPATH
```

toBeExcluded

数据类型: db2Uuint32。

指示快照备份中是否包含给定路径的标志。值:

- 0 - 快照备份中包含路径
- 1 - 快照备份中不包含路径

groupID

数据类型: db2Uuint32。

组标识。

db2ACS_PathList DB2 高级副本服务 (ACS) API 数据结构

db2ACS_PathList 包含数据库路径列表，包括有关特定于 DB2 ACS 操作的每个这些路径的一些额外信息。

```
/* =====  
 * Snapshot File List  
 *  
 * This is an array of 'numEntries' db2ACS_PathEntry's, where each path entry is  
 * a path to some storage on the DB2 server which is in use by the current  
 * database.  
 * ===== */  
  
typedef struct db2ACS_PathList  
{  
    db2UInt32          numEntries;  
    db2ACS_PathEntry  * entry;  
} db2ACS_PathList;
```

numEntries

数据类型: db2UInt32。

entry 数组中路径条目的数目。

entry 数据类型: db2ACS_PathEntry。

db2ACS_PathEntry 包含有关数据库路径的信息。

db2ACS_PostObjectInfo DB2 高级副本服务 (ACS) API 数据结构

db2ACS_DB2ID 是一组数据，在快照备份对象创建时无法知道，但必须在对象存储库中对其进行维护。

```
/* =====  
 * The PostObjectInfo is a set of data that can not be known at object  
 * creation time, but which must be maintained in the object repository. This  
 * is an optional field on the Verify() call, which may be NULL if there are  
 * no post-operation updates to be made.  
 * ===== */  
  
typedef struct db2ACS_PostObjectInfo  
{  
    /* The first active log will only be valid when creating a backup or  
     * snapshot object. It will indicate the file number and chain id of the  
     * first log required for recovery using this object.  
     * ----- */  
    db2ACS_LogDetails  firstActiveLog;  
} db2ACS_PostObjectInfo;
```

firstActiveLog

数据类型: db2ACS_LogDetails。

db2ACS_LogDetails 包含用于标识特定数据库日志文件的信息。

db2ACS_QueryInput 和 db2ACS_QueryOutput DB2 高级副本服务 (ACS) API 数据结构

db2ACS_QueryInput 包含要查询的对象的标识信息。 db2ACS_QueryOutput 包含有关快照备份对象的查询结果信息。

```
/* =====  
 * Unique Querying.  
 *  
 * When using this structure as query input, to indicate the  
 * intention to supply a 'wildcard' search criteria, DB2 will supply:  
 *  
 * -- character strings as "*".  
 * ===== */
```

```

*   -- numeric values as (-1), cast as the appropriate signed or unsigned
*   type.
* ===== */
typedef struct db2ACS_ObjectInfo db2ACS_QueryInput;

typedef struct db2ACS_QueryOutput
{
    db2ACS_ObjectID          objectID;
    db2ACS_ObjectInfo        object;
    db2ACS_PostObjectInfo    postInfo;
    db2ACS_DB2ID             db2ID;
    db2ACS_ObjectStatus      status;

    /* Size of the object in bytes.
    * ----- */
    db2UInt64                 objectSize;

    /* Size of the metadata associated with the object, if any, in bytes.
    * ----- */
    db2UInt64                 metaDataSize;

    /* The creation time of the object is a 64bit value with a definition
    * equivalent to an ANSI C time_t value (seconds since the epoch, GMT).
    *
    * This field is equivalent to the file creation or modification time in
    * a traditional filesystem. This should be created and stored
    * automatically by the BA subsystem, and a valid time value should be
    * returned with object query results, for all object types.
    * ----- */
    db2UInt64                 createTime;
} db2ACS_QueryOutput;

```

objectID

数据类型: db2ACS_ObjectID。

db2ACS_ObjectID 是由查询返回到存储库的每个存储对象的唯一标识。db2ACS_ObjectID 被保证是唯一的且仅在单一 DB2 ACS 会话时间框架中是持久存在的。

object 数据类型: db2ACS_ObjectInfo

db2ACS_ObjectInfo 包含有关使用 DB2 高级副本服务 (ACS) API 创建的对象的信息。

postInfo

数据类型: db2ACS_PostObjectInfo。

db2ACS_DB2ID 是一组数据, 在快照备份对象创建时无法知道, 但必须在对象存储库中对其进行维护。

db2ID 数据类型: db2ACS_DB2ID。

db2ACS_DB2ID 标识了 IBM 数据服务器。

status 数据类型: db2ACS_ObjectStatus。

db2ACS_ObjectStatus 包含有关快照备份操作的状态或进度的信息, 或包含有关快照备份对象的状态或可用性的信息。

objectSize

数据类型: db2UInt64。

对象大小 (以字节计)。

metaDataSize

数据类型: db2UInt64。

与对象（如果有）相关的元数据的大小（以字节计）。

createTime

数据类型: db2UInt64。

对象的创建时间。**createTime** 的值等于 ANSI C time_t 值。

db2ACS_ReadList DB2 高级副本服务（ACS）API 数据结构

db2ACS_ReadList 包含组列表。

```
/* The ReadList will only be used for snapshots where the action is READ, and
 * where one of the granularity modifiers other than BY_OBJ has been specified.
 * In the typical usage scenario of ( READ | BY_OBJ ) the ReadList parameter
 * should be ignored.
 *
 * When the action is DB2ACS_ACTION_BY_GROUP the union is to be interpreted
 * as a group list.
 * ----- */
typedef union db2ACS_ReadList
{
    db2ACS_GroupList      group;
} db2ACS_ReadList;
```

group 数据类型: db2ACS_GroupList。

db2ACS_GroupList 包含在快照备份操作中要包含的组的列表。

db2ACS_ReturnCode DB2 高级副本服务（ACS）API 数据结构

db2ACS_ReturnCode 包含诊断信息，其中包括特定于存储硬件的消息文本和错误代码。用于 DB2 ACS API 函数调用的 db2ACS_ReturnCode 参数的内容将记录在数据库管理器诊断日志中。

```
/* =====
 * Storage Adapter Return Code and Diagnostic Data.
 *
 * These will be recorded in the DB2 diagnostic logs, but are intended to be
 * internal return and reason codes from the storage layers which can be used
 * in conjunction with the DB2ACS_RC to provide more detailed diagnostic info.
 * ===== */
typedef struct db2ACS_ReturnCode
{
    int          returnCode;
    int          reasonCode;
    char         description[DB2ACS_MAX_COMMENT_SZ + 1];
} db2ACS_ReturnCode;
```

returnCode

数据类型: int。

特定于存储硬件的返回码。

reasonCode

数据类型: int。

特定于存储硬件的原因码。

description

数据类型: char[]。

长度 DB2ACS_MAX_COMMENT_SZ + 1 的字符串。

db2ACS_SessionInfo DB2 高级副本服务 (ACS) API 数据结构

db2ACS_SessionInfo 包含有关 DB2 ACS 会话的所有信息。

```
/* =====  
 * Session Info  
 * ===== */  
typedef struct db2ACS_SessionInfo  
{  
    db2ACS_DB2ID          db2ID;  
  
    /* Fields identifying the backup session originator.  
     * ----- */  
    SQL_PDB_NODE_TYPE    dbPartitionNum;  
    char                  db[SQL_DBNAME_SZ + 1];  
    char                  instance[DB2ACS_MAX_OWNER_SZ + 1];  
    char                  host[SQL_HOSTNAME_SZ + 1];  
    char                  user[DB2ACS_MAX_OWNER_SZ + 1];  
    char                  password[DB2ACS_MAX_PASSWORD_SZ + 1];  
  
    /* The fully qualified ACS vendor library name to be used.  
     * ----- */  
    char                  libraryName[DB2ACS_MAX_PATH_SZ + 1];  
} db2ACS_SessionInfo;
```

db2ID 数据类型: db2ACS_DB2ID

db2ACS_DB2ID 标识了 IBM 数据服务器。

dbPartitionNum

数据类型: SQL_PDB_NODE_TYPE

数据库分区的唯一的数字标识。

db 数据类型: char[]。

长度 SQL_DBNAME_SZ + 1 的字符串。

instance

数据类型: char[]。

长度 DB2ACS_MAX_OWNER_SZ + 1 的字符串。

host 数据类型: char[]。

长度 SQL_HOSTNAME_SZ + 1 的字符串。

user 数据类型: char[]。

长度 DB2ACS_MAX_OWNER_SZ + 1 的字符串。

password

数据类型: char[]。

长度 DB2ACS_MAX_PASSWORD_SZ + 1 的字符串。

libraryName

数据类型: char[]。

长度 DB2ACS_MAX_PATH_SZ + 1 的字符串。

db2ACS_SnapshotDetails DB2 高级副本服务 (ACS) API 数据结构

db2ACS_SnapshotDetails 包含有关快照备份操作的信息。

```
typedef struct db2ACS_SnapshotDetails
{
    char                imageTimestamp[SQLU_TIME_STAMP_LEN + 1];
} db2ACS_SnapshotDetails;
```

imageTimestamp

数据类型: char[]。

长度 SQLU_TIME_STAMP_LEN + 1 的字符串。

db2ACS_MetaData DB2 高级副本服务 (ACS) API 数据结构

db2ACS_MetaData 存储快照备份元数据。

```
/* =====
 * The metadata structure itself is internal to DB2 and is to be treated by
 * the storage interface as an unstructured block of data of the given size.
 * ===== */
typedef struct db2ACS_MetaData
{
    db2UInt64                size;
    void                    * data;
} db2ACS_MetaData;
```

size 数据类型: db2UInt32。

data 的大小 (以字节计)。

data 数据类型: void *。

指向数据库管理器用于存储快照备份元数据的内存块的指针。

db2ACS_VendorInfo DB2 高级副本服务 (ACS) API 数据结构

db2ACS_VendorInfo 包含有关 DB2 ACS API 驱动程序的信息。

```
/* =====
 * Storage Vendor Identifier
 * ===== */
typedef struct db2ACS_VendorInfo
{
    void                    * vendorCB;                /* Vendor control block */
    db2UInt32                version;                  /* Current version */
    db2UInt32                release;                 /* Current release */
    db2UInt32                level;                   /* Current level */
    char                    signature[DB2ACS_MAX_VENDORID_SZ + 1];
} db2ACS_VendorInfo;
```

vendorCB

数据类型: void *。

特定于 DB2 ACS API 驱动程序的控制块的指针。

version

数据类型: db2UInt32。

DB2 ACS API 驱动程序的版本。

release

数据类型: db2UInt32。

DB2 ACS API 驱动程序的发行版级别。

level 数据类型: db2UInt32。

DB2 ACS API 驱动程序的级别标识。

signature

数据类型: db2ACS_VendorSignature。

长度 DB2ACS_MAX_VENDORID_SZ + 1 的字符串。

DB2 高级副本服务 (ACS) 最佳实践

在安装和配置 DB2 高级副本服务 (ACS) 时, 请考虑下列最佳实践。

为日志路径指定专用卷组

建议要包括在自己的快照卷中的日志路径独立于数据库目录和数据库容器。

为每个数据库分区指定一个卷组

在数据库分区功能 (DPF) 环境中, 每个数据库分区必须位于独立于其他数据库分区的一组快照卷中。

DB2 高级副本服务 (ACS) 限制

安装和配置 DB2 高级副本服务 (ACS) 时, 要考虑一些限制。

不支持卷共享。如果一个数据库分区与任何其他数据库分区位于相同存储卷上, 那么不允许执行快照操作。

DB2 高级副本服务 (ACS) API 返回码

DB2 高级副本服务 (ACS) API 函数返回已定义的一组可能的返回码。

表 25. DB2 高级副本服务 (ACS) API 返回码

返回码	描述
DB2ACS_RC_OK	操作已成功。
DB2ACS_RC_LINK_EXIST	先前已激活会话。
DB2ACS_RC_COMM_ERROR	存在与诸如磁带机之类的存储设备的通信错误。
DB2ACS_RC_INV_VERSION	数据库管理器的 DB2 ACS 库版本与 DB2 ACS API 驱动程序版本不兼容。
DB2ACS_RC_INV_ACTION	数据库管理器请求来自 DB2 ACS API 驱动程序的无效操作。
DB2ACS_RC_NO_DEV_AVAIL	当前不存在诸如磁带机之类的存储设备可用。
DB2ACS_RC_OBJ_NOT_FOUND	DB2 ACS API 驱动程序找不到数据库管理器指定的快照备份对象。
DB2ACS_RC_OBJ_S_FOUND	DB2 ACS API 驱动程序找到多个与数据库管理器指定的规范相符的快照备份对象。
DB2ACS_RC_INV_USERID	数据库管理器将无效的用户标识传递到 DB2 ACS API 驱动程序。
DB2ACS_RC_INV_PASSWORD	数据库管理器将无效的密码传递到 DB2 ACS API 驱动程序。
DB2ACS_RC_INV_OPTIONS	数据库管理器指定了无效选项。
DB2ACS_RC_INIT_FAILED	数据库管理器尝试初始化 DB2 ACS 会话, 但初始化失败。
DB2ACS_RC_INV_DEV_HANDLE	数据库管理器传递无效的存储设备句柄。
DB2ACS_RC_BUFF_SIZE	数据库管理器指定了无效的缓冲区大小。
DB2ACS_RC_END_OF_DATA	DB2 ACS API 驱动程序找不到更多的快照备份对象。

表 25. DB2 高级副本服务 (ACS) API 返回码 (续)

返回码	描述
DB2ACS_RC_END_OF_TAPE	存储设备意外地到达磁带备份介质的末端。
DB2ACS_RC_DATA_RESEND	诸如磁带机之类的存储设备请求数据库管理器重新发送最近的数据缓冲区。
DB2ACS_RC_COMMIT_FAILED	DB2 ACS API 驱动程序无法落实事务。
DB2ACS_RC_DEV_ERROR	存在与诸如磁带机之类的存储设备有关的错误。
DB2ACS_RC_WARNING	存储硬件返回警告。查看数据库管理器诊断日志，以了解更多信息。
DB2ACS_RC_LINK_NOT_EXIST	先前未激活会话。
DB2ACS_RC_MORE_DATA	有更多的数据要从存储位置传送到数据库管理器。
DB2ACS_RC_ENDOFMEDIA_NO_DATA	存储设备到达存储介质的末端而未找到任何数据。
DB2ACS_RC_ENDOFMEDIA	存储设备到达存储介质的末端。
DB2ACS_RC_MAX_LINK_GRANT	已经建立最大数目的链接。数据库管理器无法建立更多链接。
DB2ACS_RC_IO_ERROR	DB2 ACS API 驱动程序遇到输入或输出操作所产生的错误。
DB2ACS_RC_DELETE_FAILED	DB2 ACS API 驱动程序无法成功地删除数据库管理器指定的快照备份对象。
DB2ACS_RC_INV_BKUP_FNAME	数据库管理器为快照备份对象指定了无效的文件名。
DB2ACS_RC_NOT_ENOUGH_SPACE	DB2 ACS API 驱动程序估计没有足够的存储空间来执行数据库管理器指定的数据库的快照备份。
DB2ACS_RC_ABORT_FAILED	数据库管理器尝试中止 DB2 ACS 操作，但中止尝试失败。
DB2ACS_RC_UNEXPECTED_ERROR	DB2 ACS API 驱动程序遇到严重的未知错误。
DB2ACS_RC_NO_DATA	DB2 ACS API 驱动程序未将任何数据返回到数据库管理器。
DB2ACS_RC_OBJ_OUT_OF_SCOPE	数据库管理器尝试对 DB2 ACS API 驱动程序未管理的恢复对象执行 DB2 ACS 操作。
DB2ACS_RC_INV_CALL_SEQUENCE	数据库管理器按照无效的顺序来调用 DB2 ACS API 函数。例如，数据库管理器必须先调用 db2ACSInitialize，然后才可以调用除 db2ACSQueryAPIVersion 之外的其他任何 DB2 ACS API 函数。
DB2ACS_RC_SHARED_STORAGE_GROUP	数据库管理器尝试针对另一数据库或应用程序正在使用的存储对象执行快照操作。

DB2 高级副本服务 (ACS) 支持的操作系统和硬件

集成到 IBM 数据服务器的是 DB2 ACS API 驱动程序，它支持 IBM 数据服务器所支持的一部分操作系统和硬件。

表 26. DB2 高级副本服务 (ACS) API 支持的操作系统和硬件

硬件	具有存储区域网络 (SAN) 存储器的 AIX 操作系统 ²	具有网络文件系统 (NFS) 存储器的 AIX 操作系统 ²	具有网络文件系统 (NFS) 存储器的 Linux 操作系统 ¹
IBM TotalStorage SAN Volume Controller	完全支持。	不支持。	不支持。
IBM 系统存储器 DS6000	完全支持，除了： <ul style="list-style-type: none"> 增量复制不受支持 	不支持。	不支持。

表 26. DB2 高级副本服务 (ACS) API 支持的操作系统和硬件 (续)

硬件	具有存储区域网络 (SAN) 存储器的 AIX 操作系统 ²	具有网络文件系统 (NFS) 存储器的 AIX 操作系统 ²	具有网络文件系统 (NFS) 存储器的 Linux 操作系统 ¹
IBM 系统存储器 DS8000	完全支持, 除了: • 增量复制不受支持	不支持。	不支持。
IBM 系统存储器 N 系列	完全支持。	完全支持。	完全支持。
NetApp V 系列	完全支持。	完全支持。	完全支持。
NetApp FAS 系列	完全支持	完全支持	完全支持

¹只有下列具有 DB2 ACS 的 Linux 系统受支持:

- 在 x86 (Intel Pentium[®]、Intel Xeon[®] 和 AMD) 处理器上仅限于 64 位
- POWER[™] (支持 Linux 的 IBM eServer[™] OpenPower[®]、System i 或 pSeries 系统)

²仅 AIX 5.3 对于 AIX 上的 DB2 ACS 受支持。对于 AIX 6.1 操作系统, 可以执行手动复制。此类操作的示例是暂挂写 I/O 和使用第 178 页的『将分割镜像用作备份映像』中所述的分割镜像功能。还可购买并安装 Tivoli Storage Manager for Advanced Copy Services V6.1 for AIX 6.1 的完整版本。

第 3 部分 附录

附录 A. DB2 技术信息概述

可以通过下列工具和方法获取 DB2 技术信息:

- DB2 信息中心
 - 主题（任务、概念和参考主题）
 - DB2 工具的帮助
 - 样本程序
 - 教程
- DB2 书籍
 - PDF 文件（可下载）
 - PDF 文件（在 DB2 PDF DVD 中）
 - 印刷版书籍
- 命令行帮助
 - 命令帮助
 - 消息帮助

注: DB2 信息中心主题的更新频率比 PDF 书籍或硬拷贝书籍的更新频率高。要获取最新信息, 请安装可用的文档更新, 或者参阅 [ibm.com](http://www.ibm.com) 上的 DB2 信息中心。

可以在线访问 [ibm.com](http://www.ibm.com) 上的其他 DB2 技术信息, 如技术说明、白皮书和 IBM Redbooks® 出版物。访问位于以下网址的 DB2 信息管理软件库站点: <http://www.ibm.com/software/data/sw-library/>。

文档反馈

我们非常重视您对 DB2 文档的反馈。如果您想就如何改善 DB2 文档提出建议, 请将电子邮件发送至 db2docs@ca.ibm.com。DB2 文档小组会阅读您的所有反馈, 但不能直接答复您。请尽可能提供具体的示例, 这样我们才能更好地了解您所关心的问题。如果您要提供有关具体主题或帮助文件的反馈, 请加上标题和 URL。

请不要用以上电子邮件地址与 DB2 客户支持机构联系。如果您遇到文档不能解决的 DB2 技术问题, 请与您当地的 IBM 服务中心联系以获得帮助。

硬拷贝或 PDF 格式的 DB2 技术库

下列各表描述 IBM 出版物中心（网址为 www.ibm.com/shop/publications/order）提供的 DB2 资料库。可从 www.ibm.com/support/docview.wss?rs=71&uid=swg2700947 下载 PDF 格式的 DB2 版本 9.7 手册的英文版本和翻译版本。

尽管这些表标识书籍有印刷版, 但可能未在您所在国家或地区提供。

每次更新手册时, 表单号都会递增。确保您正在阅读下面列示的手册的最新版本。

注: DB2 信息中心的更新频率比 PDF 或硬拷贝书籍的更新频率高。

表 27. DB2 技术信息

书名	书号	是否提供印刷版	最近一次更新时间
<i>Administrative API Reference</i>	SC27-2435-00	是	2009 年 8 月
<i>Administrative Routines and Views</i>	SC27-2436-00	否	2009 年 8 月
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-00	是	2009 年 8 月
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-00	是	2009 年 8 月
<i>Command Reference</i>	SC27-2439-00	是	2009 年 8 月
《数据移动指南和参考》	S151-1186-00	是	2009 年 8 月
《数据恢复及高可用性指南与参考》	S151-1187-00	是	2009 年 8 月
《数据库管理概念和配置参考》	S151-1163-00	是	2009 年 8 月
《数据库监视指南和参考》	S151-1165-00	是	2009 年 8 月
《数据库安全性指南》	S151-1188-00	是	2009 年 8 月
<i>DB2 Text Search Guide</i>	SC27-2459-00	是	2009 年 8 月
《开发 ADO.NET 和 OLE DB 应用程序》	S151-1167-00	是	2009 年 8 月
《开发嵌入式 SQL 应用程序》	S151-1168-00	是	2009 年 8 月
<i>Developing Java Applications</i>	SC27-2446-00	是	2009 年 8 月
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-00	否	2009 年 8 月
<i>Developing User-defined Routines (SQL and External)</i>	S151-1169-00	是	2009 年 8 月
<i>Getting Started with Database Application Development</i>	G151-1170-00	是	2009 年 8 月
《Linux 和 Windows 上的 DB2 安装和管理入门》	G151-1172-00	是	2009 年 8 月
《全球化指南》	S151-1189-00	是	2009 年 8 月
《安装 DB2 服务器》	G151-1174-00	是	2009 年 8 月
《安装 IBM 数据服务器客户端》	G151-1175-00	否	2009 年 8 月
《消息参考第 1 卷》	S151-1182-00	否	2009 年 8 月

表 27. DB2 技术信息 (续)

书名	书号	是否提供印刷版	最近一次更新时间
《消息参考第 2 卷》	S151-1183-00	否	2009 年 8 月
《Net Search Extender 管理和用户指南》	S151-1185-00	否	2009 年 8 月
《分区和集群指南》	S151-1190-00	是	2009 年 8 月
《pureXML 指南》	S151-1180-00	是	2009 年 8 月
<i>Query Patroller Administration and User's Guide</i>	SC27-2467-00	否	2009 年 8 月
《Spatial Extender 和地理数据管理功能部件用户指南和参考》	SC27-2468-00	否	2009 年 8 月
《SQL 过程语言: 应用程序启用和支持》	S151-1171-00	是	2009 年 8 月
<i>SQL Reference, Volume 1</i>	SC27-2456-00	是	2009 年 8 月
<i>SQL Reference, Volume 2</i>	SC27-2457-00	是	2009 年 8 月
《故障诊断和调整数据库性能》	S151-1164-00	是	2009 年 8 月
《升级到 DB2 版本 9.7》	S151-1173-00	是	2009 年 8 月
《Visual Explain 教程》	S151-1184-00	否	2009 年 8 月
《DB2 版本 9.7 新增内容》	S151-1179-00	是	2009 年 8 月
<i>Workload Manager Guide and Reference</i>	SC27-2464-00	是	2009 年 8 月
《XQuery 参考》	S151-1181-00	否	2009 年 8 月

表 28. 特定于 DB2 Connect 的技术信息

书名	书号	是否提供印刷版	最近一次更新时间
《安装和配置 DB2 Connect 个人版》	S151-1177-00	是	2009 年 8 月
《安装和配置 DB2 Connect 服务器》	S151-1178-00	是	2009 年 8 月
《DB2 Connect 用户指南》	S151-1176-00	是	2009 年 8 月

表 29. Information Integration 技术信息

书名	书号	是否提供印刷版	最近一次更新时间
<i>Information Integration: Administration Guide for Federated Systems</i>	SC19-1020-02	是	2009 年 8 月
<i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i>	SC19-1018-04	是	2009 年 8 月

表 29. Information Integration 技术信息 (续)

书名	书号	是否提供印刷版	最近一次更新时间
Information Integration: Configuration Guide for Federated Data Sources	SC19-1034-02	否	2009 年 8 月
Information Integration: SQL Replication Guide and Reference	SC19-1030-02	是	2009 年 8 月
Information Integration: Introduction to Replication and Event Publishing	GC19-1028-02	是	2009 年 8 月

订购印刷版的 DB2 书籍

如果您需要印刷版的 DB2 书籍，可以在许多（但不是所有）国家或地区在线购买。无论何时都可以从当地的 IBM 代表处订购印刷版的 DB2 书籍。请注意，DB2 PDF 文档 DVD 上的某些软拷贝书籍没有印刷版。例如，DB2 消息参考的任何一卷都没有提供印刷版书籍。

只要支付一定费用，就可以从 IBM 获取 DB2 PDF 文档 DVD，该 DVD 包含许多 DB2 书籍的印刷版。根据您下订单的位置，您可能能够从 IBM 出版物中心在线订购书籍。如果在线订购在您所在国家或地区不可用，您始终可以从当地的 IBM 代表处订购印刷版 DB2 书籍。注意，并非 DB2 PDF 文档 DVD 上的所有书籍都有印刷版。

注：最新最完整的 DB2 文档保留在 DB2 信息中心中，网址为：<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>。

要订购印刷版的 DB2 书籍：

- 要了解您是否可从所在国家或地区在线订购印刷版的 DB2 书籍，可查看 IBM 出版物中心站点，网址为：<http://www.ibm.com/shop/publications/order>。必须先选择国家、地区或语言才能访问出版物订购信息，然后再按照针对您所在位置的订购指示信息进行订购。
- 要从当地的 IBM 代表处订购印刷版的 DB2 书籍：
 1. 从下列其中一个 Web 站点找到当地代表处的联系信息：
 - IBM 全球联系人目录，网址为 www.ibm.com/planetwide。
 - IBM 出版物 Web 站点，网址为 <http://www.ibm.com/shop/publications/order>。必须先选择国家、地区或语言才能访问对应您的所在地的出版物主页。在此页面中访问“关于此站点”链接。
 2. 请在致电时说明您想订购 DB2 出版物。
 3. 请您向当地的代表提供想要订购的书籍的书名和书号。有关书名和书号的信息，请参阅第 307 页的『硬拷贝或 PDF 格式的 DB2 技术库』。

从命令行处理器显示 SQL 状态帮助

DB2 产品针对可能充当 SQL 语句结果的条件返回 SQLSTATE 值。SQLSTATE 帮助说明 SQL 状态和 SQL 状态类代码的含义。

要启动 SQL 状态帮助，请打开命令行处理器并输入：

```
? sqlstate or ? class code
```

其中，*sqlstate* 表示有效的 5 位 SQL 状态，*class code* 表示该 SQL 状态的前 2 位。例如，? 08003 显示 08003 SQL 状态的帮助，而 ? 08 显示 08 类代码的帮助。

访问不同版本的 DB2 信息中心

对于 DB2 版本 9.7 主题，DB2 信息中心 URL 为 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>

对于 DB2 版本 9.5 主题，DB2 信息中心 URL 为 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>

对于 DB2 版本 9 主题，DB2 信息中心 URL 为 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>

对于 DB2 版本 8 主题，请访问版本 8 信息中心 URL: <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>

在 DB2 信息中心中以您的首选语言显示主题

DB2 信息中心尝试以您在浏览器首选项中指定的语言显示主题。如果未提供主题的首选语言翻译版本，那么 DB2 信息中心将显示该主题的英文版。

• 要在 Internet Explorer 浏览器中以您的首选语言显示主题：

1. 在 Internet Explorer 中，单击 **工具** → **Internet 选项** → **语言...** 按钮。“语言首选项”窗口打开。
2. 确保您的首选语言被指定为语言列表中的第一个条目。
 - 要将新语言添加至列表，请单击 **添加...** 按钮。

注：添加语言并不能保证计算机具有以首选语言显示主题所需的字体。

- 要将语言移至列表顶部，请选择该语言并单击 **上移** 按钮直到该语言成为语言列表中的第一个条目。

3. 清除浏览器高速缓存然后刷新页面以便以首选语言显示 DB2 信息中心。

• 要在 Firefox 或 Mozilla 浏览器中以首选语言显示主题：

1. 在 **工具** → **选项** → **高级对话框** 中的 **语言** 部分中选择按钮。“语言”面板将显示在“首选项”窗口中。
2. 确保您的首选语言被指定为语言列表中的第一个条目。
 - 要将新语言添加至列表，请单击 **添加...** 按钮以从“添加语言”窗口中选择一种语言。
 - 要将语言移至列表顶部，请选择该语言并单击 **上移** 按钮直到该语言成为语言列表中的第一个条目。

3. 清除浏览器高速缓存然后刷新页面以便以首选语言显示 DB2 信息中心。

在某些浏览器和操作系统组合上，可能还必须将操作系统的区域设置更改为您选择的语言环境和语言。

更新安装在您的计算机或内部网服务器上的 DB2 信息中心

本地安装的 DB2 信息中心必须定期进行更新。

开始前

必须已安装 DB2 版本 9.7 信息中心。有关详细信息，请参阅《安装 DB2 服务器》中的“使用 DB2 安装向导来安装 DB2 信息中心”主题。所有适用于安装信息中心的先决条件和限制同样适用于更新信息中心。

关于此任务

可自动或手动更新现有 DB2 信息中心：

- 自动更新 - 更新现有信息中心功能和语言。自动更新的一个优点是在更新期间，信息中心不可用的时间最短。另外，自动更新可设置为作为定期运行的其他批处理作业的一部分运行。
- 手动更新 - 应该在更新过程期间要添加功能或语言时使用。例如，如果本地信息中心最初安装的是英语和法语版，而现在还要安装德语版；那么手动更新将安装德语版，并更新现有信息中心的功能和语言。但是，手动更新要求您手动停止、更新和重新启动信息中心。在整个更新过程期间信息中心不可用。

过程

此主题详细说明了自动更新的过程。有关手动更新的指示信息，请参阅“手动更新安装在您的计算机或内部网服务器上的 DB2 信息中心”主题。

要自动更新安装在您的计算机或内部网服务器上的 DB2 信息中心：

1. 在 Linux 操作系统上，
 - a. 浏览至信息中心的安装位置。缺省情况下，DB2 信息中心安装在 `/opt/ibm/db2ic/V9.7` 目录中。
 - b. 从安装目录浏览至 `doc/bin` 目录。
 - c. 运行 `ic-update` 脚本：

```
ic-update
```
2. 在 Windows 操作系统上，
 - a. 打开命令窗口。
 - b. 浏览至信息中心的安装位置。缺省情况下，DB2 信息中心安装在 `<Program Files>\IBMDB2 Information Center\Version 9.7` 目录中，其中 `<Program Files>` 表示 `Program Files` 目录的位置。
 - c. 从安装目录浏览至 `doc\bin` 目录。
 - d. 运行 `ic-update.bat` 文件：

```
ic-update.bat
```

结果

DB2 信息中心自动重新启动。如果更新可用，那么信息中心会显示新的以及更新后的主题。如果信息中心更新不可用，那么会在日志中添加消息。日志文件位于 `doc\ eclipse\ configuration` 目录中。日志文件名称是随机生成的编号。例如，`1239053440785.log`。

手动更新安装在您的计算机或内部网服务器上的 DB2 信息中心

如果已经在本地安装了 DB2 信息中心，那么您可以从 IBM 获取文档更新并安装。

手动更新在本地安装的 DB2 信息中心要求您：

1. 停止计算机上的 DB2 信息中心，然后以独立方式重新启动信息中心。如果以独立方式运行信息中心，那么网络上的其他用户将无法访问信息中心，因而您可以应用更新。DB2 信息中心的工作站版本总是以独立方式运行。
2. 使用“更新”功能部件来查看可用的更新。如果有您必须安装的更新，那么请使用“更新”功能部件来获取并安装这些更新。

注：如果您的环境要求在一台未连接至因特网的机器上安装 DB2 信息中心更新，那么通过使用一台已连接至因特网并有已安装的 DB2 信息中心的机器将更新站点镜像至本地文件系统。如果网络中有许多用户将安装文档更新，那么可以通过在本地也为更新站点制作镜像并为更新站点创建代理来缩短每个人执行更新所需要的时间。如果提供了更新包，请使用“更新”功能部件来获取这些更新包。但是，只有在单机方式下才能使用“更新”功能部件。

3. 停止独立信息中心，然后在计算机上重新启动 DB2 信息中心。

注：在 Windows 2008、Windows Vista 和更高版本上，稍后列示在此部分的命令必须作为管理员运行。要打开具有全面管理员特权的命令提示符或图形工具，请右键单击快捷方式，然后选择**以管理员身份运行**。

要更新安装在您的计算机或内部网服务器上的 DB2 信息中心：

1. 停止 DB2 信息中心。
 - 在 Windows 上，单击**开始** → **控制面板** → **管理工具** → **服务**。右键单击 **DB2 信息中心服务**，并选择**停止**。
 - 在 Linux 上，输入以下命令：

```
/etc/init.d/db2icdv97 stop
```
2. 以独立方式启动信息中心。
 - 在 Windows 上：
 - a. 打开命令窗口。
 - b. 浏览至信息中心的安装位置。缺省情况下，DB2 信息中心安装在 `<Program Files>\IBM\DB2 Information Center\Version 9.7` 目录中，其中 `<Program Files>` 表示 Program Files 目录的位置。
 - c. 从安装目录浏览至 `doc\bin` 目录。
 - d. 运行 `help_start.bat` 文件：

```
help_start.bat
```
 - 在 Linux 上：
 - a. 浏览至信息中心的安装位置。缺省情况下，DB2 信息中心安装在 `/opt/ibm/db2ic/V9.7` 目录中。
 - b. 从安装目录浏览至 `doc/bin` 目录。

c. 运行 help_start 脚本:

```
help_start
```

系统缺省 Web 浏览器将打开以显示独立信息中心。

3. 单击**更新按钮** (🔄)。(必须在浏览器中启用 JavaScript™。) 在信息中心的右边面板上, 单击**查找更新**。将显示现有文档的更新列表。
4. 要启动安装进程, 请检查您要安装的选项, 然后单击**安装更新**。
5. 在安装进程完成后, 请单击**完成**。
6. 要停止独立信息中心, 请执行下列操作:
 - 在 Windows 上, 浏览至安装目录的 doc\bin 目录并运行 help_end.bat 文件:

```
help_end.bat
```

注: help_end 批处理文件包含安全地停止使用 help_start 批处理文件启动的进程所需的命令。不要使用 Ctrl-C 或任何其他方法来停止 help_start.bat。
 - 在 Linux 上, 浏览至安装目录的 doc/bin 目录并运行 help_end 脚本:

```
help_end
```

注: help_end 脚本包含安全地停止使用 help_start 脚本启动的进程所需的命令。不要使用任何其他方法来停止 help_start 脚本。
7. 重新启动 DB2 信息中心。
 - 在 Windows 上, 单击**开始** → **控制面板** → **管理工具** → **服务**。右键单击 **DB2 信息中心服务**, 并选择**启动**。
 - 在 Linux 上, 输入以下命令:

```
/etc/init.d/db2icdv97 start
```

更新后的 DB2 信息中心将显示新的以及更新后的主题。

DB2 教程

DB2 教程帮助您了解 DB2 产品的各个方面。这些课程提供了逐步指示信息。

开始之前

可从信息中心查看 XHTML 版的教程: <http://publib.boulder.ibm.com/infocenter/db2help/>。

某些课程使用了样本数据或代码。有关其特定任务的任何先决条件的描述, 请参阅教程。

DB2 教程

要查看教程, 请单击标题。

《pureXML 指南》中的“pureXML™”

设置 DB2 数据库以存储 XML 数据以及对本机 XML 数据存储执行基本操作。

《Visual Explain 教程》中的“Visual Explain”

使用 Visual Explain 来分析、优化和调整 SQL 语句以获取更好的性能。

DB2 故障诊断信息

提供了很多故障诊断和问题确定信息以帮助您使用 DB2 数据库产品。

DB2 文档

故障诊断信息可在《DB2 故障诊断指南》或 *DB2 信息中心*的“数据库基础”部分中找到。可在该处找到有关如何使用 DB2 诊断工具和实用程序来隔离和找出问题的信息、某些最常见问题的解决方案以及有关如何解决使用 DB2 数据库产品时可能遇到的问题的建议。

DB2 技术支持 Web 站点

如果您遇到了问题并且想要获取查找可能的原因和解决方案的帮助，请参阅 DB2 技术支持 Web 站点。该“技术支持”站点具有指向最新 DB2 出版物、技术说明、授权程序分析报告（APAR 或错误修订）、修订包和其他资源的链接。可搜索此知识库并查找问题的可能解决方案。

请访问 DB2 技术支持 Web 站点：http://www.ibm.com/software/data/db2/support/db2_9/。

条款和条件

如果符合以下条款和条件，那么授予您使用这些出版物的许可权。

个人使用：只要保留所有的专有权声明，您就可以为个人、非商业使用复制这些出版物。未经 IBM 明确同意，您不可以分发、展示或制作这些出版物或其中任何部分的演绎作品。

商业使用：只要保留所有的专有权声明，您就可以仅在企业内复制、分发和展示这些出版物。未经 IBM 明确同意，您不可以制作这些出版物的演绎作品，或者在您的企业外部复制、分发或展示这些出版物或其中的任何部分。

除非本许可权中明确授予，否则不得授予对这些出版物或其中包含的任何信息、数据、软件或其他知识产权的任何许可权、许可证或权利，无论是明示的还是暗含的。

当使用这些出版物损害了 IBM 的利益，或者根据 IBM 的规定，未正确遵守上述指导说明时，那么 IBM 保留自主决定撤销本文授予的许可权的权利。

只有您完全遵循所有适用的法律和法规，包括所有的美国出口法律和法规，您才可以下载、出口或再出口该信息。

IBM 对这些出版物的内容不作任何保证。这些出版物“按现状”提供，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的关于适销和适用于某种特定用途的保证。

附录 B. 声明

本信息是为在美国提供的产品和服务编写的。有关非 IBM 产品的信息是基于首次出版此文档时的可获得信息且会随时更新。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节字符集 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区： International Business Machines Corporation“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是此 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

只要遵守适当的条款和条件，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息可能包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，与实际商业企业所用的名称和地址的任何雷同纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。此样本程序“按现状”提供，且不附有任何种类的保证。对于使用此样本程序所引起的任何损坏，IBM 将不承担责任。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明：

©（贵公司的名称）（年份）。此部分代码是根据 IBM 公司的样本程序衍生出来的。© Copyright IBM Corp.（输入年份）。All rights reserved.

商标

IBM、IBM 徽标和 ibm.com[®] 是 International Business Machines Corp. 在全球范围许多管辖区域内的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。Web 站点 www.ibm.com/legal/copytrade.shtml 上的“版权和商标信息”中提供了 IBM 商标的最新列表。

下列术语是其他公司的商标或注册商标

- Linux 是 Linus Torvalds 在美国和/或其他国家或地区的注册商标。
- Java 和所有基于 Java 的商标和徽标是 Sun Microsystems, Inc. 在美国和/或其他国家或地区的商标。
- UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。
- Intel、Intel 徽标、Intel Inside[®]、Intel Inside 徽标、Intel[®] Centrino[®]、Intel Centrino 徽标、Celeron[®]、Intel[®] Xeon[®]、Intel SpeedStep[®]、Itanium[®] 和 Pentium 是 Intel 公司或其子公司在美国和其他国家或地区的商标或注册商标。
- Microsoft、Windows、Windows NT[®] 和 Windows 徽标是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。

索引

[A]

按需归档日志 119

[B]

版本

 数据库的版本恢复 204

帮助

 配置语言 311

 SQL 语句 311

保持活动的信息包

 描述 100

备份

 操作系统限制 155

 磁带 178

 存储器注意事项 154

 递增 207

 分区数据库 180

 联机 153

 命名管道 180

 频率 153

 容器名称 173

 数据库

 自动 182

 脱机 153

 显示信息 173

 映像 173

 包括日志文件及 125

 用户出口程序 154

 自动 151, 183

 CLP 示例 186

备用服务器

 标识 21

 示例 142

备用数据库

 状态 134

本地同步复制状态 134

崩溃恢复 194

表

 关系 154

表空间

 重建 222, 229

 复原 205

 恢复 195, 196

 前滚恢复 205, 247

 容器

 重建数据库 225

并行恢复 211

不可恢复的数据库

 备份与恢复 151

不确定事务

 恢复

 不使用 DB2 同步点管理器 203

 使用 DB2 同步点管理器 202

 在主机上 202

[C]

重定向复原 218

 使用脚本 219

 使用生成的脚本 221

重建

 数据库 222

 表空间容器 225

 使用增量备份映像 230

 所选表空间 222, 229

重新定义表空间容器

 RESTORE 实用程序 218

 使用脚本 219

重新平衡

 与联机备份的兼容性 184

重组

 自动 183

重组表

 与联机备份的兼容性 184

创建

 克隆数据库 130

磁带备份 178

磁带机

 将日志文件存储于 56, 119

磁盘

 独立磁盘冗余阵列 (RAID) 197

 故障管理 197

 条带分割 197

磁盘镜像

 RAID 级别 1 197

磁盘阵列

 概述 197

存储器

 介质故障 154

 要求

 备份与恢复 154

错误

 日志满载 48

[D]

定额设备 63

订购 DB2 书籍 310

独立磁盘冗余阵列 (RAID)

 减少介质故障的影响 197

- 对等状态
 - 描述 134
- 多个实例
 - 使用 Tivoli Storage Manager 256

[F]

- 发生故障的数据库分区服务器 199
- 返回码
 - 用户出口程序 123
- 废弃的表恢复
 - 描述 193
- 分割镜像
 - 处理 15
 - 作为备份映像 178
 - 作为备用数据库 38
 - 作为克隆数据库 130
- 分区表
 - 备份 181
- 分区数据库环境
 - 备份
 - 概述 180
 - 重建数据库 231
 - 事务
 - 故障恢复 199
- 服务器
 - 备用 17, 21
- 服务器故障转移集群 104
- 服务器集群 104
- 复原
 - 从快照备份 214
 - 递增 209, 216
 - 数据库
 - 递增 207
 - 前滚恢复 205
 - 数据至现有数据库 215
 - 数据至新数据库 216
 - 自动递增
 - 局限性 210
- 复制的操作
 - 高可用性灾难恢复 (HADR) 132

[G]

- 高可用性
 - 策略
 - 概述 7
 - 故障转移 8
 - 集群 8, 100
 - 冗余 7, 131
 - 故障监视器
 - 概述 141
 - 配置 (db2fm 命令) 24
 - 配置 (db2fmc 和系统命令) 25
 - 注册表文件 23

- 高可用性 (续)
 - 管理 117
 - 脉动信号监视 141
 - 配置
 - 概述 17
 - 集群环境 57
 - AUTO_DEL_REC_OBJ 参数 168
 - NAT 44
 - 日志装入 14
 - 设计 1, 41
 - 维护
 - 概述 117
 - 手动 127
 - 自动 127
 - 最小化影响 127
 - 中断 3
 - 避免 5
 - 概述 1
 - 检测 139, 141
 - 容错 4
 - 特征符 3
 - 响应 139, 142
 - 中断成本 4
 - IBM 数据服务器功能 11
 - Microsoft Cluster Server (MSCS) 104
 - Solaris 操作系统 108
 - Sun Cluster 3.0 110
 - Tivoli System Automation for Multiplatforms 103
- 高可用性集群多处理 (HACMP)
 - 描述 100
- 高可用性灾难恢复 (HADR)
 - 备用数据库
 - 初始化 37
 - 同步 131
 - 状态 136
 - 不复制的操作 132
 - 复制的操作 132
 - 概述 12
 - 故障回退 148
 - 故障转移 145
 - 管理 136
 - 集群管理器 37
 - 监视 141
 - 卷动更新 129
 - 卷动升级 129
 - 客户机自动重新路由 28
 - 命令 137
 - 配置 30
 - 切换数据库角色 147
 - 日志归档 34
 - 设置 26
 - 数据库
 - 初始化 26
 - 激活 129
 - 取消激活 129
 - 停止 128

高可用性灾难恢复 (HADR) (续)

- 同步方式 39
- 系统要求 41
- 限制 44
- 性能 35
- 要求 43
 - 初始化 41
- 支持 41
- 执行卷动更新 129
- 执行卷动升级 129
- 主要重新集成 148
- 装入操作 30
- 更新
 - DB2 信息中心 312, 313
- 故障回退
 - 操作 148
- 故障监视器 141
 - 配置 23
 - db2fm 24
 - db2fmc 25
- 故障监视器工具 12
- 故障事务 194
- 故障诊断
 - 教程 315
 - 联机信息 315
- 故障转移 8, 139, 142
 - 高可用性灾难恢复 (HADR) 145
 - 故障转移策略
 - 本地重新启动故障转移 64
 - 定制故障转移 64
 - 相互故障转移 64
 - 循环故障转移 64
 - HADR 故障转移 64
 - N 加 M 故障转移 64
 - 空闲备用 8
 - 相互接管 8
- 故障转移集群 104
- 故障转移支持
 - 概述 108
 - AIX 100
 - Solaris 操作系统 108
 - Sun Cluster 3.0 110
 - Windows 104
- 关系
 - 表之间 154
- 关于本书
 - 数据恢复及高可用性指南与参考 vii
- 管理
 - 高可用性灾难恢复 (HADR) 136
- 管理视图
 - DB_HISTORY 161
- 管理通知日志 139, 194
- 归档日志记录 10, 56
- 归档日志文件
 - 至磁带 119

规则

- 文件 100

[H]

恢复

- 版本 204
- 崩溃 194
- 并行 211
- 操作系统限制 155
- 策略概述 151
- 从数据库分区服务器的故障 201
- 存储器注意事项 154
- 递增 207
- 减少日志记录 54
- 跨节点示例 188
- 历史记录文件 157
- 两阶段落实协议 199
- 前滚 205
- 时间点 205
- 数据库
 - 概述 187
 - 数据库重建 222
 - 损坏的表空间 195, 196
 - 所需时间 153
 - 性能 211
 - 已删除的表 193
 - 至日志末尾 205
- 恢复对象
 - 管理 167, 168
 - 自动 168
 - db2Prune 167
 - PRUNE HISTORY 167
- 恢复历史记录文件
 - 不活动
 - 条目状态 158
 - 到期
 - 条目状态 158
 - 活动
 - 条目状态 158
 - 条目
 - 防止 165
 - 修剪 163
 - 条目状态 158
 - 修剪 168
 - 自动 163
 - db2Prune API 163
 - PRUNE HISTORY 命令 163
 - do_not_delete
 - 条目状态 158, 165
- 活动日志 9

[J]

- 级联赋值 100

- 集群
 - 脉动信号监视 8
 - HACMP 100
 - IP 地址接管 8
- 集群管理
 - 高可用性灾难恢复 (HADR) 37
- 集群管理软件
 - 受支持 100
 - 资源 63
 - 资源组 63
 - db2haicu 实用程序 62
- 集群域 62
 - 安装点 65
 - 路径 65
 - 数据库分区 65
 - 网络等值 64
 - 网络接口卡 (NIC) 64
 - 网络协议 64
 - 网络 IP 地址 64
 - 子网掩码 64
- 监视
 - 高可用性灾难恢复 (HADR) 141
 - progress
 - 备份 171
 - 崩溃恢复 171
 - 复原 171
 - rollforward 171
- 减少
 - 记录
 - 使用已声明的临时表 54
 - 使用 NOT LOGGED INITIALLY 参数 54
 - 介质故障的影响 197
 - 事务故障的影响 199
- 教程
 - 故障诊断 315
 - 问题确定 315
 - Visual Explain 314
- 节点
 - 同步 115
- 介质故障
 - 降低影响 197
 - 目录分区注意事项 197
 - 日志 154
- 镜像
 - 日志 14
- 卷动更新
 - 执行 129
- 卷动升级
 - 执行 129

[K]

- 可恢复数据库
 - 描述 151
- 可伸缩性
 - 多集群数据库 100

- 客户机重新路由
 - 高可用性灾难恢复 (HADR) 28
 - 示例 142
 - 限制 21
 - 与连接之间的交互超时 20
 - 自动 17
 - IBM 数据服务器 JDBC 和 SQLJ 驱动程序 21
- 客户机通信错误 17
- 客户机自动重新路由
 - 安装 17
 - 高可用性灾难恢复 (HADR) 28, 137
 - 连接失败 20
 - 路线图 11
 - 描述 17
 - 配置 19
 - 示例 142
 - 限制 21
- 克隆数据库
 - 创建 130
- 跨节点数据库恢复示例 188
- 快照备份 177, 259
 - 复原来自 214
 - 管理快照备份对象 169

[L]

- 历史记录文件
 - 访问 161
- 联机
 - 已归档日志 10
- 联机备份
 - 与其他实用程序的兼容性 184
- 联机表重组
 - 与联机备份的兼容性 184
- 联机创建索引
 - 与联机备份的兼容性 184
- 联机检查
 - 与联机备份的兼容性 184
- 联机索引重组
 - 与联机备份的兼容性 184
- 联机装入
 - 与联机备份的兼容性 184
- 连接失败
 - 客户机自动重新路由 20
- 连续可用性 108
- 两阶段落实
 - 协议 199
- 临时表空间
 - 和数据库重建 225
- 路线图
 - 客户机自动重新路由 11

[M]

- 脉动信号
 - 高可用性集群多处理 (HACMP) AIX 版 100
 - 监视 139, 141
 - Solaris 108
- 命令
 - db2adutl
 - 跨节点恢复示例 188
- 命名管道
 - 备份 180
- 目标映像
 - 用于重建 226

[P]

- 配置
 - 高可用性灾难恢复 30
 - 故障监视器 23
 - db2fm 24
 - db2fmc 25
 - 获取高可用性 17
- 配置参数
 - 数据库日志记录 47, 48
 - auto_del_rec_obj 168
 - logarchopt1
 - 跨节点恢复示例 188
 - TCP_KEEPALIVE 20
 - vendoropt
 - 跨节点恢复示例 188
- 配置自动维护向导 183

[Q]

- 前滚恢复
 - 表空间 205, 247
 - 配置文件参数支持 48
 - 日志管理注意事项 117
 - 日志序列 117
 - 数据库 205
- 切换数据库角色
 - 高可用性灾难恢复 (HADR) 148
- 切换 HADR 数据库角色 147

[R]

- 热备用配置
 - 描述 100
- 日志
 - 按需归档 119
 - 除去 124
 - 防止丢失 126
 - 分配 124
 - 管理 117
 - 归档 10

- 日志 (续)
 - 活动 9
 - 镜像 14
 - 目录, 已满 55
 - 配置参数 47
 - 数据库 9
 - 索引
 - 高可用性灾难恢复 (HADR) 29
 - 需要的存储器 154
 - 循环 9
 - 循环日志记录 124
 - 已联机归档 10
 - 已脱机归档 10
 - 用户出口程序 154
- 日志归档
 - 配置 34
- 日志镜像 131
- 日志控制文件
 - 概述 10
- 日志文件
 - 包括在备份映像中 125
 - 管理 139
 - 归档 56
 - 日志控制文件 10
- 日志文件管理
 - ACTIVATE DATABASE 命令 117
- 日志装入 131
 - 高可用性 14
- 容错 108
- 容器
 - 名称 173
- 冗余 7
- 软件磁盘阵列 197

[S]

- 设置写
 - 与联机备份的兼容性 184
- 声明 317
- 时间
 - 数据库恢复时间 153
- 时间戳记
 - 转换
 - 客户机/服务器环境 116
- 使已归档日志脱机 10
- 事件监视器
 - 高可用性集群多处理 (HACMP) AIX 版 100
- 示例
 - 备用服务器 142
 - 客户机自动重新路由 142
- 事务
 - 当日志目录已满时分块 55
 - 故障恢复
 - 崩溃 199
 - 降低故障的影响 194
 - 在发生故障的数据库分区服务器上 199

- 事务 (续)
 - 故障恢复 (续)
 - 在活动的数据库分区服务器上 199
- 书籍
 - 印刷版
 - 订购 310
- 数据
 - 按扇区进行的带奇偶校验的条带分割 (RAID 级别 5) 197
 - 恢复
 - 概述 149
- 数据库
 - 备份
 - 策略 151
 - 自动 182
 - 不可恢复 151
 - 重建
 - 表空间容器 225
 - 分区 231
 - 概述 222
 - 示例 235
 - 限制 232
 - 选择目标映像 226
 - 增量备份映像 230
 - 恢复
 - 策略 151
 - 记录
 - 概述 9
 - 配置参数 48
 - 循环 9
 - 临时表空间 225
 - 前滚恢复
 - 概述 205
 - 在高可用性灾难恢复 (HADR) 环境中激活 129
 - 在高可用性灾难恢复 (HADR) 环境中取消激活 129
- 数据库对象
 - 表空间更改历史记录文件 151
 - 恢复历史记录文件 151
 - 恢复日志文件 151
- 数据库分区
 - 使时钟同步 115
- 数据库分区服务器
 - 故障恢复 201
- 数据库服务器
 - 备用 21
- 数据库配置参数
 - 自动重新启动 194
- 双工
 - RAID 级别 1 197
- 双记录 14
- 按需应变的日志归档 119
- 索引
 - 为高可用性灾难恢复 (HADR) 记录 29

[T]

- 特权
 - BACKUP 实用程序 184
 - RESTORE 实用程序 233
 - Rollforward 实用程序 250
- 条款和条件
 - 出版物的使用 315
- 停止
 - 高可用性灾难恢复 (HADR) 128
- 同步
 - 恢复注意事项 115
 - 节点 115
 - 数据库分区 115
- 同步点管理器 (SPM)
 - 恢复不确定事务 202
- 同步方式
 - 高可用性灾难恢复 (HADR) 39
- 统计信息概要分析
 - 自动 183
- 统计信息收集
 - 自动 183
- 脱机备份
 - 与联机备份的兼容性 184
- 脱机装入
 - 与联机备份的兼容性 184

[W]

- 维护
 - 安排 45
- 文档
 - 概述 307
 - 使用条款和条件 315
 - 印刷版 307
 - PDF 307
- 问题确定
 - 教程 315
 - 可用的信息 315

[X]

- 系统要求
 - 高可用性灾难恢复 (HADR) 41
- 限制
 - 高可用性灾难恢复 (HADR) 44
- 相互接管配置 100
- 性能
 - 高可用性灾难恢复 (HADR) 35
 - 恢复 211
- 旋转赋值 100
- 循环日志记录 9, 124

[Y]

- 样本
 - 自动维护
 - 配置 47
 - 一致点
 - 数据库 194
- 已归档日志
 - 联机 10
 - 脱机 10
- 意外中断
 - 检测 141
- 硬件
 - 磁盘阵列 197
- 映像
 - 备份 173
- 用户出口程序
 - 备份 154
 - 错误处理 123
 - 调用格式 122
 - 归档日志文件 56
 - 检索日志文件 56
 - 日志 154
 - 数据库恢复 121
 - 样本程序
 - UNIX 121
 - Windows 121
- 用户出口数据库配置参数 48
- 用户定义的事件 100
- 优化
 - 备份性能 183
 - 复原性能 232
- 远程同步复制暂挂状态 134
- 远程同步复制状态 134

[Z]

- 灾难恢复
 - 概述 203
 - 高可用性灾难恢复 (HADR)
 - 概述 12
 - 要求 43
- 暂挂的 I/O 和磁盘镜像 131
- 暂挂状态
 - 描述 134, 171
- 暂挂 I/O 以支持连续可用性 15
- 增量备份映像
 - 重建数据库时使用 230
- 增量备份与恢复 207
- 增量复原 209, 216
- 站点故障
 - 高可用性灾难恢复 (HADR) 12
- 种子数据库
 - 复原
 - 现有数据库 215
 - 新数据库 216

- 主要重新集成
 - 高可用性灾难恢复 (HADR) 148
- 注册表变量
 - DB2_CONNRETRIES_INTERVAL 19
 - db2_connretries_interval 17
 - DB2_HADR_PEER_WAIT_LIMIT 35
 - DB2_HADR_SORCVBUF 35
 - DB2_HADR_SOSNDBUF 35
 - DB2_MAX_CLIENT_CONNRETRIES 19
 - db2_max_client_connretries 17
- 状态
 - 备用数据库 134
 - 暂挂 171
- 资源 63
- 资源组 63
- 自动备份
 - 启用 182
 - 样本配置 47
- 自动重新启动 194
- 自动重组
 - 配置
 - 样本 47
- 自动收集统计信息
 - 配置
 - 样本 47
- 自动维护 183
 - 备份 151, 182
 - 策略规范
 - 样本 47
 - 配置 46
 - 检索 46
 - AUTOMAINT_GET_POLICY 46
 - AUTOMAINT_GET_POLICYFILE 46
 - AUTOMAINT_SET_POLICY 46
 - AUTOMAINT_SET_POLICYFILE 46
- 自动增量复原
 - 局限性 210

A

- AIX
 - 备份与复原支持 155
- ASYNCR
 - 同步方式 39

B

- BACKUP 实用程序
 - 概述 173
 - 故障诊断 173
 - 监视进度 171
 - 使用时需要的权限和特权 184
 - 示例 186
 - 显示信息 173
 - 限制 175

BACKUP 实用程序 (续)

性能 183

BACKUP DATABASE 命令 175

blk_log_dsk_ful 配置参数

概述 48

C

CLP (命令行处理器)

示例

备份 186

重定向复原会话 233

复原会话 233

前滚会话 250

数据库重建会话 235

connectTimeout

与客户机重新路由之间的交互 20

D

DB2 高级副本服务 (ACS)

安装 258

概述 257

激活 259

目录 261

配置 259

启用 257

设置脚本 setup.sh 261

限制 301

最佳实践 301

DB2 高级副本服务 (ACS) API

操作系统 302

返回码 301

概述 261

函数

概述 262

db2ACSBEGINOperation 267

db2ACSBEGINQuery 271

db2ACSDelete 282

db2ACSEndOperation 269

db2ACSEndQuery 274

db2ACSGetNextObject 272

db2ACSInitialize 262

db2ACSPartition 278

db2ACSPrepare 265

db2ACSQueryApiVersion 262

db2ACSRetrieveMetaData 285

db2ACSSnapshot 276

db2ACSStoreMetaData 283

db2ACSTerminate 264

db2ACSVerify 280

数据结构

概述 287

db2ACS_BackupDetails 287

db2ACS_CB 287

DB2 高级副本服务 (ACS) API (续)

数据结构 (续)

db2ACS_CreateObjectInfo 288

db2ACS_DB2ID 289, 296

db2ACS_GroupList 289

db2ACS_LoadcopyDetails 290

db2ACS_LogDetails 290

db2ACS_MetaData 300

db2ACS_ObjectInfo 291

db2ACS_ObjectStatus 292

db2ACS_OperationInfo 293

db2ACS_Options 294

db2ACS_PartitionEntry 294

db2ACS_PartitionList 294

db2ACS_PathEntry 295

db2ACS_PathList 296

db2ACS_QueryInput 296

db2ACS_QueryOutput 296

db2ACS_ReadList 298

db2ACS_ReturnCode 298

db2ACS_SessionInfo 299

db2ACS_SnapshotDetails 300

db2ACS_VendorInfo 300

硬件 302

DB2 高可用性实例配置实用程序 (db2haicu)

定额设备 63

故障诊断 98

集群环境 60

集群域 62

创建 96

维护 97

检测到的数据库路径 97

描述 66

启动方式 67

维护方式 67

先决条件 96

限制 98

运行

交互方式 68

XML 输入文件 68, 86

DB2 高可用性 (HA) 功能部件

概述 13

集群管理器

集成 58

API 100

集群配置 59

DB2 信息中心

版本 311

查看各种语言版本 311

更新 312, 313

语言 311

db2adutil 命令

跨节点恢复示例 188

db2Backup API

备份数据 175

db2fm 命令
 故障监视器概述 12

db2haicu 实用程序
 输入文件的 XML 模式 69
 ClusterDomainType 71
 ClusterNodeType 77
 DB2ClusterType 69
 DB2PartitionSetType 79
 DB2PartitionType 80
 FailoverPolicyType 78
 HADBDfn 86
 HADBType 85
 HADRDBDfn 84
 HADRDBType 83
 InterfaceType 76
 IPAddressType 76
 MountType 82
 MutualPolicyType 82
 NPlusMPolicyType 83
 PhysicalNetworkType 74
 QuorumType 73

输入文件样本
 db2ha_sample_DPF_mutual.xml 88
 db2ha_sample_DPF_NPlusM.xml 91
 db2ha_sample_HADR.xml 94
 db2ha_sample_sharedstorage_mutual.xml 87

db2inidb 命令
 创建分割镜像 178
 概述 15

db2Recover API
 恢复数据 187

db2Restore API
 恢复数据库或表空间 213

db2Rollforward API
 将事务应用于复原的备份映像 246

db2tapemgr 命令
 将日志文件归档到磁带 119

db2uext2 程序
 调用格式 122
 描述 121

DB2_CONNRETRIES_INTERVAL 注册表变量
 描述 19

DB2_MAX_CLIENT_CONNRETRIES 注册表变量
 配置客户机自动重新路由重试行为 19

DB_HISTORY 管理视图
 查看恢复历史记录文件条目 161

E

EXPORT 实用程序
 联机备份兼容性 184

G

GET SNAPSHOT 命令
 HADR 备用数据库状态 136

H

HACMP (高可用性集群多处理)
 描述 100

HADR (高可用性灾难恢复)
 请参阅高可用性灾难恢复 (HADR) 12

HP on IPF
 备份 155
 复原 155

HP-UX
 备份 155
 复原 155

I

IBM Tivoli SA MP 59

IBM Tivoli Storage Manager (TSM)
 使用 255
 使用 BACKUP DATABASE 命令
 需要的最低级别 255
 使用 RESTORE DATABASE 命令
 需要的最低级别 255

L

Linux
 备份与复原支持
 AMD64 和 Intel EM64T 155
 IA-64 155
 Power PC 155
 zSeries 155

LIST HISTORY 命令 157

logarchmeth1 配置参数
 和 HADR (高可用性灾难恢复) 34

logarchmeth2 配置参数
 和 HADR (高可用性灾难恢复) 34

logarchopt1 配置参数
 跨节点恢复示例 188

LOGBUFSZ 配置参数 48

logfilsiz 配置参数 48
 和 HADR (高可用性灾难恢复) 30

logprimary 配置参数 48

logretain 数据库配置参数 48

logsecond 配置参数
 描述 48

M

maxRetriesForClientReroute 17

Microsoft 服务器 104

Microsoft 故障转移集群服务器 104
mincommit 数据库配置参数 48
mirrorlogpath 配置参数 14
mirrorlogpath 数据库配置参数 48

N

NEARSYNC 同步方式 39
newlogpath 数据库配置参数 48
nodedown 事件 100
nodeup 事件 100

O

overflowlogpath 数据库配置参数 48

R

RAID (独立磁盘冗余阵列) 设备
 级别 1 (磁盘镜像或双工) 197
 级别 5 (按扇区进行的数据和带奇偶校验的条带分割) 197
RECOVER DATABASE 命令 187
 所需的权限和特权 212
RESTART DATABASE 命令 194
RESTORE 实用程序
 重新定义表空间容器 218
 复原到现有的数据库 215
 复原到新的数据库 216
 概述 213
 监视进度 171
 使用时需要的权限和特权 233
 示例 233
 限制 213
 性能 213, 232
 与联机备份的兼容性 184
RESTORE DATABASE 命令 213
retryIntervalForClientReroute 17
ROLLFORWARD
 最短恢复时间 180, 247
Rollforward 实用程序
 概述 245
 恢复已删除的表 193
 监视进度 171
 使用时需要的权限和特权 250
 示例 250
 限制 246
 与联机备份的兼容性 184
ROLLFORWARD DATABASE 命令 246
RUNSTATS 实用程序
 与联机备份的兼容性 184

S

Solaris 操作系统
 备份与复原支持 155

SP 帧 100
SQL 语句
 显示帮助 311
START HADR 命令 137
STOP HADR 命令 137
Sun Cluster 3.0
 高可用性 110
SYNC
 同步方式 39

T

TAKEOVER HADR 命令 137
 切换数据库角色 147
 执行故障转移操作 145
TCP_KEEPAALIVE
 操作系统配置参数 20
Tivoli Storage Manager (TSM)
 分区表 181
Tivoli System Automation for Multiplatforms
 高可用性 103

V

vendoropt 配置参数
 跨节点恢复示例 188
VERITAS Cluster Server 112
 高可用性 112
Visual Explain
 教程 314

W

Windows 操作系统
 故障转移 104

[特别字符]

instance_name.nfy 日志文件
 管理通知日志消息 139



Printed in China

S151-1187-00



Spine information:

IBM DB2 9.7 Linux 版、UNIX 版和 Windows 版

数据恢复及高可用性指南与参考

