

**GBASE<sup>®</sup>**

**GBase 8s 备份与恢复指南**



GBase 8s 备份与恢复指南，南大通用数据技术股份有限公司

GBase 版权所有©2004-2030，保留所有权利

## 版权声明

本文档所涉及的软件著作权及其他知识产权已依法进行了相关注册、登记，由南大通用数据技术股份有限公司合法拥有，受《中华人民共和国著作权法》、《计算机软件保护条例》、《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

## 免责声明

本文档包含的南大通用数据技术股份有限公司的版权信息由南大通用数据技术股份有限公司合法拥有，受法律的保护，南大通用数据技术股份有限公司对本文档可能涉及到的非南大通用数据技术股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅，并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经南大通用数据技术股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将视为侵权，南大通用数据技术股份有限公司具有依法追究其责任的权利。

本文档中包含的信息如有更新，恕不另行通知。您对本文档的任何问题，可直接向南大通用数据技术股份有限公司告知或查询。

## 通讯方式

南大通用数据技术股份有限公司

天津市高新区开华道22号普天创新产业园东塔20-23层

电话：400-013-9696

邮箱：[info@gbase.cn](mailto:info@gbase.cn)

## 商标声明

**GBASE<sup>®</sup>** 是南大通用数据技术股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由南大通用数据技术股份有限公司合法拥有，受法律保护。未经南大通用数据技术股份有限公司书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯南大通用数据技术股份有限公司商标权的，南大通用数据技术股份有限公司将依法追究其法律责任。

## 目 录

1 简介 .....	1
2 备份与恢复的概述.....	1
2.1 备份与恢复概念.....	1
2.1.1 恢复系统.....	2
2.1.2 gbackuprestore 和 gtape 实用程序的比较.....	9
2.2 计划备份与恢复.....	11
2.2.1 计划恢复策略.....	11
2.2.2 为生产数据库服务器计划备份系统.....	14
3 gbackuprestore 备份与恢复系统 .....	17
3.1 gbackuprestore 备份与恢复系统的概述 .....	18
3.1.1 gbackuprestore 组件 .....	18
3.1.2 gbackuprestore 的存储管理 .....	22
3.2 配置存储管理器和 gbackuprestore .....	22
3.2.1 配置存储管理器.....	22
3.2.2 验证存储管理器.....	30
3.2.3 配置 gbackuprestore .....	30
3.2.4 验证 gbackuprestore 和存储管理器的配置.....	32
3.2.5 gbackuprestore 和存储管理器使用的文件.....	32
3.3 使用 gbackuprestore 备份 .....	36
3.3.1 准备备份数据.....	36
3.3.2 gbackuprestore -b 语法: 备份 .....	38
3.3.3 gbackuprestore -m 语法: 监视最近的 gbackuprestore 活动.....	45
3.3.4 查看注册备份的列表.....	45
3.3.5 gbackuprestore -P 语法: 打印备份逻辑日志 .....	47
3.3.6 gbackuprestore -v 语法: 验证备份 .....	50
3.4 使用 gbackuprestore 恢复数据 .....	55
3.4.1 预恢复核对表.....	56
3.4.2 存储设备可用性.....	58
3.4.3 gbackuprestore -r 语法: 恢复数据.....	58
3.4.4 恢复期间更换磁盘.....	70
3.4.5 恢复到其他计算机.....	72

3.4.6 gbackuprestore -RESTART 语法: 重新启动失败的恢复.....	77
3.4.7 解决失败的恢复.....	79
3.5 外部备份与恢复.....	81
3.5.1 外部备份与恢复概述.....	81
3.5.2 RS 辅助服务器外部备份.....	85
3.5.3 在外部恢复中恢复数据.....	86
3.5.4 执行外部恢复.....	89
3.5.5 使用外部备份与恢复来初始化 HDR.....	92
3.6 定制和维护 gbackuprestore.....	92
3.6.1 定制 gbackuprestore 和存储管理器命令.....	92
3.6.2 使备份目录到期和同步.....	95
3.6.3 监视 gbackuprestore 和存储管理器的性能.....	102
3.7 gbackuprestore 目录表.....	105
3.7.1 bar_action 表.....	105
3.7.2 bar_instance 表.....	106
3.7.3 bar_ixbar 表.....	108
3.7.4 bar_object 表.....	110
3.7.5 bar_server 表.....	111
3.7.6 bar_syncdeltab 表.....	112
3.7.7 gbackuprestore 目录映射.....	112
3.8 gbackuprestore 消息和返回码.....	113
3.8.1 消息格式.....	113
3.8.2 消息编号.....	113
3.8.3 gbackuprestore 返回码.....	114
4 gtape 备份与恢复系统.....	123
4.1 配置 gtape.....	123
4.1.1 设置 gtape 实用程序的配置参数.....	123
4.1.2 更改 gtape 配置.....	129
4.2 使用 gtape 备份.....	129
4.2.1 gtape 任务摘要.....	129
4.2.2 更改数据库日志记录状态.....	130
4.2.3 创建备份.....	131
4.2.4 使用 gtape 备份逻辑日志文件.....	143
4.3 使用 gtape 恢复.....	146

4.3.1 物理恢复的类型.....	147
4.3.2 冷恢复、热恢复或混合恢复.....	147
4.3.3 gtape 实用程序执行恢复的语法.....	148
4.3.4 恢复整个系统.....	151
4.3.5 使用 gtape 配置连续日志恢复.....	157
4.3.6 恢复期间重命名块.....	157
4.3.7 从标准输入恢复.....	160
4.3.8 将数据恢复到远程服务器.....	161
4.3.9 使用标准 I/O 同时备份与恢复.....	161
4.4 执行外部备份与恢复.....	162
4.4.1 使用外部备份和恢复来恢复数据.....	162
4.4.2 在外部备份中备份的数据.....	163
4.4.3 准备进行外部备份.....	164
4.4.4 在外部恢复中恢复的数据.....	165
5 GBase 8s Primary Storage Manager.....	168
5.1 GBase 8s Primary Storage Manager.....	168
5.2 示例：使用 GBase 8s Primary Storage Manager 管理存储设备.....	170
5.3 设置 GBase 8s Primary Storage Manager.....	176
5.3.1 收集有关文件目录和设备的信息.....	176
5.3.2 配置 GBase 8s Primary Storage Manager.....	177
5.4 管理存储设备.....	177
5.5 用于存储管理的 gpsm 实用程序.....	178
5.5.1 gpsm -C detail 输出.....	184
5.5.2 gpsm -D list 输出.....	186
5.5.3 gpsm -O list 输出.....	187
5.6 设备池.....	187
5.7 GBase 8s Primary Storage Manager 的设备配置文件.....	188
5.8 GBase 8s Primary Storage Manager 文件命名约定.....	189
5.9 GBase 8s Primary Storage Manager 的消息日志.....	190
6 archecker 表级别恢复实用程序.....	190
6.1 archecker 表级别恢复实用程序.....	190
6.1.1 archecker 实用程序概述.....	190
6.1.2 使用 archecker 进行数据恢复.....	192
6.1.3 archecker 实用程序命令的语法.....	194

6.1.4 archecker 模式引用 .....	198
7 备份与恢复配置参数引用 .....	207
7.1 备份与恢复配置参数 .....	207
7.1.1 onconfig 文件中的 gbackupstore 和 gtape 配置参数 .....	208
7.1.2 archecker 实用程序配置参数和环境变量 .....	229
7.1.3 GBase 8s Primary Storage Manager 配置参数 .....	235
7.1.4 事件警报配置参数 .....	240
8 附录 .....	240
8.1 对一些备份与恢复错误进行故障诊断 .....	240
8.1.1 归档期间页面毁坏 .....	240
8.1.2 已在运行的日志备份 .....	241
8.1.3 恢复期间没有服务器连接 .....	242
8.1.4 恢复之前删除数据库 .....	242
8.1.5 备份或恢复期间没有数据库空间或 Blob 空间 .....	242
8.1.6 在备份系统上更改系统时间 .....	243
8.2 迁移数据、服务器和工具 .....	243
8.2.1 数据库服务器或存储管理器升级之前进行备份 .....	243
8.2.2 升级第三方存储管理器 .....	244
8.2.3 更改存储管理器供应商 .....	244
8.2.4 从 gtape 切换到 gbackupstore .....	245
8.3 GLS 支持 .....	245
8.3.1 将 GLS 与 gbackupstore 实用程序一起使用 .....	245
8.3.2 将 GL_DATETIME 环境变量与 gbackupstore 一起使用 .....	246
8.3.3 将 GLS 与 gtape 实用程序一起使用 .....	247

# 1 简介

这些主题描述如何使用 GBase 8s `gbackupstore` 和 `gtape` 实用程序来备份与恢复数据库服务器数据。这些实用程序使您可以在由于硬件或软件发生故障或事故而导致数据丢失或损坏的情况下恢复数据库。

这些主题可以为以下用户提供帮助：

- 数据库管理员
- 系统管理员
- 备份操作员
- 技术支持人员

编写这些主题时假定您掌握了以下背景知识：

- 对存储管理器（管理包含备份的存储设备和介质的应用程序）有一些经验
- 对于计算机、操作系统和操作系统提供的实用程序的应用知识
- 使用关系数据库的相关经验或者了解数据库概念

有数据库服务器管理、操作系统管理或网络管理方面的相关经验

如果您对关系数据库、SQL 或操作系统的经验有限，请参阅数据库服务器的《GBase 8s 启动指南》以获取补充标题的列表。

编写这些主题时，假定您使用的是某种 GBase 8s 数据库服务器。

## 2 备份与恢复的概述

这些主题提供备份与恢复概念的概述。它们还提供了有关计划备份与恢复操作的信息。

### 2.1 备份与恢复概念

GBase 8s 提供用于备份与恢复数据库服务器数据的两个实用程序。这两个实用程序将备份与恢复存储空间和逻辑日志。但是，它们支持不同的功能部件，因此请务必了解其差异。这些主题说明了 GBase 8s 数据库服务器的基本备份与恢复概念，并比较 `gbackupstore` 和 `gtape` 实用程序。

`gbackupstore` 使用存储管理器对存储空间（数据库空间）和逻辑文件进行备份与恢复，但 `gtape` 不使用存储管理器。

### 2.1.1 恢复系统

恢复系统包含备份与恢复系统，它允许您备份数据库服务器的数据，并在当前数据毁坏或无法访问的情况下随即恢复备份的数据。

数据毁坏或丢失的原因可能从程序错误到磁盘故障，或直到损坏整个设备的灾难性事故等。恢复系统使您可以恢复在类似的灾难性事故中丢失的数据。

#### 备份系统

备份是数据库服务器维护的一个或多个数据库空间（也称为存储空间）和逻辑日志的副本。您还可以备份 Blob 空间和智能大对象空间。

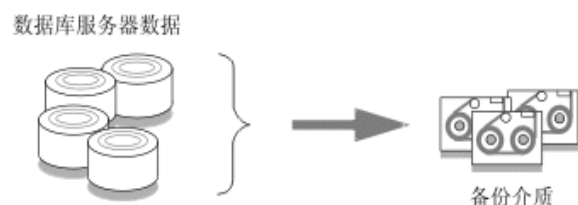
备份副本通常会写入辅助存储介质，例如磁盘或磁带。以脱机方式存储介质，并且如有可能，请保存一个非现场的副本。

备份副本通常会写入辅助存储介质，例如磁盘、磁带或光盘。以脱机方式存储介质，并且如有可能，请保存一个非现场的副本。

**重要：** 数据库备份不会替换常规的操作系统备份，后者备份除了 GBase 8s 数据库文件之外的其他文件。

下图说明了数据库备份的基本概念。

**图：数据库服务器数据的备份**



不必始终备份所有的存储空间。如果某些表每天都更改而其他一些则很少更改，那么每次备份数据库服务器时都备份包含未更改表的存储空间，这将导致效率低下。因此必须仔细地规划备份调度以避免备份或恢复数据时较长的延迟。

#### 备份级别

为了提供灵活性，gbackuprestore 和 gtape 实用程序支持三个备份级别。

##### 0 级

0 级备份将备份指定存储空间内所有包含数据的已使用的页。

您需要所有这些页将数据库恢复到备份时所处的状态。



0 级备份可能耗时比较长，因为 gbackupstore 会写入所有磁盘页面以备份介质。1 级和 2 级备份花费的时间有可能几乎与 0 级备份相同，这是因为数据库服务器必须扫描所有的数据以确定自上次备份以来更改的内容。从 0 级、1 级和 2 级备份恢复数据的时间比从 0 级备份和一长串逻辑日志备份恢复数据花费的时间要少。

## 1 级

1 级备份只备份自上次指定的存储空间进行 0 级备份后更改的数据。

所有已更改的表和索引页（包含带有已删除数据的那些页面）都将进行备份。复制到备份的数据反映 1 级备份开始时更改过的数据的状态。

1 级备份占用的空间和花费的时间比 0 级备份要少，因为前者只将上次 0 级备份后更改的数据复制到存储管理器中。

## 2 级

2 级备份只备份自上次指定的存储空间进行 1 级备份后更改的数据。

2 级备份包含自上次 1 级备份后在存储空间中更改过的每个表和索引页的副本。

2 级备份占用的空间和花费的时间比 1 级备份要少，因为前者只将上次 1 级备份后更改的数据复制到存储管理器中。

**重要：** 如果磁盘和其他介质损坏并需要更换，您需要对所有存储空间和相关逻辑日志至少进行 0 级备份，才能在更换硬件上完全恢复数据。

## 逻辑日志备份

逻辑日志备份是所有填满的逻辑日志文件在磁盘或磁带上的副本。逻辑日志文件存储发生在备份间的数据库服务器活动记录。

要释放填满的逻辑日志文件，首先要备份它们。数据库服务器将重用这些已释放的逻辑日志文件用于记录新事务。有关逻辑日志的完整描述，请参阅《GBase 8s 管理员指南》。

**限制：** 即使没有指定为数据库或表记录日志，您仍然需要备份逻辑日志，因为它们包含了管理信息，例如检查点记录和块的添加和删除。如果备份了这些逻辑日志文件，即使不为任何数据库使用日志记录仍然可以进行热恢复。

### 手动和连续逻辑日志备份

您可以手动备份逻辑日志，也可以启用连续逻辑日志备份。

手动逻辑日志备份将备份所有已满的逻辑日志文件，并在当前逻辑日志文件处停止。必须仔细监视逻辑日志，并根据需要启动逻辑日志备份。

要了解逻辑日志文件是否已准备好进行备份，请检查 `gstat -l` 的标志字段。当逻辑日志文件标记为已备份后，它可以被重新使用。当标志字段显示以下值中的任意一个时，逻辑日志文件已准备好进行备份：

U-----

## U-----L

值 **U** 表示逻辑日志文件已被使用。值 **L** 表示最近的检查点发生时指示的逻辑日志文件是当前文件。值 **C** 指示当前日志。如果 **B** 出现在第三列，那么逻辑日志文件已备份并可以重新使用。

## U-B---L

标志值 **U---C-L** 或 **U---C--** 表示当前逻辑日志。虽然允许您备份当前逻辑日志，但这样做将强制执行日志切换，从而浪费逻辑日志空间。等到逻辑日志文件填满后才备份它。

如果开启连续逻辑日志备份，数据库服务器将自动备份每个要填满的逻辑日志。如果关闭连续逻辑日志备份，那么继续填充逻辑日志文件。如果所有逻辑日志都已填满，数据库服务器会挂起，直到备份了这些日志为止。您可以通过在 `onconfig` 文件中设置

`ALARMPROGRAM` 配置参数或者通过运行 `gbackuprestore` 或 `gtape` 命令来启动连续逻辑日志备份。

### 日志回收

当数据库服务器处于脱机状态时，您可以执行特别的逻辑日志备份，称为日志回收。在日志回收中，数据库服务器直接从磁盘访问日志文件。日志回收将备份所有还未备份并且还未毁坏或损坏的逻辑日志。

日志回收使您可以将所有数据恢复到最近一个可用的并且没有被毁坏的逻辑日志文件以及最近一次完整的事务中。

### 保存逻辑日志备份

您应该频繁执行逻辑日志备份，然后从至少最近两个 0 级备份保存逻辑日志备份，这样就可以使用它们来完成恢复。

经常进行逻辑日志备份，原因如下：

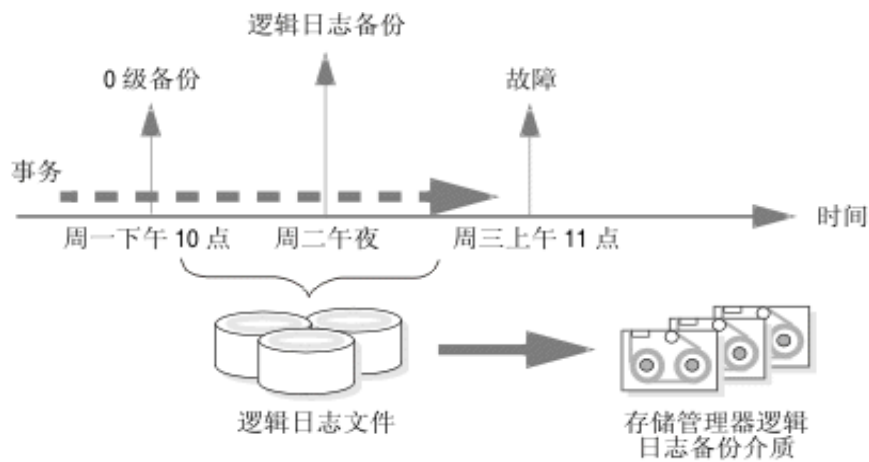
- 释放已满的逻辑日志文件
- 当包含逻辑日志的磁盘出现故障时将数据丢失降低到最小限度
- 确保恢复包含一致的以及最近的事务

您应该从最近两个 0 级备份保存逻辑日志备份，因为如果某个 0 级备份不可访问或无法使用，您可以从较旧备份中恢复数据。如果所有逻辑日志备份都是不可访问或无法使用的，那么无法从这些逻辑日志文件或任意后继逻辑日志文件中前滚这些事务。

**重要：** 您会丢失未备份或未回收的逻辑日志文件中的事务。

为举例说明，如下图所示，假定您在星期一晚上 10 点执行 0 级备份，并接着在星期二午夜备份逻辑日志。在星期三上午 11 点发生灾难性事故，数据库遭到毁坏。除非您设置了连续逻辑日志备份，否则您将不能恢复星期二午夜和星期三上午 11 点之间发生的事务。

如果包含带有逻辑日志的存储空间的磁盘受损，那么星期二午夜后的事务将丢失。要从最近的逻辑日志备份中恢复这些事务，请尝试在修理或更换坏磁盘前回收这些逻辑日志并随后执行冷恢复。

**图: 存储空间和逻辑日志备份**

## 恢复系统

恢复就是从已备份的存储空间和逻辑日志文件中重新创建数据库服务器的数据。

由于以下任一条件造成数据库服务器数据不可访问时，恢复将重新创建这些数据：

- 需要更换包含数据库服务器数据的出故障的磁盘。
- 程序的逻辑错误毁坏了数据库。
- 需要将数据库服务器数据移到新的计算机上。
- 用户意外地毁坏或破坏了数据。

要将数据恢复到发生故障时的状态，必须至少具有在发生故障之前每个存储空间的一个 0 级备份以及包含存储空间备份后所有事务的逻辑日志文件。

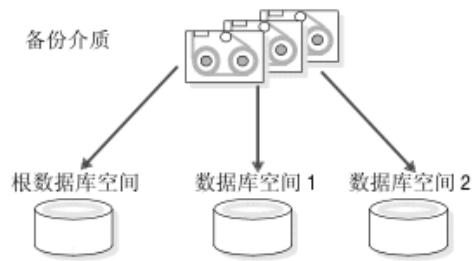
## 物理和逻辑恢复

`gbackuprestore` 和 `gtape` 分两个阶段恢复数据库服务器的数据。第一个阶段是物理恢复，它从所有或选定存储空间的备份中恢复数据。第二个阶段是逻辑恢复，它从逻辑日志备份中恢复事务。

### 物理恢复

在物理恢复期间，`gbackuprestore` 或 `gtape` 从最近的 0 级、1 级和 2 级备份恢复数据。当遇到磁盘故障时，可以只将驻留在出故障的磁盘上带有块的那些存储空间恢复到新磁盘上。下图对物理恢复进行了说明。

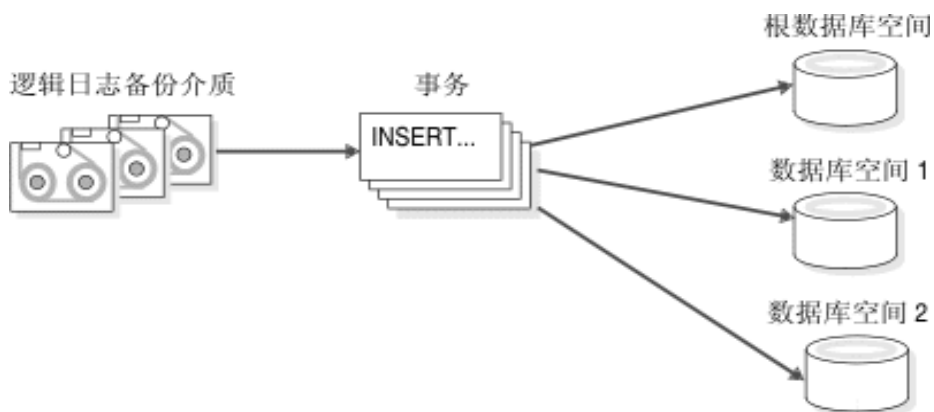
**图: 物理恢复**



### 逻辑恢复

如下图所示，数据库服务器重放逻辑日志来重新应用上次备份后发生的所有数据库事务。逻辑恢复仅应用于已物理恢复的存储空间上。

**图: 逻辑恢复**



数据库服务器将自动分辨要恢复哪些逻辑日志。

有关更多信息，请参阅使用 `gbackuprestore` 恢复数据和使用 `gtape` 恢复。

### 热恢复、冷恢复和混合恢复

恢复数据时，您必须确定是在数据库服务器处于停顿、联机还是脱机方式时进行该操作。恢复的类型取决于服务器处于其中哪一种操作方式。

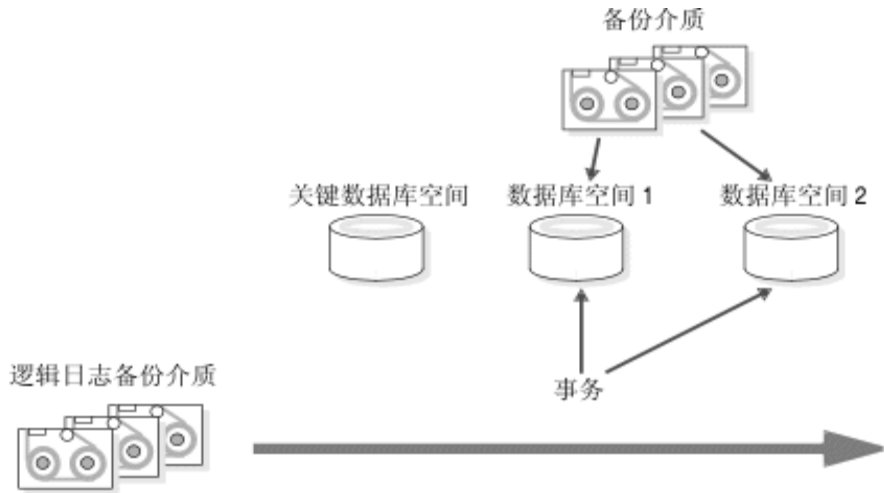
恢复的种类如下：

- 如果在数据库服务器处于联机或静默状态时恢复非关键数据库空间，那么该过程称为热恢复。
- 当 GBase 8s 处于脱机状态时，您只能执行冷恢复。
- 混合恢复是对某些存储空间进行冷恢复后接着对其余的存储空间进行热恢复。

### 热恢复

如下图所示，热恢复将恢复非关键的存储空间。热恢复由一个或多个物理恢复、一个逻辑日志备份以及一个逻辑恢复组成。

**图: 热恢复**

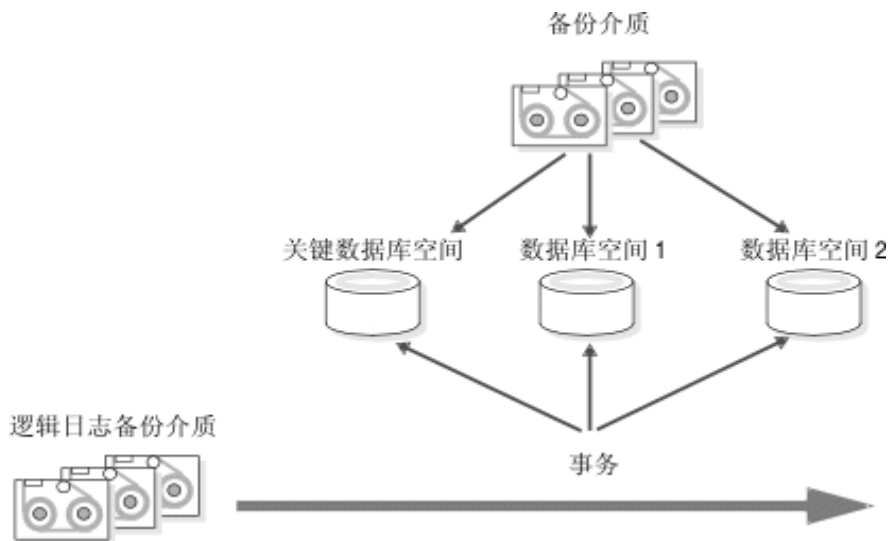


不能同时执行多个热恢复。

冷恢复

如下图所示，冷恢复会回收逻辑日志，并恢复关键数据库空间（根数据库空间以及包含物理日志和逻辑日志文件的数据库空间）、其他存储空间以及逻辑日志。

**图: 冷恢复**



通过在恢复期间为任何块提供新路径名和偏移量，可以在与执行备份的计算机不同的计算机上执行冷恢复。

恢复整个系统备份时，不需要恢复逻辑日志。整个系统备份包含在执行备份时整个实例的快照，它在所有数据库空间之间具有逻辑一致性。

当恢复标准备份时，您必须执行逻辑恢复来恢复逻辑日志。

冷恢复首先对所有关键存储空间进行物理恢复，接着恢复非关键存储空间，最后恢复逻辑日志。根数据库空间的保留页恢复后，数据库服务器进入恢复方式。当逻辑恢复完成后，数据库服务器进入停顿方式。使用 `gadmin` 命令使数据库服务器变成联机状态。

**提示：** 如果镜像关键数据库空间，那么您不太可能在磁盘故障后执行冷恢复，因为数据库服务器可以使用镜像的存储空间。如果镜像逻辑日志空间，当一个或多个磁盘出现故障时更可能回收逻辑日志数据。

**必需：** 恢复复制前，Enterprise Replication 服务器需要进行冷恢复。

### 混合恢复

混合恢复可使关键数据更快可用，但是，完整恢复需要更长的时间，因为逻辑日志将进行多次恢复和重放，初始冷恢复时一次，每个后续热恢复时各一次。

存储在冷恢复中的初始存储空间集合必须包含服务器中的所有关键存储空间。如果在初始冷恢复期间并未达到恢复所有存储空间的程度，那么可以避免恢复它们所必需的时间，从而与执行整个服务器的冷恢复相比，您可以使服务器更快地处于联机状态。然后您可以在一个或多个热恢复中恢复剩余的存储空间。

在冷恢复期间没有恢复的存储空间即使可能没有被故障损坏，也只有在热恢复中对它们进行恢复以后才可用。

### 连续日志恢复

连续日志恢复可保持辅助系统可用，以在恢复日志的主系统发生故障时替换主系统。

标准日志恢复将恢复所有可用日志文件备份并应用日志记录。在最后一个可用日志恢复和应用之后，日志恢复结束。仍处于打开状态的事务将回滚到事务清除阶段，然后服务器将处于停顿模式。服务器停顿后，将不能恢复更多的逻辑日志。

通过连续日志恢复（而不是事务清除），在恢复最后一个可用日志后，服务器将处于日志恢复暂挂状态。恢复客户机（`gtape` 或 `gbackuprestore`）退出并将控制权返回给您。对处于此状态的服务器，您可以在其他逻辑日志可用之后，启动另一个逻辑恢复。一旦将每个日志恢复作为连续日志恢复启动，您可以不断地继续该循环。

连续日志恢复的作用之一就是在主系统故障的情况下保持辅助系统可用。可以在辅助系统上恢复曾经在主系统上备份的逻辑日志（当逻辑日志可用时）。如果主系统发生故障，那么可以在辅助系统上恢复剩下的可用逻辑日志，并使辅助系统联机作为新的主系统。

连续的日志恢复所需的网络带宽远比高可用性数据复制（HDR）和企业数据复制（ER）所需的带宽少。连续的日志恢复比 HDR 和 ER 更便捷，因为您可以在任意时间启动连续的

日志恢复。因此，在不可预料的情况下（如网络中断时），连续的日志恢复比 HDR 或 ER 更稳健。

有关更多信息，请参阅通过使用 `gbackuprestore` 来配置连续日志恢复和使用 `gtape` 配置连续日志恢复。

## 2.1.2 gbackuprestore 和 gtape 实用程序的比较

本主题包含的信息可帮助您比较 `gbackuprestore` 和 `gtape` 实用程序，以便您可以决定何时使用每个实用程序。

### `gbackuprestore`

通过使用存储管理器跟踪备份和存储介质，从而备份并恢复存储空间（数据库空间）和逻辑文件。当需要执行以下操作时，请使用该实用程序：

- 选择具体存储空间
- 备份到具体时间点
- 执行单独的物理和逻辑恢复
- 并行备份与恢复多个不同的存储空间
- 并发使用多个磁带机进行备份与恢复
- 执行导入的恢复
- 执行外部备份与恢复

### `gtape`

记录、备份与恢复数据，能够更改数据库的日志记录状态。不使用存储管理器。当需要执行以下操作时，请使用该实用程序：

- 在不使用存储管理器的情况下备份与恢复数据
- 备份但不选择存储空间
- 更改数据库的日志记录方式

**重要：** `gtape` 和 `gbackuprestore` 生成的备份不兼容。不能使用 `gtape` 创建一个备份再使用 `gbackuprestore` 来将其恢复，反之亦然。

下表比较 `gbackuprestore` 与 `gtape`。

表 1. `gbackuprestore` 和 `gtape` 之间的差异

实用程序能够……	<code>gbackuprestore</code>	<code>gtape</code>
使用存储管理器来跟踪备份和存储介质吗？	是	否

实用程序能够……	gbackup restore	gtape
备份所有的数据库服务器数据吗？	是	是
备份选定的存储空间吗？	是	否
备份逻辑日志文件吗？	是	是
执行连续逻辑日志备份吗？	是	是
执行连续逻辑日志恢复？	是	是
在数据库服务器处于联机状态时进行备份吗？	是	是
在数据库服务器处于停顿方式时进行备份吗？	是	是
恢复所有的数据库服务器数据吗？	是	是
恢复选定的存储空间吗？	是	是
串行地备份并恢复存储空间吗？	是	是
是否在数据库服务器脱机时执行冷恢复？	是	是
初始化高可用性数据复制吗？	是	是
将数据恢复到特定的时间点吗？	是	否
执行单独的物理和逻辑恢复吗？	是	是
并行地备份并恢复不同的存储空间吗？	是	否
并发使用多个磁带机进行备份与恢复吗？	是	否
重新启动恢复吗？	是	否
是否在冷恢复期间重命名块路径名或设备？	是	是
执行导入的恢复吗？	是	是
执行外部备份与恢复吗？	是	是
监视性能吗？	是	否
更改数据库的日志记录方式吗？	否	是



实用程序能够……	gbackup restore	gtape
是否使用外部程序变换数据？	是	是
是否从云存储进行备份或恢复？	否	是

其他差异：

- 紧急引导文件和 sysutils 数据库

gtape 实用程序不使用 sysutils 数据库或紧急引导文件。

- 同时会话

用于 GBase 8s 主存储管理器的 gbackuprestore 支持同时发生的会话。

带有 Storage Manager 的 gbackuprestore 最多支持每个 Storage Manager 实例同时有 4 个会话。gtape 实用程序支持两个同时发生的会话，一个用于物理备份或恢复，而另一个用于日志备份。

- 设备支持和存储管理

gtape 实用程序支持其他主机上的远程备份设备。

用于 GBase 8s Primary Storage Manager 的 gbackuprestore 支持将备份生成导出至指定的目录和设备。

gbackuprestore 支持各种硬件平台上不同集合的磁带机。

也可以将 gbackuprestore 用于 Tivoli Storage Manager 或第三方存储管理器，以获取设备支持和存储管理。

- 更改数据库的日志记录方式

您不能更改 gbackuprestore 的日志记录方式；但您可以在使用 gbackuprestore 时使用 gdblogmode 实用程序来完成此任务。

您还可以使用 SQL 管理 API 备选项 ALTER LOGMODE 来更改日志记录方式。

有关每个实用程序的详细信息，请参阅使用 gbackuprestore 备份和使用 gtape 备份。

## 2.2 计划备份与恢复

这些主题描述了如何计划备份与恢复，例如，通过计划恢复策略与备份系统来进行。

### 2.2.1 计划恢复策略

使用 gbackuprestore 或 gtape 之前，请规划您的恢复目标。

### 数据丢失的类型

计划恢复策略时的第一步是确定可接受的数据丢失量（如有）。

可能发生以下类型的数据丢失：

- 以下内容的删除：
  - 行、列、表或数据库
  - 块、存储空间或逻辑日志
- 数据毁坏或产生了不正确的数据
- 硬件故障（例如包含块文件的磁盘故障或备份磁带磨损）
- 数据库服务器故障
- 自然灾害

### 确定失败严重性

确定恢复目标后再创建恢复计划。计划应该包含多级故障的恢复目标。

下表显示了针对各种数据丢失量的故障的恢复计划。

表 1. 恢复计划样本

故障严重性	数据丢失	建议的恢复计划
小	非关键数据丢失。	可以一直等到非高峰时间才恢复该数据。请使用热恢复。
中	丢失的数据对于您的业务很关键，但不驻留在关键数据库空间中。	尽快对该数据执行热恢复。
大	关键数据库空间丢失。	立即使用混合恢复来恢复关键数据，并在非高峰时间使用热恢复来恢复非关键数据。
灾难	所有数据都丢失。	尽快执行冷恢复或混合恢复。

### 数据使用情况确定备份调度

制定恢复计划后，请根据您使用数据的方式创建备份计划。

您使用数据的方式将确定您计划备份调度的方式，如下所示：

- 数据使用

用户如何使用数据？

- 关键数据库空间（根数据库空间以及包含物理日志和至少一个逻辑日志文件的数据库空间）
- 关键业务应用程序数据
- 由于法律或记录保存原因的长期数据存储
- 组之间的数据共享
- 测试数据

- 事务时间

可以丢失多长的事务时间？即，要手动重新输入丢失的事务可能花费多长时间？例如：您可以承担重新输入过去三个小时里发生的所有事务吗？

- 数量和分布

丢失多少数据是您可以承担的？例如：您丢失了客户概要文件的四分之一，或者丢失了中西部地区的销售数据而西海岸的数据仍然是完好无损的。

询问以下问题将有助于确定您希望备份数据的频率和时间：

- 您的业务是否存在可以恢复系统的停机时间？
- 如果您的系统是 24x7 全天候运行的（没有停机时间），是否存在可以进行恢复的非高峰时间？
- 如果恢复必须在高峰期内发生，该时间有多关键？
- 数据库服务器处于联机状态时可以恢复哪种数据（热恢复）？哪种数据必须在脱机状态下进行恢复（冷恢复）？
- 有多少存储设备可用于备份与恢复数据？

## 调度备份

您的恢复策略应该包含备份调度。定制备份计划以满足您系统的需要。数据更改越频繁、更改越重要，您就需要越频繁地备份该数据。

您的备份计划还应该指定备份级别。

下表显示中小型系统的备份计划样本。

表 1. 样本备份计划

备份级别	备份调度
完整备份（0 级）	星期六下午 6 点
增量备份（1 级）	星期二和星期四下午 6 点
增量备份（2 级）	每天下午 6 点
对经常更新的存储空间进行 0 级备份	每小时

**重要：** 更改物理模式（例如将块添加到存储空间）后，请执行 0 级备份。（请参阅准备备份数据。）

### 基于标号的访问控制的安全需求

对于基于标签的访问控制 (LBAC)，运行 `gbackuprestore` 或 `gtape` 的人员无需获得安全策略或其他特权的豁免，即可备份或恢复数据。

使用 `gbackuprestore` 或 `gtape` 恢复数据之后，LBAC 保护仍是完整的。

## 2.2.2 为生产数据库服务器计划备份系统

要为数据计划足够的备份保护，请分析数据库服务器配置和活动以及安装时可用的备份介质类型。

还要考虑存储介质、磁盘、计算机和控制器以及网络大小的开支。

执行 0 级备份前的操作

在执行以下任何操作之后，您必须至少为根数据库空间和已修改的存储空间执行 0 级备份：

- 添加或删除镜像。
- 移动、删除逻辑日志文件或调整逻辑日志文件的大小。
- 更改物理日志的大小或位置。
- 更改存储管理器的配置。
- 添加、移动或删除数据库空间。
- 对任意类型的存储空间添加、移动或删除块。
- 添加、移动或删除 Blob 空间或智能大对象空间。

例如，如果添加新数据库空间 `db1`，那么您会在消息日志中看到一条警告，要求您对根数据库空间和新数据库空间执行 0 级备份。如果试图对根数据库空间或新数据库空间执行增量备份作为代替，`gbackuprestore` 将自动对新数据库空间执行 0 级备份。

**提示：** 尽管在添加日志文件后不再需要立即备份，但是因为数据结构发生了变化，所以下一个备份应该是 0 级备份。

如果创建的存储空间与已删除的存储空间同名，那么会执行两次 0 级备份：

1. 在删除存储空间后并在创建同名的存储空间前备份根数据库空间。
2. 创建存储空间后，备份根数据库空间和新存储空间。

执行 0 级备份后的操作

在执行以下任何操作之前，您必须为已修改的存储空间执行 0 级备份：

- 将非日志记录数据库转换为日志记录数据库。
- 在将 RAW 表变更为 STANDARD 类型之前。此备份确保转换到日志记录表类型之前未记录的数据是可恢复的。

### 评估硬件和内存资源

当您计划备份系统时，请评估您的硬件和内存资源。

评估以下数据库服务器和硬件配置元素以确定要使用哪些存储管理器和存储设备：

- I/O 虚拟处理器数
- 可用内存量以及处理器活动的分布

另请考虑备份与恢复所需的临时磁盘空间。数据库服务器使用临时磁盘空间来存储备份期间被覆盖以及内存中发生查询处理而溢出的之前数据映像。

准备备份数据时，请确保正确设置 DBSPACETEMP 环境变量或参数，以便指定的数据库空间具有足够空间，能满足您的需求。如果指定的数据库空间中空间不足，备份将失败，并且将使用根数据库空间，或者在填满根数据库空间之后，备份将失败。

### 评估备份与恢复时间

多种因素（包括数据库服务器配置和数据库大小）会影响系统备份与恢复数据所需的时间量。

备份或恢复花费的时间取决于以下因素：

- 磁盘或磁带设备的速度  
存储设备的速度越快，备份或恢复的时间就越快。
- 当磁盘或系统故障要求您重新构建数据库时，要恢复的增量备份的数目  
增量备份比完全备份使用的存储空间少，并且还能缩短恢复时间。
- 数据库中存储空间的大小和数目

备份：许多小存储空间的备份时间比总大小相同的一些大存储空间稍微长一些。

恢复：通常恢复的时间与恢复最大存储空间和逻辑日志的时间相同。

- 存储空间是否镜像

如果存储空间被镜像，将减少必须恢复被损坏的或被破坏的数据的可能性。可以在数据库服务器联机的情况下在非高峰时间恢复镜像。

- 在备份与恢复期间用户中断的时间长度

如果在数据库服务器处于联机状态时执行备份和热恢复，用户可以继续他们的工作但可能会注意到响应变慢。如果在数据库服务器处于停顿方式下执行备份和热恢复，那么用户必须退出数据库服务器。如果在数据库服务器处于脱机状态时执行冷恢复，那么数据库服务器对于用户不可用，因此恢复进行得越快越好。外部备份与恢复将除去系统停机时间。

- 备份调度

并不是每个备份或恢复会话中都必须包含所有的存储空间。通过调度备份，相对于很少或从不更改的那些存储空间，您可以更加频繁地对快速更改的存储空间进行备份。确保对每个存储空间至少进行一次 0 级备份。

- 数据库空间中表的布局以及磁盘中数据库空间的布局

设计您的数据库服务器模式时，以能够快速恢复重要信息的目的来组织数据。例如，将关键的和常用的数据隔离在最快的磁盘的一小组存储空间中。还可以将大表分段使其分布在数据库空间中，用来平衡 I/O 并最大化多个磁盘上的吞吐量。有关更多信息，请参阅《GBase 8s 性能指南》。

- 数据库服务器和系统负载

数据库服务器或系统上的负载越大，备份或恢复的时间就越长。

- 备份与恢复配置参数的值

例如：`gbackupstore` 用来与数据库服务器交换数据所用的数据缓冲区的数目和大小可能影响性能。使用 `BAR_NB_XPORT_COUNT` 和 `BAR_XFER_BUF_SIZE` 配置参数可控制数据缓冲区的数目和大小。

## 评估日志记录和事务活动

当您计划备份系统时，另请考虑日志记录和事务活动。

以下数据库服务器使用需求将影响您针对存储管理器和存储设备的决策：

- 期望的事务活动的数量和比率
- 逻辑日志的数目和大小

如果需要从事务活动少的数据库服务器恢复数据，请定义许多小逻辑日志。由于逻辑日志备份不频繁，因此不太可能丢失数据。

- 逻辑日志文件填充的速度有多快

在日志文件填满之前进行备份，这样数据库服务器就不会挂起。

- 数据库和表的日志记录方式

当使用许多非日志记录数据库或表时，逻辑日志备份可能变得不太频繁。

## 压缩行数据

压缩行数据可使备份与恢复数据更高效。

先压缩行数据然后再备份，可以提高备份与恢复的速度并且减少所需的备份介质。备份与恢复期间，较小的数据大小比未压缩数据具有以下优势：

- 备份更快。
- 恢复更快。
- 逻辑日志更小。
- 备份映像更小。

如果使用外部压缩实用程序来压缩已压缩行数据的备份映像，那么可能不会减少备份映像大小，因为已压缩数据通常无法进一步压缩。在某些情况下，已压缩行数据的备份映像大小可能大于外部实用程序压缩的备份映像大小。

## 使用外部程序变换数据

您可以在备份之前将外部程序用作过滤器插件来将数据变换为其他格式，然后在恢复之后将数据变换回原始格式。

要压缩或变换数据，请使用 `BACKUP_FILTER` 和 `RESTORE_FILTER` 配置参数调用外部程序。

**提示：** 如果在备份行数据之前将其压缩，那么使用外部实用程序压缩备份映像可能不会生成更小的备份映像。

任何人都可以拥有该过滤器，但是非特权用户不能具有写访问权。对过滤器的许可权与对 GBase 8s 服务器或 GBase 8s 实用程序所调用的其他任何可执行文件的许可权相同。

# 3 gbackuprestore 备份与恢复系统

## 3.1 gbackuprestore 备份与恢复系统的概述

gbackuprestore 由各种组件组成，它与存储管理器一起使用来备份和恢复数据。

### 3.1.1 gbackuprestore 组件

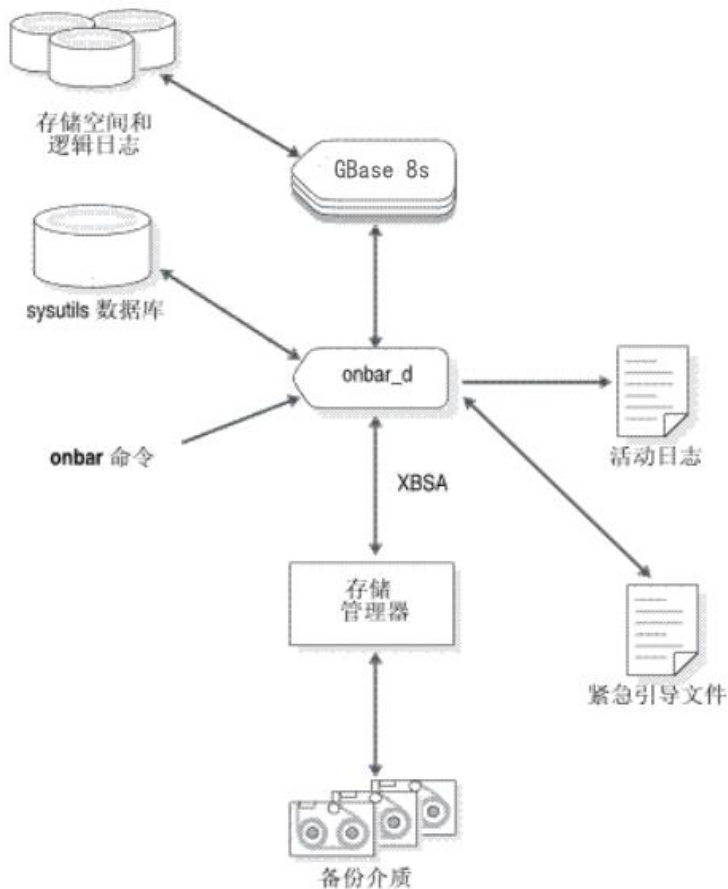
gbackuprestore 组件包含命令行实用程序、目录表、活动日志和紧急引导文件。请将 gbackuprestore 与存储管理器及其 XBSA 共享库一起使用。

下图显示了 gbackuprestore 和数据库服务器组件：

- 数据库服务器中的存储空间（数据库空间、Blob 空间和智能大对象空间）以及逻辑日志
- **sysutils** 数据库，其中包含 gbackuprestore 目录表
- gbackuprestore 和 gbackuprestore-d 命令行实用程序
- 系统上存储管理器的 XBSA 共享库
- 用于存储备份的存储介质
- gbackuprestore 活动日志
- gbackuprestore 紧急引导文件

**图: GBase 8s 的 gbackuprestore 组件**





`gbackuprestore` 与数据库服务器和存储管理器进行通信。请使用 `gbackuprestore` 命令启动备份或恢复操作。缺省情况下，`gbackuprestore` 以并行方式备份和恢复存储空间。`gbackuprestore` 始终以串行方式处理日志文件。

对于备份会话，`gbackuprestore` 从数据库服务器请求存储空间和逻辑日志的内容，并将它们传递到存储管理器。存储管理器将数据存储存储在存储介质上。对于恢复会话，`gbackuprestore` 从存储管理器请求已备份的数据，然后在数据库服务器上恢复该数据。

`gbackuprestore` 首先备份关键数据库空间，接着是剩余存储空间，最后备份逻辑日志。关键数据库空间是 `rootdbs` 以及包含逻辑日志和物理日志的数据库空间。

`gbackuprestore` 还会在备份期间将以下关键文件放到归档中：

- `onconfig` 文件
- `sqlhosts` 文件
- `oncfg_servername.servernum` 文件
- `gbackuprestore` 紧急引导文件：`ixbar.servernum`

您可以恢复存储在原始文件和格式化文件中的存储空间。如果系统包含主存储空间和镜像存储空间，那么恢复（外部恢复除外）期间 `gbackuprestore` 将同时写入主块和镜像块。

`gbackuprestore` 状态和错误消息将写入活动日志文件 `bar_act.log` 中。

### 备份服务 API (XBSA)

`gbackuprestore` 和存储管理器通过“备份服务应用程序编程接口 (XBSA)”通信，该接口支持存储管理器为数据库服务器管理介质。通过使用存储管理器的开放式系统接口，`gbackuprestore` 可以与各种同样使用 XBSA 的存储管理器一起工作。

每个存储管理器建立并分配一个唯一版本的 XBSA 共享库。必须使用随存储管理器提供的 XBSA 共享库版本。例如，如果使用 GBase 8s 主存储管理器，那么还必须使用 `gbackuprestore` 提供的 XBSA 共享库。`gbackuprestore` 和 XBSA 共享库必须以相同方式（32 位或 64 位）进行编译。例如，如果使用 Storage Manager，那么还要使用 `gbackuprestore` 提供的 XBSA 共享库。`gbackuprestore` 和 XBSA 共享库必须以相同方式（32 位或 64 位）进行编译。

`gbackuprestore` 使用 XBSA 与存储管理器交换以下类型的信息：

#### 控制数据

`gbackuprestore` 与存储管理器交换控制数据以验证 `gbackuprestore` 和 XBSA 是否兼容，并确保以正确顺序将对象恢复到数据库服务器的正确实例，以及跟踪备份对象的历史记录。

#### 备份或恢复数据

在备份与恢复期间，`gbackuprestore` 和存储管理器使用 XBSA 交换来自指定存储空间或逻辑日志文件的数据。

`gbackuprestore` 使用 XBSA 事务来确保数据的一致性。包含在一个事务中的所有操作被看作是一个单元。一个事务中的所有操作必须成功，才能恢复传送给存储管理器的对象。

### `gbackuprestore` 目录表

`gbackuprestore` 使用 `sysutils` 数据库中的目录表来跟踪备份与恢复操作。`onsmsync` 实用程序使用其他目录表来跟踪其操作。

`gbackuprestore` 在 `sysutils` 数据库中使用以下目录表来跟踪备份与恢复操作：

- **bar\_server** 表跟踪数据库服务器的实例。
- **bar\_object** 表跟踪备份对象。备份对象是数据库空间、Blob 空间、智能大对象空间或逻辑日志文件的备份。
- **bar\_action** 表跟踪所有对每个备份对象进行的备份与恢复尝试，除了某些日志回收和冷恢复事件以外。
- **bar\_instance** 表描述在成功的备份尝试中备份的每个对象。

onsmsync 实用程序使用并维护以下各表来跟踪其操作：

- **bar\_ixbar** 表包含所有时间线中所有未到期的成功备份的历史记录。
- **bar\_syncdeltab** 表通常为空白，但 onsmsync 正在运行时除外。

有关这些表内容的描述，请参阅 gbackuprestore 目录表。

### **ixbar 文件：gbackuprestore 紧急引导文件**

每次备份后会自动更新紧急引导文件。该文件包含 gbackuprestore 执行冷恢复所需的信息。

**重要：** 请勿修改紧急引导文件。如果进行了修改，将可能导致 gbackuprestore 选择错误的备份作为恢复的一部分，这可能会引起数据损坏或系统故障。

引导文件的文件名是 ixbar.servernum，其中 servernum 是 SERVERNUM 配置参数的值。

gbackuprestore 紧急引导文件位于 UNIX™ 上的 \$GBS\_HOME/etc 目录中。您可以更改 BAR\_IXBAR\_PATH 配置参数中指定的信息，从而覆盖引导文件的缺省路径和名称。

### **bar\_act.log 文件：gbackuprestore 活动日志**

gbackuprestore 将参考、进度、警告、错误和调试消息写入 gbackuprestore 活动日志 bar\_act.log 中。

gbackuprestore 备份和恢复错误不会出现在标准输出中。如果在您备份和恢复数据时发生错误，请检查 gbackuprestore 活动日志中的信息

还可以使用活动日志进行以下操作：

- 监视备份与恢复活动，例如，活动日志还将记录哪些存储空间和逻辑日志已备份或已恢复、操作的进度以及大致花了多长时间。
- 验证备份或恢复是否成功。
- 跟踪 gdblogmode 实用程序中的错误。
- 跟踪 gbackuprestore 性能统计信息

gbackuprestore 活动日志位于 UNIX™ 上的 /tmp 目录中。使用 BAR\_ACT\_LOG 配置参数指定 gbackuprestore 活动日志的位置。

### **gbackuprestore 脚本**

gbackuprestore 实用程序在 UNIX™ 上包含一个 shell 脚本，用于定制备份与恢复操作。

当您随数据库服务器安装 gbackuprestore 时，将包含一个缺省脚本。该脚本的名称和位置取决于操作系统：

### **UNIX**

`gbackuprestore shell` 脚本位于 `$GBS_HOME/bin` 目录中。

当从命令行发出 `gbackuprestore` 命令时，自变量会传递给脚本，然后传递给 `gbackuprestore_d` 实用程序。

表 1. `gbackuprestore` 实用程序

实用程序	描述
<code>gbackuprestore_d</code> 实用程序	在数据库服务器和存储管理器之间传输数据。  <code>gbackuprestore</code> 命令会调用 <code>gbackuprestore_d</code> 实用程序，以启动 <code>gbackuprestore-driver</code> 。 <code>gbackuprestore-driver</code> 启动并控制备份与恢复活动。
<code>onsmsync</code> 实用程序	同步 <code>sysutils</code> 数据库、紧急引导文件和存储管理器目录的内容。使用此实用程序以清除不再需要的备份。
<code>gdblogmode</code> 实用程序	更改数据库日志记录方式。 <code>gdblogmode</code> 实用程序将其输出记录到 <code>gbackuprestore</code> 活动日志 <code>bar_act.log</code> 中。
<code>archecker</code> 实用程序	验证备份，并从归档恢复表级别数据。

### 3.1.2 `gbackuprestore` 的存储管理

必须使用存储管理器来通过 `gbackuprestore` 执行备份与恢复操作。存储管理器是管理包含备份的存储设备和介质的应用程序。存储管理器将处理所有的介质标号、安装请求以及存储卷。

`Storage Manager` 包含在数据库服务器中。`GBase 8s` 主存储管理器包含在数据库服务器中。但是，您可以选择使用受 `gbackuprestore` 支持且与您的存储设备兼容的另一个存储管理器。

## 3.2 配置存储管理器和 `gbackuprestore`

本节中的主题提供了您在使用存储管理器来计划和设置 `gbackuprestore` 时所需的信息。

### 3.2.1 配置存储管理器

gbackuprestore 备份与恢复操作需要一个通过 XBSA 共享库接口与 gbackuprestore 集成的存储管理器。

您可以选择将 GBase 8s 主存储管理器、GBase Tivoli Storage Manager (TSM) 或第三方存储管理器与 gbackuprestore 一起使用。GBase 8s Primary Storage Manager 与 GBase 8s 捆绑在一起。如果您要使用 TSM, 那么 gbackuprestore 与 TSM 通信所需的 XBSA 共享库是与 GBase 8s 捆绑在一起的。

您可以选择将 Storage Manager、GBase Tivoli Storage Manager (TSM) 或第三方存储管理器与 gbackuprestore 一起使用。Storage Manager 是随数据库服务器一起安装的。如果使用 TSM, 那么 gbackuprestore 与 TSM 通信所需的 XBSA 共享库是与数据库 GBase 8s 捆绑在一起的。

GBase 8s Primary Storage Manager 为仅使用文件设备(磁盘)而非磁带的 gbackuprestore 备份与恢复操作(包括并行备份)管理存储。缺省情况下, GBase 8s 主存储管理器 是使用 GBase 8s Primary Storage Manager 中指定的信息和一些 gbackuprestore 配置参数自动配置的。当您使用 gpsm 实用程序时, 也会自动配置该存储管理器。您可以更改配置。有关信息, 请参阅《GBase 8s Primary Storage Manager》和配置 GBase 8s Primary Storage Manager。

### 确定您所需要的存储设备

在备份之前, 请确定需要的存储设备及其数量。

询问以下的相互关联的问题来确定您需要什么样的存储设备。例如: 存储设备的速度和类型将部分决定您所需要的存储设备数。

- 需要什么类型的存储设备?

事务卷和数据库的大小是确定您需要什么样的存储设备类型的主要因素。

Storage Manager 支持简单的磁带设备(如 QIC、4 毫米、8 毫米、DLT 磁带)、光盘设备以及磁盘的备份。如果 Storage Manager 不能管理您需要的存储设备, 那么需要购买不同的存储管理器。有关更多信息, 请参阅《GBase 8s Storage Manager 管理员指南》。

- 每个设备的可用性需求是什么?

对您的存储设备来说, 允许随机和顺序存取是否很重要? 如果重要, 那么您将不能使用磁带存储设备。

- 需要多少存储设备?

Storage Manager 针对每个主机最多支持四个设备。

需要的存储设备数取决于: 拥有的存储设备、数据库服务器上发生的事务活动量、吞吐速度、允许用于备份的时间以及其他相似因素。

## sm\_versions 文件中的存储管理器定义

有些存储管理器（除了 GBase 8s 主存储管理器）必须在 sm\_versions 文件中存在条目。存储管理器必须在 sm\_versions 文件中存在条目。

sm\_versions 文件中的存储管理器的定义使用以下格式：

```
1|XBSA_ver|sm_name|sm_ver
```

在该格式中，XBSA\_ver 是存储管理器的 XBSA 共享库的发行版，sm\_name 是存储管理器的名称，sm\_ver 是存储管理器版本。字段最大长度为 128 个字符。

在 gbackuprestore 使用 Tivoli Storage Manager、Storage Manager 和第三方存储管理器启动备份或恢复进程之前，gbackuprestore 会调用当前安装的特定于存储管理器的 XBSA 共享库版本以获取其版本号。如果该版本与当前的 gbackuprestore 版本兼容，并已在 sm\_versions 文件中定义，那么 gbackuprestore 开始执行所请求的操作。

## 配置 Storage Manager

您可以将 Storage Manager 与 gbackuprestore 一起使用。

Storage Manager 服务器与 GBase 8s 一起安装在 UNIX™ 上。多个数据库服务器实例可以共享一个 Storage Manager 实例。

**限制：** 在每台计算机上安装 Storage Manager 的一个副本，以防止可能与 XBSA 共享库的冲突。不要在同一台计算机上运行 Storage Manager 与 Legato NetWorker，因为它们相互冲突。

有关如何设置 Storage Manager 以与 gbackuprestore 一起工作的指示信息，请参阅《GBase 8s Storage Manager 管理员指南》。

在将 Storage Manager 用于备份前，请执行以下任务：

- 设置特定环境变量和配置参数
- 更新 sm\_versions 文件中存储管理器的定义。

设置 Storage Manager 环境变量和 onconfig 参数

当您使用 Storage Manager 时，必须设置特定环境变量。可以在 gbackuprestore 脚本或您的环境中设置这些必需的环境变量。有关更多信息，请参阅《GBase 8s Storage Manager 管理员指南》。

如果使用 Storage Manager，那么可以在 onconfig 文件的 ISM\_DATA\_POOL 和 ISM\_LOG\_POOL 配置参数中指定存储空间和逻辑日志的卷池名称。ISM\_DATA\_POOL 配置参数指定用于备份存储空间的卷池。ISM\_LOG\_POOL 配置参数指定用于备份逻辑日志的卷池。

如果没有设置这些配置参数，那么它们的缺省值为卷池名称 ISMData 和 ISMLogs。

相关信息请参阅《GBase 8s Storage Manager 管理员指南》。

更新 `sm_versions` 文件中的存储管理器定义以用于 Storage Manager

`gbackuprestore` 启动备份或恢复进程之前，它调用当前安装的特定于存储管理器的 XBSA 共享库版本以获取它的版本号。如果该版本与当前的 `gbackuprestore` 版本兼容，并已在 `sm_versions` 文件中定义，那么 `gbackuprestore` 开始执行所请求的操作。

要更新 `sm_versions` 文件中的存储管理器定义：

1. 将 `sm_versions.std` 模板复制到新文件 `sm_versions` 中，该文件位于 UNIX 上的 `$GBS_HOME/etc` 目录中。
2. 发出 `ism_startup -init` 命令以使用正确的版本号和存储管理器名称自动更新 `sm_versions`，或手动编辑 `sm_versions`。

**重要：** `ism_startup -init` 命令擦除先前备份的记录。

如果要安装 Storage Manager 补丁，必须手动编辑 `sm_versions`。

如果手动编辑 `sm_versions`，必须将 Storage Manager 置于 `sm_name` 字段中。

3. 停止当前运行的所有 `gbackuprestore` 进程（`gbackuprestore_d`、`gbackuprestore_w` 或 `gbackuprestore_m`），然后重新启动它们以使更改生效。

以下示例显示 `sm_versions` 文件中的 Storage Manager 定义：

```
1|1.0.1|ism|ISM.2.20.UC1.114|
```

## 配置 TSM

要将 Tivoli Storage Manager (TSM) 与 GBase 8s 数据库一起使用，您必须在数据库服务器计算机上安装和配置 Tivoli Storage Manager 客户机，并在存储计算机上安装和配置 Tivoli Storage Manager。

此外，您必须配置 GBase 8s Interface for TSM，并在 GBase 8s 数据库服务器计算机上执行其他 TSM 配置任务。要配置 TSM：

1. 编辑 TSM 客户机选项文件。
2. 指定由服务器用于备份的 TSM 管理类。
3. 设置 GBase 8s Interface for TSM 环境变量
4. 向 TSM 服务器注册。
5. 初始化 GBase 8s Interface for TSM。
6. 更新 `sm_versions` 文件中的存储管理器定义。

这些任务将在以下各部分中进行说明。

有关 TSM 的详细信息，请阅读以下手册：

- Tivoli Storage Manager 备份/归档客户机安装与用户指南

- Tivoli Storage Manager 使用应用程序界面
- Tivoli Storage Manager 管理员指南
- Tivoli Storage Manager 管理员参考大全

#### 编辑 TSM 客户机选项文件

GBase 8s Interface for Tivoli Storage Manager (TSM) 使用 TSM API 与 TSM 服务器进行通信。缺省情况下, GBase 8s Interface for TSM 使用客户机用户选项文件 (dsm.opt) 以及 (在 UNIX™ 系统上)客户机系统选项文件 (dsm.sys),这两个文件均位于 TSM API 安装目录中。

在 UNIX 系统上, 以 **root** 用户身份编辑 dsm.opt 和 dsm.sys 文件:

- 指定要在客户机用户选项文件 dsm.opt 中使用的 TSM 服务器。
- 在客户机系统选项文件 dsm.sys 中, 标识 TSM 服务器名称、通信方法和服务器选项。

使用随 TSM API 一起分发的样本 dsm.opt.smp 和 dsm.sys.smp 文件, 以帮助您快速入门。

请参阅 TSM 安装客户机和 TSM 跟踪设置指南 以获取有关您可在这些文件中指定的选项的信息。

#### 编辑 TSM 客户机用户选项文件

可以编辑 GBase 8s Tivoli Storage Manager (TSM) 客户机用户选项文件 dsm.opt。此文件必须引用正确的 TSM 服务器实例, 如 dsm.sys文件中列出的那样。

在 dsm.opt 文件中设置以下选项:

#### **SERVERNAME**

标识 GBase 8s Interface for TSM 为获取服务所访问的 TSM 服务器实例, 如 dsm.sys 文件中列出的那样。

#### **TRACEFILE**

将跟踪输出信息发送到指定的文件。

#### **TRACEFLAG**

设置特定跟踪标志

#### 编辑 TSM 客户机系统选项文件

可以编辑 GBase 8s Tivoli Storage Manager (TSM) 客户机系统选项文件 dsm.sys。此文件必须引用正确的 TSM 服务器地址和通信方法。

在 dsm.sys 文件中设置以下 TSM 选项非常重要:

#### **SERVERNAME**

指定要用于标识服务器的名称, 该服务器名称在 dsm.opt 文件中引用, 并用于创建包含该服务器的选项的实例。



**COMMMETHOD**

标识通信方法。

**TCPSERVERADDRESS**

标识 TSM 服务器。

**PASSWORDACCESS**

指定 GENERATE 来存储 TSM 密码。

dsm.opt 和 dsm.sys 文件中的 SERVERNAME 选项只定义服务器实例名称。

TCPSERVERADDRESS 选项控制所访问的服务器。

您可以在 dsm.sys 文件中设置多个服务器实例。请参阅《Tivoli Storage Manager 备份/归档客户机安装与用户指南》以获取有关多个服务器实例的信息。

**为备份指定 TSM 管理类**

当您备份数据库时，将使用节点的缺省管理类。您可以使用在 INCLUDE 选项中指定的不同值来覆盖缺省值。

INCLUDE 选项位于包含-排除选项文件中。包含-排除选项文件的文件名位于客户机系统选项文件 (dsm.sys) 中。有关更多信息，请参阅《Tivoli Storage Manager 备份/归档客户机安装与用户指南》。

将以下命名约定用于 gbackupstore 文件：

- 数据库备份：

/dbservername/dbservername/dbspacename/level

- 日志备份：

/dbservername/dbservername/server\_number/unique\_logid

对于数据库备份，INCLUDE 语句的示例如下：

```
Include /dbserverA/dbserverA/dbspaceA/* GbasedbtDbMgmt
```

对于逻辑日志备份，INCLUDE 语句的示例如下：

```
Include /dbserverA/dbserverA/55/* GbasedbtLogMgmt
```

其中数字 55 是 onconfig 文件中 SERVERNUM 参数的值。

**设置 GBase 8s Interface for TSM 环境变量**

当您使用 GBase 8s Interface for TSM 时，需要在用户的环境中设置某些环境变量。

下表描述了这些环境变量。

表 1. GBase 8s Interface for TSM 环境变量

环境变量	描述
------	----

环境变量	描述
DSMI_CONFIG	客户机用户选项文件（dsm.opt）的标准名称。在 TSM API 安装目录中，缺省值为 dsm.opt。
DSMI_DIR	在 UNIX™ 上，指向 TSM API 安装路径。只有在 TSM API 安装在不同于缺省路径的路径中时，才需要定义此环境变量。 DSMI_DIR 环境变量还用来查找 dsm.sys 文件。
DSMI_LOG	指向包含 API 错误日志文件（dsierror.log）的目录。  对于错误日志文件，创建一个要在其中创建错误日志的目录，然后将 DSMI_LOG 环境变量设置为该目录。gbasedbt 用户 或备份操作员应该具有此目录的写许可权。

以下示例显示在 TSM API 安装于 /opt/Tivoli/tsm/client/api 目录的情况下，如何对 Solaris 32 位设置这些环境变量：

```
export DSMI_CONFIG=/opt/Tivoli/tsm/client/api/bin/dsm.opt
export DSMI_DIR=/opt/Tivoli/tsm/client/api/bin
export DSMI_LOG=/home/user_a/logdir
```

#### 向 TSM 服务器注册

在备份到 GBase 8s Tivoli Storage Manager (TSM) 服务器以及从该服务器恢复之前，您必须有向 TSM 注册的节点名和密码。设置节点名和密码的过程称为注册。

在向 TSM 服务器注册 GBase 8s Interface for TSM 节点之后，您可以开始使用 GBase 8s Interface for TSM 来备份与恢复 GBase 8s 存储空间和逻辑日志。如果您的工作站向 TSM 备份/归档客户机指定了节点名，那么 GBase 8s Interface for TSM 应该具有不同的节点名。有关执行注册过程的更多信息，请参阅《Tivoli Storage Manager 备份/归档客户机安装与用户指南》。

#### 初始化 GBase 8s Interface for TSM 密码

要初始化 GBase 8s Interface for TSM 的密码，请使用 txbsapswd 程序。此程序设置与您在 dsm.opt 文件中指定的服务器实例的连接。

在使用 GBase 8s Interface for TSM 之前，您必须以 root 用户身份运行 txbsapswd 程序。

要初始化密码：

1. 启动位于 \$GBS\_HOME/bin 目录中的 txbsapswd 程序。
2. 输入密码并按 Enter 键。要保留当前密码，请在不输入任何值的情况下按 Enter 键。

#### 在 TSM 的 sm\_versions 文件中更新存储管理器定义

必须更新 `sm_versions` 文件中的存储管理器定义，以使 `gbackuprestore` 可用于 GBase Tivoli Storage Manager (TSM)。

`gbackuprestore` 启动备份或恢复进程之前，它调用当前安装的特定于存储管理器的 XBSA 共享库版本以获取它的版本号。如果该版本与当前的 `gbackuprestore` 版本兼容，并已在 `sm_versions` 文件中定义，那么 `gbackuprestore` 开始执行所请求的操作。

要更新 `sm_versions` 文件中的存储管理器定义：

1. 将 `sm_versions.std` 模板复制到新文件 `sm_versions` 中，该文件位于 UNIX™ 上的 `$GBS_HOME/etc` 目录。
2. 将 `tsm` 放在 `sm_versions` 文件的 `sm_name` 字段中。值 `adsm` 也有效，但在将来发行版中不推荐使用。
3. 停止当前运行的所有 `gbackuprestore` 进程（`gbackuprestore_d`、`gbackuprestore_w` 或 `gbackuprestore_m`），然后重新启动它们以使更改生效。

以下示例显示 `sm_versions` 文件中的 Tivoli Storage Manager 定义：

```
1|5.3|tsm|5
```

### 配置第三方存储管理器

各种存储管理器的安装与配置要求略有不同。如果使用第三方存储管理器，请确保仔细遵循制造商的指示信息。如果难以安装和配置存储管理器，请直接与制造商联系。

有关适用于您的 `gbackuprestore` 版本的已认证存储管理器列表，请咨询销售代表。

**重要：** 某些存储管理器允许您指定要备份到特定存储设备的数据。为提高备份与恢复的效率，请配置存储管理器以将逻辑日志备份到一个设备，而将存储空间备份到另一个设备。

要配置第三方存储管理器：

1. 设置 `gbackuprestore` 配置参数和环境变量。
2. 配置存储管理器使 `gbackuprestore` 可以正确地与之通信。

相关信息请参阅您的存储管理器文档。

3. 遵循存储管理器文档中的指示信息来配置您的存储设备。

存储管理器必须了解它所使用的存储设备的设备名。

4. 标记存储卷。
5. 在存储设备上安装存储卷。
6. 在 `sm_versions` 文件中创建存储管理器定义。使用第三方存储管理器的供应商提供的定义。
  - a. 将 `sm_versions.std` 模板复制到新文件 `sm_versions` 中，该文件位于 UNIX™ 上的 `$GBS_HOME/etc` 目录中。

- b. 通过使用 `sm_versions.std` 中的格式作为模板，以存储管理器的正确数据创建您自己的 `sm_versions` 文件。要找出第三方存储管理器的 `sm_versions` 中要使用哪个代码名称，请参阅存储管理器文档。
  - c. 停止当前运行的所有 `gbackuprestore` 进程（`gbackuprestore_d`、`gbackuprestore_w` 或 `gbackuprestore_m`），然后重新启动它们以使更改生效。
7. 验证存储管理器的 `BAR_BSALIB_PATH` 配置参数是否指向正确的 XBSA 共享库。
  8. `gbackuprestore` 将 `SERVERNUM` 配置参数值用作存储管理器中逻辑日志的存储路径的一部分。如果存储管理器未将通配符用于服务器编号，请为存储管理器设置相应的服务器编号环境变量。

配置了存储管理器和存储设备并为数据库服务器和逻辑日志备份标记了卷后，您就可以使用 `gbackuprestore` 启动备份或恢复操作了。

### 指定 XBSA 库的位置

缺省情况下，`gbackuprestore` 在 UNIX™ 上的 `$GBS_HOME/lib/ibsad001.s[ol]` 中查找 XBSA 共享库。要为 XBSA 共享库指定不同的名称或位置，请使用 `BAR_BSALIB_PATH` 配置参数。

#### UNIX

也可以使 `$GBS_HOME/lib/ibsad001.s[ol]` 成为指向正确库的符号链接。

例如，如果使用的是 Storage Manager，您可以执行以下任一操作：

- 将 `$GBS_HOME/lib/ibsad001.so` 链接到 `$GBS_HOME/lib/libbsa.so`
- 将 `BAR_BSALIB_PATH` 设置为 `$GBS_HOME/lib/libbsa.so`

例如，如果使用的是 TSM，您可以执行以下任一操作：

- 将 `$GBS_HOME/lib/ibsad001.so` 链接到 `$GBS_HOME/lib/libtxbsa.so`
- 将 `BAR_BSALIB_PATH` 设置为 `$GBS_HOME/lib/libtxbsa.so`

### 3.2.2 验证存储管理器

转换或还原 GBase 8s 数据库服务器时，在旧版本上使用的存储管理器可能未针对要迁移到的版本进行验证。验证存储管理器供应商是否已成功完成对数据库服务器版本和平台的 GBase 8s 验证过程。

如果尚未完成验证，那么使用 `gbackuprestore` 执行备份之前需要安装经过验证的存储管理器。

### 3.2.3 配置 `gbackuprestore`

开始第一次备份前, 请复审 onconfig 文件中的缺省 gbackuprestore 参数并按需要调整这些值。您可以通过设置 onconfig 文件中的以下配置参数来配置 gbackuprestore 的行为。

表 1. gbackuprestore 配置参数.

行为	配置参数
通过设置数据缓冲区的数量和大小以及并行进程的数量来提高 gbackuprestore 性能。	BAR_NB_XPORT_COUNT 配置参数 BAR_XFER_BUF_SIZE 配置参数 BAR_MAX_BACKUP 配置参数
设置调试级别以及调试日志文件的位置。	BAR_DEBUG 配置参数 BAR_DEBUG_LOG 配置参数
更改 gbackuprestore 引导文件的路径。	BAR_IXBAR_PATH 配置参数
保留到期的备份的历史记录。	BAR_HISTORY 配置参数
更改 gbackuprestore 活动日志的位置和内容。	BAR_IXBAR_PATH 配置参数
保留到期的备份的历史记录。	BAR_ACT_LOG 配置参数 BAR_PROGRESS_FREQ 配置参数 BAR_PERFORMANCE 配置参数
设置失败的备份或恢复的自动重试。	BAR_RETRY 配置参数
允许失败的恢复重新启动。	RESTARTABLE_RESTORE 配置参数
增大发送给存储管理器的备份大小估算值。	BAR_SIZE_FACTOR 配置参数
延长 RS 辅助服务器在外部备份期间等待检查点的时间。	BAR_CKPTSEC_TIMEOUT 配置参数
配置连续日志备份。	请参阅《GBase 8s 管理员参考》中的 ALARMPROGRAM
使用外部程序过滤或变换已备份的数据。	BACKUP_FILTER 配置参数

行为	配置参数
	RESTORE_FILTER 配置参数

不要将 LTAPEDEV 配置参数设置为 /dev/null 或 NUL，因为这样将会禁用逻辑日志备份，而您只能恢复整个系统的备份。

### **gbackuprestore 安全性**

缺省情况下，只有 UNIX™ 系统上的 gbasedb 或 root 用户才能运行 gbackuprestore 命令。

要使其他用户能够运行 gbackuprestore 命令：

- 在 UNIX 系统上，创建 bargroup 组，并向该组添加用户。有关如何创建组的指示信息，请参阅您的 UNIX 文档。

**限制：** 出于安全性考虑，建议不要使用 root 用户来运行 gbackuprestore 命令。

### **3.2.4 验证 gbackuprestore 和存储管理器的配置**

在开始使用 gbackuprestore 和存储管理器之前，请确保 gbackuprestore 和存储管理器正确设置。

通过检查以下列表中的项来验证配置：

- 安装并配置了存储管理器来管理特定的存储设备。
- 对于 UNIX™，确保 BAR\_BSALIB\_PATH 配置参数正确指定了 XBSA 共享库，或未设置该参数且库位于缺省位置。
- sm\_versions 文件中包含一行用来标识特定于存储管理器的 XBSA 共享库的版本号。

验证 gbackuprestore 和存储管理器已正确设置后，在测试数据库上运行 gbackuprestore 以确保可以备份与恢复数据。要了解更多信息，请按照使用 gbackuprestore 备份中的指示信息操作。

### **3.2.5 gbackuprestore 和存储管理器使用的文件**

gbackuprestore、GBase 8s 主存储管理器 Storage Manager 和 GBase Tivoli Storage Manager (TSM) 使用您的安装中的特定文件。

下表列出了 gbackuprestore、Storage Manager 和 GBase Tivoli Storage Manager (TSM) 使用的文件以及这些文件所在的目录。如果将 onconfig 文件设置为与缺省值不同的值，这些名称和位置将发生更改。

表 1. gbackuprestore、Storage Manager 和 TSM 使用的文件的列表

文件名	目录	用途
ac_config.std	UNIX™: \$GBS_HOME/etc	archecker 参数值的模板。  ac_config.std 文件包含缺省的archecker（归档检查）实用程序参数。要使用该模板，请将其复制到另一个文件中后修改它的值。
ac_msg.log	/tmp  %GBS_HOME%\etc	archecker 消息日志。  当您 将 archecker 与 gbackuprestore 一起使用来验证备份时，它将简要的状态和错误消息写入到gbackuprestore 活动日志中，并将详细的状态和错误消息写入到 archecker 消息日志中。技术支持使用 archecker 消息日志来诊断备份与恢复的问题。请使用 AC_MSGPATH 配置参数指定 archecker 消息日志的位置。
bar_act.log	/tmp  %GBS_HOME%	gbackuprestore 活动日志。  有关更多信息，请参阅 <a href="#">bar_act.log 文件</a> ： <a href="#">gbackuprestore 活动日志</a> 。
bldutil.process_id	/tmp  \tmp	创建 <b>sysutils</b> 数据库后，错误消息出现在该文件中。
dsierror.log	\$DSMI_LOG	TSM API 错误日志。

文件名	目录	用途
dsm.opt	\$DSMI_CONFIG	TSM 客户机用户选项文件。
dsm.sys	\$DSMI_DIR	TSM 客户机系统选项文件。
紧急引导文件 (ixbar* 文件)	\$GBS_HOME/etc %GBS_HOME%\etc	在冷恢复中使用。有关更多信息，请参阅 ixbar 文件： gbackuprestore 紧急引导文件。
Storage Manager 目录	\$GBS_HOME/ism %ISM%	记录有关 Storage Manager 使用的备份与恢复保存集及存储卷的信息。  在 ism_startup 初始化期间，Storage Manager 会创建 Storage Manager 目录。Storage Manager 目录记录存储在 mm、index 和 res 文件中。有关更多信息，请参阅《GBase 8s Storage Manager 管理员指南》。
Storage Manager 日志	\$GBS_HOME/ism/logs %ISM\logs	操作员警报消息、后端状态、其他 Storage Manager 信息。 Storage Manager 日志名称是 daemon.log、messages 和 summary。
ISMversion	\$GBS_HOME/ism DIR%\bin	标识 Storage Manager 的版本。 不要编辑该文件。
oncfg_servername.servernum	\$GBS_HOME/etc %GBS_HOME%\etc	gbackuprestore 恢复的配置信息。  当您初始化磁盘空间时，数据库服务器将创建



文件名	目录	用途
		<p>oncfg_servername.servernum 文件。每次您添加或删除数据库空间、逻辑日志文件或块时，数据库服务器都将更新该文件。当数据库服务器在冷恢复期间回收逻辑日志文件时使用 oncfg* 文件。由于数据库服务器使用 oncfg*文件，因此不要删除它们。</p>
save、savegrp、savefs	<p>\$GBS_HOME/bin</p> <p>%ISM%\bin</p>	
sm_versions	<p>\$GBS_HOME/etc</p> <p>%GBS_HOME%\etc</p>	<p>标识 Storage Manager 或第三方存储管理器的版本。</p> <p>要更新存储管理器版本，请直接编辑sm_versions 文件。</p> <p>GBase 8s Primary Storage Manager 不使用 <b>sm_versions.std</b> 文件。</p>
xbsa.messages	<p>\$GBS_HOME/ism/applogs</p> <p>%ISM%\applogs</p>	<p>XBSA 库调用信息。</p> <p>gbackuprestore 和 Storage Manager 使用 XBSA 来相互通信。技术支持使用 xbsa.messages 日志来修复 gbackuprestore 和 Storage Manager 通信中出现的问题。</p>

## 3.3 使用 gbackuprestore 备份

您可以使用 gbackuprestore 实用程序来备份并验证存储空间（数据库空间、Blob 空间和智能大对象空间）以及逻辑日志文件。

要使用 gbackuprestore 执行备份：

1. 准备备份。
2. 使用 gbackuprestore 备份
3. 监视备份进度。
4. 验证备份。
5. 备份存储管理器信息。

您可以在 shell 或批处理脚本中定制 gbackuprestore 和存储管理器命令。可以从作业调度程序调用 gbackuprestore。

### 3.3.1 准备备份数据

备份存储空间和逻辑日志之前，必须先准备系统并复制关键管理文件。

要准备备份数据：

1. 配置 gbackuprestore 和存储管理器。
2. 确保您有足够的逻辑日志空间。

gbackuprestore 在备份开始时检查可用的逻辑日志空间。如果日志几乎填满，gbackuprestore 在试图备份存储空间之前将备份并释放这些日志。如果日志包含充足的空间，gbackuprestore 将备份存储空间接着备份逻辑日志。

3. 验证您是否有足够的临时磁盘空间。

数据库服务器使用临时磁盘空间来存储备份期间被覆盖以及内存中发生查询处理而溢出的之前数据映像。验证 DBSPACETEMP 环境变量和 DBSPACETEMP 配置参数指定的数据库空间是否具有满足您需要的足够空间。如果指定的数据库空间中空间不足，备份将失败，并且将使用根数据库空间，或者在填满根数据库空间之后，备份将失败。

4. 将管理文件备份到不同的位置。
5. 运行 gcheck -cD 命令以验证所有数据库服务器数据是否一致。

不需要在每个 0 级备份之前都检查一致性。在下次验证数据库的一致性之前，请勿放弃已知为一致的备份。

**要备份的管理文件**

gbackuprestore 备份不会替换重要配置文件的正常操作系统备份。必须手动备份关键管理文件。

**重要：** 请制作关键管理文件的当前版本的备份副本，以在紧急情况下使用。如果需要更换磁盘或者如果恢复到第二个计算机系统（导入的恢复），那么必须恢复这些文件。

备份以下管理文件：

- 紧急引导文件
- onconfig 文件
- sm\_versions 文件
- sqlhosts 文件 (UNIX™)
- 存储管理器配置和数据文件
- 存储在磁盘上的 Blob 空间中的简单大对象数据
- 存储在磁盘或光盘的 Blob 空间中的简单大对象的数据
- 存储于外部的数据（例如：由 DataBlade® 维护的外部表）

**提示：** 尽管 gbackuprestore 包含 onconfig 和 sqlhosts 及其备份的文件，但较好的做法是将 onconfig 和 sqlhosts 文件包含在您的系统归档中。通过将关键文件同时包含在 GBase 8s 和系统归档中，您在需要时可以有更多选择。

尽管 gbackuprestore 不备份以下项，但 gbackuprestore 将在恢复期间自动重新创建这些项。您不需要为这些文件创建备份副本：

- 已分配给数据库服务器但还未分配给表空间范围的数据库空间页面
- 镜像块，如果相应的主要块是可以访问的
- 临时数据库空间

gbackuprestore 不备份或恢复临时数据库空间中的数据。一旦恢复，数据库服务器重新创建空的临时数据库空间。

当您执行文件的 0 级、1 级或 2 级备份时，还将备份多个关键文件。bar\_act.log 包含有关备份了哪些关键文件的信息。例如：

```
Begin backup of critical file '/opt/gbasedbt-11.70.fc3/etc/ixbar.0'.  
Completed backup of critical file '/opt/gbasedbt-11.70.fc2/etc/ixbar.0'  
  
Begin backup of critical file '/opt/gbasedbt-11.70.fc3/etc/oncfg_work_loc.0'.  
Completed backup of critical file '/opt/gbasedbt-11.70.fc3/etc/oncfg_aork_loc.0'  
  
Begin backup of critical file '/opt/gbasedbt-11.70.fc3/etc/hygia_work.wh'.  
Completed backup of critical file '/opt/gbasedbt-11.70.fc2/etc/hygia_work.wh'
```

### 3.3.2 gbackuprestore -b 语法: 备份

使用 gbackuprestore -b 命令可备份存储空间和逻辑日志。

要运行 gbackuprestore 命令, 您必须是 root 用户或 gbasedbt 用户, 或者是 UNIX™ 上 bargroup 组的成员。

用途

- 示例: 备份整个系统
- 示例: 备份所有联机存储空间和逻辑日志
- 示例: 执行增量备份
- 示例: 备份指定的存储空间和所有逻辑日志
- 示例: 备份文件中指定的存储空间列表
- 示例: 备份逻辑日志
- 示例: 物理备份

使用 gbackuprestore 备份的语法

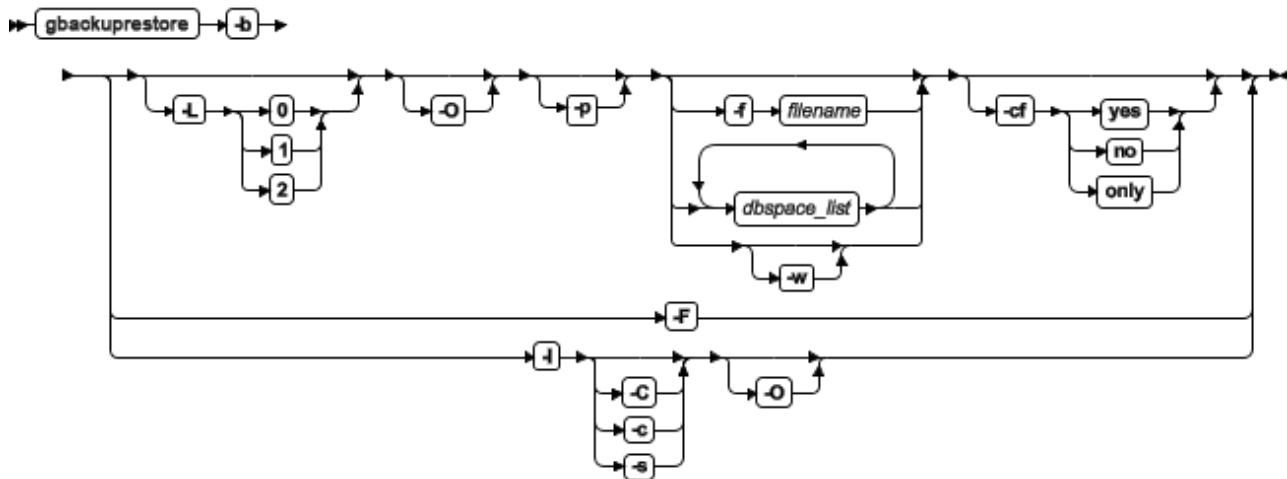


表 1. gbackuprestore -b 命令的选项

选项	描述
-b	指定一个备份 备份存储空间和逻辑日志, 包括当前逻辑日志。
dbspace_list	指定要备份的存储空间, 以空格分隔。

选项	描述
	<p>如果没有输入 <code>dbspace_list</code> 或 <code>-f filename</code>, <code>gbackuprestore</code> 会备份数据库服务器上所有联机的存储空间。</p>
-c	<p>关闭并备份当前逻辑日志和其他所有逻辑日志。</p>
-C	<p>启动连续日志备份。</p> <p>因为连续日志备份将无限期运行以等待逻辑日志填满, 所以请保留专用存储设备和终端窗口。</p> <p>要停止连续日志备份, 请使用中断命令 (例如, CTRL-C 或 SIGTERM) 停止 <code>gbackuprestore</code> 进程。</p>
-cf	<p>指定是否备份关键文件。</p> <p>有效值为:</p> <ul style="list-style-type: none"> <li>• <code>yes</code> = 备份关键文件。该值为执行 0 级、1 级或 2 级备份时的缺省值。</li> <li>• <code>no</code> = 不备份关键文件。该值为备份逻辑日志文件时的缺省值。</li> <li>• <code>only</code> = 仅备份关键文件。</li> </ul>
-ffilename	<p>备份 <code>filename</code> 值所指定的文本文件中列出的存储空间。</p> <p>使用该选项以避免每次备份时都要输入一长串存储空间。</p> <p>有关更多信息, 请参阅<a href="#">文件中的存储空间列表</a>。</p>
-F	<p>执行伪备份</p> <p>存储管理器应用程序不是必需的。实际上并没有发生备份, 因此不可能从伪备份进行恢复。如果确实需要伪备份, 请</p>

选项	描述
	<p>酌情少量使用。伪备份可能在以下情况下适用：</p> <ul style="list-style-type: none"> <li>• 更改数据库日志记录方式</li> <li>• 将 RAW 表更改为 STANDARD 表</li> <li>• 允许用户在不执行备份的情况下使用新的日志、块或镜像</li> <li>• 在您作为管理员判定不需要备份的特殊情况下</li> </ul>
-L	<p>指定要在存储空间上执行的备份级别：</p> <ul style="list-style-type: none"> <li>• 0 = 完整备份（缺省值）</li> <li>• 1 = 自上次 0 级备份以来的更改</li> <li>• 2 = 自上次 1 级备份以来的更改</li> </ul> <p>如果您请求增量备份，而 <code>gbackuprestore</code> 发现没有为特定存储空间执行前一级的备份，那么 <code>gbackuprestore</code> 会对该存储空间执行前一级的备份。例如，如果请求 1 级备份，而 <code>gbackuprestore</code> 未找到 0 级备份，那么它会转而进行 0 级备份。</p>
-l	<p>执行对所有逻辑日志文件的备份。</p> <p>当前逻辑日志文件未备份。</p> <p>如果使用的是 Storage Manager，它还会备份 Storage Manager 目录。</p>
-0	<p>覆盖常规备份限制。</p> <p>当 Blob 空间处于脱机状态时，使用该选项备份逻辑日志。</p> <p>如果 Blob 空间脱机时发生日志备份，返回码 178 将显示在 <code>gbackuprestore</code> 活动日志中。</p>
-p	<p>仅备份物理存储空间，而不备份逻辑日志。</p>

选项	描述
	<p>会将警告消息写入活动日志，以列出恢复存储空间时所需的最新日志文件的日志唯一标识。如果逻辑日志是连续进行备份的，请使用此选项。如有必要，将启动日志开关，以便可以备份此日志。如果当前的日志比最近一次存储空间的归档检查点日志新，那么不会启动日志开关。</p>
-s	<p>数据库服务器发生故障后，回收磁盘上现有的所有逻辑日志。当服务器处于脱机状态时，可以运行 <code>gbackuprestore -l -s</code> 命令。</p> <p>如果可能，替换损坏的磁盘前请使用该选项。如果使用 <code>gbackuprestore -r</code> 在未损坏的磁盘上执行冷恢复，<code>gbackuprestore</code> 将自动回收逻辑日志。</p>
-w	<p>备份整个系统，其中包含基于单个检查点的所有存储空间和逻辑日志。</p> <p>备份的时间与备份信息存储在一起。在整个系统的备份中，所有存储空间中的数据是一致的，因此，不再需要恢复逻辑日志。如果不保存逻辑日志，您必须使用 <code>-w</code> 选项。</p>

## 用途

在备份数据之前，请通过运行 `gcheck -cD` 命令来确保您的数据是一致的。

要运行 `gbackuprestore` 命令，您必须是 `root` 用户或 `gbasedbt` 用户，或者是 UNIX 上 `bargroup` 组的成员。有关更多信息，请参阅 `gbackuprestore` 安全性。

当数据库服务器处于联机、停顿或快速恢复方式时，可以备份存储空间和逻辑日志。

存储空间块可以存储在原始磁盘存储空间上、热文件中。

仅备份联机的存储空间。使用 `gstat -d` 命令来确定哪些存储空间是联机的。备份期间，如果 `gbackuprestore` 遇到关闭的数据库空间，它将跳过并随后返回一个错误。如果存储空间处于脱机状态，请在存储空间回到联机状态时重新启动备份。

开始备份后，在 `gbackuprestore` 活动日志和数据库服务器消息日志中监视其进度。

可以单独备份逻辑日志或与存储空间一起备份。一旦逻辑日志填满，请立即对其进行备份，这样就可以复用这些日志，并防止在包含这些日志的磁盘丢失时发生数据丢失。如果日志

文件填满，数据库服务器将暂停，直到备份了逻辑日志为止。可以手动备份逻辑日志，也可以运行 `gbackuprestore -b -C` 命令来启动连续逻辑日志备份。逻辑日志备份始终是 0 级的。关闭当前逻辑日志后就可以对其进行备份。

如果执行整个系统的备份与恢复，那么无需恢复逻辑日志。但是，请在使用整个系统的备份时备份逻辑日志。这些日志备份允许您将数据恢复到整个系统备份后的某一时间，从而使数据丢失降到最低限度。

如果要运行连续的逻辑日志备份，然后启动整个系统备份，那么 `gbackuprestore` 进程会尝试保存逻辑日志。由于连续的逻辑日志备份正在运行，因此会返回错误消息以指示逻辑日志备份已在运行，且整个系统备份会返回非零错误代码。在这种情况下，逻辑日志只会备份一次。要避免该错误，请使用 `gbackuprestore -b -w -p` 命令创建物理备份。

要在 `gbackuprestore` 中备份特定表或一组表，请将这些表存储在单独的数据库空间中，然后备份该数据库空间。或者，您也可以使用 `archecker` 实用程序执行表级别恢复。

示例：备份整个系统

以下命令在执行所有联机存储空间和逻辑日志的检查点后执行整个系统的 0 级备份：

```
gbackuprestore -b -w
```

以下命令执行整个系统的 1 级备份：

```
gbackuprestore -b -w -L 1
```

示例：备份所有联机存储空间和逻辑日志

以下命令执行所有联机存储空间和已用逻辑日志的标准 0 级备份：

```
gbackuprestore -b
```

示例：执行增量备份

以下命令执行标准 1 级备份：

```
gbackuprestore -b -L 1
```

示例：备份指定的存储空间和所有逻辑日志

以下命令对名为 `fin_dbSPACE1` 和 `fin_dbSPACE2` 的数据库空间以及所有逻辑日志执行 0 级备份：

```
gbackuprestore -b fin_dbSPACE1 fin_dbSPACE2
```

示例：备份文件中指定的存储空间列表



以下名为 listfile3 的样本文件包含要备份的存储空间列表：blobsp2.1、my\_dbSPACE1、blobsp2.2、dbsl.1、rootdbs.1 和 dbsl.2。

```
blobsp2.1
# a comment                ignore this text

        my_dbSPACE1        # back up this dbSPACE
; another comment
blobsp2.2                dbsl.1
rootdbs.1    dbsl.2    ; backing up two spaces
```

以下命令备份 listfile3 文件中列出的存储空间：

```
gbackuprestore -b -f listfile3
```

示例：备份逻辑日志

以下命令启动手动逻辑日志备份：

```
gbackuprestore -b -l
```

以下命令备份当前逻辑日志文件：

```
gbackuprestore -b -l -c
```

示例：物理备份

以下命令备份所有存储空间，而不备份任何逻辑日志：

```
gbackuprestore -b -p -L 0
```

会将警告消息写入 gbackuprestore 活动日志文件中，说明该日志文件备份未启动。该消息还包含恢复存储空间时所需的最新日志文件的日志唯一标识。所需的最新日志文件包含最近一次备份的数据库空间的归档检查点。

示例消息：

```
2011-12-14 09:30:35 14277 14275 (-43354) WARNING: Logical logs were
not backed up as part of this operation. Logs through log unique ID 9
are needed for restoring this backup. Make sure these logs are backed
up separately.
```

### 文件中的存储空间列表

您可以在文件中列出要备份或恢复的存储空间。

filename 值可以是任何有效的 UNIX™ 文件名：

- 简单文件名，例如：listfile\_1
- 相对文件名，例如：../backup\_lists/listfile\_2 或 ..\backup\_lists\listfile2
- 绝对文件名，例如：/usr//backup\_lists/listfile3 或 c:\\backup\_lists\listfile3

文件的格式规则如下：

- 如果要恢复块，那么列出不带路径的存储空间名称。每行可以列出多个存储空间，用空格或制表符分隔。
- 如果要重命名块，那么列出旧块路径名、旧偏移量、新块路径名和新偏移量。在每项之间放置一个空格或一个制表符。将每个块的信息置于单独的行上。
- 注释以 # 或 ; 符号开头，并继续到当前行的结尾。
- gbackuprestore 将忽略文件中的所有注释和空白行。

### 备份期间使用 Storage Manager

使用 ism\_watch 命令可监视发送到 Storage Manager 服务器的备份与恢复。在备份期间，Storage Manager 服务器自动将存储空间数据路由到 ISMData 卷池中的卷，并将逻辑日志文件路由到 ISMLogs 卷池中的卷。

始终保持 ISMLogs 池中有安装的卷，以确保存储设备总是可用于接受逻辑日志数据。如果未安装此卷，备份会暂停。有关使用设备和 Storage Manager 命令的更多信息，请参阅《GBase 8s Storage Manager 管理员指南》。

备份操作期间，Storage Manager 会创建已备份数据的保存集并在 Storage Manager 目录中输入有关已备份数据的信息。您还可以使用 ism\_catalog -create\_bootstrap 命令来备份 Storage Manager 目录：

如果使用 gbackuprestore 脚本备份存储空间和逻辑日志，将自动备份 Storage Manager 目录。如果直接调用 gbackuprestore\_d，那么必须使用 ism\_catalog -create\_bootstrap 命令。

### 备份 Blob 空间

您可以在使用事务日志记录的数据库中备份 Blob 空间。

备份新的 Blob 空间之前，确保对 Blob 空间创建进行了记录的日志文件不再是当前的日志文件。您可以运行 gstat -l 命令来验证逻辑日志状态。

当用户更新或删除 Blob 空间中的简单大对象时，直到释放了包含删除记录的日志文件后才能释放 Blob 页以重新使用。必须备份日志文件才能将其释放。

**重要：** 如果在更新或删除没有备份逻辑日志的 Blob 空间的数据后对该 Blob 空间执行热恢复，那么该 Blob 空间可能无法恢复。

要备份 Blob 空间：

1. 通过运行 `gstat -l` 或 `xctl gstat -l` 命令来验证逻辑日志的状态。
2. 通过运行 `gadmin -l` 命令来切换到下一个日志文件。
3. 备份逻辑日志：
  - 如果 Blob 空间是联机的，请运行 `gbackuprestore -b -l -c` 命令。
  - 如果 Blob 空间是脱机的，请运行 `gbackuprestore -b -l -O` 或 `gbackuprestore -b -O` 命令。如果备份成功，`gbackuprestore` 将返回 178。
4. 通过运行 `gbackuprestore -b` 或 `gbackuprestore -b -w` 命令来备份 Blob 空间。

### 3.3.3 gbackuprestore -m 语法：监视最近的 gbackuprestore 活动

您可以使用 `gbackuprestore -m` 命令来监视最近的 `gbackuprestore` 活动。只有有权执行备份与恢复操作的用户才可以使用此选项。

监视最近的 `gbackuprestore` 活动

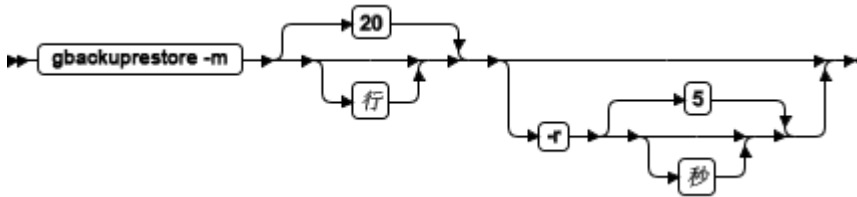


表 1. `gbackuprestore -m` 命令的选项

选项	描述
<code>-m</code>	打印活动日志文件中 <code>gbackuprestore</code> 的最近活动。
<code>lines</code>	指定要输出的行数。缺省值为 20 行。
<code>-r</code>	使 <code>gbackuprestore -m</code> 命令重复。
<code>秒</code>	指定在重复之前等待的秒数。缺省值为 5 秒。

### 3.3.4 查看注册备份的列表

您可以为系统上执行的 `gbackuprestore` 注册备份创建列表。

要查看注册备份的列表：

1. 在 `sysutils` 中创建一个视图，其中包含 `bar_action`、`bar_instance` 和 `bar_object` 目录表的信息。

在视图中包括以下字段：

- Backup\_ID: 内部生成的备份标识
- Type: 定义备份是整个系统备份、数据库空间备份还是逻辑日志备份。
- Object\_Name: 所备份对象的名称。
- Ifx\_Time: 创建对象的时间。对于数据库空间备份，表示启动备份的检查点时间。对于逻辑日志，表示日志填满的时间。
- CopyID\_HI: 在存储管理器中查找对象的标识的高位部分。
- CopyID\_LO: 在存储管理器中查找对象的标识的低位部分。
- Backup\_Start: 此对象备份启动的日期和时间。
- Backup\_End: 此对象备份结束的日期和时间。
- Verify\_Date: 上次对此对象进行验证的时间（如果有）。

2. 对视图运行 SELECT 语句。

示例

以下语句会创建包含备份信息的视图：

```
CREATE VIEW list_backups(Backup_ID, Type, Object_Name, Ifx_Time, CopyID_HI,
                        CopyID_LO, Backup_Start, Backup_End, Verify_Date)
AS SELECT * FROM (
SELECT
    act_aid AS backup_id,
    DECODE(act_type, 5, "Whole-System", DECODE(obj_type, "L",
        "Logical log", "Dbospace")) AS Type,
    substr(obj_name,1, 8) AS Object_Name,
    min(DBINFO ('utc_to_datetime', seal_time)) AS Ifx_Time,
    ins_copyid_hi AS CopyID_HI,
    ins_copyid_lo AS CopyID_LO,
    act_start AS Backup_Start,
    act_end AS Backup_End,
    ins_verify_date AS Verify_Date

FROM
    bar_action A,
    bar_instance I,
    bar_object O

WHERE
```

```
A.act_aid = I.ins_aid AND
A.act_oid = O.obj_oid AND
A.act_oid = I.ins_oid AND
O.obj_type in ("R", "CD", "ND", "L")
GROUP BY 1,2,3,5,6,7,8,9
ORDER BY lfx_Time, Backup_ID) AS view_list_backups
```

以下查询将返回所有备份:

```
SELECT * FROM list_backups
```

### 3. 3. 5 gbackuprestore -P 语法: 打印备份逻辑日志

您可以使用 gbackuprestore -P 命令来打印已使用 gbackuprestore 实用程序备份的逻辑日志。

要运行 gbackuprestore 命令, 您必须是 root 用户或 gbasedbt 用户, 或者是 UNIX™ 上 bargroup 组的成员。

- 用途
- 示例: 打印特定事务
- 示例: 打印多个逻辑日志文件

打印已备份的逻辑日志

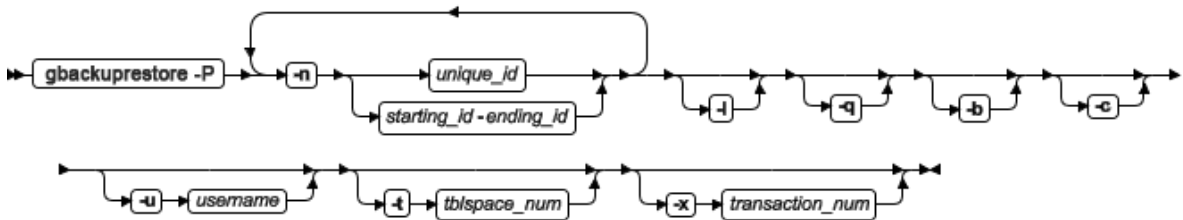


表 1. gbackuprestore -P 命令的选项

选项	用途
-b	打印与 Blob 空间 Blob 页面相关联的逻辑日志记录。  数据库服务器将这些记录作为 Blob 空间日志记录的一部分存储在逻辑日志备份介质上。
-c	使用压缩字典来展开压缩的数据。
-l	打印逻辑日志记录的长列表。

选项	用途
	日志记录的长列表包含整个日志记录的复杂十六进制和 ASCII 转储。
<code>-n starting_id-ending_id</code>	打印包含在指定的日志文件范围中的逻辑日志记录。 <code>starting_id</code> 选项是要打印的第一个日志的标识。 <code>ending_id</code> 选项是要打印的最后一个日志的标识。 <code>starting_id</code> 选项的值必须小于 <code>ending_id</code> 选项的值。  以连字符分隔开始和结束标识值。请勿包含空格。
<code>-n unique_id</code>	打印包含在指定的日志文件中的逻辑日志记录。 <code>unique_id</code> 选项是逻辑日志的唯一标识号。要确定特定逻辑日志文件的唯一标识, 请使用 <code>gstat -l</code> 命令。
<code>-P</code>	打印已备份的逻辑日志信息
<code>-q</code>	不打印程序头
<code>-tblspace_num</code>	打印与使用 <code>tblspace_num</code> 选项指定的表空间相关联的记录。  以无符号整数或十六进制值的形式指定 <code>tblspace_num</code> 值。如果您没有使用 <code>0x</code> 前缀, 那么值将解释为整数。整数必须大于零, 并且必须存在于 <code>systables</code> 系统目录表的 <code>partnum</code> 列中。
<code>-u username</code>	打印特定用户的记录。用户名必须为现有登录名, 并且必须符合登录名的特定于操作系统的规则。
<code>-xtransaction_num</code>	只打印与指定的事务相关联的记录。  <code>transaction_num</code> 必须是在零和 <code>TRANSACTIONS -1</code> 之间 (包括零和 <code>TRANSACTIONS -1</code> ) 的无符号

选项	用途
	<p>整数。</p> <p>其他信息：仅在前滚期间生成错误这一不太可能的情况下使用 <code>-x</code> 选项。当此情况发生时，数据库服务器将向包含违规事务的事务标识的消息日志发送消息。您可以将此事务标识与 <code>-x</code> 选项一起使用，以查找错误的原因。</p>

用途

要查看已备份的逻辑日志，存储管理器必须正在运行。

此命令的输出将打印到 `stdout`。

示例：打印特定事务

以下命令打印由 `gbasedbt` 用户 针对表空间 `1048722` 执行，且包含在逻辑日志文件 `2` 中的单个事务的相关信息：

```
gbackuprestore -P -n 2 -l -q -b -u "gbasedbt" -t 1048722 -x 1
```

此命令的输出可能为：

```
log uniqid: 2.
1665d0 120 DPT 1 2 0 5
00000078 0002006c 00000010 0000fefe ...x...l .....
00000001 00000000 000077e3 00000000 ..... ..w....
00000005 00000005 00002a24 00000001 ..... ..*$...
00100004 0a0c21b8 00002a48 00000001 .....!. ..*H...
00100006 0a0c2288 00002ea1 00000001 .....". .....
0010001b 0a0c3810 00002bee 00000001 .....8. ..+....
00100015 0a0c3a18 00002a3d 00000001 .....: ..*=....
00100005 0a0c57c0 .....W.
166648 60 CKPOINT 1 0 1665d0 1
0000003c 00000042 00000010 0000fefe ...<...B .....
00000001 001665d0 000077e3 00000000 .....e. ..w....
00010005 00000002 00000002 001665a0 ..... ..e.
00000007 ffffffff 00084403 ..... ..D.
```

示例：打印多个逻辑日志文件

以下命令打印标识为 `2`、`3`、`4`、`5`、`10`、`11` 和 `12` 的逻辑日志文件的逻辑日志记录：

```
gbackuprestore -P -n 2-5 -n 10-12
```

### 3.3.6 gbackuprestore -v 语法：验证备份

使用 gbackuprestore -v 命令可验证 gbackuprestore 实用程序所创建的备份是否完整且可以恢复。

要运行 gbackuprestore 命令，您必须是 root 用户或 gbasedbt 用户，或者是 UNIX™ 上 bargroup 组的成员。

必须有足够的临时空间可用。有关更多信息，请参阅用于备份验证的临时空间。

- 用途
- 示例：执行备份的时间点验证
- 示例：验证文件中列出的存储空间的备份
- 示例：gbackuprestore 活动日志验证消息
- 示例：archecker 消息日志验证消息

验证备份

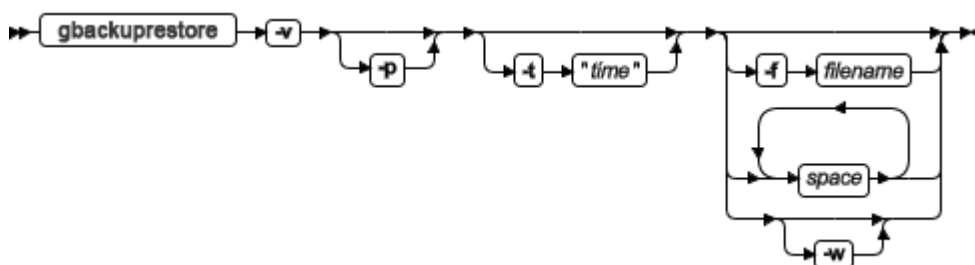


表 1. gbackuprestore -v 命令的选项

选项	描述
-v	验证备份。服务器可以采用任何方式。  如果验证成功，那么可以安全地恢复存储空间。  可以验证整个系统的备份或仅物理的备份。不能验证逻辑日志。
space	要验证的存储空间的名称。  如果输入多个存储空间名称，那么使用空格来分隔这些名称。



选项	描述
-f filename	<p>验证由 filename 提供其路径名的文本文件中列出的存储空间。</p> <p>使用该选项以避免每次验证存储空间时都要输入一长串存储空间。</p> <p>您可以使用任何有效的 UNIX 路径名和文件名。有关该文件的格式，请参阅文件中的存储空间列表。</p> <p>此文件可以在每行列出多个存储空间。</p>
-p	验证仅物理的备份。
-t "time"	<p>指定验证数据库空间的日期和时间。必须以引号括起。</p> <p>如何输入时间取决于当前的 GLS 语言环境约定。如果设置了 <code>GL_DATETIME</code> 环境变量，那么必须根据该变量指定日期和时间。如果未设置 GLS 语言环境，请使用 ANSI 样式的日期格式：YYYY-MM-DD HH:MM:SS。</p>
-w	验证整个系统的备份。

### 用途

`gbackupstore -v` 命令运行 `archecker` 实用程序。 `archecker` 实用程序验证恢复备份所需的页面是否以正确的格式存在于介质上。成功验证备份后就可以安全恢复了。

当验证备份时， `gbackupstore` 将摘要消息写入到 `bar_act.log` 中，它报告验证了哪些存储空间以及验证是成功的还是失败的。 `archecker` 实用程序将详细的消息写入到 `ac_msg.log` 中。GBase 软件支持使用 `ac_msg.log` 来诊断备份与恢复的问题。

`gbackupstore -v` 命令仅验证智能大对象空间中智能大对象的范围。使用 `gcheck -cS` 命令进行完整的检查。

`gbackupstore -v` 命令不能验证 Blob 空间中数据行和简单大型对象之间的链接。使用 `gcheck -cD` 命令作为代替来验证 Blob 空间中的链接。

示例：执行备份的时间点验证

以下命令验证某个时间点的备份：

```
gbackuprestore -v -t "2011-12-10 10:20:50"
```

示例：验证文件中列出的存储空间的备份

以下命令验证文件 `bkup1` 中列出的已备份的存储空间：

```
gbackuprestore -v -f /usr/backups/bkup1
```

示例：gbackuprestore 活动日志验证消息

以下示例显示 gbackuprestore 活动日志中有关验证的消息：

数据库空间 `db2.2` 的 0 级备份已通过验证，如下：

```
开始 db2.2 的 0 级备份验证（存储管理器复制标识：##）  
成功完成 0 级备份检验。
```

`rootdb` 的 0 级备份验证失败，如下：

```
开始 rootdb 的 0 级备份验证（存储管理器复制标识：##）。  
错误：无法关闭物理检查：error_message。
```

示例：archecker 消息日志验证消息

更多详细信息可在 `archecker` 消息日志中获得，如下：

```
状态：扫描已通过  
状态：控制页检查已通过  
状态：开始检查数据库空间 db2.2.  
状态：正在检查 db2.2:TBLSpace  
.  
.  
状态：已验证的表/分段：1  
归档验证已通过
```

### 用于备份验证的临时空间

验证备份时，必须有 15 到 25 MB 的临时空间可用。

在备份验证期间，对于中等大小的系统（40 到 50 GB），`archecker` 实用程序需要大约 15 MB 的临时空间，对于大型系统则需要大约 25 MB。这个临时空间存储在文件系统上，由 `AC_STORAGE` 参数指定的目录中，而不在数据库空间中。临时文件包含关于分区页、块中的可用页和保留页的备份和复制的位图信息，（以下可选）还可以包含 Blob 空间中的可用页和调试信息。`archecker` 实用程序必须具有该临时目录的许可权。

如果备份验证成功，这些文件将被删除。如果备份验证失败，这些文件将继续保留。将它们复制到另一个位置，以便GBase 软件支持可以进行查看。

如果数据库服务器只包含数据库空间，请使用以下公式估算 `archecker` 临时文件使用的临时空间量（以 KB 为单位）：

$$\text{空间} = (130 \text{ KB} * \text{number\_of\_chunks}) + (\text{pagesize} * \text{number\_of\_tables}) + (.05 \text{ KB} * \text{number\_of\_logs})$$

对于 GBase 8s，如果您的数据库服务器包含 Blob 空间或智能大对象空间，那么使用以下公式估计 `archecker` 临时文件的临时空间大小：

$$\text{空间} = (130 \text{ KB} * \text{number\_of\_chunks}) + (\text{pagesize} * \text{number\_of\_tables}) + (.05 \text{ KB} * \text{number\_of\_logs}) + (\text{pagesize} * (\text{num\_of\_blobpages}/252))$$

#### **number\_of\_chunks**

估计用于数据库服务器的块的最大数目。

#### **pagesize**

系统页的大小（以 KB 为单位）。

#### **number\_of\_tables**

估计用于数据库服务器的表的最大数目。

#### **number\_of\_logs**

数据库服务器上的逻辑日志数。

#### **num\_of\_blobpages**

Blob 空间中 Blob 页的数目或智能大对象空间的数目。（如果数据库服务器包含智能大对象空间，那么使用智能大对象空间的数目替换 `num_of_blobpages`。）

例如，在页面大小为 2 KB 的 50 千兆字节系统上，将需要 12.9 兆字节的临时磁盘空间。该系统不包含任何 Blob 空间，如以下语句所示：

$$13,252 \text{ KB} = (130 \text{ KB} * 25 \text{ 块}) + (2 \text{ KB} * 5000 \text{ 个表}) + (.05 \text{ KB} * 50 \text{ 个日志}) + (2 \text{ KB} * 0)$$

要将 KB 转换为 MB，请将结果除以 1024：

$$12.9 \text{ MB} = 13,252/1024$$

## 验证失败

备份验证可能因各种原因而失败。如果备份验证失败，请不要试图恢复它。

由于 `gbackuprestore` 在存储管理器上找不到备份对象，因此验证失败的原因是不可预测的，从数据库服务器毁坏到恢复失败，不一而足。实际上，表面上恢复可能成功，但它隐藏了数据或介质的真正问题。

### 损坏页的备份

如果页面损坏，问题在于数据库而不是备份或介质。

在所有产生错误的表上运行 `gcheck -cd`，然后重新执行备份和验证。要检查范围和保留页，请运行 `gcheck -ce` 和 `gcheck -cr`。

### 损坏的控制信息的备份

在这种情况下，所有数据都是正确的，但是某些备份控制信息不正确，这可能使恢复出现问题。向 GBase 软件支持请求协助。

### 缺少数据的备份

当备份缺少数据时，它可能无法恢复。数据丢失后，请尝试从较旧备份中恢复。然后恢复当前逻辑日志。

### 数据库服务器不一致数据的备份

有时会出现以下情况：`archecker` 返回“成功”给 `gbackuprestore`，但在 `archecker` 消息日志中显示“失败”。当 `archecker` 验证 `gbackuprestore` 已正确备份数据，但数据库服务器数据在备份时无效或不一致时就会出现这种情况。

### 对备份验证失败的原因进行诊断

如果备份验证失败，您可以执行步骤来诊断并尝试修正问题。

要对备份验证失败的原因进行诊断：

1. 验证 `AC_CONFIG` 环境变量和 `archecker` 配置文件的内容是否设置正确。

如果这些变量未正确设置，那么 `gbackuprestore` 活动日志将打印一条消息。

2. 将数据备份到不同的介质。

不要复用原始的备份介质，因为它可能已损坏。

不要使用基于该备份的任何备份。如果 0 级备份验证失败，不要使用相应的 1 级和 2 级备份。

3. 验证新备份。

如果验证成功，那么可以恢复存储空间。

4. 使用存储管理器使验证失败的备份到期，然后在不带参数的情况下运行 `onsmsync` 实用程序，以从 `sysutils` 和紧急引导文件中除去坏的备份。

5. 如果验证再次失败，请致电 GBase 软件支持，并向他们提供以下信息：

- 备份工具名称（gbackuprestore）
- 数据库服务器 online.log
- archecker 消息日志
- 包含备份的位图以及重要备份页副本的 AC\_STORAGE 目录

如果只有备份的一部分损坏，GBase 软件支持可帮助您确定在紧急情况下可以恢复备份的哪一部分。

GBase 软件支持可能建议您针对一组表运行 gcheck 选项。

### 验证到期备份

可以验证到期备份，以防后续备份无效。

要验证到期备份：

1. 检查存储管理器上备份保存集的状态。如果存储管理器使备份保存集到期，那么 archecker 实用程序无法对其进行验证。
2. 使用存储管理器命令激活已到期的备份保存集。请参阅您的存储管理器文档。
3. 重新运行 gbackuprestore -v 命令。
- 4.

### 在备份缺少数据时进行恢复

如果因为缺少数据导致备份验证失败，您可以从较旧备份执行恢复。

要在备份缺少数据时进行恢复：

1. 选择早于已失败备份的备份的日期和时间。要执行时间点验证，请使用 gbackuprestore -v -t time space 命令。
2. 如果较旧备份通过验证，请使用相同 time 值执行时间点物理恢复，然后执行日志恢复，如下所示：

```
gbackuprestore -r -p -t time space
gbackuprestore -r -l
```

3. 在存储管理器中使损坏的备份到期。
4. 不带自变量运行 onsmsync 命令。

onsmsync 实用程序将从紧急引导文件和 sysutils 数据库中除去存储管理器不再保留的备份，从而防止 gbackuprestore 尝试使用此类备份。

## 3.4 使用 gbackuprestore 恢复数据

可以使用 `gbackuprestore` 实用程序来恢复由 `gbackuprestore` 实用程序备份的数据。

恢复数据前，请使用恢复前的核对表来确定是否需要恢复，以及准备恢复。

要使用 `gbackuprestore` 实用程序执行恢复：

1. 使备份期间可用的存储设备可用于恢复。
2. 如有必要，添加足够的临时空间来执行恢复。热恢复的逻辑日志恢复部分需要临时空间。最小临时空间量等于分配的逻辑日志空间总量与要重放的日志文件数中较小的一个。
3. 使用合适的选项运行 `gbackuprestore -r` 命令以恢复数据。
4. 监视 `gbackuprestore` 活动日志。
5. 恢复完成后，运行 `gstat -d` 命令来验证所有存储空间是否都已恢复。`flags` 列中的字母 `O` 表示该块处于联机状态。

### 3.4.1 预恢复核对表

使用该核对表可确定是否需要恢复，并准备恢复。

要准备恢复：

- 决定您是否需要进行恢复操作。如果存在其中一个或多个问题，请执行恢复来修复该问题：
  - 数据是否已丢失或毁坏？
  - 已落实的事务错误需要撤销吗？
  - 数据库服务器是停止运行了还是发生了磁盘故障？
  - 存储空间或块是关闭了还是不一致？
- 通过使用数据库服务器监视工具来诊断问题。
- 如果需要恢复根数据库空间或包含物理日志和逻辑日志文件的数据库空间，必须执行冷恢复。在冷恢复期间，数据库服务器必须处于脱机状态。请要求您的客户机用户注销系统。
- 在执行恢复之前，联系相应的供应商以解决以下类型的问题：
  - 存储管理器
  - XBSA 连接
  - 操作系统
  - 存储介质

#### 存储空间状态和所需操作

要确定每个存储空间及其块的状态，请检查 `gstat -d` 命令的输出。存储空间状态决定解决问题所需执行的操作。数据库服务器必须处于联机状态。

下表描述了有关块状态以及解决问题所需操作的 `gstat -d` 命令输出。块状态信息位于输出的第一部分（存储空间）和第二部分（块）中 `flags` 列的第二个位置处。

表 1. 块标志描述和所需操作

块标志	存储空间或块状态	需要采取的操作
(无标志)	存储空间不再存在。	对删除存储空间前的时间执行时间点冷恢复。
D	块已关闭或存储空间被禁用。	对受影响的存储空间执行热恢复。
I	块已进行物理恢复,但还需要逻辑恢复。	执行逻辑恢复。
L	存储空间在进行逻辑恢复。	请重试逻辑恢复。
N	块已重命名并且已关闭或不一致。	当物理设备可用时对块进行热恢复。
O	块是联机的。	不需要任何操作。
P	对存储空间进行物理恢复。	如果尚未进行恢复,那么执行逻辑恢复。
R	正在恢复存储空间。	执行物理或逻辑恢复。
X	存储空间或块是最近镜像的。	不需要任何操作。

### 使用 Storage Manager 恢复保存集

如果要使用 Storage Manager，可以从存储卷上的保存集恢复数据。当 Storage Manager 服务器接收到恢复请求时，`ism_watch` 命令会提示您在存储设备上安装所需的存储卷。安装该卷后，恢复会继续。

可以为保存集或卷设置保持周期。除非卷上所有的保存集都已到期，否则可以使用 `gbackuprestore` 将其恢复。

当保存集的保留周期到期后，`gbackuprestore` 无法再恢复它。要重新创建已到期的保存集，请使用 `ism_catalog -recreate from` 命令。

如果为卷设置了保留期，那么 Storage Manager 会保留这些保存集，直到该卷上的所有保存集都到期为止。要恢复到期的卷，请使用 `ism_catalog -recover from` 命令。

### 3.4.2 存储设备可用性

验证备份中使用的存储设备和文件是否可用于恢复。

如果在 0 级备份后删除了某个数据库空间或镜像设备，那么该数据库空间或镜像设备在开始恢复时必须可用于数据库服务器。如果存储设备不可用，那么数据库服务器无法写入块中，恢复将失败。

如果在上次备份后添加了块，那么块设备在前滚逻辑日志时必须可用于数据库服务器。

### 3.4.3 gbackuprestore -r 语法：恢复数据

要执行完整的恢复，请使用 gbackuprestore -r 命令。

要运行 gbackuprestore 命令，您必须是 root 用户或 gbasedbt 用户，或者是 UNIX™ 上 bargroup 组的成员。

- 用途
- 示例：执行整个系统的恢复
- 示例：恢复特定存储空间
- 示例：分阶段执行热恢复
- 示例：时间点恢复
- 示例：法语语言环境中的时间点恢复
- 示例：分阶段的时间点恢复
- 示例：恢复已删除的存储空间和块

执行恢复

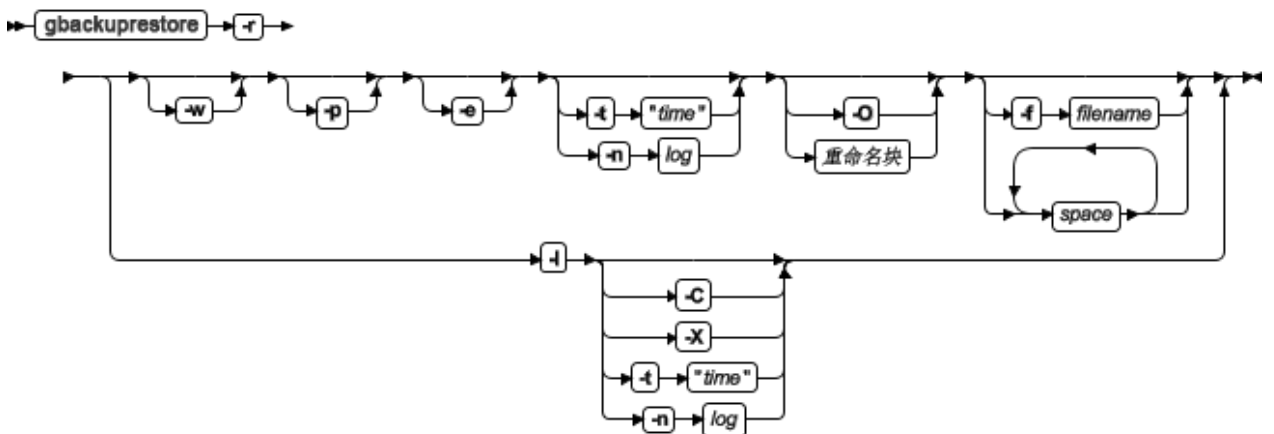


表 1. gbackuprestore -r 命令的选项。

选项	描述
----	----



选项	描述
-r	<p>指定恢复。如果数据库服务器处于脱机状态，gbackuprestore 将执行冷恢复。如果数据库服务器处于联机、停顿或快速恢复方式，gbackuprestore 将执行热恢复。</p> <p>在冷恢复中，-r 选项会恢复所有存储空间，回收并恢复逻辑日志。在热恢复中，-r 选项恢复所有脱机的存储空间并恢复逻辑日志。</p> <p>必须先指定 -r 选项。</p>
space	<p>指定将备份哪些存储空间，以空格分隔的一个或多个数据库空间、Blob 空间或智能大对象空间名称的列表表示。</p> <p>gbackuprestore 仅恢复列出的存储空间。如果数据库服务器处于脱机状态，必须列出所有关键数据库空间。不能指定临时空间。</p>
-C	<p>从当前逻辑日志磁带中连续恢复逻辑日志，而不发送安装磁带的提示。</p> <p>服务器处于暂挂日志恢复状态，在最后一个可用日志恢复后，该命令依然存在。如果日志存在于多个磁带中，那么服务器将发出提示。配置参数 RESTARTABLE_RESTORE 不会影响连续日志恢复。</p>
-e	<p>指定外部恢复。在外部恢复存储空间后，运行 gbackuprestore -r -e 命令。将存储空间标记为已物理恢复，恢复逻辑日志并使存储空间联机。</p>
-f filename	<p>指定用于列出要恢复或重命名的存储空间的文本文件的路径和文件名。</p> <p>使用该选项可避免输入一长串存储空间。</p> <p>有关更多信息，请参阅<a href="#">文件中的存储空间列表</a>。</p>

选项	描述
-l	<p>指定仅逻辑恢复。恢复并前滚逻辑日志。逻辑恢复仅应用于已物理恢复的存储空间。</p> <p><b>重要：</b> 为提高性能，在热恢复期间以并行方式重放逻辑日志事务。使用 <code>ON_RECOVERY_THREADS</code> 配置参数来设置并行线程数。要在冷恢复期间以并行方式重放逻辑日志事务，请使用 <code>OFF_RECOVERY_THREADS</code> 配置参数。有关更多信息，请参阅《GBase 8s 性能指南》。</p>
-n log	<p>指示冷恢复中要恢复的最后一个逻辑日志的唯一标识。数据库服务器必须处于脱机状态。</p> <p>要查找唯一标识，请使用 <code>gstat -l</code> 命令。</p> <p>日志点恢复是特殊的时间点恢复。必须在日志点恢复中恢复所有存储空间以保持数据的一致性。如果指定的日志后存在任何逻辑日志，<code>gbackuprestore</code> 不会恢复这些日志，它们的数据将丢失。如果特定逻辑日志应用于多个时间线，那么 <code>gbackuprestore</code> 将使用最新的时间线。</p> <p>不能与 <code>-p</code> 选项一起使用。</p>
-n new_path	<p>指定块的新路径。与 <code>-rename</code> 选项一起使用。</p>
-0	<p>覆盖内部错误检查。允许联机存储空间的恢复。强制重新创建不再存在的块文件。</p> <p>用来覆盖内部错误检查以执行以下任务：</p> <ul style="list-style-type: none"> <li>• 强制联机存储空间的恢复。如果要恢复的存储空间的列表中某个存储空间是联机的，那么 <code>gbackuprestore</code> 将使该存储空间脱机，然后将其恢复。如果此操作成功，那么 <code>gbackuprestore</code> 将完成并带有退出码 177。</li> <li>• 强制创建不存在的块文件。如果所恢复的存储空间</li> </ul>

选项	描述
	<p>的块文件不再存在，那么 <code>gbackuprestore</code> 将重新创建该文件。新创建的块文件是格式化的磁盘空间，不是原始磁盘空间。如果 <code>gbackuprestore</code> 成功地重新创建了缺少的块文件，那么 <code>gbackuprestore</code> 将完成并带有退出码 179。</p> <ul style="list-style-type: none"> <li>如果缺少关键存储空间，那么强制冷恢复继续进行。在冷恢复中，<code>gbackuprestore</code> 将检查每个关键空间是否正在恢复。此检查偶尔会引起假警告。如果警告有效，那么恢复失败。如果警告为 <code>false</code> 并且 <code>gbackuprestore</code> 成功恢复了服务器，那么 <code>gbackuprestore</code> 将完成并带有退出码 115。</li> </ul> <p>对整个系统的恢复使用 <code>-0</code> 选项仅仅是为了重新创建丢失的块文件。当数据库服务器处于联机状态时不能使用 <code>gbackuprestore -r -w -0</code> 命令，因为在整个系统恢复期间无法让根数据库空间脱机。</p>
<code>-o new_offset</code>	指定已重命名的块的偏移量。与 <code>-rename</code> 选项一起使用。
<code>-o old_offset</code>	指定要重命名的块的偏移量。与 <code>-rename</code> 选项一起使用。
<code>-p</code>	<p>指定仅物理恢复。</p> <p>除非您使用整个系统的恢复，否则在可以访问数据之前必须在逻辑恢复后执行物理恢复。该选项关闭冷恢复前的自动日志回收。如果 <code>LTAPEDEV</code> 配置参数设置为 <code>/dev/null</code> 或 <code>NUL</code>，那么必须在恢复期间使用 <code>-p</code> 选项。</p> <p>不能与 <code>-n</code> 选项一起使用。</p>
<code>-p old_path</code>	指定要重命名的块的路径。与 <code>-rename</code> 选项一起使用。

选项	描述
-rename	<p>在冷恢复期间重命名一个或多个块。数据库服务器必须处于脱机状态。</p> <p>如果需要将存储空间恢复到与完成备份的磁盘不同的磁盘上，该选项很有帮助。可以重命名任意类型的块，包括关键块和镜像块。您可以重命名具有 0 级备份的块。</p>
-t "time"	<p>指定在冷恢复中要从逻辑日志恢复的最后一个事务的时间。数据库服务器必须处于脱机状态。</p> <p>指定的所有存储空间都将恢复到相同的时间点。但是，如果先执行物理恢复，然后执行逻辑恢复，那么该逻辑恢复可以恢复到更晚的时间点。例如，您可能检测到当前备份损坏，且需要恢复先前备份。在这种情况下，请使用先前备份的时间戳记启动物理恢复，然后将逻辑恢复启动到更新的时间戳记。</p> <p>时间点恢复通常用来从错误恢复。例如：如果您意外删除了数据库，那么可以将服务器恢复到删除数据库之前的时间点。</p> <p>要确定时间点恢复的相应日期和时间，请使用 <code>glogdump</code> 实用程序。<code>glogdump</code> 实用程序输出显示了逻辑日志中已落实事务的日期和时间。在恢复命令中指定的时间后发生的所有数据事务将丢失。</p> <p>使用引号括起日期和时间。英语语言环境的格式是 <code>yyyy-mm-dd hh:mm:ss</code>。如果设置了 <code>GL_DATETIME</code> 环境变量，那么必须根据该变量指定日期和时间。</p>
-w	<p>从上次整个系统的备份对所有存储空间和逻辑日志执行整个系统的恢复。数据库服务器必须处于脱机状态。</p> <p>整个系统的恢复完成后，服务器处于停顿方式。</p>

选项	描述
	如果指定 <code>gbackuprestore -r -w</code> 而不带整个系统的备份，将返回返回码 147，因为 <code>gbackuprestore</code> 找不到作为整个系统的备份一部分而备份的任何存储空间。
<code>-X</code>	停止连续逻辑日志恢复。将服务器保留在逻辑恢复暂挂状态中的停顿方式下，而不恢复其他日志。

### 用途

您可以恢复存储在原始文件和格式化文件中的存储空间。如果系统包含主存储空间和镜像存储空间，那么恢复（外部恢复除外）期间 `gbackuprestore` 将同时写入主块和镜像块。不能指定恢复临时空间。恢复关键数据库空间（例如根数据库空间）时，数据库服务器重新创建临时数据库空间，但它们为空。

如果 `BAR_MAX_BACKUP` 配置参数设置为大于 1 的值，那么 `gbackuprestore` 将并行恢复存储空间。要加速恢复，您可以添加更多 CPU 虚拟处理器。

在以下情况下，当数据库服务器处于联机状态时，可以在热恢复中恢复非关键存储空间：

- 存储空间处于联机状态，但它的一个块处于脱机、正在恢复或不一致状态。
- 存储空间处于脱机或关闭状态。

您无法同时执行多个热恢复。如果您需要恢复多个存储空间，请对 `gbackuprestore` 指定要恢复的存储空间集，或通过不显式指定任何空间来允许 `gbackuprestore` 恢复所有关闭的存储空间。

**提示：** 要在恢复中得到更快的性能，请为备份存储空间和逻辑日志指定不同的存储设备。如果物理备份和逻辑备份在存储介质上混合在一起，这将花费较长的时间用于在恢复期间扫描介质。

在某些情况下，您可能想分阶段地执行恢复。如果有多个设备可用于恢复，那么可以分别或同时恢复多个存储空间，然后再执行单一的逻辑恢复。

缺省情况下，`gbackuprestore` 恢复最新的备份。如果不希望恢复最新的备份，可以从较旧备份中进行恢复：例如，在备份验证失败或备份介质丢失时。您可以执行时间点恢复或日志点恢复。或者，您也可以使存储管理器中的错误备份到期，运行 `onsmsync` 命令，然后从较旧备份中恢复。如果意外删除了某存储空间，您可以使用时间点恢复或日志点恢复来将其恢复。

通过使用 `-O` 选项，可以强制执行联机存储空间的恢复（除了关键数据库空间以外）。数据库服务器开始恢复每个存储空间之前会自动将其关闭。让存储空间脱机以确保在恢复进程中用户不会试图更新该空间中的表。

可以在冷恢复期间使用 `gbackuprestore` 指定新块路径和偏移量来重命名块。如果需要将存储空间恢复到与完成备份的磁盘不同的磁盘上，该选项很有帮助。可以重命名任意类型的块，包括关键块和镜像块。

示例：执行整个系统的恢复

整个系统的恢复是冷恢复，必须在服务器处于脱机状态时执行。以下命令恢复整个系统的备份：

```
gbackuprestore -r -w
```

示例：恢复特定存储空间

以下示例恢复两个特定存储空间 `fin_dbpace1` 和 `fin_dbpace2`：

```
gbackuprestore -r fin_dbpace1 fin_dbpace2
```

示例：分阶段执行热恢复

以下命令执行物理恢复，备份逻辑日志，然后执行逻辑恢复：

```
gbackuprestore -r -p  
gbackuprestore -b -l  
gbackuprestore -r -l
```

示例：时间点恢复

以下命令将数据库服务器数据恢复到其在特定日期和时间的状态：

```
gbackuprestore -r -t "2011-05-10 11:35:57"
```

在此示例中，恢复将重放在指定时间或指定时间之前落实的事务，包括带有落实时间 11:35:57 的所有事务。正在处理但未在 11:35:57 之前落实的事务将回滚。

示例：法语语言环境中的时间点恢复

法语语言环境 `fr_fr.8859-1` 的缺省日期和时间格式使用格式

“`%A %.1d %B %iY %H:%M:%S`”。

以下命令将数据恢复到针对法语语言环境格式化的特定时间点：

```
gbackuprestore -r -t "Lundi 6 Juin 2011 11:20:14"
```

可以将 `GL_DATETIME` 设置为使用日期、月份、两位数表示的年份、小时、分钟和秒的不同日期和时间格式。例如：

```
%.1d %B %iy %H:%M:%S
```

以下命令通过为法语语言环境指定两位数年份来恢复到特定时间点：

```
gbackuprestore -r -t "6 Juin 11 11:20:14"
```

示例：分阶段的时间点恢复

以下命令执行到相同时间点的物理恢复和逻辑恢复：

```
gbackuprestore -r -p -t "2011-05-10 11:35:57"  
gbackuprestore -r -l -t "2011-05-10 11:35:57"
```

示例：恢复已删除的存储空间和块

假设某个事务删除了名为 `dbspace1` 的存储空间，并在时间 `2011-05-10 12:00:00` 删除了块。以下命令在服务器处于脱机状态时恢复该存储空间并重新创建已删除的块：

```
gbackuprestore -r -t "2011-05-10 11:59:59" -O
```

### 避免回收逻辑日志

`gbackuprestore -r` 命令自动回收逻辑日志。但是，在某些情况下需要避免回收逻辑日志。

使用 `gbackuprestore -r -p` 和 `gbackuprestore -r -l` 命令跳过日志回收。

如果将 `LTAPEDEV` 配置参数设置为 `/dev/null`（UNIX™ 上），那么不会在任何 `gbackuprestore` 恢复中回收逻辑日志（例如，`gbackuprestore -r` 或 `gbackuprestore -r -w`）。

应在以下情况下避免回收逻辑日志：

- 执行导入的恢复时  
在源数据库服务器而不是在目标数据库服务器上回收逻辑日志。
- 执行冷恢复前重新初始化数据库服务器 (`oninit -i`)  
重新初始化将创建新逻辑日志，而其中不包含您要恢复的数据。
- 为包含逻辑日志的数据库空间安装新磁盘  
从旧磁盘而不是新磁盘回收日志。

### 执行冷恢复

如果关键存储空间由于磁盘故障或毁坏的数据而受损，您必须执行冷恢复。如果磁盘出故障，那么须在可以执行冷恢复之前替换该磁盘以恢复数据。

如果在没有备份的情况下尝试执行冷恢复，那么未备份的存储空间中的数据将丢失。

要执行冷恢复：

1. 通过运行 `gadmin -ky` 命令来关闭服务器。

2. 如果必须更换或维修包含逻辑日志文件的磁盘，请使用 `gbackuprestore -b -l -s` 命令回收损坏磁盘上的逻辑日志文件。  
否则，`gbackuprestore` 将自动回收逻辑日志。
3. 如有必要，请维修或更换损坏的磁盘。
4. 如果 `GBS_HOME` 中的文件已损坏，请将管理文件的备份复制到其原始位置。  
否则，不需要复制管理文件。
5. 通过运行 `gbackuprestore -r` 命令来恢复关键和非关键存储空间。  
恢复完成后，数据库服务器处于停顿方式。
6. 通过运行 `gadmin -m` 命令来启动服务器。
7. 通过运行 `onsmsync` 命令来同步存储管理器。

### 通过使用 `gbackuprestore` 来配置连续日志恢复

使用连续日志恢复可保持辅助系统（热备份）可用，以在主系统发生故障时替换主系统。主系统和辅助系统上的 GBase 8s 版本必须相同。

要使用 `gbackuprestore` 配置连续的日志恢复：

1. 在主系统上，使用 `gbackuprestore -b -L 0` 命令执行 0 级备份。
2. 导入已在辅助服务器存储管理器中创建的备份对象。
3. 在辅助系统上，使用 `gbackuprestore -r -p` 命令执行物理恢复。  
在辅助系统上完成物理恢复后，数据库服务器将在快速恢复方式下等待恢复逻辑日志。
4. 在主系统上，使用 `gbackuprestore -b -l` 命令备份逻辑日志。
5. 将备份的逻辑日志传输到辅助系统，然后使用 `gbackuprestore -r -l -C` 命令将其恢复。
6. 对可用于备份与恢复的所有逻辑日志重复步骤 4 和 5。
7. 如果您要在紧急备用的辅助系统上执行连续日志恢复，请运行以下命令来完成逻辑日志恢复并停顿服务器：
  - 如果逻辑日志可用于恢复，请运行 `gbackuprestore -r -l` 命令。
  - 在恢复所有可用逻辑日志之后，运行 `gbackuprestore -r -l -X` 命令。

### 使用混合恢复来恢复数据

在您需要恢复服务器时，紧急数据处于联机状态并且为可用之前，您可以使用混合恢复来减少时间。紧急数据是您认为对于业务运营很关键的数据。

在混合恢复中，您首先对关键数据库空间（根数据库空间以及包含物理日志和逻辑日志的数据库空间）和包含紧急数据的数据库空间执行冷恢复。由于您并不恢复所有数据库空间，因此可以使服务器更快联机。然后在一个或多个热恢复中恢复剩余的存储空间。



要执行混合恢复：

1. 通过运行 `gadmin -ky` 命令来关闭数据库服务器。
2. 通过使用关键和紧急数据库空间名称列表运行 `gbackuprestore -r` 命令来对关键和紧急数据库空间执行冷恢复。

可以指定从较旧备份进行恢复的时间点。

3. 通过运行 `gadmin -m` 命令来启动服务器。
4. 通过运行 `onsmsync` 命令来同步存储管理器。
5. 通过运行 `gbackuprestore -r` 命令来对剩余存储空间执行热恢复。

您可以执行多个热恢复以对某些存储空间划分优先级。

示例

### 示例 1：简单混合恢复

一个数据库服务器除根数据库空间之外，还有以下五个数据库空间：`logdbs`、`dbs_1`、`dbs_2` 和 `dbs_3` 和 `dbs_4`。逻辑日志存储在 `logdbs` 中，物理日志位于根数据库空间中。必须在初始冷恢复期间恢复的关键数据库空间为 `rootdbs` 和 `logdbs`。包含紧急数据的数据库空间是 `dbs_1`。以下命令将关闭数据库服务器，对关键和紧急数据库空间执行冷恢复，然后重新启动数据库服务器：

```
gadmin -ky
gbackuprestore -r rootdbs logdbs dbs_1
gadmin -m
```

数据库服务器启动后，`rootdbs`、`logdbs` 和 `dbs_1` 数据库空间中存储的任何数据都可访问。

以下命令同步存储管理器，并对剩余数据库空间 `dbs_2`、`dbs_3` 和 `dbs_4` 执行热恢复：

```
onsmsync
gbackuprestore -r
```

### 示例 2：时间点混合恢复

以下命令在初始冷恢复中对存储空间的子集（包括所有关键数据库空间）执行冷恢复，对 `dbspace_2` 和 `dbspace_3` 执行热恢复，接着对 `dbspace_4` 和 `dbspace_5` 执行热恢复，最后对所有剩余的存储空间执行热恢复：

```
gbackuprestore -r -t "2011-05-10 11:35:57" rootdbs logspace_1 dbspace_1
gadmin -m
onsmsync
gbackuprestore -r dbspace_2 dbspace_3
gbackuprestore -r dbspace_4 dbspace_5
```

```
gbackuprestore -r
```

### 针对使用混合恢复的策略

要实现混合恢复策略，请仔细选择在创建数据库和数据库对象时要将其放置到的数据库空间集。

`gbackuprestore` 会备份与恢复物理实体而非逻辑实体。因此，`gbackuprestore` 无法恢复特定数据库或一组特定的表。相反，`gbackuprestore` 将恢复一组特定的存储空间。跟踪这些存储空间中存储的内容将取决于您。

例如：考虑数据库空间 `cat_dbs` 中的带有目录的数据库：

```
create database mydb in cat_dbs with log;
```

在数据库空间 `tab_dbs_1` 和 `tab_dbs_2` 之间将对该数据库中的某个表进行分段：

```
create table mytab (i integer, c char(20))  
fragment by round robin in tab_dbs_1, tab_dbs_2;
```

表的索引存储在数据库空间 `idx_dbs` 中：

```
create index myidx on mytab(i) in idx_dbs;
```

如果您需要恢复服务器，那么只有在恢复包含数据库目录、表数据和索引的数据库空间之后，您才能访问示例数据库中的所有数据：在本例中，为数据库空间 `cat_dbs`、**`tab_dbs_1`**、**`tab_dbs_2`** 和 **`idx_dbs`**。

要简化数据的管理和跟踪，请将数据库空间集划分为用于存储特定紧急程度的数据的子集。当您创建数据库对象时，请将这些数据库对象放置在适合其紧急程度的数据库空间中。例如，如果您有三个紧急程度级别的数据，可能希望将与最紧急数据相关联的所有对象（数据库目录、表和索引）放置在特定的数据库空间集内：例如，`urgent_dbs_1`、`urgent_dbs_2`、...`urgent_dbs_n`。您会将与紧急程度较低的数据相关联的所有对象放在不同的数据库空间集内：例如，`less_urgent_dbs_1`、`less_urgent_dbs_2`、...`less_urgent_dbs_k`。最后，您会将其余数据放在不同的数据库空间集内：例如，`non_urgent_dbs_1`、`non_urgent_dbs_2`、...`non_urgent_dbs_r`。

如果您需要恢复服务器，那么您应首先对所有关键数据库空间和包含紧急数据的数据库空间（从 `urgent_dbs_1` 到 `urgent_dbs_n`）执行冷恢复。例如，假设在 `logdbsp_1` 和 `logdbsp_2` 这两个数据库空间之间分发逻辑日志，并且物理日志在 `rootdbs` 中。因此，关键的数据库空间就是 `rootdbs`、`logdbsp_1` 和 `logdbsp_2`。

通过发出以下 `gbackuprestore` 命令来执行初始冷恢复：

```
gbackuprestore -r rootdbs logdbsp_1 logdbsp_2 urgent_dbs_1 ... urgent_dbs_2
```

您可以使服务器联机，并且所有业务紧急数据都可用。

接下来，对不怎么紧急的数据执行热恢复：

```
onsmsync  
gbackuprestore -r less_urgent_dbs_1 less_urgent_dbs_2 ..... less_urgent_dbs_k
```

最后，您可以通过发出以下命令来对服务器的剩余部分执行热恢复。

```
gbackuprestore -r
```

在具有几十个数据库空间的更大型系统中，可以将混合恢复的热恢复部分划分为几个热恢复，每个热恢复只恢复系统中剩余要恢复的数据库空间的其中一小部分。

### 恢复期间重新创建块文件

如果磁盘或文件系统发生故障，数据库空间中可能会缺少一个或多个块文件。使用 `-O` 选项可在恢复期间重新创建缺少的块文件以及任何必要的目录。

如果文件系统上存在的空间不足，恢复会失败。新创建的块文件是经过格式化的文件，并为 UNIX™ 上的组 `gbasedbt` 所有。

### 在使用热块时恢复

可以在恢复期间重新创建缺少的热块文件。

**限制：** 如果逻辑日志中包含块创建记录，那么 `gbackuprestore` 不会在逻辑恢复期间重新创建块文件。

要在使用热块时恢复：

1. 安装新磁盘。
2. 基于您的系统，执行以下某个任务：
  - 。 在 UNIX™ 上将设备作为文件系统安装。
3. 为块文件分配磁盘空间。
4. 运行 `gbackuprestore -r -O space` 命令来重新创建块文件并恢复数据库空间。

### 在使用原始块时恢复

可以在恢复期间重新创建缺少的原始块文件。

要在使用原始块时恢复：

1. 安装新磁盘。
2. 对于 UNIX™，如果使用原始设备的符号链接，那么为脱机块创建指向新安装的磁盘的新链接。  
`gbackuprestore` 恢复符号链接指向的块文件。
3. 发出 `gbackuprestore -r space` 命令来恢复数据库空间。

### 重新初始化数据库服务器并恢复数据

重新初始化磁盘空间将毁坏数据库服务器管理的所有现有数据。但是，您可以根据重新初始化之前执行的备份来恢复数据。

您必须拥有所有存储空间的当前 0 级备份。

初始化期间，`gbackuprestore` 将紧急引导文件保存到其他地方，并启动一个全新的空紧急引导文件。因此，重新初始化数据库服务器之前执行的任何备份都无法识别。必须使用在初始化之前保存的紧急引导文件的副本来恢复先前的数据库服务器实例。

要重新初始化数据库服务器并恢复旧数据：

1. 将紧急引导文件、`oncfg` 文件和 `onconfig` 文件复制到不同的目录。
2. 在 `onconfig` 文件中将 `FULL_DISK_INIT` 配置参数设置为 1。
3. 关闭数据库服务器。
4. 通过运行 `oninit -i` 命令来重新初始化数据库服务器。
5. 将管理文件移至数据库服务器目录。

如果管理文件不可用，将它们从最近一次备份复制到数据库服务器目录中。

6. 通过运行 `gbackuprestore -r -p -w` 命令来执行恢复。

不要回收逻辑日志。

7. 验证是否恢复了关键和非关键存储空间的正确实例。

### 3.4.4 恢复期间更换磁盘

您可以在恢复期间通过重命名块来更换磁盘。在使用 `gbackuprestore` 进行冷恢复期间，通过指定新块路径和偏移量来重命名块。如果需要将存储空间恢复到与完成备份的磁盘不同的磁盘上，该选项很有帮助。可以重命名任意类型的块，包括关键块和镜像块。

旧块必须包含在上一个 0 级备份中。

以下准则适用于新块：

- 新块不需要存在。可以以后安装新块并对包含它的存储空间执行热恢复。如果指定不存在的块，`gbackuprestore` 将重命名信息记录在块保留页中，但不恢复数据。已重命名（但未恢复）的块处于脱机状态，在 `gstat -d` 命令的输出中由 N 标志指示。
- 新块必须有正确的许可权。
- 新块必须包含在上一个 0 级备份中。
- 新块路径名不能与现有块相同，且偏移量不能重叠。

**提示：** 如果使用块名称的符号链接，可能不需要重命名块；而只需编辑符号名称定义即可。

要在恢复期间重命名块：

1. 关闭数据库服务器。
2. 使用 `-rename` 选项和块信息选项来运行 `gbackuprestore -r` 命令。

如果要对主根块或镜像根块进行重命名，那么 gbackuprestore 将更新 ROOTPATH 和 ROOTOFFSET 或者 MIRRORPATH 和 MIRROROFFSET 配置参数的值。旧版本的 onconfig 文件将另存为 \$ONCONFIG.localtime。

3. 执行 0 级归档，以便您可以恢复重命名的块。

示例

下表列出在本部分的示例中使用的两个块的示例值。

元素	第一个块的值	第二个块的值
旧路径	/chunk1	/chunk2
旧偏移量	0	10000
新路径	/chunk1N	/chunk2N
新偏移量	20000	0

#### 示例 1: 通过在命令中提供块信息来重命名块

以下命令将块 chunk1 重命名为 chunk1N，将块 chunk2 重命名为 chunk2N:

```
gbackuprestore -r -rename -p /chunk1 -o 0 -n /chunk1N -o 20000
                -rename -p /chunk2 -o 10000 -n /chunk2N -o 0
```

#### 示例 2: 通过在文件中提供块信息来重命名块

假设您有一个名为 listfile 的文件，其中包含以下内容:

```
/chunk1 0 /chunk1N 20000
/chunk2 10000 /chunk2N 0
```

以下命令将块 chunk1 重命名为 chunk1N，将块 chunk2 重命名为 chunk2N:

```
gbackuprestore -r -rename -f listfile
```

#### 将块重命名到不存在的设备上

要将块重命名到不存在的设备，请指定新的路径名，但在安装该物理设备之后再恢复存储空间。该选项在您需要重命名块时很有用，便于您在安装新设备前执行冷恢复。当新块设备就绪后，您可以在它上面执行存储空间的热恢复。

可以在同一个重命名操作中将重命名块与现有设备结合在一起，以及将重命名块与不存在的设备结合在一起。本示例显示如何将单个块重命名到不存在的设备名上。

下表列出本示例中使用的块的示例值。

存储空间	旧块的路径	旧偏移量	新块的路径	新偏移量

sbspace1	/chunk3	0	/chunk3N	0
----------	---------	---	----------	---

要将块重命名到不存在的设备上：

1. 使用以下命令重命名块：`gbackuprestore -r -rename -p /chunk3 -o 0 -n /chunk3N -o 0`
2. 当您看到以下提示时，请输入 `y` 以继续：

块 `/chunk3N` 不存在。如果继续，那么对包含该块的数据库空间的恢复操作稍后可能会失败。

在不创建该块的情况下，是否继续？(y/n)

块 `/chunk3` 被重命名为 `/chunk3N`，但数据还未恢复到 `/chunk3N`。

3. 执行 0 级归档。
4. 为 `/chunk3N` 添加物理设备。
5. 使用 `gbackuprestore -r sbspace1` 命令来执行 `sbspace1` 的热恢复。
6. 执行 0 级归档。

### 3.4.5 恢复到其他计算机

您可以备份一台计算机上的数据并在另一台计算机上恢复该数据。对于灾难恢复或升级数据库服务器，导入恢复很有用。备份数据并移至存储管理器对象上之后，可以执行导入的恢复。导入的恢复涉及将源计算机中的文件复制到目标计算机，以及以多种方式之一执行恢复。

先决条件：

- 您的存储管理器必须支持导入的恢复。
- 整个系统的备份必须包含所有存储空间；逻辑日志是可选的。  
0 级备份必须包含所有存储空间和逻辑日志。
- 源计算机和目标计算机都必须在相同的 LAN 或 WAN 上，并且必须具有以下属性：
  - 相同的硬件和操作系统
  - 相同的数据库服务器版本
  - 相同的配置和 `ROOTPATH` 信息，但服务器名称和数量可以不同。
  - 相同的存储管理器版本
  - 兼容的 XBSA 库

**重要：** 要完成导入恢复，源计算机和目标计算机上的每个块（包括镜像）的大小、位置和偏移量都必须精确匹配。

要执行导入的恢复：

1. 在目标计算机上安装数据库服务器和存储管理器。
2. 在目标数据库服务器实例上安装存储管理器。
  - a. 设置必要的环境变量。
  - b. 定义与源实例上类型相同的存储设备。
  - c. 用正确的池名标记存储介质。
  - d. 安装存储设备。
  - e. 用存储管理器的版本更新目标计算机上的 `sm_versions` 文件。
3. 对于与源计算机上的设备和链接匹配的块，确保目标计算机已将这些设备和链接准备就绪
4. 对源数据库服务器上的所有存储空间执行 0 级备份（`gbackuprestore -b` 或 `gbackuprestore -b -w`）。  
限制：不要执行增量备份。

5. 如果在使用 Storage Manager，请遵循以下步骤：

- a. 关闭两台计算机上的存储管理器。
- b. 在源计算机上创建存储管理器目录的 `tar` 文件。
- c. 将这个 `tar` 文件复制到目标计算机上并解包。

如果使用其他存储管理器，您可以使用备份磁带或通过网络导入存储管理器目录。要了解更多信息，请参阅您的存储管理器文档。

6. 安装传送的存储卷。
  - o. 如果备份文件在磁盘上，请将它们从源计算机复制到目标计算机上。
  - o. 如果备份位于磁带上，请在连接到目标计算机的存储设备上安装传输的卷。源计算机和目标计算机都必须使用相同类型的存储设备，如 8 毫米磁带或磁盘。
  - o. 如果备份位于备份服务器上，请从该备份服务器检索该备份。
7. 使用存储管理器命令将源主机名作为客户机添加到目标计算机上。
8. 将以下文件从源计算机复制到目标计算机上。

表 1. 要复制的管理文件

文件	操作
紧急引导文件	用目标数据库服务器编号重命名紧急引导文件。例如：将 <code>ixbar.51</code> 重命名为 <code>ixbar.52</code> 。紧急引

文件	操作
	<p>导文件只需要来自源计算机上 0 级备份的条目。</p> <p>文件名为 <code>ixbar.servernum</code>。</p>
<p>oncfg 文件： <code>oncfg_servername.servernum</code></p>	<p><code>gbackuprestore</code> 需要让 <code>oncfg</code> 文件知道要检索哪些数据库空间。用目标数据库服务器的名称和编号重命名 <code>oncfg</code> 文件。例如：</p> <p>将 <code>oncfg_bostonserver.51</code> 重命名为 <code>oncfg_chicagoserver.52</code>。该文件名必须与目标计算机上的 <code>sqlhosts</code> 文件中的 <code>dbservername</code> 条目和 <code>SERVERNUM</code> 相匹配。</p>
onconfig 文件	在 <code>onconfig</code> 文件中,使用目标数据库服务器编号更新 <code>SERVERNUM</code> 参数。
存储管理器配置文件, 如果有该文件	存储管理器配置文件可能需要更新。

9. 使用以下方法之一恢复数据：

表 2. 恢复数据选项

选项	操作
如果未在目标服务器上启动 GBase 8s 实例	使用 <code>gbackuprestore -r</code> 命令来恢复数据。
如果要导入整个系统备份	使用 <code>gbackuprestore -r -w -p</code> 命令来恢复数据。
如果已在目标服务器上启动了 GBase 8s 实例。	<p>分阶段恢复数据：</p> <ul style="list-style-type: none"> <li>使用 <code>gbackuprestore -r -p</code> 命令来恢复物理数据。</li> <li>使用 <code>gbackuprestore -r -l</code> 命令来恢复逻辑日志。</li> </ul> <p>此过程可避免回收日志和对实例的任何潜在破</p>



选项	操作
	坏。

10. 在您使用 `onsmsync` 实用程序使目标计算机和存储管理器上的对象到期之前，请执行以下某个任务。

否则，`onsmsync` 将使错误对象到期。

- 手动编辑目标计算机上的 `$GBS_HOME/etc` 目录中的紧急引导文件 `viz ixbar.servernum`。将源计算机上使用的 GBase 8s 服务器名称替换为目标计算机的 GBase 8s 服务器名称。
- 在目标计算机上以 `gbasedbt` 用户身份运行 `onsmsync -b` 命令，以只从 `sysutils` 数据库重新生成紧急引导文件。重新生成的紧急引导文件反映了目标计算机的服务器名称。

#### 使用 `gbackuprestore` 和 `Storage Manager` 执行导入的恢复的示例

此示例显示如何使用 `gbackuprestore`、`Storage Manager` 和归档的文件设备来设置实例的导入恢复。

有多种方式可执行导入的恢复。此示例显示的是 `Storage Manager` 目录复制方法。另一种方法是引导程序恢复方法，该方法在《GBase 8s `Storage Manager` 管理员指南》中描述。

此示例的先决条件：

- 带有相同配置的源机器和目标机器。但是，服务器名称和编号可以不同。
- 在两台机器上相同的 `ROOTPATH`。
- 目标机器已为块准备好设备和链接，并且这些设备和链接与源机器上的设备和链接相匹配。
- `Storage Manager` 已在两台计算机上初始化。
- 设备目录路径、卷名称和池名称在两台机器上都相同。
- `root` 用户和 `gbasedbt` 用户是两台机器上的 `Storage Manager` 管理员。

在此示例中，数据库空间和日志备份的目录如下：

<目录路径>/dbspaces1

<目录路径>/logfiles1

源环境中设置的其他环境参数如下：

`ISM_server = source computer`

```
export IDS_server
```

目标机器上设置的其他环境参数如下：

```
ISM_client = source computer
```

```
export IDS_client
```

```
SM_server = target computer
```

```
export ISM_server
```

1. 以 gbasedbt 用户身份，在源机器上执行 0 级备份。
2. 以 root 用户身份，通过运行以下命令来停止两台计算机上的 Storage Manager: %ism\_shutdown
3. 以 root 用户身份，压缩源机器上的相应 Storage Manager 目录，如下所示：

```
%cd /nsr  
%tar -cvf nsr.tar index mm
```

4. 将上一步中的 nsr.tar 文件以二进制方式通过 FTP 传输至目标机器。
5. 以 root 用户身份，解压缩目标机器上的 nsr.tar 文件，如下所示：

```
%cd /nsr  
%tar -xvf nsr.tar
```

6. 以 root 用户身份，在源机器上以 tar 格式压缩备份目录（设备），如下所示：

```
%tar -cvf logfiles1.tar logfiles1  
%tar -cvf dbspaces1.tar dbspaces1
```

7. 将上一步中的归档目录以二进制方式通过 FTP 传输至目标机器。
8. 以 root 用户身份，在目标机器上将现有日志和归档目录覆盖为源机器中的目录，如下所示：

```
%tar -xvf logfiles1.tar  
%tar -xvf dbspaces1.tar
```

9. 以 root 用户身份，在目标机器上  
通过运行以下命令来启动 Storage Manager: ism\_startup  
运行 ism\_show -devices 命令以将设备显示为已安装。

10. 在目标机器上，使用以下内容创建一个文件（例如，nsr.txt）：

```
create type: NSR client; name: source_machine;  
remote access: root@target_machine, gbasedbt@target_machine;
```

11. 以 gbasedbt 用户身份，在目标机器上运行以下命令：

```
%nsradmin -s target_machine -i nsr.txt
```

该命令返回以下输出：created resource id <IP information>

12. 以 `gbasedbt` 用户身份，将源机器上 `$GBS_HOME/etc` 目录中的以下文件通过 FTP 传输至目标机器上的 `$GBS_HOME/etc` 目录

`ixbar,servernum`

`oncfg_servername,servernum`

13. 在目标机器上，将上一步中的文件的文件名更改为本地服务器的对应名称。

14. 运行恢复命令。

### 3.4.6 gbackuprestore -RESTART 语法：重新启动失败的恢复

如果在恢复期间数据库服务器、介质、存储管理器或 `gbackuprestore` 发生故障，可以从发生故障之处重新启动该恢复。要重新启动失败的恢复，当恢复失败时，`onconfig` 文件中的 `RESTARTABLE_RESTORE` 配置参数必须设置为 `ON`。

重新启动恢复

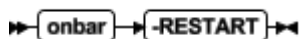


表 1. `gbackuprestore -RESTART` 命令.

选项	描述
-RESTART	<p>在数据库服务器、存储管理器或 <code>gbackuprestore</code> 发生故障后重新启动恢复。</p> <p>当恢复失败时，<code>RESTARTABLE_RESTORE</code> 配置参数必须设置为 <code>ON</code>。</p> <p>可以重新开始以下类型的恢复：</p> <ul style="list-style-type: none"> <li>• 整个系统</li> <li>• 时间点</li> <li>• 存储空间</li> <li>• 冷恢复的逻辑部分</li> </ul> <p>如果在热逻辑恢复期间发生故障，请勿使用 <code>-RESTART</code> 选项。</p>

用途

当您启用可重新启动的恢复时，如果恢复的逻辑日志很多，逻辑恢复将会变慢。但是，如果恢复失败后重新启动恢复，可以节省时间。恢复是否可重新启动不会影响物理恢复的速度。

物理恢复在发生故障的存储空间和级别处重新启动。当恢复了存储空间的某些块而不是所有块时，如果恢复失败，那么将恢复该存储空间的所有块。失败前如果存储空间和增量备份已成功恢复，那么不会再次恢复它们。

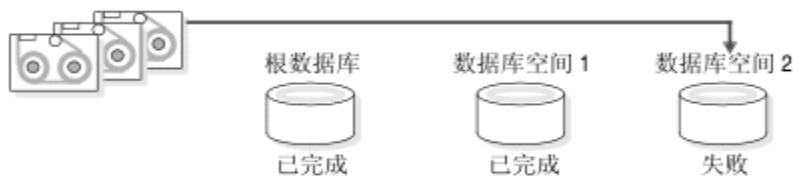
如果 `BAR_RETRY` 配置参数设置为 2，那么 `gbackuprestore` 会自动再次尝试恢复任何失败的存储空间和逻辑日志。如果恢复成功，那么不需要重新启动恢复。

如果 `BAR_RETRY` 配置参数设置为 0 或 1，那么 `gbackuprestore` 不会再次尝试恢复任何失败的存储空间和逻辑日志。如果数据库服务器还在运行，`gbackuprestore` 跳过失败的存储空间并尝试恢复剩余的存储空间。要完成恢复，请运行 `gbackuprestore -RESTART` 命令。

下图显示如果物理恢复 `dbspace2` 期间恢复失败，可重新启动的恢复是如何工作的。`rootdbs` 的 0 级、1 级和 2 级备份，`dbspace1` 和 `dbspace2` 的 0 级备份和 1 级备份都已成功恢复。恢复 `dbspace2` 的 1 级备份时数据库服务器出现故障。重新启动恢复时，`gbackuprestore` 将恢复 `dbspace1` 的 2 级备份，`dbspace2` 的 1 级和 2 级备份以及逻辑日志。

**图: 可重新启动的物理恢复**

数据库空间 2 的物理恢复过程失败:



重新启动恢复:



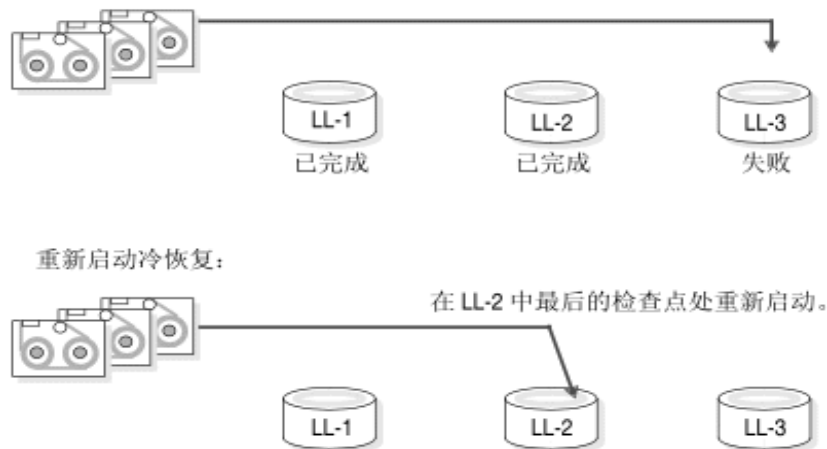
如果恢复在逻辑阶段期间失败，然后您重新启动该恢复，那么 `gbackuprestore` 会验证存储空间是否已恢复，跳过物理恢复并重新启动逻辑恢复。下图显示在恢复逻辑日志 LL-3 时冷恢复失败的情况。当重新启动冷逻辑恢复时，从最近的恢复检查点开始重放日志。在本示例中，最近的检查点在逻辑日志 LL-2 中。

如果在冷逻辑恢复期间发生故障，`gbackuprestore` 将在发生故障之处重新启动该恢复。

**重要：** 如果在热逻辑恢复期间发生故障，请从头重新启动该恢复。如果数据库服务器仍在运行，请运行 `gbackuprestore -r -l` 命令来完成恢复。

### 图：可重新启动的冷逻辑恢复

在 LL-3 的逻辑恢复期间冷恢复失败。最后的检查点位于 LL-2 中。



### 3.4.7 解决失败的恢复

解决失败的恢复的方式取决于失败的原因。

即使可重新开始的恢复被关闭，您仍然可以挽救某些失败的恢复。例如：如果恢复由于存储管理器或存储设备错误而失败，您可以修复磁带机或存储管理器的问题，重新安装磁带，然后继续进行恢复。

下表显示当物理恢复失败并且 `BAR_RETRY` 配置参数的值  $> 1$  时预期的结果。

表 1. 失败的物理恢复方案

错误类型	RESTARTABLE_RESTORE 设置	物理恢复失败时采取什么措施？
数据库服务器、 <code>gbackuprestore</code> 或存储管理器错误（数据库服务器仍在运行）	ON 或 OFF	<code>gbackuprestore</code> 重试每个失败的恢复。如果存储管理器失败，请修复存储管理器的错误。  如果尝试的恢复失败，请发出 <code>gbackuprestore -r spaces</code> 命令，其中 <code>spaces</code> 是还没有恢复的存储空间列表。使用 <code>gstat -d</code> 获取需要恢复的存储空间列表。 <code>gbackuprestore</code> 恢复每个存储空间的 0 级备份，接着是 1 级和 2 级备份（如果备份存在）。
<code>gbackuprestore</code> 或存储管理	启用	发出 <code>gbackuprestore -RESTART</code> 命令。

错误类型	RESTARTABLE_ RESTORE 设置	物理恢复失败时采取什么措施?
器错误（数据库服务器仍在运行）		如果存储管理器失败，请修复存储管理器的错误。  恢复将从第一个恢复失败的存储空间和备份级别处重新启动。如果成功恢复存储空间的 0 级备份，重新启动的恢复将跳过 0 级备份并恢复 1 级和 2 级备份（如果备份存在）。
数据库服务器故障	ON 或 OFF	由于数据库服务器已关闭，所以执行冷恢复。使用 <code>gbackuprestore -r</code> 恢复关键数据库空间以及第一次未恢复的所有非关键空间。
数据库服务器故障	启用	发出 <code>gbackuprestore -RESTART</code> 命令。  恢复将从第一个恢复失败的存储空间和备份级别处重新启动。如果成功恢复存储空间的 0 级备份，重新启动的恢复将跳过 0 级备份并恢复 1 级和 2 级备份（如果备份存在）。

下表显示当逻辑恢复失败时将出现的结果。

表 2. 失败的逻辑恢复方案

错误类型	RESTARTABLE_ RESTORE 设置	逻辑恢复失败时采取什么措施?
冷恢复中数据库服务器 或 <code>gbackuprestore</code> 发生错误（数据库服务器仍在运行）	启用	发出 <code>gbackuprestore -RESTART</code> 命令。  逻辑恢复在最近的检查点处重新启动。如果该恢复失败，关闭并重新启动数据库服务器来启动逻辑日志的快速恢复。所有未恢复的逻辑日志将会丢失。
数据库服务器 或 <code>gbackuprestore</code> 错误（数据库服务器仍在运行）	ON 或 OFF	发出 <code>gbackuprestore -r -l</code> 命令。恢复将在失败的逻辑日志处重新启动。  如果 <code>gbackuprestore -r -l</code> 仍然失败，关闭并重新启动数据库服务器。数据库服务器将完成快速恢复。所有未恢复的逻辑日志将会丢失。

错误类型	RESTARTABLE_ RESTORE 设置	逻辑恢复失败时采取什么措施?
		如果快速恢复不起作用，您必须执行冷恢复。
数据库服务器错误	启用	如果冷逻辑恢复失败，那么发出 <code>gbackuprestore -RESTART</code> 。  如果热逻辑恢复失败，那么发出 <code>gbackuprestore -r -l</code> 命令。如果该命令失败，从头开始重新启动整个恢复。
存储管理器错误	ON 或 OFF	<code>gbackuprestore</code> 重试每个失败的逻辑恢复。如果重试的恢复失败，那么逻辑恢复将暂挂。请修复存储管理器的错误。然后发出 <code>gbackuprestore -r -l</code> 命令。恢复将在失败的逻辑日志处重新启动。

## 3.5 外部备份与恢复

这些主题讨论使用外部备份与恢复来恢复数据的方法。

### 3.5.1 外部备份与恢复概述

外部备份与恢复会消除系统的停机时间，因为备份与恢复操作都是在 GBase 8s 系统外部执行的。

在备份或物理恢复期间 `gbackuprestore` 不移动数据。外部备份使您不必使用 `gbackuprestore` 就可以复制包含存储空间块的磁盘。当磁盘发生故障时，将其更换并使用供应商软件来恢复数据，然后使用 `gbackuprestore` 进行逻辑恢复。有关更多信息，请参阅在外部恢复中恢复数据。

以下是外部备份与恢复的典型情况：

- 磁盘镜像的可用性

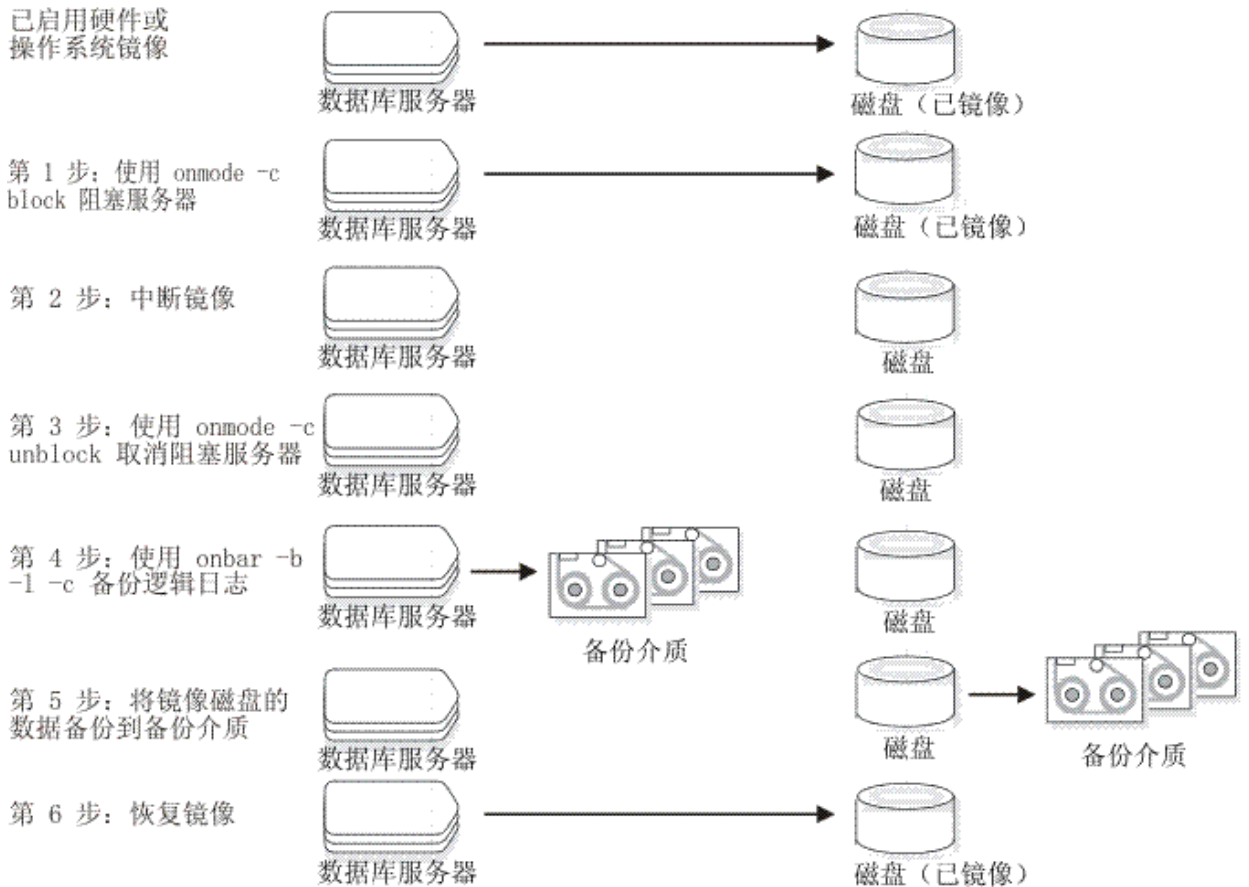
如果使用硬件磁盘镜像，与使用常规的 `gbackuprestore` 命令相比，使用外部备份与恢复可以使系统更快地联机。

- 克隆

可以使用外部备份与恢复来克隆现有的生产系统，以在不干扰生产系统的情况下对其进行测试或迁移。

下图显示如何使用镜像来执行备份。

**图: 使用镜像执行备份**



在此配置中，除了数据库服务器被阻塞而中断镜像的短时间之外，数据库服务器持续运行。镜像磁盘包含数据库服务器存储空间的副本。要创建备份，请阻塞数据库服务器以停止事务并禁用镜像。已镜像的磁盘现在包含位于特定时间点的一致数据的副本。在禁用镜像之后，请对数据库服务器取消阻塞以允许事务重新开始，然后备份逻辑日志。使用外部命令将数据从脱机的镜像磁盘复制到备份介质中。现在您可以重新开始制作镜像了。

**备份前阻塞**

开始外部备份之前，需要阻塞数据库服务器。阻塞将强制设置检查点，清空缓冲区、将其中的内容都保存到磁盘，并阻塞包含临时表的用户事务。



在阻塞操作期间，用户可采用只读方式访问数据库服务器。接着您可以使用操作系统或第三方的工具以物理方式将数据备份或复制到另一组磁盘或存储介质上。完成外部备份后，取消阻塞数据库服务器以使事务能继续进行。在外部备份中应包含每个存储空间中的所有块文件、管理文件（例如onconfig）以及紧急引导文件。

**重要：** 要使跟踪备份更加容易，您应该在每个外部备份中备份所有存储空间。

gbackuprestore 将外部备份当做 0 级备份同等对待。不能执行外部备份后接着使用 gbackuprestore 执行 1 级备份，反之亦然，因为 gbackuprestore 没有外部备份的任何记录。有关更多信息，请参阅在未对块进行镜像时执行外部备份。

## 外部备份的规则

开始外部备份之前，请查看执行外部备份的规则。

必须遵循的规则为：

- 外部备份期间数据库服务器必须处于联机状态或停顿方式。
- 用 gbackuprestore 备份包括当前日志在内的所有逻辑日志，这样您就可以在外部恢复结束时恢复逻辑日志。
- 阻塞数据库服务器以进行外部备份之前请暂挂连续逻辑日志备份。外部备份完成后，恢复连续逻辑日志备份。  
要停止连续逻辑日志备份，请使用 CTRL-C 命令。要恢复连续逻辑日志备份，请使用 gbackuprestore -b -l -C 命令。
- 等到所有 gbackuprestore 备份会话完成后再阻塞数据库服务器。如果有任何备份会话是活动的，阻塞命令将显示错误消息。
- 当数据库服务器阻塞时，所有 OLTP 工作或查询都将暂挂。直到数据库服务器取消阻塞后它们才会继续进行。
- 数据库服务器实例中所有关键的数据空间必须使用同一个 gadmin -c block ... gadmin -c unblock 命令编组一同进行备份。在不同时间对不同的关键数据库空间所做的备份无法恢复到一致的系统中。
- 在 AIX<sup>®</sup> 操作系统上，如果由于 DIRECT\_IO 配置参数设置为启动并发 I/O，而由此服务器运行时使用的是并发 I/O，那么联机的外部备份程序也必须使用并发 I/O。

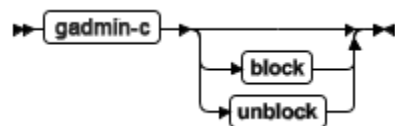
**重要：** 由于外部备份不受 gbackuprestore 的控制，因此必须手动跟踪这些备份。有关更多信息，请参阅跟踪外部备份。

## 准备进行外部备份

这些主题描述了用于准备外部备份的命令。有关过程的信息，请参阅在未对块进行镜像时执行外部备份。

**阻塞和取消阻塞数据库服务器**

本主题显示了 GBase 8s 上阻塞和取消阻塞命令的语法。



元素	用途	关键注意事项
gadmin -c	执行检查点并阻塞或取消阻塞数据库服务器	无。
block	阻止数据库服务器执行任何事务	设置数据库服务器以进行外部备份。当数据库服务器被阻塞时，用户可采用只读方式进行访问。样本命令：gadmin -c block
unblock	取消阻塞数据库服务器，从而允许数据事务和正常的数据库服务器操作可以继续进行	在外部备份完成前请勿取消阻塞。样本命令：gadmin -c unblock

**跟踪外部备份**

数据库服务器和 gbackupstore 不会跟踪外部备份。要跟踪外部备份数据，请使用第三方存储管理器或手动跟踪数据。

下表显示了建议您在外部备份中跟踪的项。gbackupstore 保留有限的外部恢复历史记录。

表 1. 使用外部备份与恢复时要跟踪的项

要跟踪的项	示例
每个已备份存储空间的各个块文件的完整路径名	/work/dbspaces/rootdbs (UNIX™)
对象类型	关键数据库空间、非关键存储空间
ins_copyid_hi 和 ins_copyid_lo	存储管理器分配给每个备份对象的副本标识
备份日期和时间	阻塞和取消阻塞数据库服务器的时间
备份介质	磁带卷编号或磁盘路径名
数据库服务器版本	从中进行备份的数据库服务器版本。

### 在未对块进行镜像时执行外部备份

外部备份期间数据库服务器必须处于联机状态或停顿方式。

要在未对块进行镜像时执行外部备份：

1. 要获取外部备份，请使用 `gadmin -c block` 命令来阻塞数据库服务器。  
系统将获取一个检查点并暂挂所有更新事务。用户能以只读方式访问数据库服务器。
2. 要备份存储空间和管理文件，请使用复制命令（如 UNIX™ 上的 `cp`、`dd` 或 `tar`）或文件备份程序。  
必须备份存储空间中的所有块。
3. 要允许正常操作继续，请使用 `gadmin -c unblock` 命令来取消阻塞数据库服务器。
4. 备份包括当前日志在内的所有逻辑日志，这样就可以将检查点信息用于外部恢复。  
**重要：** 由于外部备份并不是通过 `gbackuprestore` 完成的，因此必须确保具有从执行 `gadmin -c block` 命令时开始的当前逻辑日志的备份。没有该逻辑日志文件的备份，外部备份将是不可恢复的。
5. 执行外部备份后，使用 `gbackuprestore -b -l -c` 命令来备份当前日志。

如果丢失了磁盘或整个系统，现在便可以执行外部恢复。

### 3.5.2 RS 辅助服务器外部备份

您可以为 RS 辅助服务器执行外部备份。执行 RS 辅助服务器的备份会阻塞该 RS 辅助服务器，但不会阻塞主服务器。

您可以从主实例备份的日志中执行逻辑恢复。从辅助服务器获取的备份无法使用 1 级或 2 级备份进行恢复。

**重要：** 如果数据库实例包含以下任一项，那么表明外部备份未完成：

- 非日志记录的智能大对象
- 常规 Blob 空间
- 非日志记录数据库
- 原始表

如果在包含上述任一项的实例中执行了外部备份，那么该备份将无法完成且无法用于恢复主服务器。

如果由于主服务器检查点超时而导致备份失败，那么可以使用 `BAR_CKPTSEC_TIMEOUT` 配置参数来增加执行外部备份时 RS 辅助服务器应等待检查点从主服务器到达的时间量（以秒计）。

### 为 RS 辅助服务器执行外部备份

要为 RS 辅助服务器执行外部备份，不能启用 STOP\_APPLY 配置参数。如果启用 STOP\_APPLY，那么会返回错误。在 RS 辅助服务器上执行备份时，服务器会切换为 STOP\_APPLY 方式。处理归档检查点之后，RS 辅助服务器会停止应用逻辑日志，但继续从主服务器接收日志。

要对 DELAY\_APPLY 配置参数值大于 0 的 RS 辅助服务器执行外部备份，可能需要暂时减小此参数的值。执行备份需要 RSS 处理逻辑日志中的检查点，如果在以下过程第二步中 `gadmin -c block timeout` 命令指定的时间长度内看不到任何检查点，那么不允许进行备份。可以通过 `gadmin -wf DELAY_APPLY=setting` 命令来减小 DELAY\_APPLY 配置参数。

外部备份期间主数据库服务器必须处于联机状态或静默方式。

要执行外部备份：

1. 确保 RS 辅助服务器上的 LOG\_STAGING\_DIR 配置参数设置为有效的登台目录。
2. 要获取外部备份，请使用 `gadmin -c block timeout` 命令来阻塞数据库服务器。  
  
timeout 参数指示 RS 辅助服务器等待接收检查点的秒数。仅当在 RS 辅助服务器上运行 `gadmin -c block` 命令时，timeout 参数有效。在继续外部备份之前，您必须等待 `gadmin -c block` 命令成功返回。
3. 要备份存储空间和管理文件，请使用复制命令（如 UNIX™ 上的 `cp`、`dd` 或 `tar`）或文件备份程序。
4. 必须备份存储空间中的所有块。
5. 要恢复正常操作，请使用 `gadmin -c unblock` 命令来取消阻塞数据库服务器。
6. 执行外部备份之后，使用 `gbackuprestore` 或 `gtape` 实用程序来备份当前日志和任何新日志。

**重要：** 只能在主服务器上进行逻辑日志备份。

如果设置了 DELAY\_APPLY 配置参数，那么恢复进程所需的日志并不一定是当前在主服务器上处于活动状态的那些日志，因为某些日志可能已归档。

备份完成后，如果之前减小了 RS 辅助服务器上的 DELAY\_APPLY 设置，现在可以通过 `gadmin -wf DELAY_APPLY=setting` 命令将其设置为原始值。执行外部备份之后，如果磁盘或整个系统发生故障，可以执行外部恢复。

### 3.5.3 在外部恢复中恢复数据

仅当您在外部备份数据后，才能够在丢失磁盘或整个系统的情况下从外部进行恢复。必须对外部备份与恢复使用相同的第三方实用程序。要从外部恢复存储空间，请将已备份的数据复制到磁盘。使用 `gbackuprestore -r -e` 命令将存储空间标记为已物理恢复，接着重放逻

辑日志并让存储空间回到联机状态。如果未指定外部恢复命令，数据库服务器认为这些存储空间仍然是关闭的。

可以执行以下这些类型的外部恢复：

- 外部热恢复

将非关键存储空间标记为已物理恢复，然后对这些存储空间执行逻辑恢复。

- 外部冷恢复

将存储空间标记为已物理恢复，然后对所有存储空间执行逻辑恢复。还可以选择执行时间点外部冷恢复。

**限制：** 当执行外部冷恢复时，gbackuprestore 并不首先从数据库服务器尝试回收逻辑日志文件，因为外部备份已经复制了所有逻辑日志数据。

要回收逻辑日志，请在复制外部备份之前执行 `gbackuprestore -l -s`，接着执行外部恢复 (`gbackuprestore -r -e`)。

### 重命名块

您可以使用其他恢复方法的重命名选项语法，对外部冷恢复中的块进行重命名。恢复过程中使用以下命令指定新的块名称：

```
gbackuprestore -r -e -rename -f filename
```

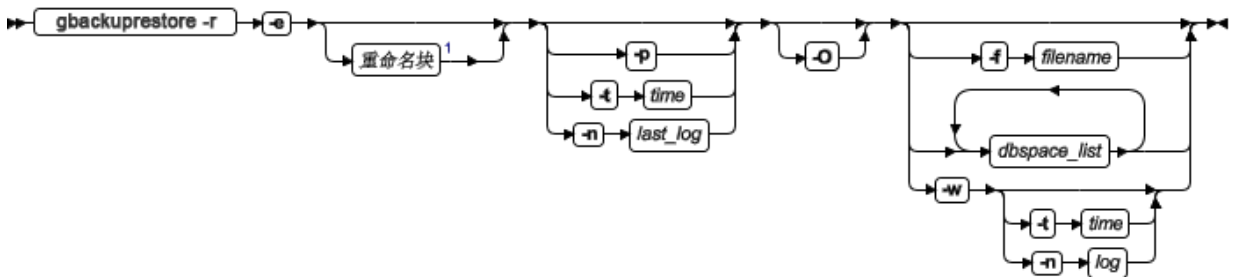
或

```
gbackuprestore -r -e rename -p old_path -o old_offset-n new_path-o new_offset
```

### 外部恢复命令

使用 `gbackuprestore -r -e` 命令来执行热或外部冷恢复。该命令将存储空间标记为已物理恢复并恢复逻辑日志。下图显示外部恢复语法。

使用 `gbackuprestore` 来执行外部恢复



元素	用途	关键注意事项
----	----	--------

元素	用途	关键注意事项
<code>gbackuprestore -r</code>	指定一个恢复	在冷恢复中，如果未指定存储空间名称，那么所有存储空间都将标记为已恢复。
<code>-e</code>	指定外部恢复	必须与 <code>-r</code> 选项一起使用。在外部热恢复中，除非指定 <code>-0</code> 选项，否则将关闭的存储空间标记为已恢复。
<code>dbspace_list</code>	在热恢复中指定要标记为已恢复的一个或多个存储空间	如果未输入 <code>dbspace_list</code> 或 <code>-ffilename</code> ，并且数据库服务器处于联机状态或停顿方式，那么 <code>gbackuprestore</code> 仅将关闭的存储空间标记为已恢复。如果输入多个存储空间名称，那么使用空格来分隔这些名称。
<code>-ffilename</code>	恢复文本文件中列出的存储空间，该文件的路径名由 <code>filename</code> 提供	要避免每次输入一长串存储空间，请使用该选项。 <code>filename</code> 可以是任何有效的 UNIX™ 文件名。
<code>-n last_log</code>	指示要恢复的最后一个日志的编号	如果该日志后还存在任何逻辑日志， <code>gbackuprestore</code> 将不恢复它们并且它们的数据也将丢失。 <code>-n</code> 选项不能与 <code>-p</code> 选项一起使用。
<code>-0</code>	恢复联机存储空间	无。
<code>-p</code>	仅指定外部物理恢复。	物理恢复完成后，您必须执行逻辑恢复。
<code>-t time</code>	恢复指定的时间点之前的最近一次备份。如果选择该时间点后所做的备份，那么恢复将失败。	只能在冷恢复中使用时间点恢复。必须恢复所有的存储空间。 如何输入时间取决于当前的 GLS 语言环境约定。如果未设置 GLS 语言环境，请使用英语样式的日期格式。
<code>-w</code>	自上次整个系统备份以来对所有的存储空间和逻辑日志	必须在冷恢复中指定 <code>-w</code> 选项。 如果不是对整个系统的备份指

元素	用途	关键注意事项
	执行整个系统的恢复	定 <code>gbackuprestore -r -w</code> ，将出现返回码 147，因为 <code>gbackuprestore</code> 找不到任何作为整个系统备份的一部分而备份的存储空间。

### 外部恢复的规则

外部恢复具有特定规则。

以下规则适用于外部恢复：

- 必须从外部备份进行外部恢复。尽管外部备份被视为 0 级备份，但它实际上可能是与 GBase 8s 无关的增量备份。
- 外部热恢复仅恢复非关键存储空间。
- 不能从外部恢复临时数据库空间。
- 不能从常规 `gbackuprestore` 备份进行外部恢复。
- 如果使用 `gbackuprestore`，您无法验证是否在从正确的备份进行恢复以及存储介质是否可读。
- 如果多个外部备份在不同的时间进行，那么外部恢复将使用最早的备份中的开始逻辑日志。
- 您不能执行混合恢复。如果必须恢复关键数据库空间，那么必须执行完整的冷恢复。

以下规则适用于外部冷恢复：

- 在切换包含关键存储空间的磁盘之前回收逻辑日志 (`gbackuprestore -b -l -s`)。
- 如果正在恢复关键数据库空间，那么数据库服务器必须处于脱机状态。
- 时间点外部恢复必须是冷恢复并且恢复所有存储空间。
- 数据库服务器实例的所有关键数据库空间的外部备份必须同时进行。必须在同一组 `gadmin -c block ... gadmin -c unblock` 命令中备份所有关键数据库空间。

### 3.5.4 执行外部恢复

本部分描述了执行冷和外部热恢复过程。

#### 执行外部冷恢复

如果在冷恢复中指定 `gbackuprestore -r -e` 命令，那么必须恢复所有的存储空间。使用 `gbackuprestore -r -e -p` 命令来恢复所有的或特定的存储空间。

要执行外部冷恢复：

1. 使用 `gadmin -ky` 命令来关闭数据库服务器。
  2. 使用 `gbackuprestore -b -l -s` 命令来回收逻辑日志。
  3. 要从外部备份恢复存储空间，请使用复制命令，例如 UNIX™ 上的 `cp`、`dd` 或 `tar`，或使用文件备份程序。
- 必须将存储空间恢复到与原始数据相同的路径中，并包含所有的块文件。
4. 要执行所有存储空间和逻辑日志的外部恢复，请使用 `gbackuprestore -r -e` 命令。
  5. 要执行所有存储空间的时间点外部恢复，请使用 `gbackuprestore -r -e -t datetime` 命令。

这一步让数据库服务器转入快速恢复方式。

`gbackuprestore` 和数据库服务器将前滚逻辑日志并使存储空间处于联机状态。

### 执行外部热恢复

在外部热恢复期间数据库服务器处于联机状态。外部热恢复仅涉及非关键存储空间。

要执行外部热恢复：

1. 要从外部备份恢复存储空间，请使用复制命令，例如 UNIX™ 上的 `cp`、`dd` 或 `tar`，或使用文件备份程序。
- 必须将存储空间恢复到与原始数据相同的路径中，并包含每个已恢复存储空间的所有块文件。
2. 对非关键存储空间执行外部热恢复以使它们处于联机状态。
    - 要恢复选定的存储空间和所有逻辑日志，请使用 `gbackuprestore -r -e dbspace_list` 命令。
    - 要分步恢复关闭的非关键存储空间（名为 `dbsp1`）和逻辑日志，请使用以下命令：
 

```
gbackuprestore -r -e -p dbsp1
gbackuprestore -r -l dbsp1
```
    - 要恢复所有非关键存储空间和逻辑日志，请使用 `gbackuprestore -r -e -O` 命令。

### 外部恢复命令的示例

下表包含外部恢复命令的示例。

外部恢复命令	操作	注释
--------	----	----



外部恢复命令	操作	注释
<code>gbackuprestore -r -e</code>	完整的外部恢复	在冷恢复中恢复所有内容。  在热恢复中，恢复所有关闭的非关键存储空间。
<code>gbackuprestore -r -e -p</code> <code>gbackuprestore -r -l</code>	物理外部恢复和单独的逻辑恢复	如果各个外部备份来自不同的时间，那么必须执行逻辑恢复。系统从最早的外部备份恢复逻辑日志。
<code>gbackuprestore -r -e dbspace_list</code>	选定的存储空间和逻辑日志的外部恢复	仅在外热恢复中使用该命令。
<code>gbackuprestore -r -e -p dbspace_list</code> <code>gbackuprestore -r -l</code>	选定存储空间外部恢复和单独的逻辑恢复	仅在外热恢复中使用该命令。
<code>gbackuprestore -r -e -t datetime</code>	外部时间点（冷）恢复	确保选择指定时间之前的备份集。
<code>gbackuprestore -r -e rename -p old_path -o old_offset -n new_path -o new_offset</code>	重命名块的外部（冷）恢复	仅在外冷恢复中使用该命令对块进行重命名。
<code>gbackuprestore -r -e -w</code> <code>gbackuprestore -r -e -p -w</code>	整个系统外部恢复	使用 <code>gbackuprestore -r -e -w -p</code> 时，在一个阻塞和取消阻塞会话中备份所有的存

外部恢复命令	操作	注释
		存储空间。那样，所有的存储空间将具有相同的检查点。

### 3.5.5 使用外部备份与恢复来初始化 HDR

您可以使用外部备份来初始化“高可用性数据复制”(HDR)。

要使用外部备份与恢复来初始化 HDR:

1. 使用 `gadmin -c block` 命令来阻塞源数据库服务器。
2. 在外部备份源数据库服务器上的所有块。
3. 备份完成时，使用 `gadmin -c unblock` 命令来取消阻塞源数据库服务器。
4. 使用以下命令来使源数据库服务器成为主服务器：`gadmin -d primary secondary_servername`
5. 在目标数据库服务器上，通过复制或文件备份程序从外部备份恢复数据。
6. 在目标数据库服务器上，使用 `gbackuprestore -r -e -p` 命令恢复所有块的外部备份。  
在 HDR 上，辅助服务器只能恢复 0 级归档。
7. 使用以下命令来使目标数据库服务器成为辅助服务器：`gadmin -d secondary primary_servername`
8. 如果从步骤 1 以来写入到主数据库服务器中的逻辑日志记录仍然驻留在主数据库服务器磁盘上，那么辅助数据库服务器将读取这些记录以执行逻辑恢复。否则，使用 `gbackuprestore -r -l` 命令来执行逻辑恢复。

数据库服务器可操作消息将显示在主服务器和辅助服务器上的消息日志中。

## 3.6 定制和维护 gbackuprestore

这些主题讨论以下内容:

- 使用 `gbackuprestore` 脚本定制 `gbackuprestore` 和存储管理器命令
- 手动启动 `gbackuprestore-worker` 进程
- 使备份历史记录到期和同步

### 3.6.1 定制 gbackuprestore 和存储管理器命令

您可以编辑与 `gbackuprestore` 一起安装的脚本，以定制备份与恢复命令以及存储管理器命令。

在 UNIX™ 操作系统上，`gbackuprestore shell` 脚本位于 `$GBS_HOME/bin` 目录中。

编辑脚本并备份原始文件的副本，以便在需要时还原。

**重要：** 请小心编辑脚本并测试更改。请勿更改脚本底部的清除代码。这样做可能导致意外行为，例如，在备份验证期间遗留临时文件。

该脚本包含用于 `Storage Manager` 的命令，并备份 `Storage Manager` 目录。如果要使用不同的存储管理器，请删除特定于 `Storage Manager` 的行，并（可选）为所使用的存储管理器添加命令。

脚本包含以下部分：

- 此处添加启动处理

如果需要，使用这一部分初始化存储管理器，并且设置环境变量。

- 此处结束启动处理

这一部分启动 `gbackuprestore_d` 驱动程序并检查返回码。将这一部分用于 `gbackuprestore_d` 和存储管理器命令。

- 此处添加清除处理

这部分中的代码在备份操作完成后将 `Storage Manager` 目录备份到 `ISMDData` 卷池中。如果使用第三方存储管理器，请删除特定于 `Storage Manager` 的信息。

如果对卷池使用不同于 `ISMDData` 的名称，请将其更改为在 `ISM_DATA_POOL` 配置参数中指定的名称。

这部分会除去 `archecker` 临时文件。

- 此处结束清除处理

使用这一部分返回 `gbackuprestore_d` 错误代码。

### 在重新安装期间更新 `gbackuprestore` 脚本

重新安装数据库服务器后，可能需要更新与 `gbackuprestore` 一起安装的脚本。安装过程备份了现有定制脚本，以便您可以复用其内容。

安装程序在 UNIX™ 上安装缺省 `gbackuprestore shell` 脚本。如果缺省脚本与本地脚本不同，那么安装程序将备份本地脚本，并发出消息通知您本地脚本已被重命名。重命名的文件的命名约定为 `gbackuprestore.date`，其中 `date` 是重命名文件时的日期。例如，文件 `gbackuprestore.2012.05.15` 于 2012 年 5 月 15 日重命名。

您可以通过从重命名的脚本向缺省脚本添加信息来更新缺省脚本。

## 打印备份引导文件

当备份成功时，使用下面示例的方法向 `gbackuprestore` 脚本添加命令来打印紧急引导文件。每次发出 `gbackuprestore -b` 命令时，都将打印紧急引导文件。

以下示例适用于 UNIX™:

```
gbackuprestore_d "$@" # receives gbackuprestore arguments from command
line return_code = $?
# check return code

# if backup (gbackuprestore -b) is successful, prints emergency boot file
if [$return_code -eq 0 -a "$1" = "-b"]; then
    servernum= `awk '/^SERVERNUM/ {print $2}' $GBS_HOME/etc/$ONCONFIG `
    lpr \${GBS_HOME}/etc/ixbar.$servernum
fi
exit $return_code
```

## 将备份的逻辑日志迁移到磁带

可以将存储管理器设置为将逻辑日志备份到磁盘上，接着编写脚本自动将这些逻辑日志从磁盘迁移到磁带，以便进行工作环境之外的保存。编辑 `gbackuprestore` 脚本使其在 `gbackuprestore_d` 进程完成后调用该迁移脚本。以下示例显示了调用迁移脚本的脚本:

以下示例适用于 UNIX™:

```
gbackuprestore_d "$@" # starts the backup or restore
EXIT_CODE=$? # any errors?

PHYS_ONLY=false #if it's physical-only, do nothing
for OPTION in $*; do
    if [ $OPTION = -p ]; then
        PHYS_ONLY = true
    fi
done
if ! PHYS_ONLY; then # if logs were backed up, call another
    migrate_logs # program to move them to tape
fi
```

### 3.6.2 使备份目录到期和同步

gbackuprestore 在 sysutils 数据库中为备份与恢复操作保留一份历史记录,并在紧急引导文件中保留备份历史记录的另一副本。当只有部分数据丢失时,gbackuprestore 将在热恢复中使用 sysutils 数据库。由于 sysutils 数据库无法访问,因此 gbackuprestore 将在冷恢复中使用紧急引导文件。您可以使用 onsmsync 来重新生成紧急引导文件和使旧备份到期。

根据您提供的命令选项,onsmsync 实用程序可从 sysutils 数据库以及紧急引导文件中除去以下各项:

- 存储管理器已终止的备份
- 基于备份存在时间的旧备份
- 基于已备份次数的旧备份

在数据库服务器处于联机状态或停顿方式下使用 onsmsync,以同步 sysutils 数据库与紧急引导文件。

要同步 sysutils 数据库:

1. 使数据库服务器处于联机状态或停顿方式。
2. 不带任何选项运行 onsmsync 实用程序。

onsmsync 实用程序按照如下方式同步 sysutils 数据库、存储管理器和紧急引导文件:

- 将备份历史记录添加到 sysutils 中,紧急引导文件中有该备份历史记录,但已从 sysutils 数据库中丢失。
- 从 sysutils 数据库中除去恢复的记录、整个系统恢复的记录、伪备份记录、成功和失败的备份的记录。
- 使不再需要的旧逻辑日志到期。
- 从 sysutils 数据库再次生成紧急引导文件。

#### 选择到期策略

可从以下三个到期策略中选择:

##### 保留时间日期 (-t)

在特定日期和时间之前删除所有备份。

##### 保留时间的间隔 (-i)

删除早于指定时段的所有备份。

##### 保留时间生成 (-g)

为每个备份保留一定数目的版本。

`gbackuprestore` 始终为每个存储空间保留最新的 0 级备份。它让所有早于指定时间的 0 级备份到期，除非从所保留的最早 1 级备份恢复时需要这些 0 级备份。

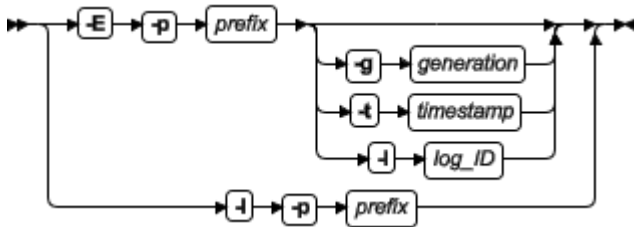
`gbackuprestore` 让所有早于指定时间的 1 级备份到期，除非从所保留的最早 2 级备份恢复时需要这些 1 级备份。

`gbackuprestore` 将保留从指定的保留时间之前开始以及指定的保留时间之后结束的整个系统备份。

**onsmsync 实用程序**

使用 `onsmsync` 实用程序可通过存储管理器目录来同步 `sysutils` 数据库和紧急引导文件。

如果您的存储管理器是 GBase 8s Primary Storage Manager，那么还可以使用 `onsmsync` 实用程序的导出和导入选项来将 GBase 8s Primary Storage Manager 备份对象导出至外部设备，并将外部设备中的对象导入到 GBase 8s Primary Storage Manager 管理的设备。不能对其他存储管理器使用导出和导入选项。



下表列出了所有 `onsmsync` 命令元素，但用于导入和导出备份生成的元素除外。用于导入和导出的命令元素在表 2 中列出。

表 1. `onsmsync` 命令的元素

元素	用途	关键注意事项
-b	将紧急引导文件 ( <code>ixbar.servernum</code> ) 和 <code>sysutils</code> 数据库通过对方重新生成。	<p>如果 <code>ixbar</code> 文件为空或不存在，那么 <code>onsmsync -b</code> 会重新创建 <code>ixbar</code> 文件并从 <code>sysutils</code> 表填充该文件。</p> <p>如果 <code>ixbar</code> 文件不为空且包含对象数据，那么 <code>onsmsync -b</code> 会更新 <code>sysutils</code> 数据库和 <code>ixbar</code> 文件，以使其同步。</p> <p>如果 <code>ixbar</code> 文件具有条目，并且 <code>sysutils</code> 数据库已重新构建，但由于不包含数据而为空，那么 <code>onsmsync -b</code> 会从 <code>ixbar</code> 文件重新创建 <code>sysutils</code> 数据。</p>

元素	用途	关键注意事项
		<p>请不要将 <code>-b</code> 元素与其他 <code>onsmsync</code> 选项一起使用。</p> <p><code>-b</code> 元素不与存储管理器同步。</p>
<code>dbspace</code>	指定要使其到期的一个或多个存储空间	如果输入多个存储空间名称，请使用空格来分隔这些名称。
<code>-f filename</code>	指定包含要到期存储空间列表的文件的名称	使用该选项可避免输入一长串存储空间。文件名可以是任何有效的 UNIX™ 文件名。
<code>-g generation</code>	指定每个 0 级备份要保留的版本数	保留最近生成的备份并使此前的所有备份都到期。
<code>-i interval</code>	指定保留备份的时间间隔。	<p>实用程序：</p> <ul style="list-style-type: none"> <li>保留在此时间间隔之后创建的备份。</li> <li>使在此时间间隔之前创建的备份到期，并在到期的对象也被除去时除去这些备份。</li> </ul> <p>如果从该间隔后的其他备份恢复时需要在此间隔之前的备份，那么该间隔之前的备份不会到期。</p> <p>使用 ANSI 或 GLS 格式来表示 <code>interval</code>： YYYY-MM 或 DD HH:MM:SS</p>
<code>-0</code>	覆盖内部错误检查，并强制实施到期策略	如果与 <code>-t</code> 、 <code>-g</code> 或 <code>-i</code> 选项一起使用，即使从发生在此到期日期之后的备份进行恢复时需要所有级别的备份中的一部分，也要使所有级别的备份到期。 <code>-0</code> 选项不影响逻辑日志到期。请参阅使所有备份到期。
<code>-s</code>	跳过对到期的备份的同步	如果提供了 <code>-s</code> 选项，那么对象到期取决于其他参数。
<code>-t timestamp</code>	在特定日期和时间之	保留在指定时间戳记后完成的备份。如果从该时

元素	用途	关键注意事项
	前让所有备份到期	<p>间戳记后发生的其他备份恢复时需要在该时间戳记之前发生的备份，那么后者不会到期。</p> <p>使用 ANSI 或 GLS_DATETIME 格式来表示时间戳记。</p>
-cf value	<p>指定是否备份关键文件</p> <p>在与 -g、-i 或 -t 一起使用时，从GBase 8s Primary Storage Manager删除关键文件备份</p>	<p>有效值为：</p> <ul style="list-style-type: none"> <li>• Yes。备份关键文件。该值为 0 级、1 级或 2 级备份的缺省值。</li> <li>• No。不备份关键文件。该值为备份逻辑日志文件的缺省值。</li> <li>• Only。仅备份关键文件。</li> </ul>

表 2. onsmsync 导出和导入命令的元素

元素	用途	关键注意事项
-E	将单一备份生成导出至 GBase 8s Primary Storage Manager 外部池	<p>仅当设置 GBase 8s Primary Storage Manager 外部池时使用此选项。</p> <p>如果导出备份生成，那么必须指定前缀，以标识导出的备份。onmsync 实用程序会在外部池中创建一个包含该前缀的子目录，并将导出的对象放入该目录。</p>
-g generation	指定要导出的备份生成。	缺省值是当前备份。
-I	从外部 GBase 8s Primary Storage Manager 池导入单一备份生成。	如果从外部池导入备份生成，那么必须指定导出的备份的前缀。该前缀用于标识要导入的备份生成。
-l log_ID	导出包含逻辑日志标识的备份生成。	
-p prefix	指定要分配给将导出的备份生成的前缀，或用于标识要导入的备份的前缀。	当 onmsync 实用程序导出备份生成时，它会使用该前缀作为放置该备份的子目录的名称。



元素	用途	关键注意事项
-t timestamp	指定包含特定日期和时间的备份生成(仅用于导出)。	使用 ANSI 或 GLS_DATETIME 格式。 当您推出应用程序的新版本时, 可能希望包含日期和时间。

### 用途

如果不指定任何选项, 那么 `onsmsync` 命令会使用存储管理器目录来同步 `sysutils` 数据库和紧急引导文件。`onsmsync` 实用程序会将 `sysutils` 数据库和紧急引导文件中的备份与存储管理器目录中的备份进行比较, 然后从 `sysutils` 数据库和紧急引导文件中除去存储管理器目录内不存在的所有备份。

**提示:** 要控制 `sysutils` 数据库是否保留已到期的备份与恢复的历史记录, 请使用 `BAR_HISTORY` 配置参数。有关信息, 请参阅 `BAR_HISTORY` 配置参数。

除了存储空间名或文件名必须放在最后之外, 命令的顺序无关紧要。

命令的顺序无关紧要, 但以下情况除外:

- 存储空间名称或文件名必须放在最后。
- 导出或导入时, `-E` 或 `-I` 选项必须放在最前。例如, 指定 `onsmsync -E -g 2`, 而不是 `onsmsync -g 2 -E`。

在不同计算机上导入和导出备份生成的先决条件如下:

- 在源计算机和目标计算机上必须具有相同版本的 GBase 8s, 并且这些计算机必须使用相同的操作系统。
- 在源计算机和目标计算机上必须设置 GBase 8s Primary Storage Manager 并创建一个外部池。

当您使用 `-E` 或 `-I` 选项导出或导入备份生成时, 必须指定用于标识放置备份生成的子目录的前缀。

如果使用 `-E` 或 `-I` 选项导出或导入备份生成, 那么不能使用与导出或导入操作无关的任何 `onsmsync` 命令选项。例如, 不能同时导出备份生成并重新生成紧急引导文件。

`onsmsync -I` 命令重命名当前 `ixbar` 文件, 并创建仅包含恢复导入的备份所必需信息的新文件。您可以将 `-cf` 选项与 `-g`、`-i` 或 `-t` 选项一起使用, 以从存储管理器删除关键文件备份。

如果应用 `-g` 选项, 并且 `onsmsync` 实用程序的对象列表仅包含逻辑日志而不包含任何空间备份, 那么这些日志备份不会到期。在此情况下, 请使用 `-t` 或 `-i` 选项来使逻辑日志备份到期。

### 示例

以下示例使 2012 年 11 月 30 日前开始的备份到期:

```
onsmsync -t "2012-11-30 00:00:00"
```

以下命令将最后一个备份生成导出到外部池中名为 `gen` 的目录：

```
onsmsync -E -p gen -g 1
```

以下命令将按最新程度排名第 4 位的备份生成导出到外部池中名为 `gen` 的目录：

```
onsmsync -E -p gen -g 4
```

以下命令将当前备份生成导出到外部池中名为 `gen` 的目录：

```
onsmsync -E -p gen
```

以下命令将生成 2 中的所有备份对象导出到外部池中名为 `gen` 的目录：

```
onsmsync -E -p gen -g 2
```

以下命令将时间戳记为 2012-12-31 12:00:00 的所有备份对象导出到外部池中名为 `gen` 的目录：

```
onsmsync -E -p gen -t "2012-12-31 12:00:00"
```

以下命令导入以前缀 `gen` 标识的子目录中的所有对象：

```
onsmsync -l -p gen
```

以下命令导入使用前缀 `gen` 和时间戳记 2012-12-31 12:00:00 导出的所有备份对象。由于前缀用于标识备份生成，因此不用指定时间戳记。

```
onsmsync -l -p gen
```

以下命令将除了最后两次生成的关键文件备份之外的所有内容删除：

```
onsmsync -g 2 -cf yes
```

#### 使 Storage Manager 上的旧备份到期

Storage Manager 和某些第三方存储管理器不允许 `onsmsync` 实用程序从存储管理器删除备份。

必须手动使存储管理器中的旧备份到期或将其删除。然后，不带任何参数运行 `onsmsync`。

要使 Storage Manager 上的旧备份到期：

1. 要手动使 Storage Manager 中的旧备份到期，请使用 `ism_config -retention #days` 命令。  
有关更多信息，请参阅《GBase 8s Storage Manager 管理员指南》。
2. 不带任何选项运行 `onsmsync`。

#### 重新生成紧急引导文件

要只重新生成紧急引导文件，请使用 `onsmsync -b` 命令。

`onsmsync -b` 命令将旧的紧急引导文件另存为 `ixbar.server_number.system_time`，并将其重新生成成为 `ixbar.server_number`。

#### 重新生成 sysutils 数据库

如果丢失 `sysutils` 数据库，请使用 UNIX™ 上的 `$GBS_HOME/etc` 中的 `bldutil` 实用程序，以重新创建带有空表的 `sysutils` 数据库。

接着使用 `onmsync` 实用程序重新创建备份并在 `sysutils` 中恢复数据。

**限制：** 如果 `sysutils` 数据库和紧急引导文件都已丢失，那么不能使用 `onmsync` 重新生成它们。一定要将紧急引导文件与其他操作系统文件一起备份。

### 删除坏备份

`onmsync` 实用程序无法分辨哪些备份未能通过验证。如果最近的备份未能通过验证，但前面的某一备份却是成功的，那么必须手动从存储管理器中删除失败的备份记录，然后不带任何选项运行 `onmsync` 以同步 `gbackupstore`。有关更多信息，请参阅 `gbackupstore -v` 语法：验证备份。

### 基于保留时间日期使备份到期

以下示例使 2006 年 11 月 24 日前开始的备份以及所有伪备份、失败的备份与恢复都到期：

```
onmsync -t "2006-11-24 00:00:00"
```

### 使生成的备份到期

以下示例保留最新的三组 0 级备份以及相关的增量备份，同时使所有之前的备份和所有恢复、伪备份和失败的备份到期：`onmsync -g 3`

### 基于保留时间的间隔使备份到期

以下示例让三天以前的所有备份以及所有伪备份、失败的备份与恢复到期：

```
onmsync -i "3 00:00:00"
```

以下示例使 18 个月前的所有备份到期（写作 1 年 + 6 个月）：

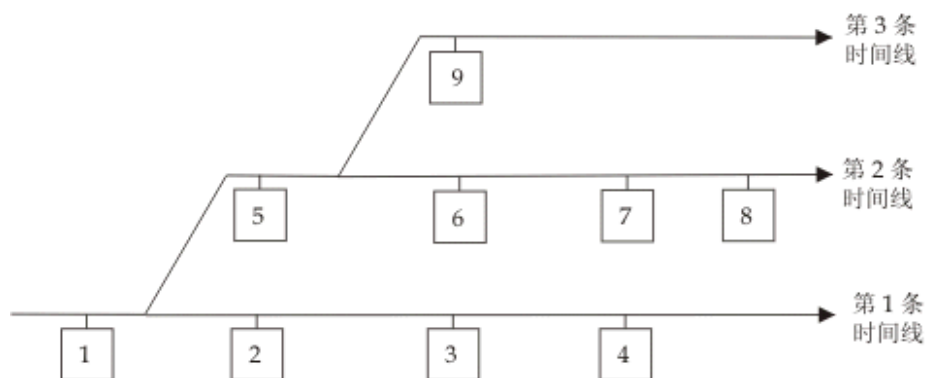
```
onmsync -i "1-6"
```

### 使用多个时间点恢复使备份到期

如果您执行多个时间点恢复，那么将存在多个备份的时间线。

下图显示了三条时间线及其备份。

**图：备份的多条时间线**



在此示例中，第二条时间线从对备份 1 的时间点恢复开始。第二条时间线包含备份 1、5、6、7 和 8。第三条时间线（粗体显示）包含备份 1、5 和 9。第三条时间线被视为当前时间线，因为它包含最新的备份。

在运行 `onmsync` 实用程序以使旧备份到期时，`onmsync` 将从当前时间线中除去旧备份，并确保当前时间线可从保留的备份对象中恢复。不在当前时间线中的所有其他备份也将过期，但是 `onmsync` 将不确保其他时间线可从保留的对象中恢复。

`onmsync` 实用程序以以下顺序应用到期策略，以确保根据指定的到期策略使当前时间线中的对象到期，并且确保当前时间线可恢复：

- 在所有备份对象的集合上应用到期策略。
- 不使属于当前时间线的备份对象到期。
- 在当前时间线上应用到期策略，以确保当前时间线可恢复。

同时，将到期策略应用于其他时间线中的备份。

例如，如果在上图的示例上执行 `onmsync -g 2` 命令，那么当前时间线中的备份 1 将到期，第一和第二个时间线中的备份 2、3、4、6 和 7 也将到期。当前时间线中的备份 1、5 和 9 将保留。其他时间线中的备份 8 将保留。

#### 使所有备份到期

除非使用 `-O` 选项，否则 `onmsync` 实用程序将保留最新的 0 级备份。如果使用 `-O` 和 `-t` 选项，即使恢复还需要所指定时间前的所有备份，也会将那些备份全部除去。如果使用 `-O` 和 `-i` 选项，即使恢复还需要所指定时间间隔前的所有备份，也会将那些备份全部除去。

例如：要使所有备份到期，请指定以下选项：

```
onmsync -O -g 0
```

**重要：** 如果将 `-O` 选项与 `-t`、`-i` 或 `-g` 选项一起使用，您可能会意外删除某些关键备份，从而使恢复无法进行。

### 3.6.3 监视 `gbackuprestore` 和存储管理器的性能

您可以监视 `gbackuprestore` 和存储管理器的性能。可以指定性能监视的级别并将统计信息写入到 `gbackuprestore` 活动日志。`BAR_PERFORMANCE` 配置参数指定是否收集统计值。收集的统计值如下：

- XBSA 调用所耗用的总时间。
- 归档 API 调用所耗用的总时间。
- `gbackuprestore` 在与 XBSA（存储管理器调用）相互传输数据时所耗用的时间。
- `gbackuprestore` 在 `gbackuprestore` 与 GBase 8s 之间传输数据时所耗用的时间。
- 与 XBSA API 之间的数据传送量。
- 与归档 API 之间的数据传送量。

### 设置 `gbackuprestore` 性能统计信息级别

要指定写入到 `gbackuprestore` 活动日志的性能统计信息的级别，请在 `onconfig` 文件中设置 `BAR_PERFORMANCE` 配置参数。

例如，`BAR_PERFORMANCE 1` 设置显示了在 GBase 8s 实例与存储管理器之间传输数据时所耗用的时间。

有关此参数选项的信息，请参阅 `BAR_PERFORMANCE` 配置参数。

### 查看 `gbackuprestore` 备份与恢复性能统计信息

要查看 `gbackuprestore` 性能结果，请打开 `gbackuprestore` 活动日志。

要确定活动日志的位置，请参阅 `BAR_ACT_LOG` 配置参数。

当 `BAR_PERFORMANCE` 设置为 1 或 3 时，活动报告显示传输率报告：

图: `gbackuprestore` 活动日志中的样本传输速率性能。

```

2009-06-03 15:38:02 8597 8595 Begin restore logical log 310 (Storage Manager
                           copy ID: 28206 0).
2009-06-03 15:38:03 8597 8595 Completed restore logical log 310.
2009-06-03 15:38:08 8597 8595 Completed logical restore.
2009-06-03 15:38:19 8597 8595 PERFORMANCE INFORMATION

                                                                    TRASFER RATES
-----
|      OBJECT      |          XBSA API          |
SERVER API          |                               |
|      NAME        | xfer-kbytes  xfer-time RATIO(kb/s)  API-TIME |
xfer-kbytes  xfer-time RATIO(kb/s)  API-TIME |

```

```

-----
-----
| 309          |          62      0.479      129      1.078 |
62      0.019      3310      0.310 |
| 310          |          62      0.407      152      1.098 |
62      0.025      2522      0.025 |
| rootdbs      |          5828      0.618      9436      1.864 |
5828      8.922      653      8.931 |
| datadbs01    |          62      0.488      127      1.768 |
62      0.004      17174      0.004 |
| datadbs02    |          62      0.306      203      1.568 |
62      0.008      8106      0.008 |
| datadbs03    |          62      0.304      204      1.574 |
62      0.007      8843      0.007 |
| datadbs04    |          62      0.306      202      1.563 |
62      0.007      8664      0.007 |
| datadbs05    |          62      0.315      197      1.585 |
62      0.007      8513      0.007 |
| datadbs06    |          62      0.310      200      1.583 |
62      0.002      25348      0.002 |
-----
...
-----
| PID =      8597 |          14722      26.758      550      107.476 |
14756      10.678      1382      15.829 |
-----
-----
2009-06-03 15:38:19 8597 8595 PERFORMANCE INFORMATION

                                PERFORMANCE

CLOCKS
-----
-----
|                                ITEM DESCRIPTION                                |
TIME SPENT                        |
-----
-----
| Time to Analyze ixbar file                                |
0.000 |
-----
-----

```

当 BAR\_PERFORMANCE 设置为 2 或 3 时，活动报告使用微秒时间戳记（如以下示例所示）：

图: gbackuprestore 活动日志中的样本处理速率（以微秒计）。

```

2009-06-03 16:34:04.15272 15270 /usr/gbasedbt/bin/gbackuprestore_d complete,
                                returning 0 (0x00)
2009-06-03 16:45:11.608424 17085 17083 /usr/gbasedbt/bin/gbackuprestore_d -r
-w
2009-06-03 16:46:07.926097 17085 17083 Successfully connected to Storage
Manager.
2009-06-03 16:46:08.590675 17085 17083 Begin salvage for log 311.
2009-06-03 16:48:07.817487 17085 17083 Completed salvage of logical log 311.
2009-06-03 16:48:08.790782 17085 17083 Begin salvage for log 312.
2009-06-03 16:48:10.129534 17085 17083 Completed salvage of logical log 312.
2009-06-03 17:06:00.836390 17085 17083 Successfully connected to Storage
Manager.
...
2009-06-03 17:07:26.357521 17085 17083 Completed cold level 0 restore
datadbs07.
2009-06-03 17:07:28.268562 17085 17083 Begin cold level 0 restore datadbs08
(Storage Manager copy ID: 28122 0).
2009-06-03 17:07:29.378405 17085 17083 Completed cold level 0 restore
datadbs08.

```

## 3.7 gbackuprestore 目录表

这些主题描述存储在 `sysutils` 数据库中的 `gbackuprestore` 表。

`gbackuprestore` 使用这些表跟踪备份并执行恢复。

可以对这些表查询备份与恢复数据，以评估性能或标识某个恢复的对象实例。

### 3.7.1 bar\_action 表

`bar_action` 表列出试图对某个对象执行的所有备份与恢复操作，除了在某些类型的冷恢复期间执行的操作之外。请使用该表中的信息跟踪备份与恢复历史记录。

表 1. `bar_action` 表列

列名	类型	解释
<code>act_aid</code>	SERIAL	操作标识。表中的唯一编号。可以与 <code>act_oid</code> 列一起使用以连接 <code>bar_instance</code> 表。

列名	类型	解释
act_oid	INTEGER	对象标识。标识备份或恢复操作所针对的备份对象。可以与 act_aid 一起使用以连接 bar_instance。 bar_action 表的 act_oid 列等同于 bar_object 表的 obj_oid 列。
act_type	SMALLINT	标识已尝试的操作：1 表示备份，2 表示恢复，3 表示外部或导入的恢复，4 表示伪备份，5 表示整个系统的备份，6 表示整个系统的恢复，7 表示到期的或已删除的对象，8 表示外部恢复。
act_status	INTEGER	标识操作的结果：如果成功则为 0，否则是特定于 gbackuprestore 的错误代码。有关更多信息，请参阅 gbackuprestore 消息和返回码。
act_start	DATETIME YEAR TO SECONDS	操作开始的日期和时间。
act_end	DATETIME YEAR TO SECONDS	操作完成的日期和时间。

### 3.7.2 bar\_instance 表

bar\_instance 表包含备份的每个对象的描述。

gbackuprestore 为每个成功的备份向 bar\_instance 表中写入一条记录。gbackuprestore 可能稍后使用该信息进行恢复操作。例如：如果指定 2 级备份，gbackuprestore 将使用该表以确保先前已成功执行 1 级备份。

表 1. bar\_instance 表列

列名	类型	解释
ins_aid	INTEGER	操作标识。标识创建该备份对象实例的成功操作。可以与 ins_oid 结合起来用于连接 bar_action 表。
ins_oid	INTEGER	对象标识。标识受影响的对象。可以用于连接 bar_object 表。可以与 ins_aid 结合



列名	类型	解释
		起来用于连接 <code>bar_action</code> 表。
<code>ins_prevtime</code>	INTEGER	时间戳记（实时时间）。此值指定先前对象的时间戳记。此值代表 1970 年 1 月 1 日 午夜（格林威治标准时间）以后的秒数。
<code>ins_time</code>	INTEGER	时间戳记（实时时钟时间）。数据库服务器创建下级备份时将使用该值。该值代表自 1970 年 1 月 1 日 午夜（格林威治标准时间）以来的秒数。  <code>ins_time</code> 的值是 0。
<code>rsam_time</code>	INTEGER	备份检查点时间戳记。不是时钟时间。数据库服务器创建下级备份时使用该值。
<code>ins_level</code>	SMALLINT	备份操作的级别：0 表示完整备份，1 表示自上次 0 级备份以来对该对象所做的所有更改的备份，2 表示自上次 1 级备份以来所有更改的备份。对于逻辑日志备份，该值始终是 0。
<code>ins_copyid_hi</code>	INTEGER	实例副本标识的高位。它是存储管理器分配的唯一值，与 <code>ins_copyid_lo</code> 结合在一起，用于将 <code>gbackuprestore</code> 对象标识与存储管理器对象标识连接在一起。
<code>ins_copyid_lo</code>	INTEGER	实例副本标识的低位。它是存储管理器分配的唯一值，与 <code>ins_copyid_hi</code> 结合在一起，用于将 <code>gbackuprestore</code> 对象标识与存储管理器对象标识连接在一起。
<code>ins_req_aid</code>	INTEGER	存储备份对象必需的操作标识。在恢复中使用，以确定与 1 级备份相伴的 0 级备份，以及与 2 级备份相伴的 1 级备份。对于 0 级备份， <code>ins_req_aid</code> 的值与表中 <code>ins_aid</code> 的值相同。 例如：如果备份是 1 级的， <code>ins_req_aid</code> 保留该对象相应的 0 级备份的操作标识。

列名	类型	解释
ins_first_log	INTEGER	在标准备份中，标识从该备份恢复必需的第一个逻辑日志。
ins_verify	INTEGER	如果备份已验证则该值为 1。如果备份未验证则该值为 0。
ins_verify_date	DATETIME YEAR TO SECOND	验证备份时插入当前日期。如果该备份尚未验证，那么用短划线表示每个日期和时间。

### 3.7.3 bar\_ixbar 表

bar\_ixbar 表用于存储所有时间线内所有未到期的成功备份的历史记录，它仅由 onmsync 实用程序维护和使用。

bar\_ixbar 表的模式与 bar\_syncdeltab 表的模式相同，但主键除外。

表 1. bar\_ixbar 表列

列名	类型	解释
ixb_sm_id	INTEGER	存储管理器实例标识。 从 \$GBS_DATA/conf/onconfig 中的 BAR_SM 创建。
ixb_copyid_hi	INTEGER	实例副本标识的高位。它是存储管理器分配的唯一值，与 ixb_copyid_lo 结合在一起，用于链接 gbackuprestore 对象标识与存储管理器对象标识。
ixb_copyid_lo	INTEGER	实例副本标识的低位。它是存储管理器分配的唯一值，与 ixb_copyid_hi 结合在一起，用于链接 gbackuprestore 对象标识与存储管理器对象标识。
ixb_aid	INTEGER	操作标识，标识创建该备份对象实例的成功操作。
ixb_old	INTEGER	对象标识。标识受影响的对象。
ixb_time	INTEGER	时间戳记（实时时钟时间）。数据库服务器创建下级备份时将使用该值。该值代表自 1970 年 1 月 1 日午夜（格林威治标准时间）以来的秒数。

列名	类型	解释
<code>ixb_prevtime</code>	INTEGER	时间戳记（实时时钟时间）。此值指定先前对象的时间戳记。此值代表自 1970 年 1 月 1 日 午夜（格林威治标准时间）以来的秒数。
<code>ixb_rsam_time</code>	INTEGER	备份检查点时间戳记。不是时钟时间。数据库服务器创建下级备份时使用该值。
<code>ixb_act_start</code>	日期和时间，从年到秒	操作开始的日期和时间。
<code>ixb_act_end</code>	日期和时间，从年到秒	操作完成的日期和时间。
<code>ixb_level</code>	SMALLINT	备份操作的级别：0 表示完整备份，1 表示自上次 0 级备份以来对该对象所做的所有更改的备份，2 表示自上次 1 级备份以来所有更改的备份。对于逻辑日志备份，该值始终是 0。
<code>ixb_req_aid</code>	INTEGER	存储备份对象必需的操作标识。在恢复中使用，以确定与 1 级备份相伴的 0 级备份，以及与 2 级备份相伴的 1 级备份。对于 0 级备份， <code>ixb_req_aid</code> 的值与此表中 <code>ixb_aid</code> 的值相同。例如：如果备份是 1 级， <code>ixb_req_aid</code> 保留该对象相应的 0 级备份的操作标识。
<code>ixb_first_log</code>	INTEGER	在标准备份中，标识需要从该备份恢复的第一个逻辑日志。
<code>ixb_chpt_log</code>	INTEGER	包含 <code>rsam_time</code> 检查点的日志的标识。在备份期间用于验证是否已备份恢复所需的日志。
<code>ixb_last_log</code>	INTEGER	在逻辑恢复期间需要的最后一个日志的日志标识，以便存储空间可以将它恢复到备份时的状态。
<code>ixb_lbuflags</code>	INTEGER	描述日志备份的标志
<code>ixb_verify</code>	INTEGER	如果备份已验证则该值为 1。如果备份未验证则该值为 0。

列名	类型	解释
<b>ixb_verify_date</b>	日期和时间，从年到秒	验证备份时插入当前日期。如果该备份尚未验证，那么用短划线表示每个日期和时间。
<b>ixb_sm_name</b>	VARCHAR(128)	存储管理器实例名。从 onconfig 文件中的 BAR_SM_NAME 参数创建。
<b>ixb_srv_name</b>	VARCHAR(128)	数据库服务器名。用于确保将对象恢复到正确的数据库服务器上。当节点上有多个数据库服务器时，用于确保对象在所属的数据库服务器实例中恢复。数据库服务器名最多可有 128 个字符。
<b>ixb_obj_name</b>	VARCHAR(128)	对象的用户名。该名称最多可以有 128 个字符。
<b>ixb_obj_type</b>	CHAR(2)	备份对象类型： <b>CD</b> 关键数据库空间 <b>L</b> 逻辑日志 <b>ND</b> 非关键数据库空间或智能大对象空间 <b>R</b> 根数据库空间 <b>B</b> Blob 空间

### 3.7.4 bar\_object 表

bar\_object 表包含每个备份对象的描述。该表是来自每个数据库服务器的所有存储空间和逻辑日志（数据库服务器至少已对它们进行了一次备份尝试）的列表。

表 1. bar\_object 表列

列名	类型	解释
<b>obj_srv_name</b>	VARCHAR(128, 0)	数据库服务器名。用于确保将对象恢复到正确的数据库服务器上。当节点上有多个数据库服务器时，用于确保对象在所属的数据库服务器实例中恢复。

列名	类型	解释
		数据库服务器名最多可有 128 个字符。
obj_oid	SERIAL	对象标识。表中的唯一编号。可用于连接 <code>bar_action</code> 和 <code>bar_instance</code> 表。
obj_name	VARCHAR(128,0)	对象的用户名。  该名称最多可以有 128 个字符。
obj_type	CHAR(2)	备份对象类型：  CD 关键数据库空间  L 逻辑日志  ND 非关键数据库空间或智能大对象空间  R 根数据库空间  B Blob 空间

### 3.7.5 bar\_server 表

bar\_server 表在安装中列出数据库服务器。该表用于确保备份对象在恢复期间返回到它们的位置。

表 1. bar\_server 表列

列名	类型	解释
srv_name	VARCHAR(128,0)	sqlhosts 文件中的 dbservername 条目指定的 dbservername 值。  数据库服务器名称最多可有 128 个字符。
srv_node	CHAR(256)	数据库服务器驻留的计算机的主机名。

列名	类型	解释
		主机名最多可有 256 个字符。
srv_synctime	INTEGER	onsmsync 运行的时间。

### 3.7.6 bar\_syncdeltab 表

bar\_syncdeltab 表只可由 onsmsync 实用程序维护和使用。该表为空，但当 onsmsync 正在运行时除外。

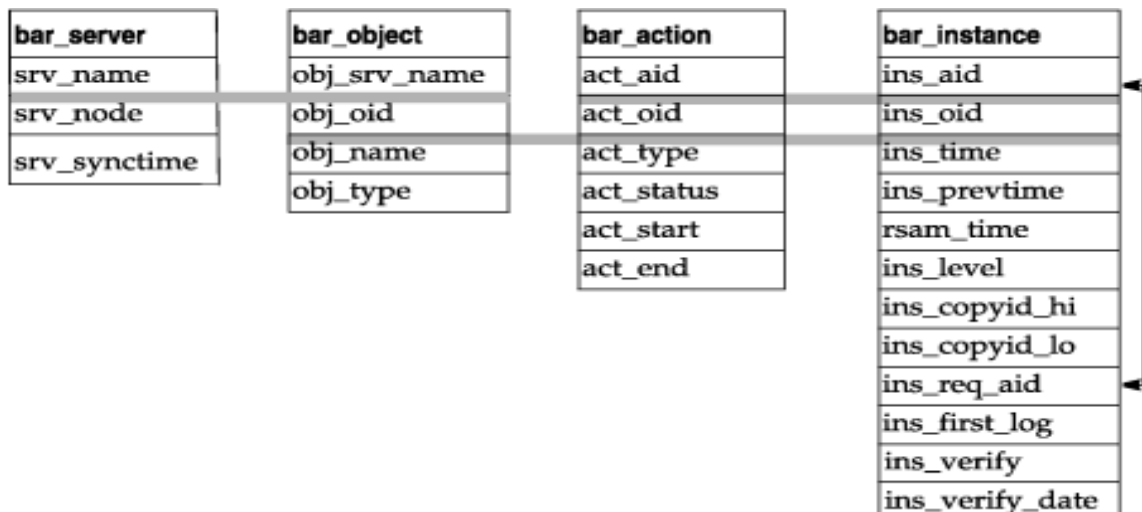
bar\_syncdeltab 表的模式与 bar\_ixbar 表的模式相同，但主键除外。

### 3.7.7 gbackuprestore 目录映射

本主题包含 gbackuprestore 表之间的映射示例。

下图映射 GBase 8s 上的 gbackuprestore 表。在该图中，灰线显示表之间的关系。箭头显示 ins\_req\_aid 的值必须是有效的 ins\_aid 值。

图：GBase 8s 上的 gbackuprestore 目录映射



## 3.8 gbackuprestore 消息和返回码

gbackuprestore 将参考、进度、警告和错误消息打印到 gbackuprestore 活动日志文件中。

gbackuprestore 返回码指示命令的状态。

有关每个 gbackuprestore 参考、进度、警告和错误消息的完整描述，请搜索 finderr 或 Error Messages 实用程序，或者搜索 GBase 8s Error Messages。

### 3.8.1 消息格式

gbackuprestore 活动日志文件中的消息具有以下格式：

```
timestamp process_id parent_process_id message
```

下表描述了消息中的每个字段。gbackuprestore 活动日志中没有出现错误消息号。

表 1. gbackuprestore 消息格式

消息字段	描述
timestamp	gbackuprestore 写入消息的日期和时间。
process_id	操作系统用来标识此 gbackuprestore 实例的编号。
parent_process_id	操作系统用来标识执行此 gbackuprestore 实例的进程的编号。
message	gbackuprestore 消息文本。

以下示例举例说明了 gbackuprestore 活动日志中的一个典型条目：

```
1999-08-18 10:09:59 773 772 已完成逻辑恢复。
```

**重要：** 如果收到 XBSA 错误消息，请查阅存储管理器日志获取关于更多详细信息。

### 3.8.2 消息编号

gbackuprestore 消息编号的范围从 -43000 到 -43421。

下表列出 gbackuprestore 消息组。由于活动日志中不显示消息编号，因此要查找有关 gbackuprestore 消息的信息，最佳方法是在 \$GBS\_HOME/msg 目录下您语言环境的子目录中的错误消息文件中搜索消息文本。

gbackuprestore 消息类型	消息编号
gbackuprestore 用法	-43000 到 -43007 和 -43357
选项检查	-43010 到 -43034

权限检查	-43035 到 -43039
紧急引导文件接口	-43040 到 -43059
onconfig 文件接口	-43060 到 -43074
操作系统接口	-43075 到 -43099
数据库服务器接口	-43100 到 -43229
备份与恢复状态	-43230 到 -43239
gbackuprestore-worker 进程	-43240 到 -43254
XBSA 接口	-43255 到 -43301
onsmsync	-43302 到 -43319
archecker	-43320 到 -43334
gdblogmode	-43400 到 -43424

### 3.8.3 gbackuprestore 返回码

您可以通过查看适用于特定返回码的活动日志消息来对问题进行故障诊断。

下表显示所有 GBase 8s 数据库服务器的 gbackuprestore 返回码。这些返回码都与 gbackuprestore 活动日志中的消息一起出现。有关错误的详细信息，请在致电技术支持前查看活动日志。

表 1. 公共 gbackuprestore 返回码

十进制值	gbackuprestore 返回码描述
2 到 34	这些返回码由 XBSA 产生。要了解更多信息，请查阅您的存储管理器文档和日志文件。
100	gbackuprestore 在 <code>sysutils</code> 、紧急引导文件或处理时需要的存储管理器目录中找不到某些内容。  请检查 gbackuprestore 活动日志中表示未找到内容的消息并尝试解决该问题。如果问题再次发生，请联系技术支持。
104	Adstar Distributed Storage Manager (ADSM) 处于生成密码方式中。



十进制值	<code>gbackuprestore</code> 返回码描述
	<code>gbackuprestore</code> 不支持 ADSM 在生成密码的方式下运行。有关更改 ADSM 安全配置的信息，请查看您的 ADSM 手册。
115	正在冷恢复的数据库空间集中缺少一个关键数据库空间。
116	<code>onsmsync</code> 实用程序已在运行。
117	<code>sysutils</code> 数据库和紧急引导文件中包含的信息不一致。
118	尝试向存储管理器提交备份对象时出错。
120	自该对象上次备份以来传送缓冲区大小已经改变。该对象无法恢复。将传输缓冲区大小设置为原始值并重试恢复。
121	<code>gbackuprestore</code> 无法确定数据库空间的列表。
122	检测到死锁。  <code>gbackuprestore</code> 命令与另一个进程发生争用。重试 <code>gbackuprestore</code> 命令。
123	根数据库空间不在冷恢复中。  在不恢复根数据库空间的情况下无法执行冷恢复。要解决问题，请尝试以下过程之一： <ul style="list-style-type: none"> <li>• 让数据库服务器处于停顿或联机方式，这样只需要恢复存储空间。</li> <li>• 如果数据库服务器处于脱机状态，请发出 <code>gbackuprestore -r</code> 命令来恢复所有的存储空间。</li> <li>• 确保根数据库空间和其他关键数据库空间在命令行或 <code>-f filename</code> 中列出。</li> </ul>
124	缓冲区在备份期间有不完整的页。  要获取帮助，请联系技术支持。

十进制值	gbackuprestore 返回码描述
126	<p>处理紧急引导文件出错。</p> <p>检查 gbackuprestore 活动日志以获取对问题的描述，并查看紧急引导文件以获取毁坏的情况，例如非 ASCII 字符或列数不同的行。如果问题来源不明显，请联系技术支持。</p>
127	<p>无法写入紧急引导文件中。</p> <p>通常，操作系统错误消息将与该问题一起出现。检查以下文件和目录的许可权：</p> <ul style="list-style-type: none"> <li>• UNIX™ 上的 \$GBS_HOME/etc</li> <li>• 紧急引导文件</li> </ul>
128	<p>对象描述中的数据丢失。</p> <p>要获取帮助，请联系技术支持。</p>
129	<p>gbackuprestore 接收到要进行恢复的对象不是预期的对象。（备份对象不匹配。）请求的备份对象可能已被删除或对于存储管理器已到期。</p> <p>运行 onsmsync 让 sysutils 数据库、紧急引导文件和存储管理器目录同步。要获取帮助，请联系技术支持。</p>
130	<p>数据服务器未响应。</p> <p>在备份或恢复期间数据库服务器可能已失败。运行 gstat - 命令检查数据库服务器状态，接着：</p> <ul style="list-style-type: none"> <li>• 如果操作是冷恢复，重新启动该操作。</li> <li>• 如果操作是备份或热备份，请重新启动数据库服务器并重试备份或热恢复。</li> </ul>
131	<p>在 gbackuprestore 和数据库服务器的接口中发生故障。</p>

十进制值	<code>gbackuprestore</code> 返回码描述
	要获取帮助，请联系技术支持。
132	<p>函数不在 XBSA 共享库中。</p> <p>请验证是否为存储管理器使用了正确的 XBSA。相关信息请查阅存储管理器手册。</p>
133	<p>装入 XBSA 库函数失败。</p> <p>请验证是否为存储管理器使用了正确的 XBSA。确保 <code>onconfig</code> 文件中的 <code>BAR_BSALIB_PATH</code> 值指向 XBSA 共享库的正确位置。相关信息请查阅存储管理器手册。</p>
134	<p>用户希望恢复的逻辑日志文件太早。</p> <p>您可能在执行单独的物理恢复后尝试了日志点恢复 (<code>gbackuprestore -r -l -n</code>)。指定的逻辑日志太旧而无法与物理恢复中使用的备份相匹配。请执行以下两个步骤之一：</p> <ul style="list-style-type: none"> <li>• 从一组较旧的物理备份重新运行物理恢复。</li> <li>• 当您重新运行日志点恢复时，在 <code>-n</code> 选项中指定较晚的逻辑日志。要查找您可以使用的最早的逻辑日志，请查看紧急引导文件。要获取帮助，请联系技术支持。</li> </ul>
136	<p><code>gbackuprestore</code> 无法对关键数据库空间执行热恢复。</p> <p>请执行以下两个步骤之一：</p> <ul style="list-style-type: none"> <li>• 不列出任何关键数据库空间的情况下重新发出热恢复命令。</li> <li>• 关闭数据库服务器并执行冷恢复。</li> </ul>
137	<p>超过了 <code>MAX_DBSPACE_COUNT</code>。</p> <p>要获取帮助，请联系技术支持。</p>

十进制值	<code>gbackuprestore</code> 返回码描述
138	<p>发生了 XBSA 错误。</p> <p>请验证是否为存储管理器使用了正确的 XBSA。同时还检查 <code>bar_act.log</code> 中的 XBSA 错误消息。相关信息请查阅存储管理器手册。</p>
139	<p>可能是 XBSA 版本不在 <code>sm_versions</code> 文件中或 <code>sm_versions</code> 文件中存在不正确的 XBSA 版本。</p> <p>将正确的 XBSA 版本插入到 <code>sm_versions</code> 文件中。要了解更多信息，请查阅存储管理器手册。</p>
140	<p>伪备份失败。</p> <p>使用 <code>gbackuprestore -b -F</code> 命令重试该伪备份。仅 GBase 8s 支持伪备份。如果伪备份再次失败，请与技术支持联系。</p>
141	<p><code>gbackuprestore</code> 接收到操作系统信号。最可能的情况是，用户输入 <code>Ctrl-C</code> 命令停止了 <code>gbackuprestore</code> 进程。</p> <p>修复导致中断的问题，然后重试 <code>gbackuprestore</code> 命令。</p>
142	<p><code>gbackuprestore</code> 无法打开文件。</p> <p>请验证指定的文件存在并且许可权也是正确的。检查 <code>gbackuprestore</code> 活动日志中操作系统的错误消息。</p>
143	<p><code>gbackuprestore</code> 无法创建子进程。</p> <p>如果 <code>BAR_MAX_BACKUP</code> 不是 0，<code>gbackuprestore</code> 将无法创建子进程来执行并行备份或恢复。很可能操作系统的资源已用尽。可能是没有足够的内存可用于启动新进程或进程表中没有空槽。</p> <p>检查操作系统日志、<code>gbackuprestore</code> 活动日志或控制台。</p>
144	<p>日志备份由于一个或多个 Blob 空间关闭而停止。</p>

十进制值	<code>gbackuprestore</code> 返回码描述
	尝试恢复 Blob 空间。如果恢复失败，请使用 <code>gbackuprestore -l -0</code> 命令重试日志备份。执行此命令可能使 Blob 空间无法恢复。
145	<code>gbackuprestore</code> 无法获取更多的内存空间。  等待系统资源释放出空间，然后重试 <code>gbackuprestore</code> 命令。
146	<code>gbackuprestore</code> 无法连接到数据库服务器。  网络或数据库服务器可能已关闭。要获取帮助，请联系技术支持。
147	<code>gbackuprestore</code> 找不到任何要进行备份或恢复的存储空间或逻辑日志。  例如：如果指定时间点恢复但使用第一次标准备份前的 <code>datetime</code> 值，那么 <code>gbackuprestore</code> 将无法构建要恢复的存储空间列表。如果指定了整个系统的恢复而没有执行整个系统的备份，也将显示该返回码。  请验证数据库服务器和存储空间是否针对备份或恢复请求而处于正确的状态。请与技术支持联系。
148	发生内部 SQL 错误。  将来自 <code>gbackuprestore</code> 活动日志的信息提供给技术支持。
149	在命令行上输入了错误的 <code>gbackuprestore</code> 语法，或者为 GLS 环境输入了无效或错误的 <code>datetime</code> 值。  根据 <code>gbackuprestore</code> 活动日志中的用法消息检查您尝试过的命令。如果没有用，请将 <code>datetime</code> 值放在引号中，然后重试该命令。如果您的数据库语言环境不是英语，请使用 <code>GL_DATE</code> 或 <code>GL_DATETIME</code> 环境变量来设置日期和时间格式。
150	从 <code>onconfig</code> 文件收集数据时出错。  检查 <code>onconfig</code> 文件中的许可权、格式和值。

十进制值	gbackuprestore 返回码描述
151	<p>对于该备份或恢复请求数据库服务器处于不正确的状态，或在确定数据库服务器状态时发生错误。</p> <p>可能是尝试了与数据库服务器方式不兼容的操作，或者可能是 gbackuprestore 无法确定数据库服务器状态。例如：无法在数据库服务器处于恢复方式时执行物理备份。</p> <p>检查 gbackuprestore 活动日志中的错误消息。如果发生了 ASF 错误，那么在 gbackuprestore 活动日志中显示以下消息：</p> <pre data-bbox="417 757 1318 808">初始化 ASF 时发生致命错误；asfcode = code</pre> <p>要确定 ASF 错误的原因，请参阅该消息中 ASF 错误代码并重复备份或恢复命令。如果未发生 ASF 错误，请更改数据库服务器状态并重复备份或恢复命令。</p>
152	<p>gbackuprestore 无法备份逻辑日志。</p> <p>可能由于以下原因之一无法备份逻辑日志：</p> <ul data-bbox="451 1227 1114 1391" style="list-style-type: none"> <li>• 当前正在运行另一个日志备份。</li> <li>• 您所执行的逻辑日志备份中 LTAPEDEV 参数设置为了 /dev/null (UNIX)。</li> </ul> <p>没有日志备份完成时您将接收到该返回码。</p> <p>要启用日志备份，请将 LTAPEDEV 参数更改为有效值。</p>
153	<p>gbackuprestore 无法设置进程组标识。如果将 BAR_MAX_BACKUP 设置为除 1 以外的任意值，并且 gbackuprestore 在设置进程组标识时发生错误，将返回该值。</p> <p>该消息是对可能的操作系统问题的警告。</p>
154	<p>gbackuprestore 用户没有正确的许可权。</p>

十进制值	<code>gbackuprestore</code> 返回码描述
	要执行 <code>gbackuprestore</code> 命令，您必须是 <code>root</code> 用户、 <code>gbasedbt</code> 用户、UNIX 上 <code>bargroup</code> 组的成员。
156	未执行备份或恢复，因为 <code>LTAPEDEV</code> 参数值无效。  如果未设置 <code>LTAPEDEV</code> 或在 UNIX 中设置为 <code>/dev/null</code> ，那么将不备份逻辑日志，并且 <code>gbackuprestore</code> 会返回警告 152。
157	试图将 <code>GBASEDBTSHMBASE</code> 环境变量设置为 <code>-1</code> 出错。  <code>gbackuprestore</code> 无法将 <code>GBASEDBTSHMBASE</code> 设置为 <code>-1</code> 。要获取帮助，请与系统管理员或技术支持联系。
158	发生内部 <code>gbackuprestore</code> 错误。  请与技术支持联系。
159	发生意外错误。  请与技术支持联系。
160	外部恢复失败。  要确定外部恢复失败的原因，请查看 <code>bar_act.log</code> 和 <code>online.log</code> 文件。确保已执行外部恢复的手动部分，然后再重试 <code>gbackuprestore -r -e</code> 命令以完成外部恢复。如果不起作用，请尝试从不同的外部备份进行外部恢复。
161	重新启动的恢复失败。  请验证 <code>RESTARTABLE_RESTORE</code> 是否设置为 <code>ON</code> 并重试原始恢复。要了解更多信息，请查看 <code>gbackuprestore</code> 活动日志和数据库服务器消息日志。
162	<code>gbackuprestore</code> 日志文件不能是符号链接。  除去符号链接或更改 <code>onconfig</code> 文件，使 <code>gbackuprestore</code> 参数

十进制值	<code>gbackuprestore</code> 返回码描述
	BAR_DEBUG_LOG 或 BAR_ACT_LOG 指向非符号链接文件。
163	<p><code>gbackuprestore</code> 日志文件的所有者必须是 <code>gbasedbt</code> 用户。</p> <p>将日志文件的所有者更改为 <code>gbasedbt</code> 用户 或者更改 <code>onconfig</code> 文件中 BAR_ACT_LOG 或 BAR_DEBUG_LOG 的值以指向不同的日志文件。</p>
164	<p>无法打开文件。</p> <p>由于该文件或其目录的许可权问题，无法创建或打开该文件。验证该文件及其目录的许可权。</p>
177	<p>联机的数据库空间已恢复。该返回码通知用户 <code>-0</code> 选项在 <code>gbackuprestore</code> 中重设了内部检查。</p> <p>您不需要采取任何操作。</p>
178	<p>当一个或多个 Blob 空间关闭时备份了逻辑日志。该返回码通知用户 <code>-0</code> 选项在 <code>gbackuprestore</code> 中重设了内部检查。</p> <p>检查 Blob 空间中的数据以确定需要重新创建哪些简单大对象。这些 Blob 空间可能无法恢复。要获取帮助，请联系技术支持。</p>
179	<p><code>gbackuprestore</code> 创建了恢复数据库空间所需要的块。该返回码通知用户 <code>-0</code> 选项在 <code>gbackuprestore</code> 中重设了内部检查。</p> <p>您不需要采取任何操作。</p>
180	<p><code>gbackuprestore</code> 无法创建恢复数据库空间所需要的块。</p> <p>请手动创建块文件。不带 <code>-0</code> 选项重试该恢复。</p>
181	<p><code>gbackuprestore</code> 使备份或恢复所需的对象到期。</p> <p><code>onsmsync</code> 实用程序使恢复可能需要的对象到期。您可能使用 <code>-0</code> 指定了 <code>onsmsync</code>。如果错误地使用了 <code>-0</code> 选项，请与技术支持联系以便从存</p>



十进制值	<code>gbackuprestore</code> 返回码描述
	储管理器恢复对象。
183	<code>gbackuprestore</code> 无法从存储管理器获得逻辑日志的唯一标识。  缺少指定逻辑日志的备份。查询存储管理器以确定指定的逻辑文件的备份是否存在，是否可恢复。
247	在 UNIX 上, 请查看 <code>/tmp/bar_act.log</code> 以及由 <code>BAR_ACT_LOG</code> 参数指向的文件, 以获得线索。( <code>gbackuprestore-merger</code> 向 <code>/tmp/bar_act.log</code> 中写入直到它有足够的信息来读取 <code>onconfig</code> 文件。) 解决 <code>bar_act.log</code> 描述的问题并重试冷恢复。如果冷恢复仍然失败, 请联系技术支持。
252	要获取帮助, 请联系技术支持。

## 4 gtape 备份与恢复系统

### 4.1 配置 gtape

这些主题说明如何设置 `gtape` 实用程序用来备份存储空间和逻辑日志的配置参数。

这些主题描述了以下任务：

- 设置 `gtape` 的配置参数
- 检查 `gtape` 的配置参数
- 更改 `gtape` 的配置参数

使用 `gtape` 备份描述如何使用 `gtape` 实用程序备份存储空间和逻辑日志文件。

有关 `gtape` 与 `gbackuprestore` 不同之处的描述, 请参阅 `gbackuprestore` 和 `gtape` 实用程序的比较。

#### 4.1.1 设置 `gtape` 实用程序的配置参数

`gtape` 实用程序使用 `onconfig` 文件中的八个配置参数。其中有两个配置参数用于为备份与恢复期间的数据变换指定过滤器程序；其他六个配置参数用于创建存储空间和逻辑日志备份。

`onconfig` 文件位于 `$GBS_DATA/conf` 目录中。

## gtape 的数据变换过滤器参数

BACKUP\_FILTER 和 RESTORE\_FILTER 配置参数指定了可用于在备份之前和恢复之后变换数据的外部程序的名称。

### BACKUP\_FILTER

指定数据变换中所用外部过滤器程序的名称和位置。该过滤器在备份数据之前对其进行变换（例如，将其压缩）。然后变换的数据将被备份，并且会存储成单个文件。过滤器路径指向 \$GBS\_HOME/bin 目录（缺省值）或程序的绝对路径

### RESTORE\_FILTER

指定数据变换中所用外部过滤器程序的名称和位置。在数据返回至服务器之前，该过滤器会将数据变换为备份之前的原始状态，例如，将其解压缩。过滤器路径指向 \$GBS\_HOME/bin 目录（缺省值）或程序的绝对路径

先决条件：该数据先前必须已使用 BACKUP\_FILTER 参数进行变换。

有关语法和使用信息（与 gbackuprestore 和 gtape 相同），请参阅 BACKUP\_FILTER 配置参数和 RESTORE\_FILTER 配置参数。

## gtape 的磁带和磁带设备参数

第一组配置参数指定用于存储空间备份的磁带设备和磁带的特征；第二组指定用于逻辑日志备份的磁带设备和磁带的特征。

以下列表显示备份磁带设备以及相关的磁带参数。

### TAPEDEV

用于存储空间备份的目录文件系统或磁带设备的绝对路径名称。指定 gtape 在归档过程中写入存储空间数据的目标位置，以及 gtape 在恢复过程中读取数据的源位置。

要配置 gtape 以使用 stdio，将 TAPEDEV 设置为 STDIO。

在云环境中备份或恢复时，将以下语法用于 TAPEDEV 配置参数：

```
TAPEDEV 'local_path, keep=option, cloud=cloud_vendor, url=url'
```

- local\_path 是将存储空间备份对象临时存储在的目录的完整路径名。
- option 可以设置为 yes 或 no。如果 keep 设置为 yes，那么 gtape 实用程序将备份对象保留在本地目录中。如果 keep 设置为 no，那么在云存储位置中传输备份对象后，会将这些对象删除。
- cloud\_vendor 是云存储供应商的名称。
- url 是将存储空间备份数据永久存储在的云存储位置。

### TAPEBLK

用于存储空间备份的磁带的块大小（以千字节为单位）。

## TAPESIZE

用于存储空间备份的磁带的大小（以千字节为单位）。值可以为 0 - 2,097,151。

以下列表显示逻辑日志磁带设备以及相关的磁带参数。

## LTAPEDEV

逻辑日志磁带设备或文件系统目录。

在云环境中备份或恢复时，将以下语法用于 LTAPEDEV 配置参数：

```
LTAPEDEV 'local_path, keep=option, cloud=cloud_vendor, url=url'
```

- local\_path 是将日志备份对象临时存储在的目录的完整路径名。
- option 可以设置为 yes 或 no。如果 keep 设置为 yes，那么 gtape 实用程序将备份对象保留在本地目录中。如果 keep 设置为 no，那么在云存储位置中传输备份对象后，会将这些对象删除。
- cloud\_vendor 是云存储供应商的名称。
- url 是将日志备份数据永久存储在的云存储位置。

## LTAPEBLK

用于逻辑日志备份的磁带的块大小（以千字节为单位）。

## LTAPESIZE

用于逻辑日志备份的磁带的大小（以千字节为单位）。值可以为 0 - 2,097,151。

以下主题中包含如何为存储空间和逻辑日志备份设置磁带设备、磁带块大小和磁带大小参数的相关信息。

### 设置磁带设备参数

通过以下方法指定 TAPEDEV 和 LTAPEDEV 的值：

- 尽可能使用不同的磁带设备。
- 使用符号链接。
- 指定文件系统的目录。
- 对于磁带设备，请指定 /dev/null。
- 将磁带设备倒带。
- 配置参数以对云存储执行备份。

下面几部分分别说明这些要点。

### 为存储空间和逻辑日志备份指定不同的设备

当备份到磁带设备时,请在 `onconfig` 文件中为 `LTAPEDEV` 和 `TAPEDEV` 参数指定不同的设备。您可以单独调度这些备份。可以在一个设备上创建备份同时在另一个设备上连续地备份逻辑日志文件。

如果为 `LTAPEDEV` 和 `TAPEDEV` 指定相同的设备,逻辑日志可能会填满,这将导致数据库服务器在备份过程中停止处理。在这种情况下,您有两个选项:

- 停止备份,释放磁带设备,然后备份逻辑日志文件。
- 使标准处理处于暂挂状态,直到备份完成。

使用一个磁带设备时需要采取的预防措施

如果只有一个磁带设备,并要在数据库服务器处于联机状态下创建备份时,请采取以下预防措施:

- 通过将许多或大型的逻辑日志文件组合起来,为数据库服务器配置大量的逻辑日志空间。(请参阅《GBase 8s 管理员指南》。)
- 将所有显式创建的临时表存储在专用的数据库空间中,然后在备份前删除该数据库空间。
- 在数据库活动较少时创建备份。
- 在开始备份前释放尽可能多的逻辑日志文件。

在备份完成前,逻辑日志可能会被填满。备份与检查点同步。备份可以等待检查点以使活动同步,但直到所有虚拟处理器都退出关键段后,该检查点才会发生。当数据库服务器的处理由于某个逻辑日志文件填满而发生暂挂时,虚拟处理器无法退出其关键段,从而导致死锁。

#### 将磁带设备指定为符号链接

可以将 `LTAPEDEV` 和 `TAPEDEV` 的值指定为符号链接。使用符号链接,您可以在不更改 `onconfig` 文件中的路径名的情况下,切换到其他磁带或与磁带兼容的设备。例如:可以为磁带设备 `/dev/rst0` 指定以下符号链接:

```
In -s /dev/rst0 /dbfiles/logtape
```

当您如以下示例所示设置 `LTAPEDEV` 配置参数时,可以在不更改 `LTAPEDEV` 参数的情况下切换到另一个设备:

```
LTAPEDEV /dbfiles/logtape
```

您只需要更改符号链接,如以下示例所示:

```
In -s /usr/backups /dbfiles/logtape
```

使用磁带设备进行备份时,如果用户只具有一个磁带设备,那么可以将逻辑日志备份重定向到磁盘文件。

### 指定文件系统目录

通过使用 `gtape` 实用程序，您可以在文件系统的目录中执行存储空间（0 级、1 级或 2 级）归档或逻辑日志备份。对于每个存储空间归档和逻辑日志备份，`gtape` 会在指定目录中创建文件。

要指定文件系统目录，请将 `LTAPEDEV` 和 `TAPEDEV` 配置参数设置为目录的绝对路径名。

当 `gtape` 重复归档操作时，它会重命名现有文件，以便不会重写旧的文件。时间戳记会添加到文件名中，以便将相关存储空间或逻辑日志文件组织到一起。

要了解文件命名模式，请参阅重命名现有文件。

### 指定远程设备

在网络中执行存储空间或逻辑日志备份时，可以备份到与 UNIX™ 和 Linux™ 平台上另一个主机相连的远程设备。

不应当执行到远程设备上的连续备份。

远程设备和数据库服务器计算机必须具有信任关系，以便 `rsh` 或 `rlogin` 实用程序从数据库服务器计算机连接到远程设备计算机，而无需询问密码。您可以通过在远程设备计算机上配置 `/etc/hosts.equiv` 文件、`~/.rhosts` 文件或系统的任何对等机制，建立信任关系。如果想要使用您的平台所使用的缺省实用程序之外的其他实用程序来处理远程会话，那么可以将 `DBREMOTECMD` 环境变量设置为要使用的特定实用程序。

要指定另一个主机上的磁带设备，请使用以下语法来设置 `TAPEDEV` 或 `LTAPEDEV` 配置参数：

```
host_machine_name:tape_device_pathname
```

以下示例指定主机 `kyoto` 上的磁带设备：

```
kyoto:/dev/rmt01
```

有关远程设备上磁带大小的信息，请参阅远程设备的磁带大小。

### 为磁带设备指定 `/dev/null`

最好不要将 `/dev/null` 用作备份时使用的设备。但是，如果您决定不需要从逻辑日志恢复事务，那么可以指定 `/dev/null` 作为逻辑日志备份的磁带设备。

将 `/dev/null` 指定为备份磁带设备时，可以避免在某些操作（如更改数据库的日志记录状态）后所需的 0 级备份的开销。显然，不能将存储空间从某个备份恢复到 `/dev/null` 上。

将磁带设备指定为 `/dev/null` 时，将忽略块大小和磁带大小。如果您将 `LTAPEDEV` 配置参数设置为 `/dev/null` 或从 `/dev/null` 更改为其他值，那么必须重新启动数据库服务器以使新设置生效。

**重要：** 将 LTAPEDEV 配置参数设置为 /dev/null 时，一旦逻辑日志文件填满，数据库服务器就将其标记为已备份，从而有效地放弃逻辑日志信息。

### 将 TAPEDEV 设置为 stdio

要将 gtape 实用程序配置为从标准输入读取或写入标准输出，请将 TAPEDEV 配置参数设置为 stdio。

### 打开前和关闭时将磁带设备倒带

对于 gtape，必须使用可倒带的磁带设备。从磁带读取或向磁带写入之前，数据库服务器将执行一系列需要倒带的检查。

### 指定磁带块大小

使用 TAPEDEV 和 LTAPEBLK 配置参数可指定您的磁带设备允许的最大块大小（以千字节为单位）。

将磁带参数设置为 /dev/null 时，将忽略相应的块大小。

指定块大小时，gtape 实用程序将不检查磁带设备。请验证磁带设备是否可以读取您指定的块大小。如果不能，您将无法恢复该磁带。

### 指定磁带大小

使用 TAPESIZE 和 LTAPESIZE 配置参数可指定可以写入磁带的最大数据量。

要写入到或读取到磁带设备的末端，请将 TAPESIZE 和 LTAPESIZE 设置为 0。您不能使用远程设备的该选项。

将磁带设备指定为 /dev/null 时，将忽略相应的磁带大小。

这些参数值的范围为 0 - 2,097,151。

### 远程设备的磁带大小

您可以估算能够存储在远程设备上的磁带中的连续逻辑日志备份数据量。

在对远程设备执行连接逻辑日志备份时，写入磁带的数量是 LTAPESIZE 和通过以下公式得出的数量二者中的较小值：

（磁盘上所有逻辑日志文件占用的总空间） -

（最大逻辑日志文件）

远程设备的 I/O 完成后，数据库服务器在日志填满条件发生前释放这些逻辑日志文件。

**限制：** 远程设备的磁带大小不能设置为 0。

### 4.1.2 更改 gtape 配置

在执行备份并复审当前和新的参数值后，您可以更改 TAPEDEV、TAPEBLK 和 TAPESIZE 或 LTAPEDEV、LTAPEBLK 和 LTAPESIZE 配置参数的值。

**先决条件：**更改 gtape 的参数之前：

- 执行 0 级备份。
- 通过在运行数据库服务器时执行 `gstat -c` 来检查配置文件。
- 如果计划将 TAPEDEV 或 LTAPEDEV 更改为不同的磁带设备，请验证该磁带设备是否可以读取使用 TAPEBLK 或 LTAPEBLK 配置参数指定的块大小。如果不能，您将无法恢复该磁带。（指定块大小时，gtape 实用程序不会检查该磁带设备。）
- 请确保您以 root 用户或 gbasedbt 的身份登录。

当数据库服务器处于联机状态时可以更改 gtape 参数的值。

**要更改 TAPEDEV、TAPEBLK 和 TAPESIZE 或 LTAPEDEV、LTAPEBLK 和 LTAPESIZE 的值：**

1. 从命令行，使用文本编辑器来编辑您的 `onconfig` 文件。
2. 保存该文件。

更改将立即生效。但是，如果您将 TAPEDEV 参数或 LTAPEDEV 参数设置为 `/dev/null`，那么必须重新启动数据库服务器。

## 4.2 使用 gtape 备份

这些主题描述如何使用 gtape 实用程序来备份存储空间和逻辑日志文件，以及如何更改数据库日志记录的状态。gtape 实用程序可以备份与恢复数据库服务器所支持的最大的块文件。gtape 实用程序不能备份临时数据库空间和临时智能大对象空间。

### 4.2.1 gtape 任务摘要

gtape 实用程序允许您完成各种各样的任务：

- 更改数据库日志记录状态
- 创建备份
- 启动连续的逻辑日志文件备份
- gtape 实用程序执行恢复的语法
- 使用外部恢复命令

## 启动 gtape

如果在备份期间需要多盘磁带，gtape 实用程序会在每次需要添加一盘磁带时进行提示。

如果数据库服务器处于维护方式（例如，在转换期间），那么 gtape 实用程序只能由以下某个用户启动：

- root
- gbasedbt
- 启动过数据库服务器的用户（如果不是用户 root 或 gbasedbt）

**限制：** 请勿在后台方式下启动 gtape 实用程序（即，在命令行上使用 UNIX™ & 运算符来启动）。您可能还需要从终端或窗口提供输入。如果在后台方式下执行 gtape，您可能无法看到提示而延迟操作。

gtape 实用程序不包含用于用户交互的缺省值，也不支持重试操作。当 gtape 期待“是/否”的响应时，它将所有未识别为“是”的响应都假定为“否”。

## gtape 的退出码

gtape 实用程序具有下面两个退出码：

**0**

表示正常退出 gtape。

**1**

表示出现异常情况。

## 4.2.2 更改数据库日志记录状态

您可以使用 gtape 实用程序来更改数据库的日志记录状态。大部分情况下，更改日志记录方式都需要完整的 0 级备份。

符合 ANSI 标准的数据库的日志记录方式无法更改。

无缓冲记录或缓冲记录的数据库可以更改为未记录的数据库，这样则无需备份。

使用 0 级备份可以进行以下日志记录方式更改：

- 从无缓冲记录或缓冲记录的数据库更改为 ANSI 数据库
- 从无缓冲记录的数据库更改为缓冲记录的数据库
- 从缓冲记录的数据库更改为无缓冲记录的数据库

示例

以下命令将名为 stores7 的数据库日志记录方式更改为无缓冲日志记录：



```
gtape -s -L 0 -U stores7
```

以下命令将数据库日志记录方式更改为符合 ANSI 标准的日志记录方式:

```
gtape -s -L 0 -A stores7
```

以下命令将数据库日志记录方式更改为未记录的方式:

```
gtape -N stores7
```

### 4.2.3 创建备份

这些主题说明如何计划和创建数据库服务器数据的备份。

#### gtape 支持的备份级别

gtape 实用程序支持 0 级、1 级和 2 级备份。

有关调度备份的信息，请参阅计划恢复策略。

**提示：** 创建备份调度，这样可保持较小的 1 级和 2 级备份。调度常用 0 级备份，以避免恢复大型的 1 级和 2 级备份或许多逻辑日志备份。

#### 0 级备份

例如：当火灾或洪水完全毁坏计算机时，就需要通过 0 级备份将数据库服务器数据完全恢复到替换的计算机上。对于联机备份，备份磁带上的数据反映了 0 级备份开始时存储空间中的内容。（备份开始的时间可以反映备份开始前的最后一个检查点。）

0 级备份可能花费较长的时间，因为 gtape 必须将所有页都写入磁带中。

#### 1 级备份

1 级备份通常比 0 级备份花费的时间短，因为您只将数据库服务器的部分数据复制到备份磁带中。

#### 2 级备份

在 1 级备份后执行的 2 级备份通常比再执行另一个 1 级备份花费的时间短，因为它只将上次 1 级备份后（而不是上次的 0 级备份）发生的更改复制到备份磁带中。

#### 更改物理模式后备份

必须执行 0 级备份以确保在更改物理模式后可以恢复数据。

进行以下管理更改后，执行 0 级备份：

- 将 TAPEDEV 或 LTAPEDEV 配置参数更改为 /dev/null 之外的值
- 将日志记录添加到数据库中

- 在可以使用小于整个系统恢复的任何恢复方式进行恢复之前，添加数据库空间、Blob 空间或智能大对象空间
- 对包含逻辑日志文件的数据库空间开始镜像
- 删除逻辑日志文件
- 移动一个或多个逻辑日志文件
- 更改物理日志的大小或位置，然后设置共享内存
- 在可以重新使用包含某个块的数据库空间前删除该块
- 在冷恢复期间重命名块

请考虑等到下个定期调度的 0 级备份之后再进行这些更改。

**提示：** 尽管在添加逻辑日志文件后不再需要立即备份，但是因为数据结构发生了变化，所以下一个备份应该是 0 级备份。

## 准备备份

创建备份时，请采取以下预防措施：

- 出现大量活动时避免使用临时表。
- 确保具有足够的逻辑日志空间。
- 保留配置文件的副本。
- 执行 0 级备份之前验证一致性。
- 以适当的方式运行数据库服务器。
- 规划操作符的可用性。
- 与其他管理任务同步。
- 不要使用后台方式。
- 如有必要，适当标记磁带。
- 如有必要，准备写入到标准输出中。

### 出现大量活动时避免临时表

当您在备份期间使用 `gtape` 实用程序创建临时表时，该表将置于 `DBSPACETEMP` 中。在备份进程中出现大量活动时，临时表可能持续增长并最终填满 `DBSPACETEMP`。发生这种情况时，备份将停止并且监视器会显示“没有可用磁盘”的错误消息。

### 确保具有足够的逻辑日志空间

当逻辑日志中总的可用空间少于单个逻辑日志文件的一半时，数据库服务器将不创建备份。此时必须备份逻辑日志文件并再次尝试进行备份。

在备份过程中不能添加镜像。

**重要：** 如果只使用一个可用的磁带设备，请确保开始备份前对所有的逻辑日志文件进行备份，以减少备份期间填满逻辑日志的可能性。

### 保留配置文件的副本

创建 0 级备份时请保留当前 `onconfig` 文件的副本。从备份磁带恢复数据库服务器数据时需要该信息。

### 执行 0 级备份之前验证一致性

要确保备份的完整性，请在创建整个系统的 0 级备份前定期验证所有的数据库服务器数据和开销信息是否一致。不需要在每个 0 级备份前都检查该信息，但建议您保留成功验证数据库服务器的一致性之后随即创建的最新备份。有关一致性检查的信息，请参阅《GBase 8s 管理员指南》。

### 联机 and 静默备份

可以在数据库服务器处于联机状态或静默方式时创建备份。用于启动备份命令的终端专用于进行备份（显示消息），直到备份完成为止。一旦备份开始，数据库服务器必须始终处于相同的方式直到备份完成；对方式进行更改将终止备份活动。

#### 联机备份

如果希望创建备份时数据库服务器是可访问的，那么可以使用联机备份。

联机备份期间可能会有些微的不方便。联机备份可能减慢检查点活动，从而可能造成性能下降。然而这种性能的下降造成的代价远小于拒绝用户在备份期间访问数据库服务器而造成的代价。

联机备份期间，存储空间中某些磁盘页的分配会暂时冻结。在您备份块中已使用的页之前，磁盘页的分配在某个时间对该块是阻塞的。

#### 静默备份

当数据库服务器处于停顿方式时创建静默备份。如果要除去备份中的部分事务，请使用静默备份。

如果用户需要连续访问数据库则不要使用静默备份。

### 备份到磁带

当备份到磁带时，您必须确保有操作员在场，而且有足够多的介质。

在备份期间请保持操作符可用，以便根据提示安装磁带。备份可能需要若干卷磁带。当一个磁带变满而没有可用的操作符来安装新磁带时，备份将等待。如果备份是联机备份，那么在此等待期间，物理日志空间可能会填满，这将导致数据库服务器停止备份。因此，请确保操作员在场。

在一个磁带填满后，`gtape` 实用程序会将该磁带倒带，显示用于标记的磁带编号，并在需要另一个磁带时提示操作员安装下一个磁带。请按照提示来标记并安装新磁带。当备份完成时有消息通知您。

#### 标记使用 `gtape` 创建的磁带

当您标记用 `gtape` 实用程序创建的磁带时，标签必须包含以下信息：

- 备份级别
- 日期和时间
- `gtape` 提供的磁带编号

以下示例显示标签的形式：

```
Level 1: Wed Nov 27, 2001 20:45 Tape # 3 of 5
```

每个备份都以编号为 1 的第一卷磁带开始。随后再对每个附加的磁带卷进行连续编号。将五个磁带的备份编号为 1 到 5。（当然，有可能在备份完成前您无法得知这是五个磁带的备份。）

#### 备份到标准输出

到标准输出的备份在操作系统提供的内存缓冲区中创建归档文件。如果您选择备份到标准输出，您无须提供磁带或其他存储介质。

备份到标准输出具有以下优点：

- 无需代价高昂的磁盘、磁带读写操作。
- 可以使用操作系统实用程序对数据进行压缩或其他处理。
- 可以通过将数据直接恢复到另一台数据库服务器，用归档创建该服务器的副本。

如果您备份到了标准输出，那么您也必须从标准输入恢复。

当 `gtape` 执行备份到标准输出的操作时，数据被写入一个输出文件。输出的目录必须有足够的磁盘空间来容纳备份的数据。可以使用操作系统实用程序压缩数据。此外，执行 `backup` 命令的用户必须拥有备份转向到的文件的写许可权或创建该文件的权限。

当您备份到标准输出时，`gtape` 不提示用户交互。错误和参考消息将写入 `stderr`，而不是被指向到标准输出。

将不使用 `TAPESIZE` 配置参数，因为标准输出的容量假定为无限。但是 `TAPEBLK` 配置参数有效，因为它定义了后端服务器与 `gtape` 客户端之间的传送缓冲区大小。您可以通过将 `TAPEBLK` 设置为适当的值来优化吞吐量。

您可以用同时备份与恢复的方法克隆一台数据库服务器或建立高可用性数据复制。有关更多信息，请参阅使用标准 I/O 同时备份与恢复。

### 备份到目录

如果选择备份到目录，那么无需提供磁带。此外，您可以将数据备份到本地文件系统的目录或安装在本地系统上的目录。

运行备份的人员必须具有目录的写许可权。目录必须有足够的磁带空间来容纳备份的数据。可以在数据备份后使用该操作系统实用程序对数据进行压缩。

备份到目录具有以下优点：

- 多个实例可以同时备份到相同的目录文件系统。
- 可以使用操作系统实用程序对数据进行压缩或其他处理。
- 当文件满时，您可以轻松地配置您的系统来自动备份日志文件。

### 设置文件目录路径

请使用 `TAPEDEV` 配置参数来指定用于存储空间归档文件的文件系统目录的绝对路径名。这是 `gtape` 在归档过程中写入存储空间数据的目标位置，同时也是 `gtape` 在恢复过程中读取数据的源位置。您可以使用 `LTAPEDEV` 配置参数指定写入逻辑日志备份文件的目录。

**提示：** 备份到目录文件系统时，指定 `-d` 选项来关闭 `gtape` 交互式提示。

### 重命名现有文件

当 `gtape` 重复归档操作时，它会重命名现有文件，以便不会重写旧的文件。时间戳记会添加到文件名中，以便将相关存储空间或逻辑日志文件组织到一起。

重命名约定：

- 存储空间归档文件  
将添加归档检查点，并且其格式为 `servername_YYYYMMDD_hhmmss_archive-level`。
- 逻辑日志备份文件  
将添加备份时间，并且其格式为 `servername_YYYYMMDD_hhmmss`。

例如，文件 `My_instance_L0` 将重命名为 `My_instance_20080913_091527_L0`

从文件系统目录恢复时，`gtape` 需要按照 `TAPEDEV` 和 `LTAPEDEV` 参数的指定来命名存储空间归档和逻辑日志备份文件。如果由于重复归档和备份，已重命名文件（包括由 `gtape` 重命名），那么必须将文件手动重命名为其原始文件名。

### 覆盖归档文件的缺省名称

可以覆盖归档文件的缺省名称。TAPEDEV 或 LTAPEDEV 是目录路径，缺省的永久文件名由 `hostname_servernum_Ln`（对于级别）和 `hostname_servernum_Lognnnnnnnnnn`（对于日志文件）。您可以通过设置环境变量 `IFX_ONTAPE_FILE_PREFIX` 来覆盖永久文件名的前缀部分 (`hostname_servernum`)。

例如，如果将 `IFX_ONTAPE_FILE_PREFIX` 设置为“My\_Instance”，那么在归档的过程中，文件将命名为 `My_Instance_L0`、`My_Instance_L1`、`My_Instance_L2` 和 `My_Instance_Log0000000001`、`My_Instance_Log0000000002` 等。在恢复过程中，`gtape` 将在 `TAPEDEV` 目录中搜索文件名类似于 `My_Instance_L0` 的文件，并且还将在 `LTAPEDEV` 目录中搜索文件名类似于 `My_Instance_Log0000000001` 的文件。

### 执行备份

使用 `gtape` 实用程序命令选项可备份到磁带。

**先决条件：**开始备份前，请执行以下步骤：

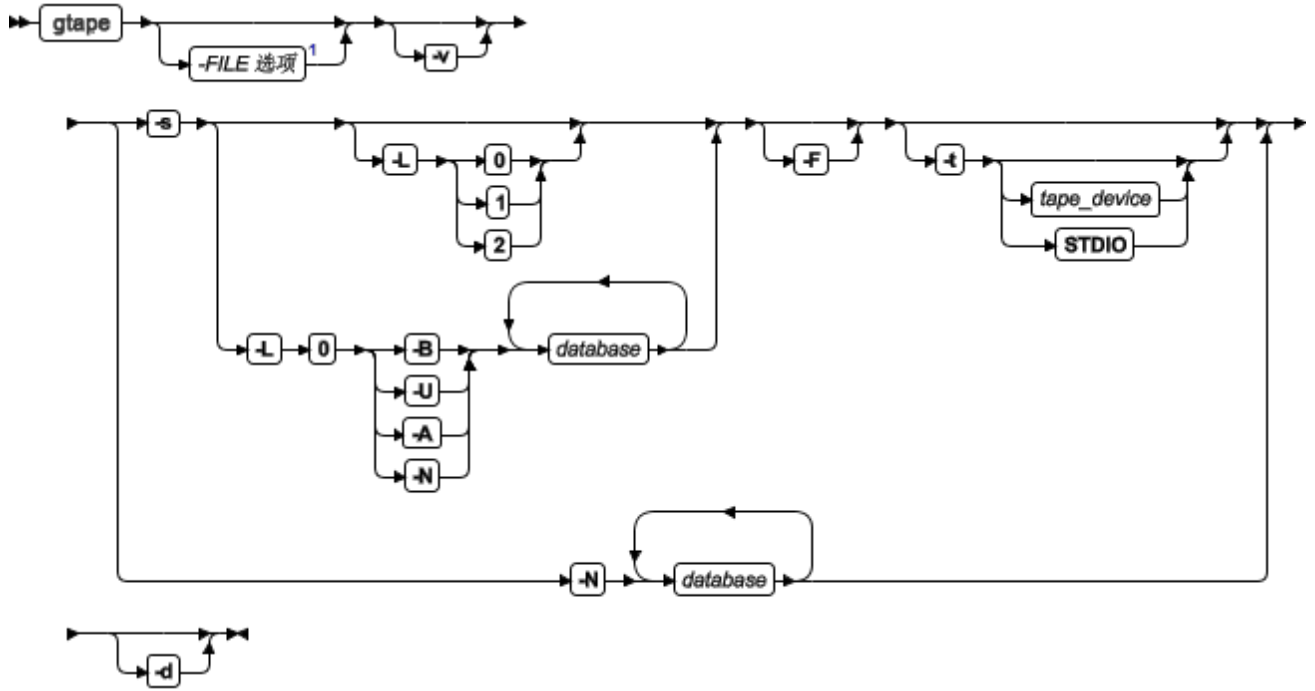
- 如有必要，将可写的磁带放在 `TAPEDEV` 指定的磁盘机设备中。  
如果将 `TAPEDEV` 设置为 `STDIO`，要确保为备份数据留有足够内存量。
- 使用适当的操作系统命令使该设备联机。
- 使数据库服务器处于联机状态或静默方式。

不要在同一磁带上存储多个备份；每次备份开始时都使用不同的磁带。（通常，一次备份将使用多卷磁带。）

要创建备份，请使用 `gtape` 命令的 `-s` 选项。

语法

创建备份



元素	用途	关键注意事项
-A	指示 gtape 将指定数据库的状态更改为符合 ANSI 标准的日志记录。	具有符合 ANSI 标准的日志记录的数据库无法更改为其他日志记录方式。需要 0 级备份才能更改此日志记录方式。
-B	指示 gtape 将指定数据库的状态更改为缓冲日志记录。	需要 0 级备份才能更改此日志记录方式。
-d	指示执行 gtape, 同时没有交互式提示。	如果要备份到文件系统目录或从文件系统目录中恢复, 那么可以关闭提示。此选项不适用于磁带设备, 在您更换磁带时, 必须暂停备份。
database	要更改日志记录方式的数据库的名称。	数据库名称中不能包含数据库服务器名称。可以在同一命令中指定多个数据库名称。
-F	指示 gtape 执行伪备份。	伪备份仅当备份到标准输出期间适用。  伪备份可用于克隆服务器中的数据。例如, 填充高可用性集群中的辅助服务器。  为了不危害正常的备份活动, 请不要保留假备

元素	用途	关键注意事项
		<p>份的记录。</p> <p>此外,您也可以使用与 SQL 管理 API 等效的 ARCHIVE FAKE。有关更多信息,请参阅《GBase 8s 管理员参考》。</p>
-L	指示 <code>gtape</code> 创建指定级别的备份。	<p>如果您要备份到磁带,使用 <code>-L</code> 选项将备份级别指定为命令的一部分,就可以避免出现与其相关的提示。</p> <p>如果您要备份到标准输出而不指定备份级别,<code>gtape</code> 将执行 0 级备份。</p>
-N	指示 <code>gtape</code> 终止指定数据库的日志记录。	更改此日志记录方式时可选择使用备份。
-s	指示 <code>gtape</code> 创建备份。	如果您未使用 <code>-L</code> 选项提供一个值, <code>gtape</code> 将提示您提供要创建的备份级别 (0、1 或 2)。
-t	指示 <code>gtape</code> 为当前备份或恢复使用其他磁带设备。	<code>-t</code> 选项为当前备份或恢复覆盖 TAPEDEV 配置参数的值。 <code>-t STUDIO</code> 选项指示 <code>gtape</code> 备份到标准输出或从标准输入恢复。
tape_device	存储备份的磁带设备的名称。	
-U	指示 <code>gtape</code> 将指定数据库的状态更改为无缓冲日志记录。	需要 0 级备份才能更改此日志记录方式。
-v	指示 <code>gtape</code> 在备份到标准输出时向 <code>stderr</code> 写入参考消息。	详细模式对于监视备份到标准输出的进程很有用。

`gtape` 实用程序按照以下次序备份存储空间: 根数据库空间、Blob 空间、智能大对象空间和数据库空间。



### 备份示例

执行以下命令以在不指定级别的情况下启动备份到磁带：`gtape -s`

可以使用 `-L` 选项将备份级别指定为命令的一部分，如以下示例所示：`gtape -s -L 0`

当备份到目录或从目录中复原时，请使用 `-d` 选项来避免交互式提示：`gtape -s -L 0 -d`

如果不在命令行上指定备份级别，`gtape` 将提示您输入它。下图举例说明了一个简单的 `gtape` 备份会话。

图: 使用 `gtape` 创建的简单备份的示例

```
gtape -s
请输入要执行的归档级别 (0、1 或 2) 0

请在 /dev/rst0 上安装磁带 1 并按“返回”以继续...
16:23:13 检查点已完成：持续时间是 2 秒
16:23:13 rootdbs 上的 0 级归档已启动
16:23:30 rootdbs 上的归档已完成。
16:23:31 检查点已完成：持续时间是 0 秒

请在弧形磁带序列中将该磁带标记为编号 1。
该磁带包含以下逻辑日志：

3

程序结束。
```

以下示例显示了如何创建所有存储空间的 0 级归档到标准输出，并转向到 `/home` 目录中的名为 `level_0_archive` 的文件：

```
gtape -s -L 0 >/home/level_0_archive -t STDIO
```

以下示例假定 `onconfig` 中有 `TAPEDEV STDIO`，并创建了一个 1 级归档到标准输出，并转向到一个管道：

```
gtape -v -s -L 1|compress -c >/home/compressed/level_1_archive
```

`compress` 系统实用程序从管道读取数据作为输入，压缩数据，并将数据写入 `/home/compressed` 目录中的 `level_1_archive` 文件。`gtape` 参考消息被送到 `stderr`。

### 备份原始表

可以使用 `gtape` 备份原始表，但是不为原始表记录日志。因此当您复原原始表时，无法复原自上次备份以来发生的所有更改。建议只在最初装入数据时使用原始表，随后在执行事务前改用标准表。有关更多信息，请参阅《GBase 8s 管理员指南》。

## 备份到 Amazon Simple Storage Service

您可以使用 `gtape` 实用程序在 Amazon Simple Storage Service (S3) 中备份与恢复数据。您应该遵守 Amazon Simple Storage Service 的相关条款并承担任何相关费用。

先决条件：

- 您必须具有 Amazon 帐户，才能执行云存储备份。请参阅 Amazon Web 站点，以获取有关设置帐户的指示信息。
- 需要 Java™ V1.5 或更高版本。
- 备份对象必须为 5 GB 或更低。

以下步骤显示如何使用 `gtape` 备份与恢复实用程序来将数据备份到 Amazon Simple Storage Service (S3) 系统并从中恢复。在此上下文中，云存储指一种通过因特网提供的联机存储服务。如果选择备份到云存储，那么无需提供磁带。取而代之的是将数据备份到虚拟设备，大多是位于因特网中。

1. 配置联机存储设备。
  - a. 使用 Web 浏览器，浏览到 Amazon S3 Web 站点并登录。
  - b. 获取访问密钥标识和访问密钥。
  - c. 将访问凭证存储到文件中。将文件的许可权设置为仅执行 `gtape` 实用程序的用户可访问。
    - 在 UNIX™ 系统上，将值存储到该文件中：  
`$GBS_HOME/etc/ifxbkcloud.credentials`

文件必须具有以下格式：

```
secretKey=secret_access_key  
accessKey=access_key_ID
```

使用 `ifxbkcloud.jar` 实用程序在准备用于存储 GBase 8s 数据的区域中创建存储设备并命名。Amazon 使用词汇存储区来描述备份数据的容器。选择存储设备的名称与 Amazon S3 中存储区的名称具有相同的限制要求，且必须是唯一的。

例如，以下命令在 Amazon S3 上的 US Standard 区域中创建名为 `mytapedevice` 的存储设备。从 UNIX 系统上的 `$GBS_HOME/bin` 目录运行该命令。

```
java -jar ifxbkcloud.jar CREATE_DEVICE amazon mytapedevice US_Standard
```

2. 将 `onconfig` 文件中的 `TAPEDEV` 和 `LTAPEDEV` 配置参数设置为云存储的位置。  
例如：

```
TAPEDEV '/opt/gbase/gbasedbt/tapedev_dir, keep = yes, cloud = amazon,  
url = https://mytapedevice.s3.amazonaws.com'  
LTAPEDEV '/opt/gbase/gbasedbt/ltapedev_dir, keep = yes, cloud = amazon,  
url = https://mylogdevice.s3.amazonaws.com'
```

3. 使用 `gtape` 实用程序将数据备份到联机存储设备中。

```
gtape -s -L 0
```

使用以下命令可以从云存储恢复数据：

```
gtape -r
```

将数据传输到云存储时应使用 `https` 保密数据传输。将数据传输到云映像之前应加密数据。要加密数据，请使用 `BACKUP_FILTER` 和 `RESTORE_FILTER` 配置参数来调用外部加密程序。`archecker` 实用程序不支持云存储的数据表级别恢复。

### ifxbkpcld.jar 实用程序

使用 `ifxbkpcld.jar` 实用程序来配置 Amazon Simple Storage Service 的联机存储设备。

`ifxbkpcld.jar` 实用程序支持以下选项：

- `CREATE_DEVICE provider device [region]`
- `DELETE_DEVICE provider device`
- `LIST_DEVICES provider`
- `DELETE_FILE provider device file`
- `LIST_FILES provider device`

`ifxbkpcld.jar` 命令的参数定义如下所示：

- `provider` 是 `amazon`。
- `device` 是存储设备的名称。
- `region` 是以下某个区域：`US_Standard`、`US_West`、`EU_Ireland` 或 `AP_Singapore`。
- `file` 是在 Amazon S3 上存储的备份对象（键）的名称。

`ifxbkpcld.jar` 的错误消息会写入到 UNIX™ 机器上的 `$GBS_HOME/ifxbkpcld.log`。

### 云存储文件命名约定

与云存储备份相关联的文件具有唯一文件名。

数据空间备份文件使用以下格式进行保存：

```
hostname_servernum_Larchive_level
```

日志备份文件名使用以下格式进行保存：

```
hostname_servernum_lognnnnnnnnnn
```

如果云存储位置中已存在对象，那么会对文件重命名以避免覆盖旧对象。重命名文件是将时间戳记添加到对象名中。

数据空间备份文件使用以下格式进行保存：

```
hostname_servernum_YYYYMMDD_hhmmss_Larchive_level
```

日志备份文件名使用以下格式进行保存：

```
hostname_servernum_lognnnnnnnnnn_YYYYMMDD_hhmmss
```

### 备份期间逻辑日志文件填满

如果在备份期间逻辑日志文件填满，控制台显示一条消息，备份暂挂正常处理。如何处理逻辑日志填满取决于您是使用一个还是两个磁带设备。

可以使用两个磁带设备时

如果可以将两个磁带设备与数据库服务器一起使用，那么在可用终端上以用户 `gbasedbt` 登录。

验证 `LTAPEDEV` 和 `TAPEDEV` 是否指定了与单独的磁带设备对应的不同路径名。如果是，那么备份逻辑日志文件。请参阅创建备份。

如果 `LTAPEDEV` 和 `TAPEDEV` 是相同的，请对逻辑日志磁带设备指定不同的值 (`LTAPEDEV`) 并启动逻辑日志文件的备份。否则，可以选择让正常的数据库服务器处理暂挂直到备份完成，或选择取消备份。

只有一个磁带设备可用时

可以使用一个可用的磁带设备创建备份时，在完成备份前不能备份任何逻辑日志文件。如果在备份期间逻辑日志文件填满，正常的数据库服务器处理将停止。您可以停止备份（使用 `Ctrl-C` 即可）来释放磁带设备并备份逻辑日志以继续处理，或者让正常的处理暂挂直到备份完成为止。

可以采取一些步骤来防止这种情况的发生。启动自动逻辑日志备份这一部分描述了这些步骤。

### 备份过早终止

如果取消或中断备份，有时备份进行到了您可能认为已经完成的位置。如果监视信息列出备份已完成（如使用 `gcheck` 监视备份历史记录所述），那么备份确实已完成。

### 使用 `gcheck` 监视备份历史记录

可以使用 `gcheck` 监视上次整个系统备份的历史记录。

执行 `gcheck -pr` 命令可显示根数据库空间的保留页信息。最后一对保留页包含最近备份的以下信息：

- 备份级别（0、1 或 2）
- 备份的有效日期和时间
- 描述备份何时开始的时间戳记（用十进制数表示）
- 备份开始时当前逻辑日志的标识号
- 检查点记录（备份开始时写入的）在逻辑日志中的物理位置

备份的有效日期和时间等于作为此次备份开始点的检查点的日期和时间。此日期和时间与备份进程启动的时间可能明显不同。

例如：如果在星期二下午 7 点以后再没有人访问数据库服务器，您在星期三早上创建了一个备份，那么该备份的有效日期和时间是星期二晚上，即最近一个检查点的时间。也就是说，如果在上一个检查点后没有活动发生，数据库服务器开始备份时将不执行另外的检查点。

#### 4.2.4 使用 `gtape` 备份逻辑日志文件

当您使用 `gtape` 制作备份磁带时，便只能使用 `gtape` 备份逻辑日志文件。

除了备份逻辑日志文件外，还可以使用 `gtape` 切换到下一个日志文件、将逻辑日志文件移动到其他数据库空间或更改逻辑日志的大小。有关那些任务的指示信息出现在《GBase 8s 管理员指南》中。

##### 备份逻辑日志文件之前

备份逻辑日志文件之前，必须了解以下问题：

- 是否需要备份逻辑日志文件
- 何时需要备份逻辑日志文件
- 需要执行自动备份还是连续备份

有关这些问题的更多信息，请参阅逻辑日志备份。

##### 使用 Blob 空间 TEXT 和 BYTE 数据类型以及逻辑日志文件

在使用事务日志记录的数据库中使用 TEXT 和 BYTE 数据类型时，必须切记以下两点：

- 为确保能及时重用 Blob 页，请备份逻辑日志文件。当用户删除 Blob 空间中的 TEXT 或 BYTE 值时，直到您释放包含已删除记录的日志文件以后 Blob 页才会释放以供重新使用。必须备份日志文件才能将其释放。

- 必须备份不可用的 Blob 空间时，gtape 将跳过它，这将造成必要时不可能恢复 TEXT 或 BYTE 的值。（不过，即使没有备份 TEXT 或 BYTE 的值，一旦 Blob 空间可用时，已删除的 TEXT 或 BYTE 值的 Blob 页也可用了。）

此外，不管数据库是否使用事务日志记录，当您创建 Blob 空间或向 Blob 空间添加块时，该 Blob 空间或新块不可用，直到记录事件的逻辑日志文件不是当前的逻辑日志文件为止。有关切换逻辑日志文件的信息，请参阅《GBase 8s 管理员指南》。

#### 在不需要恢复时使用 /dev/null

当您确定在两次备份之间不需要恢复事务或管理数据库活动时，可以将数据库服务器配置参数 LTAPEDEV 设置为 /dev/null。

**重要：** 将 LTAPEDEV 设置为 /dev/null 时，它具有以下含义：

- 只能将数据库服务器管理的数据恢复到最近的备份以及任何先前备份的逻辑日志文件之时的状态。
- 进行恢复时，必须始终执行整个系统恢复。（请参阅全面系统恢复。）不能执行部分恢复或在数据库服务器处于联机状态时进行恢复。

当您把 LTAPEDEV 设置为 /dev/null 时，逻辑日志文件一填满，数据库服务器就将其标记为已备份（状态 B）。这样数据库服务器就可以接着重用该逻辑日志文件而不需要等待您将其备份。其结果是数据库服务器不保留任何逻辑日志记录。

将 /dev/null 用作日志文件备份设备时，快速恢复和回滚事务不会受到不利影响。有关快速恢复的描述，请参阅《GBase 8s 管理员指南》。有关回滚事务的信息，请参阅 **GBase 8s SQL 指南：语法** 中的 ROLLBACK WORK 语句。

#### 备份逻辑日志文件的时间

您必须试着在每个逻辑日志文件填满时尽快将其备份。由于每个逻辑日志文件都具有已使用状态，因此您可以判断何时可以备份逻辑日志文件。有关监视逻辑日志文件状态的更多信息，请参阅《GBase 8s 管理员指南》。

#### 启动自动逻辑日志备份

备份逻辑日志文件时，数据库服务器可以在联机状态下运行。要备份所有已满的逻辑日志文件，请使用 gtape 命令的 -a 选项。

请求逻辑日志备份

```
▶▶ gtape -a ▶▶
```

-a 选项备份所有已满的逻辑日志文件，并提示您通过某个选项切换逻辑日志文件以及备份当时的最新日志。

如果在逻辑日志文件结束前，安装在 LTAPEDEV 上的磁带变满，gtape 将提示您安装新磁带。

如果在备份发生时按下中断键，数据库服务器将结束备份并将控制权交回给用户。其他任何已满的逻辑日志文件都收到已使用状态。

要备份所有已满的逻辑日志文件，请执行 `gtape -a` 命令。

### 启动连续的逻辑日志文件备份

如果不希望监视逻辑日志文件并在逻辑日志文件填满时开始备份，您可以启动连续备份。

启动连续备份后，数据库服务器将在每个逻辑日志文件填满后自动将其备份。执行连续逻辑日志文件备份时，即使在最坏的情况下，即包含逻辑日志文件的块出现故障而造成介质故障时，数据库服务器也能提供保护，防止您丢失一个以上的部分逻辑日志文件。

要启动逻辑日志文件的连续备份，请使用 `gtape -c` 命令。`-c` 选项启动逻辑日志文件的连续备份。每个逻辑日志文件一填满，数据库服务器就对其进行备份。连续备份将不对当前逻辑日志文件进行备份。启动连续备份后，数据库服务器可以在联机方式下运行。

逻辑日志文件是备份到磁带还是目录中取决于 LTAPEDEV 配置参数的设置：

- 如果 LTAPEDEV 配置参数设置为磁带设备，那么必须有人始终提供用于备份进程的介质。如果指定的已安装磁带被填满而逻辑日志文件还未结束，那么数据库服务器会提示操作员安装新磁带。同时还必须为备份进程指定备份设备。
- 如果 LTAPEDEV 配置参数设置为目录，那么可以在无人照管的情况下备份逻辑日志文件。逻辑日志填满时会进行备份，并且会在目录中为每个逻辑日志创建新文件。备份受限于新文件可用的空间。

要备份到目录，您可以从 ALARMPROGRAM 配置参数指定的脚本调用 `gtape -a -d` 自动逻辑日志备份命令，而不需要使用 `gtape -c` 命令。您可以使用 `alarmprogram` 脚本或 `log_full` 脚本，这两个脚本都位于 `$GBS_HOME/etc` 目录中。

要使用 `alarmprogram` 脚本来将逻辑日志备份到目录：

1. 将 LTAPEDEV 参数设置为现有目录。

确保 `gbasedbt` 和组 `gbasedbt` 具有此目录。

2. 编辑 ALARMPROGRAM 脚本（UNIX™ 或 Linux™ 上的 `$GBS_HOME/etc/alarmprogram.sh`），如下所示：
  - a. 将文件内的 `BACKUPLGOS` 参数设置为 `Y`。
  - b. 将备份程序从 `gbackupstore -b -l` 更改为 `gtape -a -d`。
3. 重新启动数据库服务器。

### 结束连续的逻辑日志备份

要结束连续逻辑日志备份，请按中断键 (CTRL-C)。

如果在数据库服务器将逻辑日志文件备份到本地设备时按下中断键，那么中断前备份的所有日志都将捕获到磁带上，并由数据库服务器标记为已备份。

如果数据库服务器等待逻辑日志文件填满(因而不备份任何逻辑日志文件)时按下中断键，所有中断前已备份的日志将驻留在磁带上，并由数据库服务器标记为已备份。

如果数据库服务器执行到远程设备的连续备份时按下中断键，那么此操作过程中备份的所有逻辑日志文件可能会也可能不会驻留在磁带上，并且数据库服务器不将其标记为已备份（这是您为什么不应进行连续远程备份的一个适当原因）。

停止连续日志记录后，必须对随后的日志备份操作开始使用新磁带。

必须显式地请求逻辑日志备份（使用 `gtape -a`），直到重新启动连续日志记录。

### 逻辑日志备份必须使用的设备

`gtape` 实用程序使用在 `onconfig` 文件中定义的参数来定义用于逻辑日志备份的磁带设备。然而，在选择逻辑日志备份设备时请考虑以下问题：

- 如果逻辑日志设备与备份设备不同，在规划备份时可以不考虑备份调度中的争用需要。
- 如果将 `/dev/null` 指定为配置参数 `LTAPEDEV` 中的逻辑日志备份设备，那么不必安装和维护备份磁带。不过您只能将数据恢复到最近的备份磁带中的内容。无法恢复备份后完成的工作。请参阅在不需要恢复时使用 `/dev/null` 中有关将 `LTAPEDEV` 设置为 `/dev/null` 的相关警告。

如果高可用性集群中任意服务器节点上的日志备份设备设置为 `/dev/null`（在 Linux<sup>™</sup> 或 UNIX<sup>™</sup> 上），那么集群内所有其他服务器（包括主服务器和任何 HDR、RSS 或 SDS 辅助服务器）的备份设备都必须设置为 `/dev/null`（或 `NUL`）。

- 当磁带设备运行减慢时，逻辑日志填充的速度可能大于将其复制到磁带上的速度。在这种情况下，可以考虑执行到磁盘的备份，然后将磁盘上的备份复制到磁带上。

## 4.3 使用 `gtape` 恢复

这些主题为以下过程提供使用 `gtape` 实用程序恢复数据的指示信息：

- 恢复整个系统
- 对选定的数据库空间、Blob 空间和智能大对象空间进行恢复



开始恢复数据前，必须理解恢复系统中的概念。如该节中说明的那样，对数据库服务器数据的完整恢复通常包含物理恢复和逻辑恢复。

### 4.3.1 物理恢复的类型

如果故障导致数据库服务器脱机，您必须恢复所有的数据库服务器数据。此类型的恢复是整个系统恢复。您仅可以将数据恢复到相同版本的 GBase 8s。如果故障没有导致数据库服务器脱机，那么您可以只恢复发生故障的存储空间。有关恢复类型的举例说明，请参阅热恢复、冷恢复和混合恢复。

#### 全面系统恢复

当数据库服务器由于磁盘故障或数据毁坏而脱机时，这意味着关键数据库空间受损。以下列表显示关键数据库空间：

- 根数据库空间
- 包含物理日志的数据库空间
- 包含逻辑日志文件的数据库空间

如果需要恢复任一关键数据库空间，必须执行整个系统恢复以恢复您的数据库服务器管理的所有数据。必须用冷恢复来启动整个系统恢复。请参阅冷恢复、热恢复或混合恢复。

#### 对数据库空间、Blob 空间和智能大对象空间进行恢复

当数据库服务器没有因为磁盘故障或数据毁坏而脱机时，将对非关键数据库空间、Blob 空间或智能大对象空间造成损害。

如果不需要恢复关键数据库空间，您可以只恢复包含已受损的块的那些存储空间。如果存储空间跨越多个块，当其中一块发生介质故障时，必须终止该存储空间的所有活动事务，这样数据库服务器才可以恢复它。您可以在数据库服务器完成事务前启动恢复操作，但是该恢复将延迟，直到数据库服务器验证您已经完成故障发生时所有活动的事务时为止。

### 4.3.2 冷恢复、热恢复或混合恢复

当您恢复数据库服务器的数据时，必须确定是在数据库服务器脱机还是联机状态下进行。该决定部分取决于您打算恢复的数据。

#### 冷恢复

当数据库服务器处于脱机状态时执行冷恢复。它由物理恢复和逻辑恢复组成。要恢复所有关键数据库空间必须执行冷恢复。

开始冷恢复时数据库服务器处于脱机状态，但当它恢复保留页后将进入恢复方式。从这时起，它一直停留在恢复方式直到逻辑恢复完成为止（之后它工作在停顿方式下），或者使用 `gadmin` 实用程序将其转换为另一种方式。

可以在冷恢复期间指定新块路径和偏移量来重命名块。如果需要将存储空间恢复到与完成备份的磁盘不同的磁盘上，该选项很有帮助。可以重命名任意类型的块，包括关键块和镜像块。有关更多信息，请参阅恢复期间重命名块。您还可以对外部冷恢复的块进行重命名，请参阅重命名块以获取更多信息。

在已重命名数据库空间，并且执行了 0 级备份或 `rootdbs` 和重命名数据库空间的备份之后，可以执行冷恢复。

### 热恢复

热恢复在数据库服务器处于联机状态或静默方式时恢复非关键存储空间。它由一个或多个物理恢复操作（同时恢复多个存储空间时）、一个逻辑日志备份和一个逻辑恢复组成。

在热恢复期间，数据库服务器为您恢复的存储空间重放已备份的逻辑日志文件。为避免覆盖当前逻辑日志，数据库服务器将您指定要重放的逻辑日志文件写入到临时空间中。因此，热恢复要求有足够的临时空间，用来保存逻辑日志或被重放的日志文件的数目这两者中较小的那个。有关数据库服务器如何查找临时空间的信息，请参阅《GBase 8s 管理员指南》中 `DBSPACETEMP` 的讨论。

**重要：** 确保有足够的临时空间用于热恢复的逻辑日志部分；数据库服务器需要的最大临时空间量等于所有逻辑日志文件的大小。

在已重命名数据库空间，并且执行了 `rootdbs` 和重命名数据库空间的 0 级归档之后，可以执行热恢复。

### 混合恢复

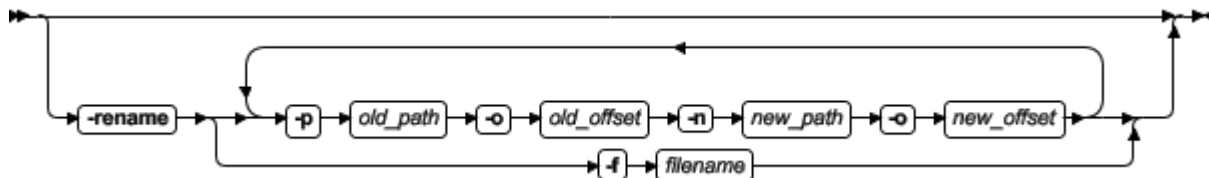
混合恢复是在冷恢复后再进行热恢复。混合恢复在冷恢复期间恢复一些存储空间（数据库服务器处于脱机状态），然后在热恢复期间再恢复一些存储空间（数据库服务器处于联机状态）。可以在执行整个系统恢复时进行混合恢复，但需尽快提供对特殊表或表集合的访问权。在这种情况下，执行冷恢复可以恢复关键数据库空间和包含重要表的数据库空间。

尽管在混合恢复的一部分时间里数据库服务器处于联机状态，但冷恢复恢复所有数据使用的总时间比混合恢复少，这是因为混合恢复需要两个逻辑恢复（一个用于冷恢复，另一个用于热恢复）。不过，混合恢复要求数据库服务器处于脱机状态的时间比冷恢复少。

对于冷恢复中未恢复的数据库空间，即使关键数据库空间可能并未损坏它们，它们也必须等到数据库服务器在热恢复中恢复了它们才可以使用。

## 4.3.3 gtape 实用程序执行恢复的语法

将 `-r` 选项和 `gtape` 一起使用对数据库服务器数据执行完全物理恢复和逻辑恢复。使用 `-D` 选项恢复选定的存储空间。使用 `-rename` 选项在恢复期间重命名块。



元素	用途	关键注意事项
<code>-C</code>	从当前逻辑日志磁带中恢复日志，同时不发送安装磁带的提示。	服务器处于暂挂日志恢复状态，在最后一个可用日志恢复后，该命令依然存在。如果日志存在于多个磁带中，那么服务器将发出提示。
<code>-D</code>	指示 <code>gtape</code> 只恢复您指定的存储空间。	数据库服务器必须处于联机状态或停顿方式才能执行热恢复。当您使用 <code>-D</code> 选项时，可以恢复选定的存储空间。  如果不指定 <code>-D</code> 选项，那么 <code>gtape</code> 将执行整个系统恢复。数据库服务器必须处于脱机状态才能执行整个系统恢复。有关更多信息，请参阅 <a href="#">恢复选定的存储空间</a> 。
<code>dbspace</code>	要恢复的存储空间的名称。	可以指定多个存储空间，但是必须包括根数据库空间。
<code>-e</code>	指示 <code>gtape</code> 执行外部恢复	有关更多信息，请参阅 <a href="#">执行外部备份与恢复</a> 。  该选项与外部冷恢复的其余块相兼容。
<code>-f filename</code>	指定一个文件，该文件包含要重命名的块的名称和偏移量，以及它们的新	文件名可以是任何有效的 UNIX™ 文件名，包括简单文件名 ( <code>listfile_1</code> )、相对文件名

元素	用途	关键注意事项
	位置。用于同时为很多块重命名。	<p>(../backup_lists/listfile_2 或 ..\backup_lists\listfile2) 以及绝对文件名</p> <p>(/usr/gbasedbt/backup_lists/listfile3 或 c:\gbasedbt\backup_lists\listfile3)。</p> <p>文件中列出旧块路径名和偏移量以及新块新路径名和偏移量，每项以空格或制表符分隔。将每个块的信息置于单独的行上。忽略空白行。注释行以 # 号开始。</p>
-l	指示 gtape 执行逻辑恢复。	-l 选项从上次 0 级备份以后创建的逻辑日志备份磁带中恢复数据（含上次 0 级备份）。
-p	指示 gtape 执行物理数据恢复。	-p 选项从上次 0 级备份以后创建的备份磁带中恢复数据（含上次 0 级备份）。恢复期间，数据库服务器处于单用户方式。
-p old_path -o old_offset-new_path -o new_offset	指定要重命名的块及其新位置。用于同时重命名一个或多个块。	<p>用于该元素的参数是：</p> <p><b>old_path</b> 块的当前路径和文件名。</p> <p><b>old_offset</b> 块的当前偏移量（以千字节计）。</p> <p><b>new_path</b> 块的新路径和文件名。</p> <p><b>new_offset</b> 块的新偏移量。</p>
-r	指示 gtape 执行数据	-r 选项将从上次 0 级备份以后（并

元素	用途	关键注意事项
	恢复（物理和逻辑恢复）。	包含）创建的备份磁带和逻辑日志备份磁带中恢复数据。
-rename	指示 gtape 重命名指定的块。	有关在恢复期间重命名块的更多信息，请参阅 <a href="#">恢复期间重命名块</a> 。
-S	指示 gtape 执行逻辑日志回收。	如果想要回收逻辑日志，那么在执行从标准输入恢复之前必须使用 -S 选项。LTAPEDEV 配置参数必须设置为逻辑日志磁带设备。
-t STDIO	指示 gtape 从标准输入恢复。	-t 选项为当前恢复覆盖 TAPEDEV 配置参数的值。
-v	指示 gtape 在从标准输入恢复期间向 stderr 写入参考消息。	详细模式对于监视从标准输入恢复的进度很有用。
-X	使服务器停顿在逻辑恢复暂挂状态，但不恢复其他日志。	包括带有 -r -l 的该选项，以结束逻辑日志的连续日志恢复。

<sup>1</sup> 请参阅 《GBase 8s 管理员参考》中的 -FILE 选项。

#### 4.3.4 恢复整个系统

本部分概述了使用 gtape 恢复整个数据库服务器之前所需完成的先决条件和步骤。

以下列表汇总了完全系统恢复中的主要步骤：

1. 收集适当的备份和逻辑日志磁带。
2. 决定是执行完整的冷恢复还是混合恢复。
3. 验证您的数据库服务器的配置。
4. 执行冷恢复。

尝试整个系统恢复之前请先熟悉这些指示信息。

#### 在恢复前收集备份和逻辑日志磁带

在恢复整个数据库系统前，您必须收集备份和逻辑日志磁带。如果您更改了备份和逻辑日志文件的名称，那么还必须手动将这些文件重命名为其原始文件名。

### 备份磁带

从包含要恢复的存储空间的最新 0 级备份以及任何后续 1 级或 2 级备份中收集所有磁带。

标识出其中有根数据库空间最近一次 0 级备份的磁带；必须先使用该磁带。

### 逻辑日志磁带

如果在归档检查点时启动了一个打开的事务，请在执行 0 级备份前收集所有逻辑日志磁带。

从要恢复的存储空间的最新 0 级备份后的备份中，收集所有逻辑日志磁带。

当使用 `gtape` 创建系统的归档备份时，该归档中会包含逻辑日志的快照。在归档结束时，系统将显示一条消息以指示归档中包含哪些逻辑日志。归档中包含该快照，这样如果备份时存在任何打开的事务，这些事务可以在恢复归档时进行调整。然后：

- 如果您决定不重放任何逻辑日志，系统即可进入一致的状态。
- 如果您决定重放逻辑日志，那么将放弃归档备份中包含的日志，然后您必须从逻辑日志备份重放事务。

起始日志文件是最旧的日志文件，其中包含上次恢复归档时的打开事务。您可以从上次恢复归档时显示的消息识别该日志文件。

### 示例：

- `gtape -s -L 0` 命令执行系统的 0 级备份，并显示消息以说明归档包含日志 2 到 4。
- `gtape -s -L 1` 命令执行系统的 1 级增量备份，并显示消息以说明归档包含日志 8 到 9。

如果您只恢复 0 级归档并且希望重放日志，那么需要从日志 2 开始的日志备份。如果恢复 0 级和 1 级归档并且希望重放日志，那么需要从日志 8 开始的日志备份。

逻辑日志文件的恢复使用归档格式，而不是日志文件格式。但是，恢复的归档中包含的日志使用日志文件格式，而不是归档格式。

从目录恢复时的文件名

从文件系统目录恢复时，`gtape` 要求如 `TAPEDEV` 和 `LTAPEDEV` 配置参数所指定那样命名存储空间归档和逻辑日志备份文件。如果由于重复的归档和备份而重命名了文件（包括由 `gtape` 重命名），那么必须手动将这些文件重命名为其原始文件名。要了解存储空间归档文件和逻辑日志备份文件的命名约定，请参阅备份到目录以获取这些文件的命名约定。

### 决定是执行完整的冷恢复还是混合恢复

如冷恢复、热恢复或混合恢复中提到的，如果是恢复整个数据库服务器，可以在冷恢复期间恢复关键数据库空间（以及需要迅速联机的任何其他存储空间），接着在热恢复期间恢复剩余的存储空间。启动恢复之前确定是要进行冷恢复还是混合恢复。

### 验证数据库服务器的配置

在冷恢复期间，不能设置共享内存、添加块或更换磁带设备。因此，开始恢复时，当前数据库服务器的配置必须保持与最近一次备份后指定的所有参数值一致，并能接受这些值。

有关指导信息，请使用每次备份时创建的配置文件的副本。但是，不要将当前所有参数设置为与最后一次备份时记录的值相同的那些值。请注意以下三组参数：

- 共享内存参数
- 镜像参数
- 设备参数

### 将共享内存参数设置为指定的最大值

确保将当前共享内存参数设置为 0 级备份后指定的最大值。例如：如果您在 0 级备份后的某个时间将 `USERTHREADS` 的值从 45 减少到 30，那么开始恢复时必须将 `USERTHREADS` 设置为 45 而不是 30，即使最近一次备份的配置文件副本可能注册了设置为 30 的 `USERTHREADS` 值。（如果在 0 级备份后不处理 `USERTHREADS` 的最大值的记录，请将该值设置为您需要的最高值。也可以为 `BUFFERPOOL`、`LOCKS` 和 `TBLSPACES` 重新指定值，因为这三个参数的最小值都是基于 `USERTHREADS` 的值。）

### 将镜像配置设置为 0 级备份状态

请验证当前的镜像配置是否与最近的 0 级备份时的有效配置相匹配。由于建议您在每次更改镜像配置后都创建 0 级备份，因此这不会产生任何问题。最关键的参数是出现在配置文件中的镜像参数 `MIRRORPATH` 和 `MIRROROFFSET`。

### 验证原始设备或文件是否可用

验证 0 级备份后用于存储（属于要恢复的存储空间）的原始设备或文件是否可用。

例如：如果在 0 级备份后删除了数据库空间或数据库空间的镜像，那么必须在开始恢复时确保此数据库空间或镜像块设备对于数据库服务器是可用的。如果数据库服务器试图写入块中但找不到该块，那么恢复无法完成。同样，如果自上次备份后添加了块，那么必须在数据库服务器开始前滚逻辑日志时，让块备份对于数据库服务器可用。

## 执行冷恢复

要执行冷恢复，数据库服务器必须是脱机的。

要使用 `gtape`，必须以 `gbasedbt` 用户 或 `root` 登录。执行以下 `gtape` 命令可恢复所有存储空间：`gtape -r`

如果执行混合恢复，那么在冷恢复期间只恢复部分存储空间。必须至少恢复所有的关键数据库空间，如以下示例所示：

```
gtape -r -D rootdbs llogdbs plogdbs
```

## 回收逻辑日志文件

冷恢复开始前，控制台提示您回收磁盘上的逻辑日志文件。要回收逻辑日志文件，请使用新磁带。它将保存您未备份的日志记录，并使您可以将数据库服务器的数据恢复到故障发生前的情形。

下面显示了一个日志回收示例：

```
...
继续恢复? (y/n) y
是否要备份日志? (y/n) y

请在 /dev/ltapedev 上安装磁带 1 并按“返回”以继续。
希望备份日志 31 - 32 中的任意一个吗? (y/n) y
逻辑日志 31 - 32 可能已备份。
请输入您希望备份的最旧的日志的标识? 31

请将该磁带标记为日志磁带序列中的编号 1。

该磁带包含以下逻辑日志：
    31-32
日志回收已完成，继续归档的恢复。
恢复 1 级归档 (y/n) y
准备 1 级磁带
...

```

## 恢复期间安装磁带

在冷恢复期间，`gtape` 将提示您安装带有适当备份文件的磁带。



从目录恢复时，提示中会指定目录的绝对路径名。响应提示前，您可以复制或重命名目录中的文件。

使用 `gtape -d` 选项可以避免出现提示。使用此选项时，请确保存储空间归档和逻辑日志备份文件存在于 `TAPEDEV` 和 `LTAPEDEV` 参数指定的目录中。`gtape` 实用程序会扫描这些目录，获取目录中的文件并将这些文件用于恢复。恢复适用的最新逻辑日志备份文件后，`gtape` 会自动落实恢复并将 GBase 8s 实例更改为静默方式。

#### 恢复逻辑日志文件

执行混合恢复时，必须恢复自上次 0 级备份以后备份的所有逻辑日志文件。

执行完全恢复时，可以选择不恢复逻辑日志文件。如果不备份逻辑日志文件或选择不恢复它们，那么只能将数据恢复到最近一次进行备份时所处的状态。有关更多信息，请参阅使用 `gtape` 备份逻辑日志文件。

要恢复逻辑日志，请使用 `gtape -l` 命令。

#### 恢复完成时使数据库服务器处于联机状态

冷恢复结束后，数据库服务器处于停顿方式。可以让数据库服务器处于联机状态并继续常规处理。

如果在冷恢复期间仅恢复部分存储空间，那么可以在数据库服务器处于联机状态后对其余的存储空间启动热恢复。

#### 恢复选定的存储空间

这些主题概述当数据库服务器处于联机状态或静默方式时，使用 `gtape` 恢复选定存储空间（热恢复）过程中必须执行的步骤。在热恢复期间，与冷恢复不同的是不需要担心共享内存参数。

尝试恢复前，请先熟悉这些指示信息。

以下列表描述热恢复中采取的主要步骤：

1. 收集适当的磁带
2. 确保需要的设备可用
3. 备份逻辑日志文件
4. 执行热恢复

#### 收集适当的磁带

收集适当的备份和逻辑日志磁带。

## 备份磁带

开始恢复之前，从包含要恢复的存储空间的最近 0 级备份以及所有随后的 1 级或 2 级备份中收集所有磁带。

## 逻辑日志磁带

从要恢复的存储空间的最近一次 0 级备份后的逻辑日志备份中，收集所有逻辑日志磁带。

## 确保需要的设备可用

开始恢复前请验证存储设备和文件可用。例如：当您在 0 级备份后删除数据库空间或数据库空间的镜像时，必须确保开始恢复时数据库空间或镜像块设备对于数据库服务器是可用的。如果存储设备不可用，那么数据库服务器无法写入块中，恢复将失败。

在上次备份后添加块时，必须确保数据库服务器前滚逻辑日志时，块备份对于数据库服务器是可用的。

## 备份逻辑日志文件

开始热恢复前（即使是作为混合恢复的一部分执行热恢复），必须备份逻辑日志文件。请参阅使用 `gtape` 备份逻辑日志文件。

执行热恢复后，必须前滚逻辑日志文件，使您要恢复的数据库空间与系统中其他数据库空间的状态一致。恢复选定的数据库空间后未能前滚逻辑日志，那么 `gtape` 将生成以下消息：

```
部分系统恢复未完成。
```

## 执行热恢复

要执行热恢复，数据库服务器必须在联机状态或停顿方式下运行。

要使用 `gtape`，必须以 `gbasedbt` 用户 或 `root` 登录。要恢复选定的存储空间，请执行 `gtape` 命令，并使用以下示例显示的选项：

```
gtape -r -D dbspace1 dbspace2
```

无法在热恢复期间恢复关键数据库空间；必须将其作为冷恢复的一部分进行恢复，如恢复整个系统中所述。

在恢复期间，`gtape` 将提示您安装带有适当备份文件的磁带。

热恢复结束时，原来关闭的存储空间又处于联机状态。

## 恢复原始表

使用 `gtape` 恢复原始表时，该表中只包含上次备份时磁盘上存在的数据。由于原始表不记录日志，因此无法恢复自上次备份以来发生的所有更改。要了解更多信息，请参阅备份原始表和《GBase 8s 管理员指南》。

### 4.3.5 使用 `gtape` 配置连续日志恢复

确保主系统和辅助系统上的 GBase 8s 版本相同。

所有当前可用日志恢复后，请使用连续日志恢复来重新启动带有新可用日志的日志恢复。有关更多信息，请参阅连续日志恢复。

要使用 `gtape` 配置连续的日志恢复：

1. 在主系统上，使用 `gtape -s -L 0` 命令执行 0 级归档。
2. 在辅助系统上，复制文件或安装磁带（由 `LTAPEDEV` 分配），然后使用 `gtape -p` 命令执行物理恢复。
3. 对以下提示作出响应：

```
继续恢复? Y
```

```
是否要备份日志? N是否要恢复 1 级归档? N
```

物理恢复完成后，数据库实例等待快速恢复方式来恢复逻辑日志。

4. 在主系统上，使用 `gtape -a` 命令备份逻辑日志。
5. 在辅助系统中，复制文件或安装包含主系统中备份逻辑日志的磁带。使用 `gtape -l -C` 命令执行逻辑日志恢复。
6. 对可用于备份与恢复的所有逻辑日志重复步骤 4 和 5。
7. 如果您要在紧急备用的辅助系统上执行连续日志恢复，请运行以下命令来完成逻辑日志恢复并停顿服务器
  - 。 如果逻辑日志可用于恢复，请使用 `gtape -l` 命令。
  - 。 在恢复所有可用逻辑日志之后，使用 `gtape -l -X` 命令。

### 4.3.6 恢复期间重命名块

可以在冷恢复期间使用 `gtape` 重命名块。如果需要将存储空间恢复到与完成备份的磁盘不同的磁盘上，该选项很有帮助。可以重命名任意类型的块，包括关键块和镜像块。

`gtape` 重命名块恢复仅对冷恢复可用。

冷恢复时必须恢复关键的数据库空间（如 `rootdbs`）。如果不指定要恢复的数据库空间列表，那么服务器会恢复关键的数据库空间以及其他所有数据库空间。但如果指定了要恢复的数据库空间列表，那么其中必须包括关键的数据库空间。

有关使用 `gtape` 重命名块的语法，请参阅 `gtape` 实用程序执行恢复的语法。

**提示：** 如果使用到块名称的符号链接，可以不需要重命名块；而只需编辑符号名称定义即可。有关更多信息，请参阅《GBase 8s 管理员指南》。

您可以在外部冷恢复的过程中对块进行重命名。有关更多信息，请参阅重命名块。

### 重命名块的验证顺序

在冷恢复期间，`gtape` 执行以下验证来重命名块：

- 验证在归档保留页中是否存在旧块的路径名和偏移量。
- 验证各个新块之间或新块与现有块之间的路径名与偏移量是否相互重叠。
- 如果对主根块或镜像根块进行重命名，那么将更新 `onconfig` 文件参数 `ROOTPATH` 和 `ROOTOFFSET`，或 `MIRRORPATH` 和 `MIRROROFFSET`。旧版本的 `onconfig` 文件将另存为 `$ONCONFIG.localtime`。
- 将旧块中的数据恢复到新块（如果新块存在）中。
- 将每个块的重命名信息写入到联机日志中。

如果任意一个验证步骤失败，剩余的进程停止并且 `gtape` 将一条错误消息写入到 `gtape` 活动日志中。

### 重要：

- 重命名块后请执行 0 级归档；否则下次恢复将失败。
- 如果在执行 0 级归档后添加了块，那么恢复期间无法重命名该块。同时也不能安全地在映射列表中将该块指定为新路径。
- 为参与 HDR 的数据库服务器重命名块的操作中，两个数据库服务器都要长时间地脱机。有关更多信息，请参阅《GBase 8s 管理员指南》。

### 新块需求

要重命名块，请遵循以下新块的准则：

- 新块不是必须存在的

可以以后安装新块并对包含它的存储空间执行热恢复。如果指定不存在的块，`gtape` 将重命名信息记录在块的保留页中，但不恢复数据。已重命名（但未恢复）的块处于脱机状态，在 `gstat -d` 块状态命令输出中由 `D` 指示。

- 新块必须有适当的权限。

除非新块有适当的权限，否则重命名操作将失败。有关更多信息，请参阅《GBase 8s 管理员指南》。

### 使用命令行选项重命名块

要通过提供命令行的信息来重命名块，请使用此命令：

```
gtape -r -rename -p /chunk1 -o 0 -n /chunk1N -o 20000  
-rename -p /chunk2 -o 10000 -n /chunk2N -o 0
```

在重命名和恢复操作完成后执行 0 级归档。

### 使用文件重命名块

要通过提供名为 listfile 的文件来重命名块，请使用以下命令：gtape -r -rename -f listfile

listfile 文件的内容是：

```
/chunk1 0 /chunk1N 20000  
/chunk2 10000 /chunk2N 0
```

在重命名和恢复操作完成后执行 0 级归档。

### 指定其他选项时重命名块

要在 dbspace1 和 dbspace2 上执行恢复时使用命令行选项来重命名块，并且其中 rootdbs 是 rootdbs 的名称，请使用以下命令：

```
gtape -r -rename -p /chunk1 -o 0 -n /chunk1N -o 20000  
-rename -p /chunk2 -o 10000 -n /chunk2N -o 0  
-D rootdbs dbspace1 dbspace2
```

此外，要在 dbspace1 和 dbspace2 上执行恢复时使用文件重命名块，请使用以下命令：

```
gtape -r -rename -f listfile -D rootdbs dbspace1 dbspace2
```

在重命名和恢复操作完成后执行 0 级归档。

### 将块重命名到不存在的设备上

要将块重命名到还不存在的设备上，首先指定新的路径名，但直到您安装了该物理设备后，才能恢复它的存储空间。该选项在您需要重命名块时很有用，便于您在安装新设备前执行冷恢复。当新块设备就绪后，您可以在它上面执行存储空间的热恢复。

可以在同一个重命名操作中将重命名块与现有设备结合在一起，以及将重命名块与不存在的设备结合在一起。本示例显示如何将单个块重命名到不存在的设备名上。

下表列出本示例中使用的块的示例值。

存储空间	旧块的路径	旧偏移	新块的路径	新偏移量
------	-------	-----	-------	------

		量		
sbspace1	/chunk3	0	/chunk3N	0

#### 将块重命名到不存在的设备上

要将块重命名到不存在的设备上：

1. 重命名块：使用以下命令：`gtape -r -rename -p /chunk3 -o 0 -n /chunk3N -o 0`
2. 当以下提示出现时，请输入 `y` 以继续：

块 `/chunk3N` 不存在。如果继续，那么对包含该块  
的数据库空间的恢复操作稍后可能会失败。  
在不创建该块的情况下，是否继续？(y/n)

块 `/chunk3` 被重命名为 `/chunk3N`，但数据还未恢复到 `/chunk3N` 上。

3. 执行 0 级归档。
4. 为 `/chunk3N` 添加物理设备。
5. 使用 `gtape -r -D sbspace1` 命令来执行 `sbspace1` 的热恢复。
6. 执行 0 级归档。

### 4.3.7 从标准输入恢复

可以从标准输入执行恢复，必须首先已经执行到标准输出的备份。

当您从标准输入恢复时，`gtape` 不会提示您提供选项或信息。如果 `gtape` 不能根据您在恢复命令中提供的信息执行该操作，`gtape` 将退出并生成相应的错误。从标准输入中恢复和从磁带恢复在以下几个方面不同：

- 不发生逻辑恢复或逻辑日志回收。  
要执行逻辑恢复，在物理恢复后使用 `gtape -l` 命令。  
要回收逻辑日志，请在物理恢复之前使用 `gtape -S` 命令。
- 不会提示您确认恢复操作。关于归档的参考消息被送到 `stderr`。  
如果您发现有问題，您可以在归档信息完成而数据库服务器尚未启动的 10 秒延迟期间中断恢复。

示例

在以下示例中，`gtape` 从文件 `level_0_archive` 执行了物理恢复，该文件中包含上次执行的到标准输出的归档。

```
cat /home/level_0_archive | gtape -p
```

在以下示例中，gtape 执行了 0 级归档的恢复，随后是 1 级归档的恢复：

```
cat /home/level_0_archive /home/level_1_archive | gtape -r
```

在以下示例中，gtape 执行了 sbspace1 的恢复：

```
cat/home/level_0_archive | gtape -r -D spspace1 -t STUDIO
```

当这些恢复完成时，数据库服务器处于单用户方式。

### 4.3.8 将数据恢复到远程服务器

可以用以下命令将数据恢复到远程服务器：

```
gtape -s -L 0 -F | rsh remote_server "gtape -p"
```

但在成功完成后，过程可能会挂起。您有三个主要选项：

- 终止远程 shell 过程
- 用以下命令执行远程服务器中的远程 shell：

```
rsh local_server "gtape -s -L 0 -F" | gtape -p
```

- 使用 sh 或 bash shell 中的以下命令对远程服务器上的标准输出 (stdout) 和标准错误 (stderr) 进行重定向：

```
gtape -p >/dev/null 2>&1
```

- 通过将上述命令放置在远程服务器上的 shell 脚本 (gtape.sh) 中，您可以简化此重定向。可以从本地服务器发布以下命令：

```
gtape -s -L 0 -F | rsh remote_server /my/path/gtape.sh
```

- shell 脚本 gtape.sh 包含以下文本：

```
#!/bin/sh
#define some Gbasedbt environment variables, such as

GBS_HOME=/... ; export GBS_HOME
GBS_DATA=/... ; export GBS_DATA
PATH=/...; export PATH
# 用 stdout/stderr redirection 调用 gtape

gtape -p >/dev/null 2>&1
```

### 4.3.9 使用标准 I/O 同时备份与恢复

为了克隆数据库服务器或快速建立高可用性数据复制 (HDR)，您可以同时执行向标准输出的备份和从标准输入的恢复。如果您执行备份与恢复只是为了复制数据库服务器，使用 `-F` 选项以避免保存归档。

在 HDR 上，辅助服务器只能恢复 0 级的归档。

要使用标准 I/O 来执行备份与恢复，请将 TAPEDEV 配置参数设置为 `STDIO`，也可以从命令行指定 `-t STDIO`。

例如，如果 TAPEDEV 配置参数设置为 `STDIO`，以下命令会将数据装入一个 HDR 对上的辅助服务器（名为 `secondary_host`）。

```
gtape -s -L 0 -F | rsh secondary_host "gtape -p"
```

在下一个示例中，假设未设置 TAPEDEV 配置参数。以下命令将数据装入一个 HDR 对的辅助服务器（名为 `secondary_host`）：

```
gtape -s -L 0 -F -t STDIO | rsh secondary_host "gtape -t STDIO -p"
```

这些示例在本地计算机上执行数据库服务器的伪 0 级归档，使用 `rsh` 系统实用程序通过管道将数据传输到远程计算机，并直接从管道中读取数据从而在远程计算机上执行物理恢复。

## 4.4 执行外部备份与恢复

这些主题将讨论如何执行外部备份以及通过使用 `gtape` 实用程序将其恢复来恢复数据。

### 4.4.1 使用外部备份和恢复来恢复数据

您可以执行外部备份与恢复，以消除系统的停机时间，因为备份与恢复操作都是在 GBase 8s 系统外部执行的。

在备份或物理恢复期间 `gtape` 实用程序不移动数据。外部备份使您不必使用 `gtape` 就可以复制包含存储空间块的磁盘。当磁盘发生故障时，将其更换并使用供应商软件来恢复数据，然后使用 `gtape` 进行逻辑恢复。有关更多信息，请参阅在外部恢复中恢复的数据。

以下是外部备份与恢复的典型情况：

- 磁盘镜像的可用性

如果您使用硬件磁盘镜像，那么与使用常规的 `gtape` 命令相比，使用外部备份与恢复方法可以使系统更快地联机。

- 克隆

可以使用外部备份与恢复来克隆现有的生产系统，以在不干扰生产系统的情况下对其进行测试或迁移。



## 4.4.2 在外部备份中备份的数据

开始外部备份之前，需要阻塞数据库服务器。阻塞将强制设置检查点，清空缓冲区、将其中的内容都保存到磁盘，并阻塞包含临时表的用户事务。在阻塞操作期间，用户可采用只读方式访问数据库服务器。接着您可以使用操作系统或第三方的工具以物理方式将数据备份或复制到另一组磁盘或存储介质上。完成外部备份后，取消阻塞数据库服务器以使事务能继续进行。在外部备份中应包含每个存储空间中的所有块文件以及管理文件（例如 onconfig）。

**重要：** 要使跟踪备份更加容易，建议您在每个外部备份中备份所有存储空间。

gtape 实用程序将外部备份视为 0 级备份。您不能在执行外部备份后使用 gtape 执行 1 级备份，反之亦然，因为 gtape 没有外部备份的任何记录。有关更多信息，请参阅执行外部冷恢复。

### 外部备份的规则

开始外部备份之前，记住以下规则：

- 在执行外部备份期间，数据库服务器必须处于联机或停顿方式下。
- 用 gtape 备份包括当前日志在内的所有逻辑日志，这样您就可以在外部恢复结束时恢复逻辑日志。
- 阻塞数据库服务器以进行外部备份之前请暂挂连续逻辑日志备份。外部备份完成后，恢复连续逻辑日志备份。
- 等到所有 gtape 备份会话完成后再阻塞数据库服务器。如果有任何备份会话是活动的，阻塞命令将显示错误消息。
- 当数据库服务器阻塞时，所有 OLTP 工作或查询都将暂挂。直到数据库服务器取消阻塞后它们才会继续进行。
- 数据库服务器实例中所有关键的数据库空间必须使用同一个 gadmin -c block ... gadmin -c unblock 命令编组一同进行备份。在不同时间对不同的关键数据库空间所做的备份无法恢复到一致的系统中。

**重要：** 由于外部备份不受 gtape 的控制，因此您必须手工跟踪这些备份。有关更多信息，请参阅跟踪外部备份。

### 执行外部备份

外部备份期间数据库服务器必须处于联机状态或停顿方式。

要在不使用磁盘镜像的情况下执行外部备份：

1. 要获取外部备份，请使用 gadmin -c block 命令来阻塞数据库服务器。  
系统将获取一个检查点并暂挂所有更新事务。用户能以只读方式访问数据库服务器。

- 要备份存储空间和管理文件，请使用复制命令（如 UNIX™ 上的 cp、dd 或 tar，）或文件备份程序。

必须备份存储空间中的所有块。

- 要允许正常操作继续，请使用 `gadmin -c unblock` 命令来取消阻塞数据库服务器。
- 备份包括当前日志在内的所有逻辑日志，这样就可以将检查点信息用于外部恢复。

**重要：** 由于外部备份并不是通过 `gtape` 完成的，因此必须确保具有从执行 `gadmin -c block` 命令时开始的当前逻辑日志的备份。没有该逻辑日志文件的备份，外部备份将是不可恢复的。

- 执行外部备份后，请备份当前日志。使用 `gtape -a` 命令。

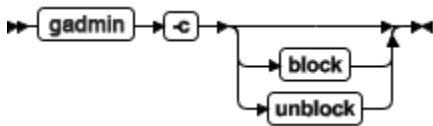
如果您丢失磁盘或整个系统，现在便可以执行外部恢复。

### 4.4.3 准备进行外部备份

这些主题描述了用于准备外部备份的命令。有关过程的信息，请参阅执行外部备份。

#### 阻塞和取消阻塞 GBase 8s

本部分介绍 `block` 和 `unblock` 命令的语法。



元素	用途	关键注意事项
-c	执行检查点并阻塞或取消阻塞数据库服务器	无。
block	阻止数据库服务器执行任何事务	设置数据库服务器以进行外部备份。当数据库服务器被阻塞时，用户可采用只读方式进行访问。样本命令： <code>gadmin -c block</code>
unblock	取消阻塞数据库服务器，从而允许数据事务和正常的数据库服务器操作可以继续进行	在外部备份完成前请勿取消阻塞。样本命令： <code>gadmin -c unblock</code>

#### 跟踪外部备份

数据库服务器和 `gtape` 不会跟踪外部备份。要跟踪外部备份数据，请使用第三方存储管理器或手动跟踪数据。下表显示了建议您在外部分备份中跟踪的项。

表 1. 使用外部备份与恢复时要跟踪的项

要跟踪的项	示例
每个已备份存储空间的各个块文件的完整路径名	UNIX™: /work/dbspaces/rootdbs
对象类型	关键数据库空间、非关键存储空间
ins_copyid_hi 和 ins_copyid_lo	存储管理器分配给每个备份对象的副本标识
备份日期和时间	阻塞和取消阻塞数据库服务器的时间
备份介质	磁带卷编号或磁盘路径名
数据库服务器版本	V8.8

#### 4.4.4 在外部恢复中恢复的数据

仅当您在外部备份数据后，才能够在丢失磁盘或整个系统的情况下从外部进行恢复。必须对外部备份与恢复使用相同的第三方实用程序。要从外部恢复存储空间，请将已备份的数据复制到磁盘。使用 `gtape -p -e` 命令将存储空间标记为已物理恢复，通过 `gtape -l` 命令重放逻辑日志，并使存储空间重新联机。如果未指定外部恢复命令，那么数据库服务器无法将这些存储空间的状态更新为联机。

您只能使用 `gtape` 执行外部冷恢复。外部冷恢复将存储空间标记为已物理恢复，然后对所有存储空间执行逻辑恢复。

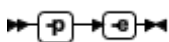
当执行外部冷恢复时，`gtape` 并不首先尝试从数据库服务器回收逻辑日志文件，因为外部备份已经复制了所有逻辑日志数据。

要回收逻辑日志，请在复制外部备份之前执行 `gtape -S`，然后执行外部恢复 (`gtape -p -e`)。

##### 使用外部恢复命令

`gtape -p -e` 命令用于执行外部冷恢复。该命令将存储空间标记为已物理恢复。下图显示外部物理恢复语法。

执行外部物理恢复



元素	用途	关键注意事项
-e	指定外部恢复	必须与 <code>-p</code> 选项一起使用。
-p	指定物理恢复。	在冷恢复中，如果未指定存储空间名称，那么所有存储空间都将标记为已恢复。物理恢复完成后，您必须执行逻辑恢复。

`gtape -l` 命令用于执行逻辑恢复。有关更多信息，请参阅 `gtape` 实用程序执行恢复的语法。

## 外部恢复的规则

开始外部恢复之前，请了解从外部备份可恢复以及无法恢复的内容，并注意外部恢复的规则。

这些需求和规则如下：

- 必须从外部备份进行外部恢复。尽管外部备份被视为 0 级备份，但它实际上可能是非 GBase 8s 增量备份。
- 不能从外部恢复临时数据库空间。
- 不能从常规 gtape 备份执行外部恢复。
- 如果使用 gtape，您无法验证是否在从正确的备份进行恢复以及存储介质是否可读。
- 如果多个外部备份在不同的时间进行，那么外部恢复将使用最早的备份中的开始逻辑日志。
- 应在切换包含关键存储空间的磁盘之前回收逻辑日志 (gtape -l)。
- 如果正在恢复关键数据库空间，那么数据库服务器必须处于脱机状态。
- 如果正在恢复 rootdbs，那么在恢复期间应禁用镜像。
- 数据库服务器实例的所有关键数据库空间的外部备份必须同时进行。必须在同一个 gadmin -c block ... gadmin -c unblock 命令编组中备份了所有关键的数据库空间。

## 重命名块

您可以使用其他恢复方法的重命名选项语法，对外部冷恢复中的块进行重命名。

在外部冷恢复过程中使用以下命令对块进行重命名：

```
gtape -p -e -rename -f  
filename
```

或

```
gtape -p -e -rename -p old_path -o old_offset -n new_path -o new_offset
```

## 执行外部冷恢复

如果在冷恢复过程中指定 gtape -p -e 命令，那么必须恢复所有存储空间。使用 gtape -p -e 命令来恢复所有存储空间。

要执行外部冷恢复：

1. 使用 gadmin -ky 命令来关闭数据库服务器。

2. 要从外部备份恢复存储空间，请使用复制命令，例如 UNIX™ 上的 `cp`、`dd` 或 `tar`，或使用文件备份程序。

您必须将存储空间恢复为与原始数据相同的路径。

3. 要对所有存储空间执行外部恢复，然后执行逻辑恢复，请使用以下命令：
  - o `gtape -p -e`
  - o `gtape -l`

#### 外部恢复命令的示例

下表包含外部恢复命令的示例。

外部恢复命令	操作	注释
<code>gtape -p -e</code> <code>gtape -l</code>	物理外部恢复和逻辑恢复	系统从最早的外部备份恢复逻辑日志。
<code>gtape -p -e -rename -f</code>	使用已重命名的块执行外部冷恢复	

#### 使用外部备份与恢复来初始化 HDR

您可以使用外部备份来初始化“高可用性数据复制”(HDR)。

要使用外部备份与恢复来初始化 HDR：

1. 使用 `gadmin -c block` 命令来阻塞源数据库服务器。
2. 在外部备份源数据库服务器上的所有块。
3. 备份完成时，使用 `gadmin -c unblock` 命令来取消阻塞源数据库服务器。
4. 使用以下命令来使源数据库服务器成为主服务器：`gadmin -d primary secondary_servername`
5. 在目标数据库服务器上，通过复制或文件备份程序从外部备份恢复数据。
6. 在目标数据库服务器上，使用 `gtape -p -e` 命令恢复所有块的外部备份。
7. 使用以下命令来使目标数据库服务器成为辅助服务器：`gadmin -d secondary primary_servername`
8. 如果从步骤 1 以来写入到主数据库服务器中的逻辑日志记录仍然驻留在主数据库服务器磁盘上，那么辅助数据库服务器将读取这些记录以执行逻辑恢复。否则，使用 `gtape -l` 命令来执行逻辑恢复。

数据库服务器可操作消息将显示在主服务器和辅助服务器上的消息日志中。

## 5 GBase 8s Primary Storage Manager

GBase 8s 主存储管理器 管理使用文件设备（磁盘）的 `gbackuprestore` 备份与恢复操作（包括并行备份）的存储。

### 5.1 GBase 8s Primary Storage Manager

GBase 8s 主存储管理器 是管理存储设备的应用程序，这些存储设备用于 `gbackuprestore` 发出的备份与恢复请求。该存储管理器支持备份与恢复请求的串行和并行两种处理。

GBase 8s Primary Storage Manager 由以下组件组成：

#### **gpsm 实用程序**

可用于执行以下任务的命令行实用程序：

- 创建、修改和删除存储设备
- 定义和修改设备的最大大小
- 将备份信息从设备池中的一个设备移至另一个设备
- 确定卷、存储对象和设备是否已锁定或繁忙
- 释放锁定的卷、存储对象和设备
- 验证卷名称和卷标

#### **XBSA 共享库**

`gbackuprestore` 和 GBase 8s Primary Storage Manager 用于相互通信的唯一版本的 X/Open 备份服务 API (XBSA) 共享库。当 `gbackuprestore` 存储或检索存储在存储设备上的数据时，存储管理器将通过设备级别的 XBSA 接口协调请求。使用 `BAR_BSALIB_PATH` 配置参数指定 XBSA 共享库的位置。

#### **存储目录表**

一组平面文件，用于跟踪有关所有存储对象、设备和设备池的信息。恢复 GBase 8s Primary Storage Manager 创建的备份对象时需要这些文件。缺省情况下，这些文件存储在 `$GBS_HOME/etc/psm` 目录中。您可以使用 `PSM_CATALOG_PATH` 配置参数来指定存储目录表的其他位置。

**重要：**

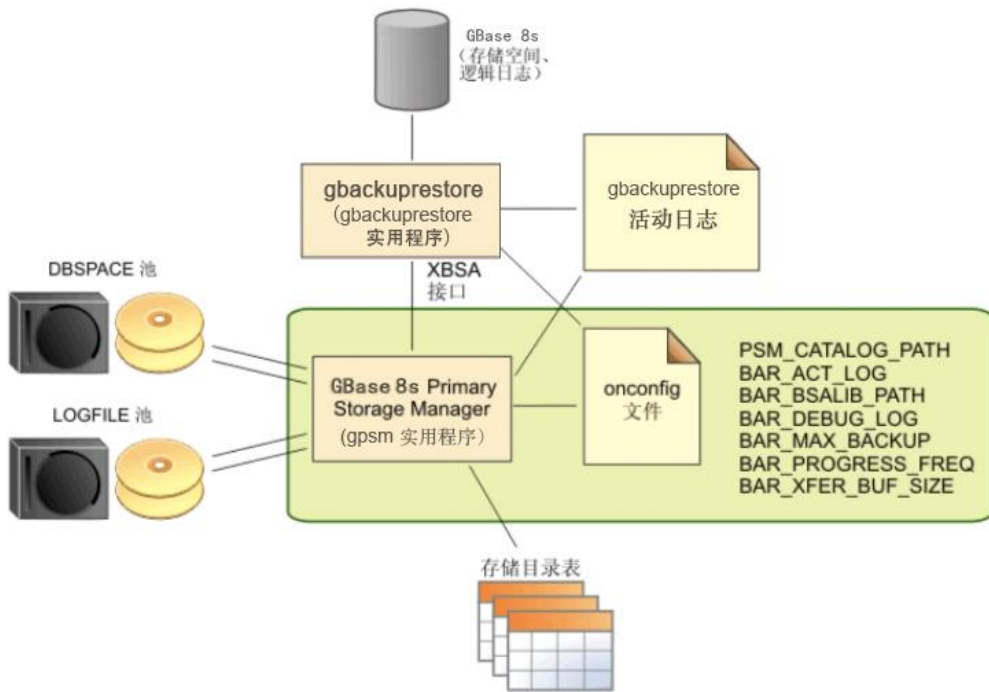
- 作为灾难恢复策略的一部分，使用您的操作系统工具来备份存储目录表。存储目录表并不是使用数据库实例备份的，也不与 GBase 8s 系统目录表关联。
- 要防止存储目录表变得过大，请定期删除旧的备份生成。使用 onsmsync 实用程序来管理到期策略。

用于配置 GBase 8s Primary Storage Manager 的配置参数位于 onconfig 文件中。

请使用 gpsm 命令行实用程序来定义和维护存储设备。可以一次配置一个设备，也可以生成一个设备配置文件来配置多个设备。在备份期间，GBase 8s Primary Storage Manager 会从可用设备池中选择一个设备。如果该设备变满或发生故障，那么存储管理器会自动移至相同池中的另一个设备。

GBase 8s Primary Storage Manager 将参考消息、警告和错误消息写入存储管理器活动日志。您可以使用 PSM\_ACT\_LOG 配置参数来指定活动日志的位置。如果 PSM\_ACT\_LOG 配置参数未包含信息，那么存储管理器会将活动信息放入通过 BAR\_ACT\_LOG 配置参数指定的目录中。

**图: GBase 8s Primary Storage Manager 的组件**



GBase 8s Primary Storage Manager 的功能部件

该表显示存储管理器的关键功能部件

存储管理器的 功能部件	解释
----------------	----

存储管理器的功能部件	解释
与存储管理器一起使用的存储设备	仅文件设备  创建目录时，存储管理器将自动创建缺省设备。缺省设备为 \$GBASEBTDIR/backups。您可以除去缺省设备。
缓冲区传输大小	无限
加密和压缩	使用 gbackuprestore 中的 BACKUP_FILTER、RESTORE_FILTER 过滤器完成（存储管理器不提供加密或压缩。）
存储管理器的到期策略	无到期策略。（请使用 onsmsync 实用程序从存储管理器手动使备份对象到期。onmsync 对象到期命令会从存储管理器中除去对象。）

可以使用 gbackuprestore 和 GBase 8s Primary Storage Manager 执行导入的恢复。在导入的恢复中，您将在一台机器上备份 GBase 8s 实例，而在另一台机器上恢复该实例。使用 onsmsync 导出和导入选项来从备份机器上的存储管理器导出备份对象，并将备份对象导入到恢复机器上的存储管理器中。

## 5.2 示例：使用 GBase 8s Primary Storage Manager 管理存储设备

了解如何设置并使用 GBase 8s Primary Storage Manager 来管理 gbackuprestore 实用程序用于备份与恢复实例的存储设备。每个示例显示如何针对特定备份策略使用存储管理器。

先决条件：

- GBase 8s 与 gbackuprestore 实用程序一起安装。
- 环境变量 GBS\_HOME 设置为数据库服务器的安装路径。
- 环境变量 GBS\_DATA 设置为数据库服务器实例的存储路径
- gbasedbt 或 root 用户特权。

示例 1：存储实例的备份

示例 2：存储两个实例的备份

示例 3：将备份导出到另一个目录并从其恢复备份



示例 4: 从一个服务器导出备份, 并将其导入到另一个服务器

在这些示例中, 存储管理器是指 GBase 8s Primary Storage Manager。

#### 示例 1: 存储实例的备份

本例显示如何设置并使用 GBase 8s Primary Storage Manager 来将单个数据库服务器实例的数据和逻辑日志备份到目录 \$GBS\_HOME/backups 中。

在本例中, 您将更新配置文件以使 GBase 8s Primary Storage Manager 能够与 gbackuprestore 通信, 并指定您希望存储备份的目录。然后, 使用 gbackuprestore 实用程序来执行所有联机存储空间和已用逻辑日志的标准 0 级备份。通过检查记录的消息并使用 gpsm 实用程序来确认已创建了存储对象, 从而验证该备份。

1. 将 BAR\_BSALIB\_PATH 配置参数设置为存储管理器的共享库的完整路径和名称。

例如, 在 Linux<sup>™</sup>、Solaris 上:

```
BAR_BSALIB_PATH $GBS_HOME/lib/libbsapsm.so
```

必须使用为 GBase 8s Primary Storage Manager 提供的 XBSA 共享库版本。如果您不使用 BAR\_BSALIB\_PATH 配置参数指定路径, 那么必须确保 XBSA 库位于您操作系统上的缺省位置中。

2. 如果需要, 请创建用于存储备份对象的目录。

缺省情况下, 存储管理器包含缺省池 LOGPOOL 和 DBSPOOL, 每个池中使用缺省目录 \$GBS\_HOME/backups。

- o 如果要使用缺省备份目录, 请验证 \$GBS\_HOME/backups 目录是否存在。
- o 如果要使用其他备份目录, 请使用 gpsm -D add 命令为 LOGPOOL 和 DBSPOOL 添加新的备份目录。例如, 运行以下命令为 LOGPOOL 和 DBSPOOL 池添加其他备份目录:

```
gpsm -D add /backups/infx/logs -g LOGPOOL -p HIGHEST -t FILE  
gpsm -D add /backups/infx/spaces -g DBSPOOL -p HIGHEST -t FILE
```

对于应该首先填充的设备, 请使用 HIGHEST 优先级。池中只能有一个设备的优先级设置可以为 HIGHEST。

3. 运行 gbackuprestore 实用程序来执行所有联机存储空间和已用逻辑日志的标准 0 级备份。

```
gbackuprestore -b -L 0
```

如果存储目录表不存在, 那么将在 \$GBS\_HOME/etc/psm 目录中进行创建。

4. 验证存储管理器是否已设置, 以及是否创建了备份对象。
  - a. 在 gbackuprestore 活动日志中查找以确认存储管理器已就绪且 gbackuprestore 可识别存储管理器。

例如，第一条消息来自存储管理器，第二条消息来自备份实用程序：

```
2012-01-03 15:51:23 11193 2569 Gbasedbt PSM is ready.
2012-01-03 15:51:23 11193 2569 Using Gbasedbt PSM version 12.10.FC1
as the Storage Manager. XBSA API version is 1.0.3.
```

缺省情况下，存储管理器会将消息发布至 gbackuprestore 活动日志。活动日志的位置由 BAR\_ACT\_LOG 配置参数设置。如果要分开记录存储管理器消息，必须设置 PSM\_ACT\_LOG 配置参数。

b. 运行 gpsm -O list 命令来列出已创建的存储对象：

列表（如以下示例中所示）包含存储对象标识、创建存储对象的日期、存储对象的大小，以及存储对象在存储设备中的位置。对象标识也存储在 ixbar 文件中，并由 gbackuprestore 用于查找对象。

```
=====
对象列表报告

                                逻辑路径
对象标识  创建日期                大小 (MB)  名称.版本（省略片段号）
-----
      1  2012-08-06 12:02:10        12.5  /serv1/rootdbs/0/serv1.1
      2  2012-08-06 12:02:12         0.1  /serv1/logdbs/0/serv1.1
      3  2012-08-06 12:02:12         0.1  /serv1/dbs2/0/serv1.1
      4  2012-08-06 12:02:12         0.1  /serv1/dbs1/0/serv1.1
      5  2012-08-06 12:02:13         0.1  /serv1/physdbs/0/serv1.1
      6  2012-08-06 12:02:14         0.3  /serv1/10/9/serv1.1
      7  2012-08-06 12:02:14         0.0  /serv1/crit_files/ixbar/serv1.1
      8  2012-08-06 12:02:14         0.0  /serv1/crit_files/oncfg/serv1.1
      9  2012-08-06 12:02:14         0.1  /serv1/crit_files/onconfig/serv1.1
     10  2012-08-06 12:02:14         0.0  /serv1/crit_files/sqlhosts/serv1.1
=====
```

c. 运行 gpsm -D list 命令来显示一个列表，其中显示该设备已添加到 DBSPOOL 和 LOGPOOL 池。以下示例显示该命令的输出：

```
类型  优先级    块/大小 (MB)  池名称  设备名---
FILE  HIGHEST    --/--        DBSPOOL  /backups/infx/logs

FILE  HIGHEST    --/--        LOGPOOL  /backups/infx/spaces
```

通过几个简单步骤，您已配置了存储管理器并执行了某个实例到文件设备的完全备份。所需的配置极少，因为存储管理器使用各种 `gbackuprestore` 配置参数的缺省设置。

GBase 8s 备份中不包含存储目录表。作为灾难恢复策略的一部分，请确保使用您的操作系统工具来备份存储目录表。如果存储目录表丢失，那么 `gbackuprestore` 实用程序无法恢复 GBase 8s Primary Storage Manager 创建的备份对象。存储目录表的位置由 `PSM_CATALOG_PATH` 配置参数设置（缺省值为 `$GBS_HOME/etc/psm`）。

要从备份对象恢复实例，请使用 `gbackuprestore` 实用程序。存储管理器将为您跟踪备份对象和存储设备。

## 示例 2：存储两个实例的备份

此示例显示如何配置一个 GBase 8s Primary Storage Manager 实例来管理多驻留环境中的两个数据库服务器实例的存储设备。

在此示例中，您将在同一台计算机上设置两个独立的数据库服务器环境。每个数据库服务器都安装在一个独立的目录中：`/usr/gbasedbt/ids1210fc1` 和 `/usr/gbasedbt/ids1210fc1b`，并且具有一个数据库服务器实例。这两个数据库服务器实例上的备份操作的存储由一个 GBase 8s Primary Storage Manager 实例管理。为每个实例配置了物理和逻辑数据的存储设备池。

1. 对于每个实例，编辑 `onconfig` 文件以配置 `gbackuprestore` 的存储管理。

此表列出了您必须设置的配置参数以及要为每个参数指定的值。

表 1. 配置参数及其关联的值。

配置参数	值
<code>BAR_BSALIB_PATH</code> 指定存储管理器的共享库的完整路径和名称。	<code>/usr/gbasedbt/ids1210fc1b/lib/libbsapsm.so</code>
<code>PSM_CATALOG_PATH</code> 指定存储目录表的路径。	<code>/usr/gbasedbt/ids1210fc1b/etc/psm</code>
<code>PSM_DBS_POOL</code> 指定用于存储联机数据（数据库空间）备份的设备组的名称。	FC1: DBSPPOOL_FC1 FC1B: DBSPPOOL_FC1B
<code>PSM_LOG_POOL</code> 指定用于存储联机逻辑日志备份的设备组的名称。	FC1: LOGPOOL_FC1

配置参数	值
	FC1B: LOGPOOL_FC1B

2. 对于每个实例，创建用于存储备份对象的目录。

```
mkdir $GBS_HOME/backups/dev_for_1201fc1
mkdir $GBS_HOME/backups/dev_for_1201fc1b
```

3. 运行 `gpsm` 实用程序来为每个实例创建设备池。例如，指定：

```
gpsm -P add DBSPOOL_FC1 -t DBSPACE
gpsm -P add LOGPOOL_FC1 -t LOGFILE
gpsm -P add DBSPOOL_FC1B -t DBSPACE
gpsm -P add LOGPOOL_FC1B -t LOGFILE
```

4. 运行 `gpsm` 实用程序来添加存储设备。

```
gpsm -D add $GBS_HOME/backups/dev_for_1201fc1 -t FILE -g DBSPOOL_FC1
gpsm -D add $GBS_HOME/backups/dev_for_1201fc1 -t FILE -g LOGPOOL_FC1
gpsm -D add $GBS_HOME/backups/dev_for_1201fc1b -t FILE -g
DBSPOOL_FC1B
gpsm -D add $GBS_HOME/backups/dev_for_1201fc1b -t FILE -g
LOGPOOL_FC1B
```

5. 对于每个实例，运行 `gbackuprestore` 实用程序来执行所有联机存储空间和已用逻辑日志的标准 0 级备份。

```
gbackuprestore -b -L 0
```

6. 验证存储管理器是否已设置，以及是否创建了备份对象。

a. 对于每个实例，在 `gbackuprestore` 活动日志中查找以确认存储管理器已就绪且 `gbackuprestore` 可识别该存储管理器。例如，查找以下信息：

```
2012-01-03 15:51:23 11193 2569 Gbasedbt PSM is ready.
2012-01-03 15:51:23 11193 2569 Using Gbasedbt PSM version
12.10.FC1
as the Storage Manager. XBSA API version is 1.0.3.
```

b. 使用 `gpsm` 实用程序可列出已创建的存储对象：

```
gpsm -O list
```

报告包含存储对象标识、创建存储对象的日期、存储对象的大小，以及存储对象在存储设备中的位置。

示例 3：将备份导出到另一个目录并从其恢复备份

此示例显示如何将备份导出到新目录，并从该目录导入备份对象。

假设您保留了五个备份生成。作为额外的预防措施，您还将最近几个备份的副本保留在一个单独的目录中。在此示例中，您将使用 `onmsync` 实用程序将最近备份导出到单独目录中的 GBase 8s Primary Storage Manager 外部池，并从中导入该备份。

存储管理器跟踪外部设备池 (EXTPOOL) 中的设备，以便可以与外部设备之间复制对象。

(虽然存储管理器会跟踪设备，但它不会跟踪存储管理器目录中 EXTPOOL 池内部的文件和对象。)

1. 按照示例 1：存储实例的备份中的步骤存储实例的备份。
2. 运行 `gpsm -D list` 命令来检查 EXTPOOL 池中是否有设备。
  - a. 如果 EXTPOOL 池中没有设备，请使用 `gpsm -D add` 命令添加一个设备。

以下示例显示如何将路径为 `/export/gbasedbt/psm_exportdir` 的设备添加到 EXTPOOL 池。

```
$ gpsm -D add /export/gbasedbt/psm_exportdir -g EXTPOOL -t FILE
```

3. 运行 `onmsync` 命令来导出第一次生成的 0 级备份中的所有备份对象，同时使用前缀 `pw_sept5`，该前缀将成为实用程序放置备份的子目录的名称：

```
onmsync -E -p pw_sept5 -g 1
```

运行 `onmsync -E` 命令来导出备份对象之后，将在 EXTPOOL 目录中看到个子目录，其中包含保存备份对象和名为 `export.bom` 的文件的目录。

假设主备份目录中存储的备份生成出现问题，而您希望从第二个目录导入 `pw_sept5` 备份生成。要导入备份生成：

1. 运行 `onmsync` 命令来导入 `pw_sept5` 子目录中的所有备份对象：

```
onmsync -l -p pw_sept5
```

使用您自己的文件传输方法，根据需要将导出的备份移至其他机器。

示例 4：从一个服务器导出备份，并将其导入到另一个服务器

此示例显示如何使用 `onmsync` 实用程序从名为 `gbasedbt_serv1` 的数据库服务器导出备份。

此示例还显示了如何使用 `onmsync` 实用程序将数据导入到名为 `gbasedbt_serv2` 的服务器。

1. 设置并导出数据库服务器 `gbasedbt_serv1` 上的文件：
  - a. 设置 `gbasedbt_serv1` 的 `GBS_HOME`、`GBS_DATA` 环境变量。
  - b. 运行 `gpsm -D list` 命令来检查 EXTPOOL 池中是否有设备。如果 EXTPOOL 池中  
没有设备，请使用 `gpsm -D add` 命令添加一个设备。
  - c. 运行 `onmsync` 命令来导出第一次生成的 0 级备份中的所有备份对象，同时使用  
前缀 `serv1_20120810`，该前缀将成为实用程序放置备份的子目录的名称：

```
$ onmsync -E -p serv1_20120810 -g 1
```

2. 准备导入第二个数据库服务器 `gbasedbt_serv2` 上的文件，如下所示：

- a. 设置 `gbasedbt_serv2` 的 `GBS_HOME`、`GBS_DATA` 环境变量。
- b. 运行 `gpsm -D list` 命令以确定 `EXTPOOL` 是否具有您在步骤 1b 中查看或添加的相同设备。（对于共享设备，可能发生这种情况。）如果 `EXTPOOL` 池中没有设备，请使用 `gpsm -D add` 命令添加一个设备。
- c. 将先前导出的备份对象（例如，子目录 `serv1_20120810`）复制到将从中导入备份对象的 `EXTPOOL` 设备。
- d. 运行以下命令来从 `EXTPOOL` 设备导入备份对象：

```
$ onsmsync -I -p serv1_20120810
```

运行 `onsmsync -I` 命令以导入备份对象之后，这些对象将存储在新的 `LOGPOOL` 和 `DBSPOOL` 池中。

- e. 运行 `gpsm -O list` 命令来查看导入的对象。

请注意，导入命令还会在 `$GBS_HOME/etc/` 目录中创建新的 `ixbar` 文件。

```
$ ls -l $GBS_HOME/etc/*ixbar*
```

```
-rw-rw-- 1 gbasedbt gbasedbt    0 Aug 10 19:44
/usr/gbasedbt/etc/ixbar.12.20120810.194441
-rw-rw-- 1 gbasedbt gbasedbt 2704 Aug 10 19:44
/usr/gbasedbt/etc/ixbar.12
```

新的 `ixbar` 文件将列出导入的备份对象，以便您能够执行 `gbackuprestore` 冷恢复以将第一个数据库服务器中的 `gbasedbt_serv1` 实例恢复到第二个数据库服务器上的 `gbasedbt_serv2` 实例。

## 5.3 设置 GBase 8s Primary Storage Manager

设置操作包括收集和指定有关存储设备的信息，以及根据需要更改存储管理器的缺省配置。

### 5.3.1 收集有关文件目录和设备的信息

必须针对每个 `DBSPOOL` 和 `LOGPOOL` 池的至少一个文件目录或设备收集其相关信息并进行配置，然后 `gbackuprestore` 才能使用 GBase 8s 主存储管理器。

在定义目录或设备前，请收集以下信息：

- 您计划用于备份存储的设备的完整路径名和类型。
- 您希望在每个设备上落实到 `gbackuprestore` 备份的空间量。

有些磁带设备不返回“设备已满”消息，因此对于这些设备，必须指定大小，以便 GBase 8s Primary Storage Manager 能够正确使用该设备。对于任何返回“设备已满”消息的设备，大小是阻止存储管理器使用设备上的所有空间的可选方法。请参阅您设备的用户指南，以确定该设备在变满时是否返回设备已满代码。

- 每个磁带设备的块大小。

### 5.3.2 配置 GBase 8s Primary Storage Manager

缺省情况下，GBase 8s 主存储管理器 是使用存储管理器中指定的信息和一些 `gbackupstore` 配置参数自动配置的。当您使用 `gpsm` 实用程序时，也会自动对其进行配置。您可以更改配置。

GBase 8s Primary Storage Manager 仅使用文件设备（磁盘），而不使用磁带。不能将存储管理器配置为使用磁带。

#### 要手动配置 GBase 8s Primary Storage Manager:

1. 更新 `BAR_BSALIB_PATH` 配置参数以指向存储管理器库。  
例如，在 Linux™ 或 Solaris 上，指定：  

```
BAR_BSALIB_PATH      $GBS_HOME/lib/libbsapsm.so
```
2. 通过使用 `gpsm` 实用程序，指定备份与恢复操作的目标和源设备。
3. 根据需要更改您环境的存储管理器的缺省配置：
  - a. 要覆盖存储管理器日志文件和目录的位置、调试活动和池名称的缺省值，请在 GBase 8s Primary Storage Manager 配置参数中指定新值。
  - b. 要使用 `gbackupstore` 和 GBase 8s Primary Storage Manager 指定更大的传输缓冲区，请增大 `BAR_XFER_BUF_SIZE` 配置参数中的大小。
  - c. 要更改 `gbackupstore` 活动日志中进度消息的频率，请更新 `BAR_PROGRESS_FREQ` 配置参数中指定的值。
  - d. 要更改 `gbackupstore` 并行运行的进程数，请更新 `BAR_MAX_BACKUP` 配置参数中指定的值。

## 5.4 管理存储设备

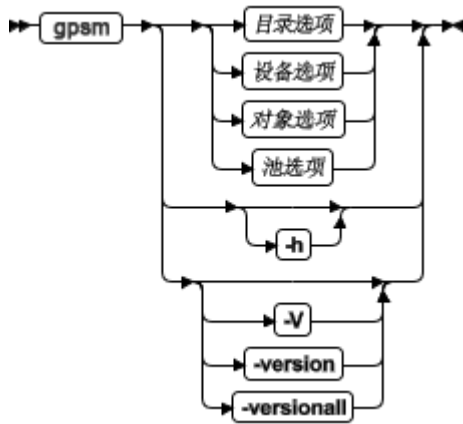
使用 `gpsm` 实用程序可添加、监视和除去存储设备，并管理 GBase 8s 主存储管理器 目录、锁定和对象。使用 `onmsync` 实用程序可导出 `gbackupstore` 备份，从外部池导入备份，以及使备份到期。

## 5.5 用于存储管理的 gpsm 实用程序

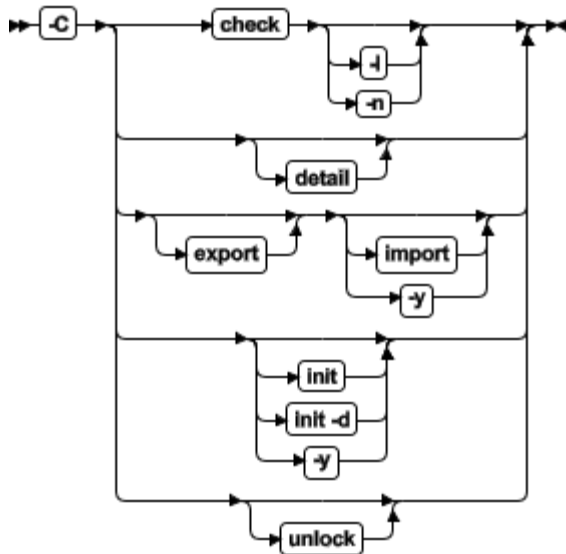
使用 gpsm 实用程序可管理 GBase 8s 主存储管理器 目录、设备、锁定和对象。

**先决条件:** 要运行 gpsm 实用程序, 您必须是 root 用户或 gbasedbt, 或者是 bargroup 组的成员。

语法

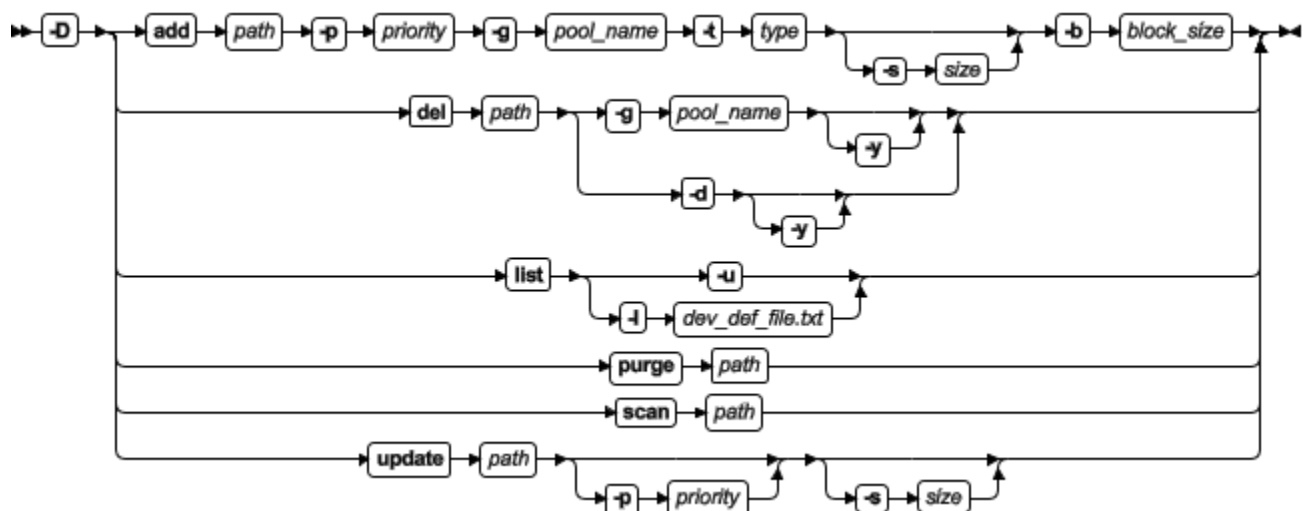


目录选项

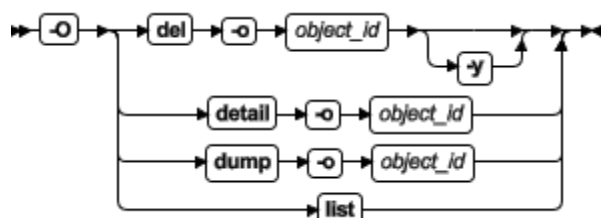


设备选项





对象选项



池选项



表 1. gpsm 实用程序目录选项

元素	用途	重要注意事项
- C check	检查存储管理器目录表，这些表用于存储有关存储管理器所管理的池和设备的元数据	此命令可识别有问题的文件。
- C check -1	检查目录表时显示索引键	
- C check -n	指示存储管理器不修正发现的错误	

元素	用途	重要注意事项
- C detail	显示有关存储管理器目录表的详细信息	
- C export	将 GBase 8s Primary Storage Manager 目录表导出到名为 psm_catalog.exp 的目录	
- C import	将当前 GBase 8s Primary Storage Manager 目录替换为从 psm_catalog.exp 目录中的文件重新创建的目录。	仅当发生系统问题、丢失当前目录以及需要还原到导出的目录时，才应导入目录。如果需要导入目录，请先运行 gpsm -C init 命令，然后再运行gpsm -C import 命令。
- C init	删除存储管理器的目录表	
- C init -d	删除文件设备中的存储管理器目录表和备份对象	
- C unlock	对存储管理器目录解锁	如果存储管理器因为发生故障而从备份或恢复会话异常退出，那么存储管理器目录表可能会保持锁定。如果目录表处于锁定状态，您可以释放这些锁定。
-y	指定在删除目录表之前不要请求确认	

表 2. gpsm 实用程序设备选项

元素	用途	重要注意事项
-D add	将设备添加到 -g 选项指定的池中	在添加设备前，收集有关设备的信息。请参阅收集有关文件目录和设备的信息。
-D del	除去设备： <ul style="list-style-type: none"> <li>如果使用 -g 选项，将从 -g 选项指定的池中除去设备，而保留 GBase 8s Primary Storage Manager 目录中的设备对象。</li> </ul>	如果使用 -g 选项删除设备，可以根据需要复原对象。 如果除去设备，存储管理器将无法向该设备添加新对象。

元素	用途	重要注意事项
	<ul style="list-style-type: none"> <li>如果使用 <code>-d</code> 选项，将从所有池中除去设备，并从该设备中的文件系统中除去所有备份对象</li> </ul>	
<code>-D list</code>	显示系统中所有设备的列表	
<code>-D purge</code>	从 GBase 8s Primary Storage Manager 目录中除去缺少的存储管理器对象	
<code>-D scan</code>	<p>扫描设备中的对象以验证 GBase 8s Primary Storage Manager 目录中是否存在这些对象，以便可以根据需要复原这些对象</p> <p>如果对象不在目录中，此命令将向目录添加该对象。</p>	<p>如果命令无法向目录添加对象，此命令将忽略缺少的文件。</p> <p>要能够向目录添加缺少的对象，必须满足以下条件：</p> <ul style="list-style-type: none"> <li>该对象标识不能分配给存储管理器中的其他任何对象。</li> <li>文件不能重命名或重定位到设备内部的其他目录中</li> <li>该对象版本不能分配给存储管理器中的其他任何对象。</li> </ul>
<code>-D update</code>	修改有关设备的信息	如果要修改有关多个设备的信息，请针对每个设备运行独立的命令。
path	设备（针对 TAPE 设备）或目录（针对 FILE 设备）的全名和路径	<p>路径的格式必须适合设备连接到的操作系统。</p> <p>设备的名称在池中必须唯一。</p> <p>您可以在多个池中包含相同的设备。</p> <p>如果要删除、列出、清除、扫描或更新信息，路径必须是现有设备的路径。</p>
<code>-b block_size</code>	（仅限磁带设备）向设备写入数据前需要积累的数据的最小字节数	块大小对于磁带设备是必需的。
<code>-d</code>	从所有池中删除池并删除备份对象	块大小对于磁带设备是必需的。
<code>-g pool_name</code>	要将设备添加到的池，即 DBSPOOL、LOGPOOL 或 EXTPool	有关池的信息存储在 GBase 8s Primary Storage Manager 目录中。

元素	用途	重要注意事项
		<p>如果未提供池名称，命令将失败。</p> <p>指定：</p> <ul style="list-style-type: none"> <li>• 用于备份数据库空间、Blob 空间和智能大对象空间的 DBSPOOL</li> <li>• 用于备份逻辑日志的 LOGPOOL</li> <li>• 用作登台区域的 EXTPool，从中可以将特定备份或备份生成移至永久存储器或移至其他计算机上。</li> </ul>
-l dev_def_file.txt	从设备定义文件装入有关设备的信息	
-p priority	设备的优先级，即 HIGHEST、HIGH、LOW 或 READ-ONLY	<p>存储管理器会先填满池中的高优先级设备，然后再将数据放入该池中的低优先级设备。当存储管理器准备填充池时，如果高优先级设备繁忙，那么存储管理器会使用低优先级设备。</p> <p>池中只能有一个设备的优先级可以为 HIGHEST。</p> <p>如果在同一池中多个设备具有相同的优先级，存储管理器将确定要首先使用的设备。</p> <p>设备变满时，存储管理器会将其优先级更改为 READ-ONLY。向设备添加更多空间后，可以更改优先级。</p>
-s size	(仅限磁带设备)设备的最大存储容量(以千字节为单位)	<p>该大小对于磁带设备是可选的。如果未指定大小，或指定为 0，那么存储管理器会将大小解释为无限制。当大小为无限制时，只有设备返回错误以指定设备已满时，才会将设备视为已满。</p> <p>要指定大小，请输入大小的数字值，后跟后缀 B、K、M、G、T 或 P (分别表示字节、千字节、兆字节、千兆字节、太字节或皮特字节)。后缀可以是大写，也可以是小写。</p>
-t type	设备的类型，可以是 FILE 或 TAPE	有关设备的信息存储在 GBase 8s Primary Storage Manager 目录中。
-u	将有关设备的信息卸载到设备定义文件	设备定义文件是一个具有特定格式的文本文件。使用装入选项运行 gpsm命令时，存储管理器会使用该文件重新创建信息。

元素	用途	重要注意事项
-y	指定不要求确认完成请求的操作	

表 3. gpsm 对象选项

元素	用途	重要注意事项
-O del	从池删除物理对象	
-O detail	显示有关指定对象的详细信息。详细信息包括对象的位置。	
-O dump	将对象数据抽取到当前目录中的某个文件	
-o object_id	标识特定对象	可以使用单一命令来删除或转储一个或多个对象，如用途中所示。
-O list	显示池中的所有对象	对于每个对象，该列表包含创建对象的日期和时间、对象的大小和对象的路径名。
-y	指定不要求确认完成请求的操作。	

表 4. gpsm 池选项

元素	用途	重要注意事项
-P add pool_name	添加新池	
P del pool_name	删除指定的池	
-P list	列出系统中的所有池	
-y	指定不要求确认完成请求的操作。	

表 5. gpsm 实用程序一般选项

元素	用途	重要注意事项
-h	显示帮助信息	
-V	显示软件版本号和序列号	有关标准 GBase 8s -V 和 -version 选项的更多详细信息，请参阅《GBase 8s 管理员参考》中的获取实用程序的版本信息。
-version	显示软件版本号、序列号和其他信息，例如主机、操作系统、构建日期和 Global Language Support (GLS) 版本	有关标准 GBase 8s -V 和 -version 选项的更多详细信息，请参阅《GBase 8s 管理员参考》中的获取实用程序的版本信息。

元素	用途	重要注意事项
<code>-version</code> <code>all</code>	显示 <code>gpsm</code> 版本信息和有关 PSM 共享库的信息	

### 用途

当您运行 `gpsm` 命令来定义设备时，如果不存在存储管理器目录，那么存储管理器会自动创建这些目录。

存储管理器的缺省设备为 `$GBS_HOME/backups`。此低优先级设备在创建目录时会自动创建。您可以除去缺省设备。

创建设备时，如果该设备的目录不存在，存储管理器将自动创建该目录。存储管理器将使用 `gpsm -D add` 命令中指定的目录路径。

可以使用单一命令来删除一个或多个对象，例如，通过运行以下格式的命令：

```
gpsm _O del -o obj_1 -o obj_2
```

还可以使用单一命令来转储一个或多个对象，例如，通过运行以下格式的命令：

```
gpsm _O dump -o obj_1 -o obj_2
```

如果不需要数据，请运行 `onsmsync` 实用程序来从 GBase 8s Primary Storage Manager 删除备份对象。

有些第三方存储管理器不允许 `onsmsync` 实用程序从存储管理器删除备份对象。如果您有第三方存储管理器，那么可能需要手动删除不再需要的备份对象。

### 示例

以下命令在 `DBSPOOL` 池中添加路径名为 `$GBS_HOME/backups` 的文件设备：

```
gpsm -D add $GBS_HOME/backups -g DBSPOOL -t FILE -p HIGH
```

以下命令检查 GBase 8s Primary Storage Manager 目录表，并指示存储管理器不修正检查期间发现的任何错误：

```
gpsm -C check -n
```

以下命令列出池中的对象，包含创建对象的日期和时间、对象的大小和对象的路径名。

```
gpsm -O list
```

## 5.5.1 `gpsm -C detail` 输出

使用 `gpsm -C detail` 命令可查看有关存储管理器目录表的详细信息。

样本 `gpsm -C detail` 命令输出

```
D:\IFMXDATA\gacpsm>gpsm -C detail
```

**GBase 8s 主存储管理器状态:**

PSM 唯一标识 : 1358735848  
 目录位置 : D:\GBase\_8s\ids1210\etc\psm\  
 目录状态 : 已锁定  
 目录所有者 : 2  
 目录锁定方式 : 常规

**会话:**

会话标识	进程标识
2	1576
3	4556
4	5555

**GBase 8s PSM 锁定的对象**

会话	对象标识	日期	时间	服务器	对象名
3	116	2012-12-09	20:54:34	/gacpsm_tcp	/gacpsm_tcp/168/242
4	117	2012-12-09	21:07:44	/gacpsm_tcp	/gacpsm_tcp/168/242

输出包含以下部分:

**GBase 8s 主存储管理器状态**

显示有关系统中处于活动状态的所有会话的常规信息以及目录是否已锁定。

**PSM 唯一标识**

目录的标识

**目录位置**

目录的路径

**目录状态**

指示目录是否已锁定

**目录所有者**

GBase 8s Primary Storage Manager 会话标识

**目录锁定方式**

锁定类别

例如，常规表示锁定是用户锁定。

## 会话

列出系统中处于活动状态的所有会话以及与这些会话匹配的进程标识。

### 会话标识

会话的标识。此标识与显示在目录所有者字段中的标识相同。

### 进程标识

gbackuprestore archecker 或锁定目录的存储管理器进程的标识。

## GBase 8s PSM 锁定的对象

显示设备或对象中由存储管理器会话持有的锁定。对于每个锁定，输出会显示会话编号、对象标识、进行锁定的日期和时间、服务器以及对象名。

### 5.5.2 gpsm -D list 输出

gpsm -D list 命令显示有关每个 GBase 8s 主存储管理器池中的所有设备的信息。您可以使用此列表来确定是否需要更改有关设备的信息。

样本 gpsm -D list 命令输出

类型	优先级	块/大小 (MB)	池名称	设备名
FILE	LOW	--/--	DBSPOOL	/gbasedbt/backups
FILE	LOW	--/--	LOGPOOL	/gbasedbt/backups

#### 类型

设备的类型，即 FILE 或 TAPE（目前仅支持 FILE 类型。）

#### 优先级

设备的优先级，即 HIGH、HIGHEST、LOW 或 READ-ONLY

如果未指定优先级，HIGH 为缺省优先级。池中只能有一个设备的优先级可以为 HIGHEST。

#### 块大小

设备的大小（仅适用于 TAPE 类型的设备）

#### 池名称

池的名称（DBSPOOL、LOGPOOL 或 EXTPOOL）

在上面的示例输出中，没有 EXTPOOL 设备。



## 设备名

设备的完整路径名

### 5.5.3 gpsm -O list 输出

gpsm -O list 命令显示池中存储的所有对象。

样本 gpsm -O list 命令输出

对象列表报告		逻辑路径	
对象标识	创建日期	大小 (MB)	名称.版本 (省略片段号)
1	2012-07-06 14:39:47	12.0	/gacpsm_tcp/rootdbs/0/gacpsm_tcp.1
2	2012-07-06 14:41:18	12.0	/gacpsm_tcp/rootdbs/0/gacpsm_tcp.2
3	2012-07-11 13:42:10	3.9	/gacpsm_tcp/160/14/gacpsm_tcp.1
4	2012-07-11 13:42:13	3.9	/gacpsm_tcp/160/15/gacpsm_tcp.1
5	2012-07-11 13:42:15	3.9	/gacpsm_tcp/160/16/gacpsm_tcp.1
6	2012-07-11 13:42:15	3.9	/gacpsm_tcp/160/17/gacpsm_tcp.1
7	2012-07-11 13:42:15	3.9	/gacpsm_tcp/160/18/gacpsm_tcp.1
8	2012-07-11 13:42:15	3.9	/gacpsm_tcp/160/19/gacpsm_tcp.1
9	2012-07-11 13:42:16	3.9	/gacpsm_tcp/160/20/gacpsm_tcp.1
10	2012-07-11 13:42:16	3.9	/gacpsm_tcp/160/21/gacpsm_tcp.1

#### 对象标识

所存储对象的标识

#### 创建日期

创建对象的日期和时间

#### 大小

对象的大小（以兆字节为单位）

#### 名称.版本

对象的路径名，后跟 . 和对象的版本（例如，.2 表示 V2）

## 5.6 设备池

GBase 8s 主存储管理器 池是一个指定的磁盘设备组，您可以将其用作备份的存储库。

在存储备份对象时，GBase 8s Primary Storage Manager 会从池中选择特定设备，并在设备已满或发生故障时自动从一个设备移至另一个设备。通过使用 `gpsm` 实用程序在池中添加、修改、查看和删除设备来维护池。

可以使用以下三个池：

#### **DBSPOOL**

保留数据库空间、Blob 空间和智能大对象空间的联机备份

#### **LOGPOOL**

保留逻辑日志的联机备份

#### **EXTPOOL**

充当登台区域，用于将对象的备份集导出到单个大型外部逻辑对象，或用于导入备份的对象。可以将此池中的特定备份或备份生成移至永久存储器或移至其他计算机上。EXTERNAL 池中的文件处于脱机状态。这些文件对于 `gbackuprestore` 不可视，并且 GBase 8s Primary Storage Manager 不会跟踪这些文件。

## 5.7 GBase 8s Primary Storage Manager 的设备 配置文件

`gpsm` 实用程序可以生成一个设备配置文件，它是一个文本文件，其中包含有关存储设备的信息。该实用程序使用此信息来重新创建设备。

配置文件包含有关每个设备的以下信息：

#### **DEVICE**

设备的完整路径

#### **TYPE**

设备类型

FILE = 磁盘设备上的文件目录

TAPE = 磁带设备

#### **POOL**

包含设备的池，可以是 DBSPOOL、LOGPOOL 或 EXTPPOOL

#### **BLOCKSIZE**

不适用于磁盘设备。

（仅限磁带设备）向设备写入数据前需要积累的数据的最小字节数（如果未指定或指定为 0，那么存储管理器会将大小解释为无限制。）

**SIZE**

不适用于磁盘设备。

对于磁带设备，是指设备的存储容量（以千字节为单位）。

**PRIORITY**

设备的优先级

例如，设备配置文件可能包含以下信息

```
DEVICE=/vobs/tristarm/sqlldist/psm_backup/dbspace
TYPE=FILE
POOL=DBSPOOL
BLOCKSIZE=0
SIZE=0
PRIORITY=HIGH
```

## 5.8 GBase 8s Primary Storage Manager 文件命名约定

当创建用于存储备份数据的文件时，GBase 8s 主存储管理器使用特定的文件命名约定。

对于 DBSPOOL 和 LOGPOOL 设备池，存储文件的路径名由以下部分组成：

1. 类别信息：
  - 。 对于空间备份，类别包含设备名、数据库服务器名称和空间名称
  - 。 对于日志备份，类别包含设备名、数据库服务器名称、服务器编号和日志文件号
2. 对于空间备份，备份级别，即 0、1 或 2
3. 版本号，为一个整数，从 1 开头，表示该类别的对象，并针对该类别的对象的每个后续备份进行递增。
4. 用于标识已备份对象的片段的标识

对于空间备份，文件名的格式如下：

```
/device/DBSERVERNAME/dbspace/backup_level/DBSERVERNAME.version.piece
```

例如，服务器 SERVER1 的数据库空间 rootdbs 的第二个 0 级备份的第三段的文件名为：

```
/my_device/SERVER1/rootdbs/0/SERVER1.2.3
```

对于日志备份，文件名的格式如下：

```
/device/DBSERVERNAME/SERVERNUM/LOG_UNIQUE_ID/DBSERVERNAME.version.piece
```

如果在 GBase 8s Primary Storage Manager 中使用带 -E 选项的 onsmsync 实用程序来导出备份生成，那么 onsmsync 实用程序将创建备份文件并将其放入存储管理器 EXTPOOL 设备的子目录中。当您使用带 -E 或 -I 选项的 onsmsync 实用程序时，必须提供前缀（子目录的名称）。gbackuprestore 使用指定的路径作为与存储管理器通信以存储和检索对象的关键字。

## 5.9 GBase 8s Primary Storage Manager 的消息

### 日志

GBase 8s Primary Storage Manager 会将消息写入存储管理器的活动日志和调试日志。

消息日志存储在 BAR\_DEBUG\_LOG 或 BAR\_ACT\_LOG 配置参数中指定的目录内。您可以使用 PSM\_ACT\_LOG 和 PSM\_DEBUG\_LOG 配置参数来为其中每个日志指定其他目录。

PSM\_DEBUG 配置参数指定调试日志中捕获的调试活动的级别。

## 6 archecker 表级别恢复实用程序

### 6.1 archecker 表级别恢复实用程序

可以使用 archecker 实用程序执行时间点表级别恢复，这种恢复将从归档和逻辑日志中抽取表或表的一部分。

archecker 实用程序通过指定要从中抽取数据的源表、放置数据的目标表和链接两个表的 INSERT 语句来恢复表。

有关使用 archecker 实用程序验证备份的信息，请参阅 gbackuprestore -v 语法：验证备份。

#### 6.1.1 archecker 实用程序概述

当需要恢复数据库的一部分、一张表、表的一部分或一组表时，archecker 实用程序很有用。在需要跨服务器版本或平台移动表的情形下，它也很有用。

在以下情况中使用 archecker：

- 恢复数据

可以使用 `archecker` 实用程序恢复先前已使用 `gbackuprestore` 或 `gtape` 备份的特定表或一组表。这些表可以被恢复到特定的时间点。这一点很有用，例如，它可用于恢复被意外删除的表。

不能从远程设备恢复数据。

在执行表级别恢复时不能使用共享内存连接。

- 复制数据

`archecker` 实用程序也可被用作一种复制数据的方法。例如，可以把表从生产系统移动到另一个系统。

`archecker` 实用程序比其他复制数据的机制更高效。由于 `archecker` 以文本形式抽取数据，它可以在不同平台或服务器版本之间复制数据。

- 迁移数据

也可以将 `archecker` 实用程序用作迁移工具，以将表移动到其他 GBase 8s 服务器。

`archecker` 实用程序旨在恢复特定表或表集合。其他情况需要使用不同的实用程序。例如，在以下数据恢复方案中使用 `gbackuprestore` 或 `gtape`：

- 整个系统恢复
- 从磁盘故障中恢复

要配置 `archecker` 实用程序的行为，请使用 `archecker` 配置文件。要定义 `archecker` 恢复的数据的模式，请使用 `archecker` 模式命令文件。这些文件在以下部分中描述。

### **archecker 配置文件**

`archecker` 实用程序使用配置文件来设置某些参数。

将 `AC_CONFIG` 环境变量设置为 `archecker` 配置文件的完整路径名。缺省情况下，`AC_CONFIG` 环境变量设置为 `$GBS_HOME/etc/ac_config.std`。如果将 `AC_CONFIG` 设置为用户定义的文件，必须指定包括文件名在内的完整路径。

有关该文件中使用的配置参数的信息，请参阅 `archecker` 实用程序配置参数和环境变量。

### **模式命令文件**

`archecker` 实用程序使用模式命令文件指定以下各项：

- 源表
- 目标表
- 表的模式
- 数据库

- 外部表
- 表恢复到的时间点
- 其他选项

该文件使用一种类似 SQL 的语言提供 `archecker` 用于执行数据恢复的信息。有关支持的语句和语法的完整信息，请参阅 `archecker` 模式引用。

有两种设置模式命令文件的方法：

- 在 `archecker` 配置文件中设置 `AC_SCHEMA` 配置参数。有关更多信息，请参阅 `AC_SCHEMA` 配置参数。
- 使用 `-f cmdname` 命令行选项。有关更多信息，请参阅 `archecker` 实用程序命令的语法。

如果同时指定了两种方法，`-f` 命令行选项优先。

### 表级别恢复和语言环境

对于表级别恢复，如果要恢复的表（归档中的表）的语言环境代码集不同于缺省语言环境的代码集（`en_US.8859-1`），那么必须设置 `DB_LOCALE` 环境变量来使其代码集与要恢复的归档表的语言环境相同。

在表级别恢复过程中将不执行任何代码集转换，数据库或要恢复的表的语言环境代码集必须与此数据库或数据将恢复到的表的语言环境代码集相匹配。此外，相同 `DB_LOCALE` 信息将用于通过相同表级别恢复命令模式文件进行恢复的所有表。

## 6.1.2 使用 `archecker` 进行数据恢复

使用 `archecker` 实用程序可执行两种类型的恢复操作。

`archecker` 实用程序执行的两种恢复类型如下：

- 基于 0 级归档的物理恢复。
- 先执行物理恢复，然后执行逻辑恢复，逻辑恢复使用 0 级归档和逻辑日志以将数据恢复到特定时间点。

`archecker` 在读取命令文件时，确定是只执行物理恢复，还是在物理恢复后执行逻辑恢复。缺省情况下，`archecker` 执行物理和逻辑恢复。如果使用了 `WITH NO LOG` 子句，`archecker` 将不执行逻辑恢复。

`archecker` 在单纯的物理恢复中使用的过程和资源不同于在物理及逻辑恢复中所使用的。以下各部分概述了这些过程。

### 物理恢复

当 `archecker` 实用程序执行物理恢复时，该实用程序将从 0 级归档抽取数据。

执行物理恢复时，`archecker` 执行以下任务：

- 在数据恢复之前，禁用所有约束（包括引用此目标表的外部约束）、索引和触发器。如果表没有约束、索引或触发器，那么恢复性能更佳。
- 读取模式命令文件，以确定以下各项：
  - 源表
  - 目标表
  - 所有表的模式
  - 表位于的数据库空间的名称
  - 从中抽取数据的特定归档
- 扫描归档，寻找属于被恢复的表的页面
- 处理数据页的每一行，以确定该行是完整的还是部分的。

如果该行目前是部分的，那么 `archecker` 将确定该行剩余部分是否已分阶段，如未分阶段，那么将其分阶段以便于后续处理。
- 对于单纯的物理恢复，对该行应用过滤器并拒绝不需要的行。
- 将该行插入目标表。

要使用原始模式恢复表，必须指定源模式。要使用不同模式恢复表，目标模式中的表名称必须与源模式中的表名称不同。使用不同模式恢复之后，可以使用 `rename table` 语句重命名表。

## 逻辑恢复

物理恢复后，逻辑恢复可以进一步将表恢复到用户指定的时间点。为此，`archecker` 实用程序读取已备份的逻辑日志，将它们转换成 SQL 语句，然后重放这些语句以恢复数据。

在执行逻辑恢复之前，确保所要恢复的事务都包含在已备份的逻辑日志中。`archecker` 实用程序不能从当前日志中重放事务。不能对外部表执行逻辑恢复。

如果逻辑恢复期间一个表被更改、删除或截断，那么恢复因该表终止。终止发生在执行更改的点上。`archecker` 消息日志文件中的消息将记录发生了更改操作。

`archecker` 实用程序无法在逻辑恢复非记录数据库中的压缩表期间处理压缩字典。如果逻辑恢复发现为表创建了新的压缩字典，那么恢复将为该表停止。

执行逻辑恢复时，`archecker` 使用两个同时运行的进程：

### **stager**

汇编逻辑日志并将其保存在表中。

## applier

将日志记录转换为 SQL 语句并执行这些语句。

## stager

为收集相关的逻辑日志记录，stager 将执行以下步骤：

1. 仅扫描已备份的逻辑日志

stager 读取逻辑日志备份文件并汇编成完整的日志记录。

2. 测试逻辑日志记录

任何不适用于被恢复的表的日志记录将被拒绝。

3. 将逻辑日志信息插入表中

如果没有拒绝逻辑日志记录，那么会将它插入到一个阶段表中。

## applier

applier 从 stager 创建的控制表中读取数据。它开始处理必需的事务，并更新控制表以显示该事务正在处理。接下来，它按顺序对每条日志记录逐行进行操作，直到事务落实。

所有对控制表的更新在同一个事务中以日志记录修改的形式出现。这使得所有的工作可以作为一个独立单元完成或撤销，始终保持了完整性。如果发生了错误，就会回滚事务，该错误将记录到该事务的控制表条目中。

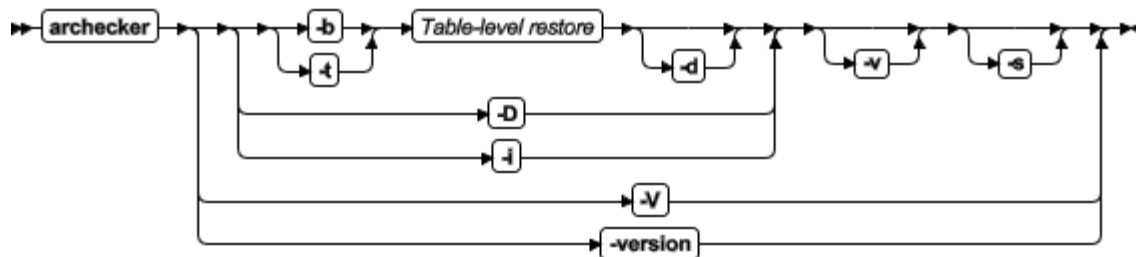
如果数据正在恢复而数据库管理员决定包含逻辑恢复，那么两个附加的工作列和一个索引将被添加到目标表中。这些列包含了原始行标识及原始部件号。这些列提供了在原始的源归档中识别该行位置的唯一键。要控制索引的存储，可使用 SET WORKSPACE 命令（请参阅 SET 语句）。否则会将索引和表存储在同一个空间。

applier 结束且恢复完成之后，这些列以及对它们创建的任何索引都将从目标表中删除。

### 6.1.3 archecker 实用程序命令的语法

archecker 实用程序为从归档中恢复数据提供了一个命令行界面。要使用 archecker，您必须指定一个配置文件和一个模式命令文件。





元素	描述
-b	提供使用 <code>gbackuprestore</code> 创建的备份的直接 XBSA 访问。
-d	删除先前的 <code>archecker</code> 恢复文件，但 <code>archecker</code> 消息日志除外。有关更多信息，请参阅 <a href="#">删除恢复文件的时间</a> 。
-D	删除先前的 <code>archecker</code> 恢复文件，但 <code>archecker</code> 消息日志除外，然后退出。  -D 选项可以和 -X 选项一起使用，以删除先前的恢复文件以及 <code>sysutils</code> 数据库中的任何表级别恢复工作表。有关更多信息，请参阅 <a href="#">删除恢复文件的时间</a> 。
-f cmdfile	指定 <code>archecker</code> 使用由 <code>cmdfile</code> 指定的命令文件。该选项将覆盖 <code>AC_SCHEMA</code> 配置参数的值。有关更多信息，请参阅模式命令文件。
-i	手动初始化系统。
-lphys, stage, apply	指定逻辑恢复的级别：  <b>phys</b>  启动系统的逻辑恢复，但在物理恢复完成后停止。备份的逻辑日志必须可用。  <b>stage</b>  物理恢复完成后，从存储管理器抽取逻辑日志并将其登台到对应的表中，再启动 <code>stager</code> 。  <b>apply</b>

元素	描述
	<p>启动 <code>applier</code>。<code>applier</code> 提取存储在登台表中的事务，将其转换为 SQL，并重放操作。</p> <p>如果未列出 <code>-l</code> 选项，那么缺省的逻辑恢复级别为 <code>-lphys,stage,apply</code>。您可以指定逻辑恢复级别的任意组合，以逗号分隔。<code>-l</code> 和级别之间不允许有空格。</p> <p>有关更多信息，请参阅手动控制逻辑恢复。</p>
<code>-s</code>	将状态消息打印到屏幕。
<code>-t</code>	指定 <code>gtape</code> 作为备份实用程序。
<code>-v</code>	指定详细方式。
<code>-X</code>	指定表级别恢复。
<code>-V</code>	显示 GBase 8s 版本信息。
<code>-version</code>	显示有关 GBase 8s 的构建操作系统、构建版本号和构建日期的附加版本信息。

当您使用 `gbackuprestore` 时，可以使用 `gbackuprestore` 命令来访问 `archecker` 信息，以验证备份。有关此命令的语法的信息，请参阅 `gbackuprestore -v` 语法：验证备份。

### 手动控制逻辑恢复

您可以使用 `-l` 命令行选项手动控制 `stager` 和 `applier`。

以下示例显示了如何执行逻辑恢复。在所有示例中，模式命令文件名都是 `cmdfile`。

下面是一个典型的用法示例：

```
archecker -bvs -f cmdfile
```

该命令等同于以下命令：

```
archecker -bvs -f cmdfile -lphys,stage,apply
```

物理恢复完成后，`archecker` 实用程序启动 `stager`。`stager` 启动后，`applier` 自动启动。

在以下示例中，`-lphys` 选项执行单纯的物理恢复：

```
archecker -bvs -f cmdfile -lphys
```

在以下示例中, `-lstage` 选项启动 `archecker stager`。`stager` 从存储管理器中抽取逻辑日志记录, 并将适用的记录保存到表中。

```
archecker -bvs -f cmdfile -lstage
```

`stager` 只应在物理恢复完成后才启动。

在以下示例中, `-lapply` 选项启动了 `archecker applier`。它在 `acu_control` 表中寻找要恢复的事务。`applier` 应该在 `stager` 启动后才启动。

```
archecker -bvs -f cmdfile -lapply
```

### 使用多个存储管理器执行恢复

如果您使用多个存储管理器, 您可以通过在每个节点上配置 `archecker`, 用 `archecker` 执行表级别恢复。

要执行涉及多个存储管理器的表级别恢复:

1. 在每个节点上创建一个 `archecker` 配置文件。
2. 在每个节点上创建一个模式命令文件。
3. 通过在单个节点上执行 `archecker -DX` 命令来除去旧的恢复。
4. 通过在每个节点上执行 `archecker -bX -lphys` 命令来启动物理恢复。

**限制:** 不要使用 `-d` 选项。

5. 物理恢复完成后, 通过在包含逻辑日志记录的每个节点上执行 `archecker -bX -lstage` 命令来启动逻辑恢复。

**限制:** 不要使用 `-d` 选项。

6. 启动所有 `stager` 之后, 通过在单个节点上执行 `archecker -bX -lapply` 命令来完成恢复。

### 执行并行恢复

如果在分开的数据库空间中驻留了一张分段表, 您可以通过为每个数据库空间使用不同的模式命令文件执行多个 `archecker` 命令来并行地执行物理的表级别恢复。

在 0 级归档期间, 不能有任何会更改表的模式的打开事务。正在恢复的表或表的分段必须存在于 0 级归档中。在逻辑恢复期间不能创建或添加该表或分段。逻辑恢复期间创建的表或添加的分段将被忽略。

因为拆离的分段不再是原始表的一部分, `applier` 从该点起不会处理拆离的分段的日志记录或该分段的任何其他日志记录。`archecker` 消息日志文件中的消息将指示出现了拆离。

在本示例中, 该表跨越三个数据库空间分段。相应的模式命令文件名为 `cmdfile1`、`cmdfile2` 和 `cmdfile3`。以下命令删除先前的恢复, 然后在每个数据库空间上并行地执行物理恢复。

- `archecker -DX`
- `archecker -bvs -f cmdfile1 -lphys`
- `archecker -bvs -f cmdfile2 -lphys`
- `archecker -bvs -f cmdfile3 -lphys`

不能并行地执行逻辑恢复。

### 使用大对象恢复表

`gbackuprestore` 支持对智能大对象和二进制大对象进行表级别恢复。

- 智能大对象

表级别恢复还支持仅针对物理恢复的智能大对象（从 0 级归档恢复）。

必须用 `CREATE TABLE` 语句的 `PUT` 子名来指定要恢复智能大对象列的存储位置。恢复的智能大对象用 `create-time` 标志 `LO_NOLOG` 和

`LO_NOKEEP_LASTACCESS_TIME` 创建。如果在智能大对象列的对象表中指定这些标志，那么它们将覆盖 `LOG` 和 `KEEP ACCESS TIME` 列属性。

- 二进制大对象

表级别恢复支持恢复表空间二进制大对象，但不支持 `Blob` 空间二进制大对象。如果尝试恢复 `Blob` 空间二进制大对象，那么该值设置为 `NULL` 并发出警告。

### 删除恢复文件的时间

如果重复运行相同的 `archecker` 表级别恢复，那么必须清除先前运行中的 `archecker` 表级别恢复工作文件和表。这些工作表是指 `sysutils` 数据库中在 `archecker` 表级别恢复期间创建的 `acu_` 表。在 `archecker` 表级别恢复完成之后，将保留这些 `archecker` 表级别恢复工作文件和表，以供诊断问题之需。

通过显式运行命令 `archecker -DX` 或在运行下一条 `archecker` 表级别恢复命令时使用 `-d` 选项，可以除去这些工作文件和表。`-d` 选项表示在新的恢复开始之前除去前一次 `archecker` 表级别恢复运行中的所有文件和表。

- `gtape` 示例：`archecker -tdvs -fschema_command_file`
- `gbackuprestore` 示例：`archecker -bdvs -fschema_command_file`

## 6.1.4 archecker 模式引用

本节中的主题描述了 `archecker` 模式命令文件使用的类似 `SQL` 的语句。该文件提供 `archecker` 实用程序用于执行数据恢复的信息。

使用模式命令文件可指定源表和目标表，并定义表模式。

有关指定 `archecker` 使用哪个命令文件的信息，请参阅模式命令文件。

以下是 `archecker` 支持的语句：

- CREATE TABLE
- DATABASE
- INSERT INTO
- RESTORE
- SET

**重要：** `archecker` 实用程序文件中允许出现标准 SQL 注释，在处理中将忽略这些注释。

以下主题中描述了这些语句的语法。

### CREATE TABLE 语句

CREATE TABLE 语句描述了源表和目标表的模式。如果目标表是外部的，那么使用 CREATE EXTERNAL TABLE 语句部分中描述的 CREATE EXTERNAL TABLE 语句。

#### 语法

`archecker` 模式命令文件中使用的 CREATE TABLE 的语法与对应 GBase 8s SQL 语句相同。有关此语法的描述，请参阅《GBase 8s SQL 指南：语法》。

#### 用途

您必须在 `archecker` 模式命令文件中包含源表的模式。该模式必须和创建归档时源表的模式相同。

源表的模式未由 `archecker` 验证。如未提供准确的模式，会发生不可预测的结果。

源表不可以是同义词或视图。源表的模式仅需列列表和存储选项。将忽略诸如扩展数据块大小和锁定方式等其他属性。对于 `gbackuprestore` 归档，`archecker` 使用源表的存储空间列表来创建要从存储管理器中检索的对象列表。如果源表是分段的，必须列出所有包含源表数据的数据库空间。`archecker` 实用程序只从模式命令文件中列出的数据库空间中抽取数据。

如果源表包含约束、索引或触发器，那么在恢复期间将会自动禁用这些约束、索引或触发器。还会禁用参考目标表的外部约束。恢复完成后，约束、索引和触发器将被启用。为了顺利地执行恢复，在执行恢复前除去约束、索引和触发器。

您必须还在命令文件中包含目标表的模式。如果执行恢复时目标表不存在，将使用所提供的模式创建它。

如果目标表存在，其模式必须和命令文件中指定的模式匹配。随后将数据附加到现有的表。

#### 示例

源表和目标表的模式不一定要相同。以下示例显示了如何能够在执行数据抽取之后对源数据重新分区：

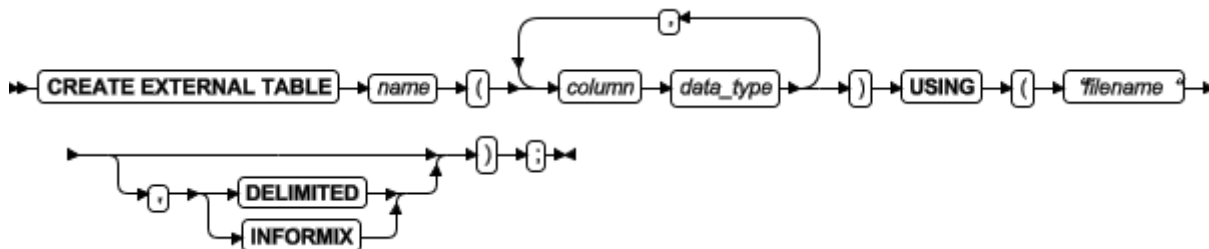
```
CREATE TABLE source (col1 integer, ...) IN dbspace1;
CREATE TABLE target (col1 integer, ...)
  FRAGMENT BY EXPRESSION
    MOD(col1, 3) = 0 in dbspace3,
    MOD(col1, 3) = 1 in dbspace4,
    MOD(col1, 3) = 2 in dbspace5;
INSERT INTO target SELECT * FROM source;
```

### CREATE EXTERNAL TABLE 语句

CREATE EXTERNAL TABLE 语句描述了外部目标表的模式。

语法

archecker 模式文件的 CREATE EXTERNAL TABLE 语句的语法和 SQL CREATE EXTERNAL TABLE 语句不同。



元素	描述
column	列名。必须符合 SQL 标识语法规则。有关更多信息，请参阅《GBase 8s SQL 指南：语法》。
data_type	列的内置数据类型。有关数据类型的更多信息，请参阅《GBase 8s SQL 指南：参考》。
filename	放置数据的文件名或管道设备名。管道设备必须在启动 archecker 实用程序之前就已存在。
名称	存储外部数据的表的名称。在当前数据库中的表、视图和同义词的名称中必须是唯一的。必须符合 SQL 数据库对象命名规则。有关更多信息，请参阅《GBase 8s SQL 指南：语法》。

## 用途

当使用 CREATE EXTERNAL TABLE 语句向外部表发送数据时，数据只从 0 级归档中抽取。逻辑日志不在外部表上前滚。

您可以为外部文件指定以下两种格式之一：

**DELIMITED:** ASCII 定界文件。这是缺省的格式。

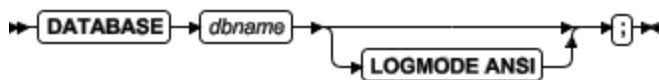
**GBASEDBT:** 内部二进制表示法。为优化性能，不应该将过滤器应用于外部表。如果过滤器存在，将出现警告以指示会将其忽略。

有关使用 CREATE EXTERNAL TABLE 语句的示例，请参阅恢复到外部表。

## DATABASE 语句

在 archecker 实用程序中，DATABASE 语句用于设置当前的数据库。

### 语法



元素	描述
dbname	当前数据库的名称。

### 用途

可以使用多条 DATABASE 语句。所有该语句后引用的表的名称都和当前的数据库关联。

如果源表的日志记录方式是 ANSI 而表的模式中使用了缺省的十进制列，那么必须声明数据库的日志记录方式。

如果未声明源数据库的日志记录方式，不会返回错误，但可能出现意外的结果和数据。

### 示例

在以下示例中，源表和目标表都驻留在同一个数据库 dbs 中。

```

DATABASE dbs;
CREATE TABLE source (...);
CREATE TABLE target (...);
INSERT INTO target SELECT * from source;

```

您可以使用多重数据库语句将表从一个数据库抽取到另一个数据库。

```

DATABASE dbs1;
CREATE TABLE source (...) IN dbspace1;

```

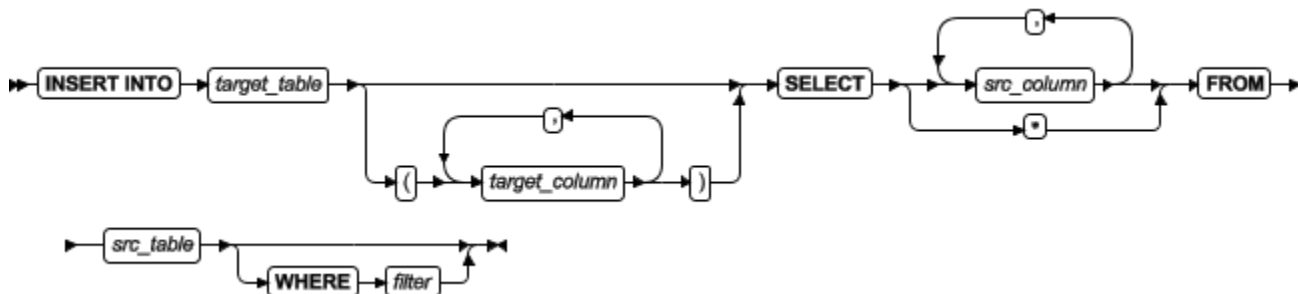
```

DATABASE dbs2;
CREATE TABLE target (...) IN dbspace2;
INSERT INTO dbs2:target SELECT * FROM dbs1:source;
    
```

**INSERT 语句**

INSERT 语句告知 archecker 实用程序要抽取的表以及用于放置抽取数据的位置。

语法



元素	描述
过滤器 (filter)	<p>INSERT 语句支持以下过滤器：</p> <ul style="list-style-type: none"> <li>• =, !=, &lt;&gt;</li> <li>• &gt;, &gt;=, &lt;, &lt;=</li> <li>• [NOT] MATCHES, [NOT] LIKE</li> <li>• IS [NOT] NULL</li> <li>• AND, OR</li> <li>• TODAY, CURRENT</li> </ul> <p>archecker 实用程序不支持以下操作程序：</p> <ul style="list-style-type: none"> <li>• 聚集</li> <li>• 函数和过程</li> <li>• 下标</li> <li>• 子查询</li> <li>• 视图</li> <li>• 连接</li> </ul> <p>过滤器只能应用于纯物理的恢复。</p>
src_column	要抽取的列的列表。
src_table	从中恢复数据的归档中的源表。



元素	描述
target_column	数据将被恢复的一个或多个目标列。
target_table	数据将被恢复的目标表。

示例

以下示例演示了 INSERT 语句最简单的格式。该语句从源表中抽取所有行和列到目标表。

```
INSERT INTO target SELECT * FROM source;
```

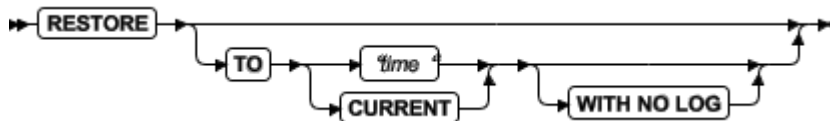
也可以抽取一部分列。在以下示例中，源表中只有两列被插入到目标表中。

```
CREATE TABLE source (col1 integer, col2 integer, col3 integer, col4 integer);
CREATE TABLE target (col1 integer, col2 integer);
INSERT INTO target (col1, col2) SELECT col3, col4 FROM source;
```

### RESTORE 语句

RESTORE 语句是将表恢复到特定时间点的可选命令。

语法



元素	描述
"time"	该表要恢复到的日期和时间。

用途

TO 子句用来将表恢复到特定的时间点，该时间点由日期和时间指定，或由保留字 CURRENT 指定。

一个命令文件中只能指定一个 RESTORE 语句。如果命令文件中未出现该语句，那么系统将被恢复到最近一次使用逻辑日志的时间。

如果出现了 WITH NO LOG 子句，那么只执行物理恢复。此外，两个额外列及索引未添加到目标表。纯物理的恢复只基于 0 级归档。

**提示：** 当没有逻辑日志时，请使用该选项。您将收不到任何有关逻辑恢复的消息。

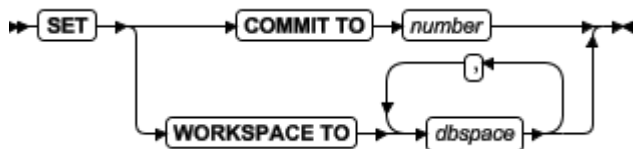
示例

```
RESTORE TO CURRENT WITH NO LOG;
```

## SET 语句

SET 语句控制表级卸装库中的不同功能。

语法



元素	描述
number	设置物理恢复期间在提交前插入的记录数量。缺省值为 1000。
dbspace	用作工作存储空间的数据库空间。缺省值是根数据库空间。不能用临时数据库空间作为工作存储空间。

archecker 实用程序在逻辑恢复中为逻辑日志记录的登台创建了几个表。这些表在 sysutils 数据库中创建，并存储在工作存储空间中。

示例

```
SET COMMIT TO 20000;
SET WORKSPACE to dbspace1;
```

### 模式命令文件示例

本节包含为不同数据恢复方案显示不同命令文件语法的示例。

#### 简单模式命令文件

本示例中的模式命令文件从 dbspace1 的最近一次 0 级备份中抽取一个表。数据放置在表 test1:tlr 中，并且应用了日志，以使表 tlr 的时间变为当前时间点。

```
database test1;
create table tlr (
  a_serial serial,
  b_integer integer,
  c_char char,
  d_decimal decimal
) in dbspace1;
insert into tlr select * from tlr;
```

从先前备份中恢复表

本示例中的模式命令文件从 `dbspace1` 的 0 级备份中抽取一个表。使用了逻辑日志将表的时间变为“2003-01-01 01:01:01”。数据放在表 `test1:tlr` 中。

```
database test1;
create table tlr (
  a_serial serial,
  b_integer integer,
  c_char char,
  d_decimal decimal
) in dbspace1;
insert into tlr select * from tlr;
restore to '2003-01-01 01:01:01';
```

#### 恢复到不同的表

本示例中的模式命令文件从 `dbspace1` 的最近备份中抽取名为 `test1:tlr` 的表,并将数据放到表 `test1:tlr_dest` 中。

```
database test1;
create table tlr (
  a_serial serial,
  b_integer integer,
  c_char char(20),
  d_decimal decimal,
) in dbspace1;
create table tlr_dest (
  a_serial serial,
  b_integer integer,
  c_char char(20),
  d_decimal decimal
) in dbspace2;
insert into tlr_dest select * from tlr;
```

#### 抽取列的子集

本示例中的模式命令文件从 `dbspace1` 的最近备份中抽取表 `test1:tlr`,并将数据子集放到表 `test1:new_dest` 中

```
database test1;
create table tlr (
  a_serial serial,
  b_integer integer,
  c_char char(20),
```

```
d_decimal decimal
) in dbspace1;
create table new_dest (
  X_char char(20),
  Y_decimal decimal,
  Z_name char(40)
) in dbspace2;
insert into new_dest (X_char, Y_decimal) select c_char,d_decimal from tlr;
```

#### 使用数据过滤

本示例中的模式命令文件从 dbspace1 的最近备份中抽取表 test1:tlr，并且只将数据放到表 test1:tlr 中列出条件为真的地方。

**重要：** 过滤器只能应用于物理恢复。

```
database test1;
create table tlr (
  a_serial serial,
  b_integer integer,
  c_char char(20),
  d_decimal decimal,
) in dbspace1;
insert into tlr
select * from tlr
where c_char matches 'john*'
and d_decimal is NOT NULL
and b_integer > 100;
restore to current with no log;
```

#### 恢复到外部表

本示例中的模式命令文件从 dbspace1 的最近备份中抽取名为 test1:tlr 的表，并将数据放到名为 /tmp/tlr.unl 的文件中。

```
database test1;
create table tlr
(a_serial serial,
  b_integer integer
) in dbspace1;
create external table tlr_dest
(a_serial serial,
  b_integer integer
```

```
) using ("/tmp/tlr.unl", delimited );  
insert into tlr_dest select * from tlr;  
restore to current with no log;
```

#### 恢复多个表

本示例中的模式命令文件从 dbspace1 的最近备份中抽取表 test1:tlr 和 test1:tlr\_2，并且将数据放到 test1:tlr\_1\_dest 和 test1:tlr\_2\_dest。这是恢复多个表的有效方法，因为它只需要扫描一次归档和逻辑日志文件。

```
database test1;  
create table tlr_1  
  ( columns ) in dbspace1;  
create table tlr_1_dest ( columns );  
create table tlr_2  
  ( columns ) in dbspace1;  
create table tlr_2_dest ( columns );  
insert into tlr_1_dest select * from tlr_1;  
insert into tlr_2_dest select * from tlr_2;
```

#### 执行分布式的恢复

本示例中的模式命令文件从 dbspace1 的最近备份中抽取表 test:tlr\_1，并将数据放到数据库服务器 rem\_srv 上的表 rem\_dbs:tlr\_1 中。

```
database rem_dbs  
create table tlr_1  
  ( columns );  
database test1;  
create table tlr_1  
  ( columns ) in dbspace1;  
insert into rem_dbs@rem_srv.tlr_1  
  select * from tlr_1;
```

## 7 备份与恢复配置参数引用

### 7.1 备份与恢复配置参数

这些主题描述了用于 gbackuprestore、gtape 和 archecker 实用程序的配置参数。

其中大部分配置参数可在 `onconfig` 文件中设置。但是，某些 `archecker` 配置参数要在 `AC_CONFIG` 文件中设置。

请务必配置您的存储管理器。根据您的存储管理器，可以设置不同的 `gbackuprestore` 配置参数。如果您使用的是第三方存储管理器，那么在启动 `gbackuprestore` 之前请参阅配置第三方存储管理器。

下表描述了每个参数的以下属性（如果相关）。

属性	描述
ac_config.std 值	对于 <code>archecker</code> 配置变量。 <code>ac_config.std</code> 文件中显示的缺省值。
onconfig	<code>onconfig</code> 配置变量。 <code>onconfig</code> 文件中显示的缺省值。
如果值不存在	如果 <code>onconfig</code> 文件缺少该参数，那么数据库服务器将提供该值。 如果该值出现在 <code>onconfig</code> 中，数据库服务器将使用 <code>onconfig</code> 值。如果 <code>onconfig</code> 中不存在该值，数据库服务器将使用该值。
单位	表示该参数用的单位
值的范围	该参数的有效值
生效	对参数值的更改将影响 <code>gbackuprestore</code> 操作的时间。 除非另有指定，否则可以在备份与恢复之间更改参数值。
引用	对进一步讨论的交叉引用

### 7.1.1 onconfig 文件中的 gbackuprestore 和 gtape 配置参数

这些主题包含有关 `gbackuprestore` 和 `gtape` 实用程序配置参数的参考信息

**重要：** `gbackuprestore` 不使用 `TAPEDEV`、`TAPEBLK`、`TAPESIZE`、`LTAPEBLK` 和 `LTAPESIZE` 配置参数。`gbackuprestore` 检查 `LTAPEDEV` 是否设置为 `/dev/null`（UNIX™ 上）。

#### BACKUP\_FILTER 配置参数

使用 `BACKUP_FILTER` 配置参数可指定用于 `gbackuprestore` 或 `gtape` 实用程序的外部过滤器程序的路径名和任何选项。

#### onconfig.std 值

未设置。备份数据未过滤。

#### 值

命令的路径名和任何选项。缺省情况下，路径名相对于 `$GBS_HOME/bin` 目录，否则，路径名必须是程序的绝对路径。如果您要包括命令行选项，那么必须使用单引号括住过滤器名称和选项。

## 生效

编辑 `onconfig` 文件，并且 `gbackuprestore` 或 `gtape` 启动后。

## 用途

该过滤器在备份数据之前对其进行变换（例如，将其压缩）。然后变换的数据将被备份，并且会存储成单个文件。执行恢复时，必须将数据变换回其原始格式。在通过设置 `RESTORE_FILTER` 配置参数进行恢复之前，指定相应程序来变换数据。

为了安全，过滤器不能有非特权用户写许可权。对过滤器的许可权与对 GBase 8s 服务器或实用程序所调用的其他可执行文件的许可权相同。

注：如果在 `onconfig` 文件中设置 `BACKUP_FILTER` 参数，那么 `LTAPESIZE` 配置参数不能设置为 0。否则，在将逻辑日志备份到磁盘上的目录时，`gbackuprestore` 或 `gtape` 实用程序会返回错误。错误消息为：

```
当已设置 BACKUP_FILTER 配置参数时，LTAPESIZE 配置  
参数无法设置为 0；请更改 LTAPESIZE 的值。
```

程序结束。

变通方法是将 `LTAPESIZE` 配置参数设置为高值。日志文件不会大幅高于 `LOGSIZE` 配置参数。使用 `LOGSIZE` 中的值作为此数据库的上限。

在 `BACKUP_FILTER` 配置参数中指定过滤器信息时，请指定过滤器程序的路径名和任何选项，如以下示例中所示：

```
BACKUP_FILTER /bin/compress
```

此过滤器所生成的输出将作为单个对象保存到存储管理器中。

`BACKUP_FILTER` 配置参数可包括命令行选项以及过滤器名称。例如，指定：

```
BACKUP_FILTER 'my_encrypt -file /var/adm/encryption.pass'
```

## BAR\_ACT\_LOG 配置参数

使用 `BAR_ACT_LOG` 配置参数可指定 `gbackuprestore` 活动日志的完整路径名。

### onconfig 值

`none`

### 值的范围

完整路径名

## 生效

`gbackuprestore-driver` 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

## 用途

您应该指定具有适当可用空间量的现有目录的路径，或者使用 `$GBS_HOME/bar_act.log`。

无论何时发生备份或恢复活动或者出错，`gbackuprestore` 都会向活动日志写入一条简要说明。文件格式与数据库服务器消息日志的格式相似。可以检查活动日志以确定 `gbackuprestore` 操作的结果。

如果 `BAR_ACT_LOG` 配置参数指定的文件不存在，那么会创建该文件。如果 `gbackuprestore` 命令（或与 `gbackuprestore` 相关的任何实用程序，如 `onmsync` 实用程序）从未在系统上运行，那么该文件不存在。

`sysbaract_log` 表是从 `BAR_ACT_LOG` 指定的文件读取数据的系统监视接口伪表。如果尝试在不存在 `BAR_ACT_LOG` 文件的系统上查询 `sysbaract_log`，那么会返回以下错误：

```
244: 无法执行物理顺序读来访问下一行。
```

```
101: ISAM 错误: 文件未打开。
```

仅指定文件名时的用法

如果在 `BAR_ACT_LOG` 配置参数中仅指定文件名，`gbackuprestore` 将在您启动 `gbackuprestore` 的工作目录中创建 `gbackuprestore` 活动日志。例如：如果您从 UNIX™ 上的 `/usr/mydata` 启动 `gbackuprestore`，那么会将活动日志写入该目录中。

对于 UNIX™，如果数据库服务器启动连续逻辑日志备份，`gbackuprestore` 会写入该数据库服务器的工作目录中的 `gbackuprestore` 活动日志。

## BAR\_BSALIB\_PATH 配置参数

使用 `BAR_BSALIB_PATH` 配置参数可为所使用的存储管理器的 XBSA 共享库指定路径名和文件名。

### onconfig 值

UNIX™: none

### 生效

`gbackuprestore-driver` 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

## 用途

`gbackuprestore` 和存储管理器依靠共享库来彼此结合。请为您的存储管理器库配置 `BAR_BSALIB_PATH` 配置参数。对 `BAR_BSALIB_PATH` 的支持特定于平台。请检查您的机器说明，以确定是否可以对您的操作系统使用该配置参数。可以在备份与恢复之间更改 `BAR_BSALIB_PATH` 的值。

要确保发生这种集成，请指定共享库的路径名。请指定以下选项之一：



UNIX:

- 将存储管理器库放到缺省目录中。  
例如，Solaris 的后缀是 so，因此在 Solaris 系统上指定 \$GBS\_HOME/lib/ibsad001.so。  
例如，Solaris 的后缀是 so，因此在 Solaris 系统上指定 \$GBS\_HOME/lib/libbsapsm.so。
- 将存储管理器库放在任意目录下，并创建一个从 \$GBS\_HOME/lib/ibsad001.platform\_extension 到它的符号链接。  
如果您使用 Storage Manager，那么创建 \$GBS\_HOME/lib/libbsa.platform\_extension 的符号链接或将 BAR\_BSALIB\_PATH 设置为此绝对路径值。  
如果使用 GBase Tivoli Storage Manager (TSM)，那么创建 \$GBS\_HOME/lib/libtxbsa.platform\_extension 的符号链接或将 BAR\_BSALIB\_PATH 设置为此绝对路径值。
- 设置 LD\_LIBRARY\_PATH 环境变量。例如，将 LD\_LIBRARY\_PATH 设置为 \$GBS\_HOME/lib。

如果参数 BAR\_BSALIB\_PATH 缺失或没有值，并且数据库服务器无法打开您的平台的 XBSA 共享库，那么 GBACKUPRESTORE 会尝试使用 GBase 8s 主存储管理器 作为所有平台中的存储管理器。

**提示：** 确保在恢复中共享库可以访问存储管理器中的备份数据。不能备份到一个存储管理器而从其他存储管理器恢复。

### **BAR\_CKPTSEC\_TIMEOUT 配置参数**

BAR\_CKPTSEC\_TIMEOUT 配置参数指定在执行外部备份时，RS 辅助服务器应等待检查点从主服务器到达的时间量（以秒为单位）。

#### **onconfig 值**

15

如果值不存在

15

单位

秒

值的范围

5 到 CKPTINTVL 配置参数值的两倍

生效

编辑 `onconfig` 文件并重新启动数据库服务器后。

通过运行 `gadmin -wf` 命令在 `onconfig` 文件中动态重置该值时。

通过运行 `gadmin -wm` 命令重置会话的值时。

#### 用途

当在 RS 辅助服务器上执行外部备份时，辅助服务器必须等待检查点从主服务器的逻辑日志中到达。检查点会清空缓冲区，将其中的内容保存到磁盘中，并阻塞包含临时表的用户事务。如果主服务器上的检查点在超时时段内未完成，那么 RS 辅助服务器上的备份将失败。您可以将 `BAR_CKPTSEC_TIMEOUT` 配置参数设置为执行外部备份时，RS 辅助服务器应等待检查点从主服务器到达的时间量（以秒计）。

### BAR\_DEBUG 配置参数

使用 `BAR_DEBUG` 配置参数可指定数据库服务器在 `gbackupstore` 活动日志中捕获的调试信息量。

#### `onconfig` 值

`BAR_DEBUG 0`

#### 值

0 = 不显示调试信息。

1 = 打印少量信息

2 = 每次 `gbackupstore` 符合以下条件时打印消息：

- 进入函数。
- 退出函数。该消息包括函数的返回码。

3 = 打印退出和进入信息，以及其他详细信息。

4 = 还打印有关 `gbackupstore` 并行操作的信息。

5 = 还打印有关以下项的信息：

- 要备份或恢复的对象。
- 与 `bar_action` 表对应的 `act_node` 结构。

6 = 打印有关以下项的其他信息：

- 要备份或恢复的对象。
- 与 `bar_action` 表对应的 `act_node` 结构。

7 = 还打印：

- 与 `bar_instance` 表对应的 `ins_node` 结构的内容相关信息。

- 对 `bar_action` 表的修改的相关信息。
- 有关恢复的逻辑日志和对象的信息。
- `sysutils` 数据库上完成的 SQL 语句以及返回的 `SQLCODES`。

8 = 还打印已归档且已恢复的所有页面的页面标题。此设置需要大量空间。

9 = 打印以下项的内容：

- 初始化后的 `bar_ins` 结构。
- 冷恢复的对象描述符。

## 生效

针对任何当前正在执行的 `gbackuprestore` 命令和任何后续命令编辑 `onconfig` 文件后立即生效。当您更新 `BAR_DEBUG` 时当前正在执行的任何 `gbackuprestore` 命令都会读取 `BAR_DEBUG` 的新值并以新的级别打印调试消息。

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

## 用途

缺省值 0 不显示调试信息。将 `BAR_DEBUG` 配置参数设置为较高值，以在 `gbackuprestore` 活动日志中显示更详细的调试信息。

在会话期间，您可以在 `onconfig` 文件中动态更新 `BAR_DEBUG` 的值。

## BAR\_DEBUG\_LOG 配置参数

使用 `BAR_DEBUG_LOG` 参数可指定 `gbackuprestore` 调试日志的位置和名称。

### onconfig 值

```
/usr/gbasedbt/bar_dbug.log
```

### 如果值不存在

```
UNIX™: /tmp/bar_dbug.log
```

## 生效

`gbackuprestore` 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

## 用途

出于安全性原因，请将 `BAR_DEBUG_LOG` 配置参数设置为带有受限许可权的目录，如 `$GBS_HOME` 目录。

## BAR\_HISTORY 配置参数

使用 BAR\_HISTORY 配置参数可指定当您使用 onsmsync 使旧备份到期时，sysutils 数据库是否保留备份历史记录。

### onconfig.std 值

none

### 如果值不存在

0

### 值的范围

0 = 从 sysutils 数据库中除去到期的备份对象的记录

1 = 在 sysutils 数据库中保留到期的备份对象的记录

### 生效

onsmsync 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 onconfig 文件中动态重置该值时。

### 用途

如果将值设置为 0, 那么 onsmsync 从 sysutils 数据库中将已到期的备份对象的 bar\_object、bar\_action 和 bar\_instance 行除去。如果将值设置为 1, 那么 onsmsync 将 bar\_action 行中的 act\_type 值设置为 7, 并在 sysutils 数据库中为已到期的备份对象保留 bar\_action 和 bar\_instance 行。如果未将 BAR\_HISTORY 设置为 1, 那么恢复历史记录将被除去。

不管 BAR\_HISTORY 的值是什么, 当存储管理器使该对象到期时, onsmsync 都将从紧急引导文件中除去描述备份对象的行并从存储管理器中除去该对象。

有关 onsmsync 的更多信息, 请参阅 onsmsync 实用程序。

## BAR\_IXBAR\_PATH 配置参数

使用 BAR\_IXBAR\_PATH 配置参数来更改 gbackuprestore 引导文件的路径和名称。

### onconfig 值

UNIX<sup>™</sup> 或 Linux<sup>™</sup>: \$GBS\_HOME/etc/ixbar.servernum

### 值的范围

gbackuprestore 引导文件的完整路径名

### 生效

当 gbackuprestore 或 onsmsync 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

#### 用途

缺省情况下，将在 UNIX 或 Linux 上的 `$GBS_HOME/etc` 文件夹中创建 `gbackuprestore` 引导文件。此文件的缺省名称为 `ixbar.servernum`，其中 `servernum` 是 `SERVERNUM` 配置参数的值。

例如，对于 `SERVERNUM` 配置参数等于 41 的情况，在 UNIX 中，缺省情况下将使用此路径和名称创建 `gbackuprestore` 引导文件：

```
BAR_IXBAR_PATH $GBS_HOME/etc/ixbarboot.41
```

您可以更改路径以在其他位置中创建该文件。例如，如果要在目录 `/usr/gbasedbt` 中使用名称 `ixbar.new` 创建 `gbackuprestore` 引导文件，请指定：

```
BAR_IXBAR_PATH=/usr/gbasedbt/ixbar.new
```

### **BAR\_MAX\_BACKUP 配置参数**

使用 `BAR_MAX_BACKUP` 参数可指定每个 `gbackuprestore` 命令允许的最大并行进程数。

#### **onconfig 值**

0

如果值不存在

4

#### **单位**

`gbackuprestore` 进程

#### **值**

0 = 系统上允许的最大进程数

1 = 串行备份或恢复

n = 所创建进程的指定数量

#### **生效**

`gbackuprestore` 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

虽然数据库服务器的 `BAR_MAX_BACKUP` 缺省值是 4，但 `onconfig.std` 的值是 0。

UNIX™ 支持并行备份。

指定串行备份与恢复

要执行串行备份或恢复(包括串行的整个系统备份或恢复),请将 `BAR_MAX_BACKUP` 设置为 1。

#### 指定并行备份与恢复

要指定并行备份与恢复(包括并行的整个系统备份与恢复),请将 `BAR_MAX_BACKUP` 设置为高于 1 的值。例如,如果将 `BAR_MAX_BACKUP` 设置为 5 并执行 `gbackuprestore` 命令,那么 `gbackuprestore` 同时创建的最大进程数为 5。`BAR_MAX_BACKUP` 最大可以配置为与存储设备最大数或物理备份与恢复可用的最大流数相等。`gbackuprestore` 按并行资源有效使用的大小对数据库空间进行分组。

如果将 `BAR_MAX_BACKUP` 设置为 0,系统将按照需要的数目创建 `gbackuprestore` 进程。`gbackuprestore` 进程的数目只受存储空间的数目或数据库服务器可用内存大小的限制,取两个值中较小的一个。

可用内存大小基于 `SHMTOTAL`。`gbackuprestore` 执行以下计算,其中 `N` 是允许执行的 `gbackuprestore` 进程的最大数目:

$$N = \text{SHMTOTAL} / (\text{传送缓冲区数} * \text{传送缓冲区的大小} / 1024)$$

如果 `SHMTOTAL` 是 0,那么 `BAR_MAX_BACKUP` 将重置为 1。如果 `N` 大于 `BAR_MAX_BACKUP`,那么 `gbackuprestore` 将使用 `BAR_MAX_BACKUP` 的值。否则,`gbackuprestore` 将启动 `N` 个备份或恢复进程。

#### **BAR\_NB\_XPORT\_COUNT 配置参数**

使用 `BAR_NB_XPORT_COUNT` 配置参数可指定每个 `gbackuprestore_d` 进程可以用来与数据库服务器交换数据的数据缓冲区的数量。

##### **onconfig 值**

20

##### **如果值不存在**

20

##### **单位**

缓冲区

##### **值的范围**

3 到无穷大

##### **生效**

`gbackuprestore` 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

此参数的值将影响 gbackupstore 的性能。例如，如果将 BAR\_NB\_XPORT\_COUNT 设置为 5 并发出 5 个 gbackupstore 命令，那么生成的 25 个 gbackupstore 进程将总共使用 125 个缓冲区。

要计算每个 gbackupstore\_d 进程需要的内存大小，请使用以下公式。有关系统的页大小的信息，请参阅发行说明：

$$\text{required\_memory} = (\text{BAR\_NB\_XPORT\_COUNT} * \text{BAR\_XFER\_BUF\_SIZE} * \text{page\_size}) + 5 \text{ MB}$$

## BAR\_PERFORMANCE 配置参数

使用 BAR\_PERFORMANCE 配置参数可指定性能统计信息的类型，以报告至备份与恢复操作的 gbackupstore 活动日志。

### onconfig 值

0

### 单位

统计信息的级别

### 值

0 = 不收集性能统计信息

1 = 报告在 GBase 8s 实例与存储管理器之间传输数据时所耗用的时间。

2 = 采用活动日志和错误日志中的时间戳记以微秒为单位报告 gbackupstore 处理性能

3 = 报告这两种微秒时间戳记并传输统计信息。

### 生效

gbackupstore 启动时

通过运行 gadmin -wf 或等效 SQL 管理 API 命令在 onconfig 文件中动态重置该值时。

### 用途

例如，如果将 BAR\_PERFORMANCE 设置为 3，gbackupstore 将在活动日志中报告 GBase 8s 实例与存储管理器之间传输数据时所耗用的时间。如果将 BAR\_PERFORMANCE 设置为 0 或不设置，gbackupstore 将不报告性能统计信息。

- 要关闭性能监视，请将该值设置为 0。这是缺省值。
- 要显示 GBase 8s 实例和存储管理器之间传输数据时所耗用的时间，请将该参数设置为 1。
- 要显示时间戳记（以微秒计），请将该参数设置为 2。

- 要显示时间戳记和传输统计信息，请将该参数设置为 3。

### BAR\_PROGRESS\_FREQ 配置参数

使用 BAR\_PROGRESS\_FREQ 配置参数可指定 gbackuprestore 活动日志中出现备份与恢复操作的进度消息的频率（以分钟为单位）。

#### onconfig 值

0

#### 如果值不存在

0

#### 单位

分钟

#### 值的范围

0，然后是 5 到无穷大

#### 生效

gbackuprestore 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 onconfig 文件中动态重置该值时。

#### 用途

示例：如果将 BAR\_PROGRESS\_FREQ 设置为 5，那么 gbackuprestore 每 5 分钟报告一次该对象的已备份或恢复的百分比。如果将 BAR\_PROGRESS\_FREQ 设置为 0 或不设置该参数，那么 gbackuprestore 不将任何进度消息写入到活动日志中。

指定一个大于等于 5 分钟的值。请不要将 BAR\_PROGRESS\_FREQ 设置为 1、2、3 或 4，gbackuprestore 会将其自动重置到 5 以防止 gbackuprestore 活动日志溢出。

如果 gbackuprestore 无法确定备份或恢复对象的大小，它将报告发送到数据库服务器的传送缓冲区数，而不是报告对象已备份或恢复的百分比。

### BAR\_RETRY 配置参数

使用 BAR\_RETRY 配置参数可指定如果第一次尝试失败，gbackuprestore 应该尝试数据备份、逻辑日志备份或恢复操作的次数。

#### onconfig 值

1

#### 如果值不存在



1

## 单位

整数

## 值的范围

0 = BAR\_ABORT, 停止备份/恢复的其余部分

1 = BAR\_CONT, 继续备份/恢复的其余部分

n = 2 到 32766

## 生效

gbackuprestore 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

## 用途

BAR\_RETRY 参数的设置按照以下方式确定 gbackuprestore 的行为:

- 如果设置为 0 (BAR\_ABORT), 那么在存储空间或逻辑日志出错时, gbackuprestore 将停止备份或恢复会话, 返回错误并退出。如果 gbackuprestore 以并行方式运行, 那么已经运行的进程将结束但是不启动新进程。
- 如果设置为 1 (BAR\_CONT), 那么 gbackuprestore 将对特定存储空间停止备份或恢复尝试, 返回错误, 并尝试备份或恢复任何剩余的存储空间或逻辑日志。
- 如果设置为具体数字 (重试备份与恢复操作 2 到 32766 次), 那么 gbackuprestore 将尝试备份或恢复该存储空间或逻辑日志指定次数, 然后放弃尝试并继续到下一个存储空间或逻辑日志。

## BAR\_SIZE\_FACTOR 配置参数

在备份之前, 使用 BAR\_SIZE\_FACTOR 配置参数来提高备份对象大小的估计值。

## onconfig 值

0

## 值的范围

正整数

## 生效

数据库服务器启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

## 用途

此估计值在备份之前进行处理，并以特定方式进行计算，以便存储管理器可以相应分配存储介质。由于备份是联机进行的，因此要备份的页数可能会在备份过程中发生变化。一些存储管理器非常严格，如果备份估计值过低，备份会出现错误。

在将 `BAR_SIZE_FACTOR` 值传达到存储管理器前，`BAR_SIZE_FACTOR` 被视为是原有备份对象大小的百分比加上估计值。`BAR_SIZE_FACTOR` 只用于数据库空间备份对象，不用于逻辑日志备份对象。

计算新的估计备份对象大小的公式为：

$$\text{new\_estimate} = \text{original\_estimate} \times (1 + (\text{BAR\_SIZE\_FACTOR} / 100))$$

在特定服务器环境中将此参数设为何值取决于在备份或归档期间系统上的活动。因此，要确定值的大小需要根据个人对该系统的经验而论。

## `BAR_XFER_BUF_SIZE` 配置参数

使用 `BAR_XFER_BUF_SIZE` 配置参数可指定每个传输缓冲区的大小。

### onconfig 值

31（如果 `PAGESIZE` 是 2 千字节）

15（如果页面大小是 4 千字节）

### 单位

页数

## `GBACKUPRESTORE` 与 GBase 8s 主存储管理器一起使用时的值的范围

1 - 32768

`BAR_XFER_BUF_SIZE` 的值以 GBase 8s 基页大小为单位。对于 Linux、Solaris 和 HP，GBase 8s 基页大小为 2K。

1 - 15 个页面（当 `PAGESIZE` 是 4 千字节时）

1 - 31 个页面（当 `PAGESIZE` 是 2 千字节时）

最大缓冲区大小是 64 千字节，因此 `BAR_XFER_BUF_SIZE * 页面大小 <= 64 千字节`。

## `GBACKUPRESTORE` 与其他存储管理器一起使用时的值的范围

1 - 15（如果 GBase 8s 基页大小为 4 千字节）

1 - 31（如果 GBase 8s 基页大小为 2 千字节）

最大缓冲区大小是 64 千字节，因此 `BAR_XFER_BUF_SIZE * 页面大小 <= 64 千字节`。

## 生效

`gbackuprestore` 启动时

通过运行 `gadmin -wf` 或等效 SQL 管理 API 命令在 `onconfig` 文件中动态重置该值时。

## 用途

数据库服务器将缓冲区传递到 `gbackuprestore` 和存储管理器。

要计算存储空间或逻辑日志备份中传输缓冲区的大小，请使用以下公式：

```
一个传输缓冲区大小 = BAR_XFER_BUF_SIZE * 基页 pagesize
```

要计算数据库服务器对于每个传输缓冲区需要多少内存，请使用以下公式：

```
内存 = (BAR_XFER_BUF_SIZE * 基页 PAGESIZE) + 500
```

额外的 500 字节是系统开销。例如，如果 `BAR_XFER_BUF_SIZE` 为 15，那么传输缓冲区应该为 61,940 字节。

传输缓冲区的数量（针对每个备份流）由 `BAR_NB_XPORT_COUNT` 指定，而并行备份流的数量由 `BAR_MAX_BACKUP` 指定。

**限制：**不能在备份与恢复之间更改缓冲区的大小。`AC_TAPEBLOCK` 和 `AC_LTAPEBLOCK` 的值需要与归档时的 `BAR_XFER_BUF_SIZE` 值相同。

例如，对于 Linux 上大小为 128\*2048（值为 262144）的传输缓冲区，请指定：

```
BAR_XFER_BUF_SIZE 128
```

## ISM\_DATA\_POOL 配置参数

`ISM_DATA_POOL` 配置参数（当在数据库服务器的 `onconfig` 文件中列出时）指定用于备份存储空间的卷池。

### onconfig 值

```
ISMData
```

## 生效

`gbackuprestore` 启动时

## 用途

该参数的值可以是 Storage Manager 识别的任意卷池。如果此参数不存在，那么 Storage Manager 使用 `ISMData` 卷池。有关详细信息，请参阅《GBase 8s Storage Manager 管理员指南》。

## ISM\_LOG\_POOL 配置参数

ISM\_LOG\_POOL 参数（当在数据库服务器的 onconfig 文件中列出时）指定用于备份逻辑日志的卷池。

### onconfig 值

ISMLogs

### 生效

gbackuprestore 启动时

### 用途

该参数的值可以是 Storage Manager 识别的任意卷池。如果此参数不存在，那么 ISM 使用 ISMLogs 卷池。有关详细信息，请参阅《GBase 8s Storage Manager 管理员指南》。

## LTAPEBLK 配置参数

使用 LTAPEBLK 配置参数可指定当使用 gtape 进行数据库空间备份时将逻辑日志备份到的设备的块大小。

LTAPEBLK 还指定当使用 gload 或 gunload 的 -l 选项时将数据装入或卸载到的设备的块大小。如果要使用 gload 或 gunload，那么可以在命令行中指定不同的块大小。

### onconfig 值

在 UNIX™ 上：32

### 单位

千字节

### 值的范围

大于 (页面大小/1024) 的值

要获取页面大小，请运行 gstat -b 命令。

### 生效

对于 gtape:

- 当您执行 gtape 时。
- 通过运行 gadmin -wf 命令在 onconfig 文件中动态重置该值时。

对于 gload 和 gunload: 当数据库服务器关闭并重新启动时

### 用途

将 LTAPEBLK 指定为磁带设备所允许的最大块大小。当您指定块大小时，数据库服务器不检查磁带设备。验证 LTAPEDEV 磁带设备是否可以读取所指定的块大小。如果不能，您可能无法读取磁带。

**仅限 UNIX:** UNIX dd 实用程序可以验证 LTAPEDEV 磁带设备是否可以读取块大小。大多数 UNIX 系统都提供了该实用程序。

如果指定 LTAPEBLK 值，那么 gbackuprestore 将忽略该值。

### LTAPEDEV 配置参数

使用 LTAPEDEV 配置参数可指定当使用 gtape 进行备份时将逻辑日志备份到的设备或目录文件系统。

LTAPEDEV 配置参数还指定当使用 gload 或 gunload 的 -l 选项时将数据装入或卸载到的设备。如果要使用 LTAPEDEV 指定用于 gunload 或 gload 的设备，那么 TAPEDEV 的相同信息对于 LTAPEDEV 也相关。

### onconfig 值

在 UNIX™ 上: /dev/tapedev

### 如果不存在

在 UNIX 上: /dev/null

### 生效

对于 gtape:

- 当执行 gtape 时，设置为磁带设备的情况下。
- 当数据库服务器关闭并重新启动时，在 UNIX 上设置为 /dev/null 的情况下。
- 通过运行 gadmin -wf 命令在 onconfig 文件中动态重置该值时。
- 通过运行 gadmin -wm 命令重置会话的值时。

对于 gload 和 gunload: 当数据库服务器关闭并重新启动时

### 用途

**警告:** 当您使用 gbackuprestore 备份逻辑日志时，请勿将 LTAPEDEV 设置为 /dev/null 或 nul。

如果在 LTAPEDEV 配置参数中指定磁带设备，那么 gbackuprestore 将忽略该值。

**重要:** 仅当不需要备份逻辑日志时，才将 LTAPEDEV 设置为 /dev/null 或保留为空白（UNIX 上）。将 LTAPEDEV 的值更改为 /dev/null 之前，必须使数据库服务器处于脱机状态。

当您 will LTAPEDEV 设置为 /dev/null 时:

- 数据库服务器将释放逻辑日志，而不要求您备份这些日志。虽然逻辑日志未被标记为已释放，但数据库服务器可以重用它们。

- `gbackuprestore` 活动日志将显示警告和返回码 152。由于数据库服务器把不再是当前的逻辑日志标记为已备份的日志，所以 `gbackuprestore` 找不到要备份的逻辑日志。那些日志中的所有事务都将丢失，并且无法恢复。

如果在 `LTAPEDEV` 设置为空 (`null`) 的情况下执行整个系统的备份，必须在恢复期间使用 `gbackuprestore -r -w -p` 命令通知 `gbackuprestore` 不需要恢复日志。

### LTAPESIZE 配置参数

使用 `LTAPESIZE` 配置参数可指定当使用 `gtape` 进行备份时将逻辑日志备份到的设备的最大磁带大小。

`LTAPESIZE` 配置参数还指定当使用 `gload` 或 `gunload` 的 `-l` 选项时将数据装入或卸载到的设备的最大磁带大小。如果要使用 `gload` 或 `gunload`，那么可以在命令行上指定不同的磁带大小。如果要使用磁带的全部容量，请将 `LTAPESIZE` 设置为 0。

#### onconfig 值

0

#### 单位

千字节

#### 值的范围

0 或任何正数。真实值依赖于操作系统。

#### 生效

对于 `gtape`:

- 当您执行 `gtape` 时。
- 通过运行 `gadmin -wf` 命令在 `onconfig` 文件中动态重置该值时。

对于 `gload` 和 `gunload`: 当数据库服务器关闭并重新启动时

#### 用途

`LTAPESIZE` 指定当使用 `gtape` 进行备份时将逻辑日志备份到的设备的最大磁带大小。

`LTAPESIZE` 还指定当使用 `gload` 或 `gunload` 的 `-l` 选项时将数据装入或卸载到的设备的最大磁带大小。如果要使用 `gload` 或 `gunload`，那么可以在命令行上指定不同的磁带大小。如果要使用磁带的全部容量，请将 `LTAPESIZE` 设置为 0。

注：如果在 `ONCONFIG` 文件中设置 `BACKUP_FILTER` 参数，那么 `LTAPESIZE` 无法设置为 0。否则，在将逻辑日志备份到磁盘上的目录时，`gtape` 实用程序会返回错误。错误消息为：

```
当已设置 BACKUP_FILTER 配置参数时，LTAPESIZE 配置  
参数无法设置为 0；请更改 LTAPESIZE 的值。  
程序结束。
```

变通方法是将 LTAPESIZE 配置参数设置为非常高的值。日志文件不会大幅高于 LOGSIZE 配置参数。使用 LOGSIZE 中的值作为此数据库的上限。

如果指定 LTAPESIZE 值，那么 gbackuprestore 将忽略该值。

### RESTARTABLE\_RESTORE 配置参数

使用 RESTARTABLE\_RESTORE 配置参数来启动或禁用可重新开始的恢复。

#### onconfig 值

RESTARTABLE\_RESTORE ON

#### 值

禁用

禁用可重新开始的恢复。如果恢复失败并且 RESTARTABLE\_RESTORE 为 OFF，那么您无法重新启动该恢复。

启用

启用可重新开始的恢复。开始恢复前将 RESTARTABLE\_RESTORE 设置为 ON。否则，您将无法在失败后重新启动恢复。

#### 生效

编辑 onconfig 文件后。如果需要重新启动物理恢复，那么不需要在可以使用 RESTARTABLE\_RESTORE 之前重新启动数据库服务器。如果需要重新启动逻辑恢复，那么必须在使用可重新开始的恢复前重新启动数据库服务器。

打开 RESTARTABLE\_RESTORE 将降低逻辑恢复的性能。有关更多信息，请参阅 gbackuprestore -RESTART 语法：重新启动失败的恢复。

### RESTORE\_FILTER 配置参数

使用 RESTORE\_FILTER 配置参数可指定过滤器程序的路径名以及任何选项。

#### onconfig 值

未设置。恢复的数据未过滤。

#### 值

命令的路径名和任何选项。缺省情况下，路径名相对于 \$GBS\_HOME/bin 目录，否则，路径名必须是程序的绝对路径。如果您要包括命令行选项，那么必须使用单引号括住过滤器名称和选项。

#### 生效

编辑 onconfig 文件，并且 gbackuprestore 或 gtape 启动后。

## 用途

此过滤器在恢复之前，将备份期间变换的数据变换至其原始格式。RESTORE\_FILTER 配置参数指定的过滤器必须与 BACKUP\_FILTER 配置参数指定的过滤器相匹配。例如，如果在备份期间压缩了数据，那么必须在恢复期间解压缩该数据。

为了安全，过滤器不能有非特权用户写许可权。该过滤器的许可权与 GBase 8s 服务器或实用程序所调用的其他可执行文件的许可权相同。

例如，如果您要压缩备份的数据，可以将 BACKUP\_FILTER 和 RESTORE\_FILTER 配置参数设置为以下值：

```
BACKUP_FILTER /bin/compress
RESTORE_FILTER /bin/uncompress
```

RESTORE\_FILTER 配置参数可包括命令行选项以及过滤器名称。例如，指定：

```
RESTORE_FILTER 'my_decrypt - file /var/adm/encryption.pass'
```

在此示例中，引号中的命令用作过滤器。

## TAPEBLK 配置参数

使用 TAPEBLK 配置参数可指定 gtape 在存储空间备份期间所写入的设备的块大小。

### onconfig 值

在 UNIX™ 上：32

### 单位

千字节

### 值的范围

大于 页面大小/1024 的值

要获取页面大小，请运行 gstat -b 命令。

### 生效

对于 gtape:

- 当您执行 gtape 时。
- 通过运行 gadmin -wf 命令在 onconfig 文件中动态重置该值时。
- 通过运行 gadmin -wm 命令重置会话的值时。

对于 gload 和 gunload: 当数据库服务器关闭并重新启动时

## 用途

TAPEBLK 还指定了当使用 gload 或 gunload 实用程序时将数据装入或卸载到的设备的缺省块大小。如果要使用 gload 或 gunload，那么可以在命令行上指定不同的块大小。



当您指定块大小时，数据库服务器不检查磁带设备。验证 TAPEBLK 磁带设备是否可以读取您所指定的块大小。如果不能，您可能无法读取磁带。

如果指定 TAPEBLK 值，那么 gbackupstore 将忽略该值。

### TAPEDEV 配置参数

使用 TAPEDEV 配置参数可指定 gtape 实用程序将存储空间备份到的设备或目录文件系统。

#### onconfig 值

在 UNIX™ 上：/dev/tapedev

#### 如果不存在

在 UNIX 上：/dev/null

#### 单位

路径名

#### 生效

对于 gtape 实用程序：

- 如果在 UNIX 上设置为 /dev/null，那么当数据库服务器关闭并重新启动时
- 如果设置为磁带设备，那么当您运行 gtape 实用程序时
- 通过运行 gadmin -wf 命令在 onconfig 文件中动态重置该值时。
- 通过运行 gadmin -wm 命令重置会话的值时。

对于 gload 和 gunload 实用程序：当数据库服务器关闭并重新启动时

#### 用途

开始处理时，gtape 实用程序读取 TAPEDEV 参数的值。如果您将 TAPEDEV 设置为 /dev/null，必须在启动 gtape 以请求备份之前进行此设置。当您为 /dev/null 并请求备份时，数据库服务器将忽略该备份，但仍使用新的备份时间戳记更新数据库空间。

您可以将 TAPEDEV 配置参数设置为 STDIO，以指示 gtape 实用程序对标准 I/O 而不是设备执行备份与恢复操作。

TAPEDEV 配置参数还指定了当使用 gload 或 gunload 实用程序时将数据装入或卸载到的缺省设备。但是，如果 TAPEDEV 设置为 STDIO，那么 gunload 实用程序将无法卸载数据。

如果更改磁带设备，请验证新设备的 TAPEBLK 和 TAPESIZE 配置参数值是否正确。

如果指定 TAPEDEV 值，那么 gbackupstore 将忽略该值。

#### 远程设备 (UNIX)

在网络中执行存储空间备份时，可以备份到与 UNIX 和 Linux™ 平台上另一个主机相连的远程设备。远程设备和数据库服务器计算机必须具有信任关系，以便 rsh 或 rlogin 实用程序从数据库服务器计算机连接到远程设备计算机，而无需询问密码。您可以通过在远程设备计算机上配置/etc/hosts.equiv 文件、用户的 ~/.rhosts 文件或您系统的任何等效机制，从而建立信任关系。如果想要使用您的平台所使用的缺省实用程序之外的其他实用程序来处理远程会话，那么可以将 DBREMOTECMD 环境变量设置为要使用的特定实用程序。

远程设备的符号链接 (UNIX)

TAPEDEV 配置参数可以是符号链接，从而允许您在磁带设备之间切换，而不必更改 TAPEDEV 配置参数指定的路径名。

请使用以下语法指定连接到另一台主机上的磁带设备：

```
host_machine_name:tape_device_pathname
```

以下示例指定主机 kyoto 上的磁带设备：

```
kyoto:/dev/rmt01
```

打开前和关闭时将磁带设备倒带

TAPEDEV 配置参数指定的磁带设备必须在其打开前和关闭时执行倒带。数据库服务器要求执行该操作，因为它在写入磁带之前要执行一系列检查。

当数据库服务器尝试写入多卷数据库空间或逻辑日志备份中第一盒磁带之外的任何磁带时，数据库服务器首先读取磁带头以确保该磁带可供使用。然后设备将关闭并重新打开。数据库服务器假定磁带在关闭时已倒带，然后数据库服务器开始写入。

每当数据库服务器尝试读取磁带时，它首先读取头并查找正确的信息。在写入过程中，如果磁带设备未在关闭时倒带，那么数据库服务器在磁带开头处将找不到正确的头信息。

## TAPESIZE 配置参数

使用 TAPESIZE 参数指定 gtape 将存储空间备份到的设备的大小。

### onconfig 值

0

### 单位

千字节

### 值的范围

0 或任何正数。真实值依赖于操作系统。

### 生效

对于 gtape:

- 当您执行 `gtape` 时。
- 通过运行 `gadmin -wf` 命令在 `onconfig` 文件中动态重置该值时。
- 通过运行 `gadmin -wm` 命令重置会话的值时。

对于 `gload` 和 `gunload`：当数据库服务器关闭并重新启动时

用途

`TAPESIZE` 还指定使用 `gload` 或 `gunload` 时将数据装入或卸载到的缺省设备的大小。如果要使用 `gload` 或 `gunload`，那么可以在命令行上指定不同的磁带大小。如果要使用磁带的全部物理容量，请将 `TAPESIZE` 设置为 0。

**注：** 如果 `TAPEDEV` 设置为 `STDIO`，那么将与磁带大小无关。

如果指定 `TAPESIZE` 值，那么 `gbackupstore` 将忽略该值。

### 7.1.2 archecker 实用程序配置参数和环境变量

这些主题描述了 `AC_CONFIG` 环境变量和用于 `archecker` 实用程序的配置参数。

`archecker` 实用程序使用 `ac_config.std` 模板中的配置参数来验证备份或执行表级别恢复。如果需要更改这些参数，请将 `ac_config.std` 模板复制到 `AC_CONFIG` 文件。`AC_CONFIG` 环境变量指定了 `AC_CONFIG` 文件的位置。

由于 `gbackupstore` 调用 `archecker` 实用程序来验证备份，因此必须配置 `archecker` 环境变量和参数，然后才能使用 `gbackupstore -v` 选项。

您也可以使用在 `ac_config.std` 文件中没有缺省值，但在该文件中有效的其他 `archecker` 配置参数。

表 1. `archecker` 实用程序使用的配置参数

配置参数	描述
<code>AC_DEBUG</code>	在 <code>archecker</code> 消息日志中打印调试消息。
<code>AC_IXBAR</code>	指定 <code>IXBAR</code> 文件的路径名。  如果未在 <code>ac_config</code> 文件中设置，将使用 <code>BAR_IXBAR_PATH</code> 配置参数的值。
<code>AC_LTAPEBLOCK</code>	指定用于读取逻辑日志的 <code>gtape</code> 块大小。  如果未在 <code>ac_config</code> 文件中设置，将使用 <code>LTAPEBLOCK</code> 配置参数的值。

配置参数	描述
AC_LTAPEDEV	指定 <code>gtape</code> 用于读取逻辑日志的本地设备名。  如果未在 <code>ac_config</code> 文件中设置，将使用 <code>LTAPEDEV</code> 配置参数的值。
AC_MSGPATH	指定 <code>archecker</code> 消息日志的位置。  该配置参数位于缺省 <code>ac_config</code> 文件中。
AC_SCHEMA	指定 <code>archecker</code> 模式命令文件的路径名。
AC_STORAGE	指定 <code>archecker</code> 构建的临时文件的位置。  该配置参数位于缺省 <code>ac_config</code> 文件中。
AC_TAPEBLOCK	指定磁带的块大小（以千字节为单位）。  如果未在 <code>ac_config</code> 文件中设置，将使用 <code>TAPEBLOCK</code> 配置参数的值。
AC_TAPEDEV	指定 <code>gtape</code> 实用程序使用的本地设备名。  如果未在 <code>ac_config</code> 文件中设置，将使用 <code>TAPEDEV</code> 配置参数的值。
AC_TIMEOUT	如果 <code>gbackuprestore</code> 和 <code>archecker</code> 进程之一过早退出，请为它们指定超时值。
AC_VERBOSE	为 <code>archecker</code> 消息指定详细或简洁方式。  该配置参数位于缺省 <code>ac_config</code> 文件中。
BAR_BSALIB_PATH	与 <code>onconfig.std</code> 文件中的 <code>BAR_BSALIB_PATH</code> 服务器配置参数相同。  有关更多信息，请参阅 <code>BAR_BSALIB_PATH</code> 配置参数。

如果使用 `gtape` 实用程序，并且 `AC_TAPEDEV`、`AC_TAPEBLK`、`AC_LTAPEDEV` 和 `AC_LTAPEBLK` 配置参数未在 `AC_CONFIG` 文件中设置，那么 `archecker` 实用程序将使用 `onconfig` 文件中指定的 `TAPEDEV`、`TAPEBLK`、`LTAPEDEV`、`LTAPEBLK` 配置参数所指定的值。

### AC\_CONFIG 文件环境变量

将 `AC_CONFIG` 环境变量设置为 `archecker` 配置文件（`ac_config.std` 或用户定义）的完整路径名。



#### 缺省值

UNIX™: `$GBS_HOME/etc/ac_config.std`

#### 生效

`gbackuprestore` 启动时

以下是有效的 `AC_CONFIG` 路径名的示例：

UNIX: `/usr/gbasedb/etc/ac_config.std` 和 `/usr/local/my_ac_config.std`

如果未设置 `AC_CONFIG`，那么 `archecker` 实用程序会将 `archecker` 配置文件的缺省位置设置为 UNIX 上的 `$GBS_HOME/etc/ac_config.std`。

**要点：**如果未在 `AC_CONFIG` 文件中指定包括配置文件的完整路径，那么 `archecker` 实用程序可能无法正常工作。

### AC\_DEBUG 配置参数

`AC_DEBUG` 配置参数使调试消息打印在 `archecker` 消息文件中。 仅在技术支持指导下使用该参数。

使用该配置参数可能导致 `archecker` 消息日志文件增长得非常大，并可能显著减慢 `archecker` 处理的速度。

#### 缺省值

禁用

#### 范围

1-16

### AC\_IXBAR 配置参数

使用 `AC_IXBAR` 配置参数可指定 `IXBAR` 文件的位置。

**缺省值**

无

**范围**

任何有效的路径名

**AC\_LTAPEBLOCK 配置参数**

使用 AC\_LTAPEBLOCK 配置参数可指定用于读取逻辑日志的 gtape 块大小。

**缺省值**

32 千字节

**范围**

0 - 2,000,000,000

**用途**

当使用以下命令执行归档时：

- gbackuprestore -b, AC\_TAPEBLOCK 的值应为 BAR\_XFER\_BUF\_SIZE 配置参数值乘以当前页面大小的结果。有关更多信息，请参阅 BAR\_XFER\_BUF\_SIZE 配置参数。
- gtape -t, AC\_LTAPEBLOCK 的值应为归档时 TAPEBLK ONCONFIG 配置参数设置为的值。有关更多信息，请参阅指定磁带块大小。

**AC\_LTAPPEDEV 参数**

使用 AC\_LTAPPEDEV 配置参数可指定 gtape 实用程序使用的本地设备名。

如果磁带设备设置为 STDIO，archecker 将从标准输入接收输入。

**缺省值**

无

**范围**

任何有效路径名或 STDIO

**AC\_MSGPATH 配置参数**

使用 AC\_CONFIG 文件中的 AC\_MSGPATH 参数可指定 archecker 消息日志 (ac\_msg.log) 的位置。

**ac\_config.std 值**

UNIX™: AC\_MSGPATH /tmp/ac\_msg.log

### 生效

gbackuprestore 启动时

### 用途

必须在 AC\_CONFIG 文件中指定消息日志的完整路径，否则 archecker 实用程序将无法正常工作。

当您使用 gbackuprestore -v 验证备份时，archecker 实用程序将摘要消息 写入到 bar\_act.log 中并指示验证是成功还是失败。它将详细消息写入到 ac\_msg.log 中。如果备份验证失败，请放弃该备份并尝试其他备份，或将 ac\_msg.log 提供给 GBase 软件支持。有关样本消息，请参阅gbackuprestore -v 语法：验证备份。

## AC\_SCHEMA 配置参数

使用 AC\_SCHEMA 配置参数可指定 archecker 模式命令文件的路径名。

### 缺省值

无

### 范围

任何有效的路径名

-f cmdfile 命令行选项会覆盖该配置参数。

## AC\_STORAGE 配置参数

使用 AC\_CONFIG 文件中的 AC\_STORAGE 配置参数可指定 archecker 存储其临时文件的目录的位置。

### ac\_config.std 值

UNIX™: /tmp

### 生效

gbackuprestore 启动时

### 用途

您必须在 AC\_CONFIG 文件中指定存储位置的完整路径，否则 archecker 实用程序可能无法正常工作。

下表列出了 archecker 构建的目录和文件。如果验证成功，这些文件将被删除。

表 1. archecker 临时文件

目录	文件
CHUNK_BM	每个已备份的存储空间的位图信息。
信息	备份的统计分析和调试信息。
保存	PT. ##### 文件中的分区页。 FL. ##### 文件中的可用块页。 RS. ##### 文件中的保留页。 BF. ##### 文件中的可用 Blob 映射页

要计算所需要的可用空间大小，请参阅用于备份验证的临时空间。建议您将 `AC_STORAGE` 设置为具有足够可用空间的位置。

### AC\_TAPEBLOCK 配置参数

使用 `AC_TAPEBLOCK` 配置参数可在使用 `gbackuprestore -b` 命令或 `gtape -t` 命令执行归档时指定磁带块的大小（以千字节为单位）。

#### 缺省值

32 千字节

#### 范围

0 - 2,000,000,000

#### 用途

当使用以下命令执行归档时：

- `gbackuprestore -b`，`AC_TAPEBLOCK` 的值应为 `BAR_XFER_BUF_SIZE` 配置参数值乘以当前页面大小的结果。有关更多信息，请参阅 `BAR_XFER_BUF_SIZE` 配置参数。
- `gtape -t`，`AC_LTAPEBLOCK` 的值应为归档时 `TAPEBLK ONCONFIG` 配置参数设置为的值。有关更多信息，请参阅指定磁带块大小。

### AC\_TAPEDEV 配置参数

使用 `AC_TAPEDEV` 配置参数可指定 `gtape` 实用程序使用的本地设备名。

如果磁带设备设置为 `STDIO`，`archecker` 将从标准输入接收输入。

#### 缺省值



无

#### 范围

任何有效路径名或 STDIO

#### AC\_TIMEOUT 配置参数

如果 gbackuprestore 和 archecker 进程之一过早退出，请使用 AC\_TIMEOUT 配置参数为它们指定超时值。

#### ac\_config.std 值

UNIX<sup>™</sup>: 300

#### 单位

秒

#### 生效

当 gbackuprestore-v 命令启动时

引入了 AC\_TIMEOUT 配置参数以在 gbackuprestore 和 archecker 进程之一过早退出时避免它们无限期地互相等待，从而避免在数据服务器初始化期间创建孤立的 zombie 进程。

#### AC\_VERBOSE 配置参数

使用 AC\_CONFIG 文件中的 AC\_VERBOSE 参数可指定 archecker 消息日志 (ac\_msg.log) 中是详细输出还是简洁输出。

#### ac\_config.std 值

1

#### 值的范围

1 = ac\_msg.log 中的详细消息

0 = ac\_msg.log 中的简洁消息

#### 生效

gbackuprestore 启动时

### 7.1.3 GBase 8s Primary Storage Manager 配置参数

GBase 8s 主存储管理器 使用某些特定配置参数中的信息。

## PSM\_ACT\_LOG 配置参数

如果您不希望在 gbackupstore 活动日志中包含日志信息,那么可以使用 PSM\_ACT\_LOG 配置参数指定 GBase 8s 主存储管理器 活动日志的位置。

### onconfig 值

none

### 如果值不存在

将使用 BAR\_ACT\_LOG 配置参数的值

### 值的范围

完整路径名

### 生效

当 gpsm 实用程序启动时

通过运行 gadmin -wf 命令在 onconfig 文件中动态重置该值时。

通过运行 gadmin -wm 命令重置会话的值时。

### 用途

指定具有适当可用空间量的现有目录的路径,或者使用 \$GBS\_HOME/psm\_act.log。如果仅指定文件名,那么存储管理器将在您启动存储管理器的工作目录中创建活动日志。

如果 PSM\_ACT\_LOG 配置参数未设置,那么 GBase 8s Primary Storage Manager 会将活动信息放入 BAR\_ACT\_LOG 配置参数中指定的目录内。要清楚地将 gbackupstore 与 GBase 8s Primary Storage Manager 活动信息区分开,请使用 PSM\_ACT\_LOG 来为存储管理器的活动日志指定不同的位置。

文件格式与数据库服务器消息日志的格式相似。可以检查活动日志以确定存储管理器操作的结果。

如果 PSM\_ACT\_LOG 配置参数指定的文件不存在,那么会创建该文件。

您也可以使用 PSM\_ACT\_LOG 环境变量来指定您环境(例如,单个会话)的 GBase 8s Primary Storage Manager 活动日志的位置。

## PSM\_CATALOG\_PATH 配置参数

使用 PSM\_CATALOG\_PATH 配置参数可指定包含 GBase 8s 主存储管理器 目录表的目录的完整路径。这些目录表包含有关存储管理器所管理的池、设备和对象的信息。

### onconfig 值

UNIX™ 或 Linux™: \$GBS\_HOME/etc/psm

### 值的范围

包含 GBase 8s Primary Storage Manager 目录表的目录的完整路径名

## 生效

当 gbackupstore 或 gpsm 实用程序启动时

通过运行 gadmin -wf 命令在 onconfig 文件中动态重置该值时。

通过运行 gadmin -wm 命令重置会话的值时。

## 用途

您可以将缺省路径更改为其他位置。GBase 8s Primary Storage Manager 会将包含有关设备和对象的信息的文件放在您指定的任何目录中。

如果将备份文件移至其他位置，请更改 PSM\_CATALOG\_PATH 配置参数的值。

您可以随时备份该文件的内容。

如果您有多个实例，并且每个实例中的 PSM\_CATALOG\_PATH 都设置为相同的路径，那么所有实例都包含相同的目录表。可以为每个实例指定不同的路径。

当您首次运行 gpsm 实用程序命令或首次使用 XBSA 共享库时，存储管理器会自动创建目录表。

您也可以使用 PSM\_CATALOG\_PATH 环境变量来指定您环境（例如，单个会话）的 GBase 8s Primary Storage Manager 目录表的位置。

## PSM\_DBSPool 配置参数

使用 PSM\_DBSPool 配置参数可更改 GBase 8s 主存储管理器 用于放置备份与恢复数据库空间数据的池的名称。

### onconfig 值

DBSPool

## 生效

当 gpsm 实用程序启动时

通过运行 gadmin -wf 命令在 onconfig 文件中动态重置该值时。

通过运行 gadmin -wm 命令重置会话的值时。

## 用途

存储管理器会自动将数据库空间数据放入 DBSPool 或您指定的池内。您可以使用字母和数字的任意组合。

也可以使用 PSM\_DBSPool 环境变量来更改您环境（例如，单个会话）的池的名称。

## PSM\_DEBUG 配置参数

如果您要使用的调试级别不同于 `gbackupstore` 所使用的级别，请使用 `PSM_DEBUG` 配置参数指定在 `GBase 8s Primary Storage Manager` 调试日志中打印的调试信息量。

### onconfig 值

0

### 单位

一位数字，代表所需的调试信息的级别

### 值的范围

0 = 无调试消息。

1 = 仅打印内部错误。

2 = 打印有关函数入口和出口的信息，并打印内部错误。

3 = 打印 1-2 指定的信息以及其他详细信息。

4 = 打印有关并行操作的信息，以及 1-3 指定的信息。

5 = 打印有关 `GBase 8s Primary Storage Manager` 中内部状态的信息。

6 = 打印 1-5 指定的信息以及其他详细信息。

7 = 打印 1-6 指定的信息以及其他详细信息。

8 = 打印 1-7 指定的信息以及其他详细信息。

9 = 打印所有调试信息。

### 如果值不存在

将使用 `BAR_DEBUG` 配置参数的值。

### 生效

当 `gpsm` 实用程序启动时

当 `gbackupstore` 实用程序执行命令并读取 `BAR_DEBUG` 配置参数中指定的信息时  
通过运行 `gadmin -wf` 命令在 `onconfig` 文件中动态重置该值时。

通过运行 `gadmin -wm` 命令重置会话的值时。

### 用途

如果将 `PSM_DEBUG` 配置参数设置为大于 0 的有效值，那么 `GBase 8s Primary Storage Manager` 会将调试消息记录到其调试日志中。

您可以使用不同的调试值进行试验，以找出合适的信息量。通常，如果 `PSM_DEBUG` 配置参数设置为 5，存储管理器即可打印足够的信息用于跟踪和调试。

设置为 8 和 9 需要大量空间。

您也可以使用 `PSM_DEBUG` 环境变量来指定您环境（例如，单个会话）的存储管理器调试日志中打印的调试信息量。

### PSM\_DEBUG\_LOG 配置参数

如果您不希望在 `gbackupstore` 调试日志中包含日志信息，那么可以使用 `PSM_DEBUG_LOG` 配置参数指定 GBase 8s 主存储管理器 将调试消息写入的调试日志的位置。

#### onconfig 值

UNIX<sup>™</sup> 或 Linux<sup>™</sup>: `/usr/gbasedbt/bar_dbug.log`

#### 如果值不存在

将使用 `BAR_DEBUG_LOG` 配置参数的值。

#### 生效

当 `gpsm` 实用程序启动时

通过运行 `gadmin -wf` 命令在 `onconfig` 文件中动态重置该值时。

通过运行 `gadmin -wm` 命令重置会话的值时。

#### 用途

如果 `PSM_DEBUG_LOG` 配置参数未设置，那么 GBase 8s Primary Storage Manager 会将活动信息放入 `BAR_DEBUG_LOG` 配置参数中指定的目录内。要清楚地将 `gbackupstore` 与 GBase 8s Primary Storage Manager 活动信息区分开，请使用 `PSM_DEBUG_LOG` 来为 GBase 8s Primary Storage Manager 活动日志指定不同的位置。

出于安全性原因，请将 `PSM_DEBUG_LOG` 配置参数设置为带有受限许可权的目录，如 `$GBS_HOME` 目录。

如果保留调试文件的目录变得太大，那么您可以擦除该文件。仅当有问题需要调试时，才需要保留信息。

您也可以使用 `PSM_DEBUG_LOG` 环境变量来指定您环境（例如，单个会话）的调试日志的位置。

### PSM\_LOG\_POOL 配置参数

使用 `PSM_LOG_POOL` 配置参数可更改 GBase 8s 主存储管理器 用于放置备份与恢复日志数据的池的名称。

#### onconfig 值

LOGPOOL

## 生效

当 `gpsm` 实用程序启动时

通过运行 `gadmin -wf` 命令在 `onconfig` 文件中动态重置该值时。

通过运行 `gadmin -wm` 命令重置会话的值时。

## 用途

存储管理器会自动将日志数据放入 `LOGPOOL` 或您指定的池内。您可以使用字母和数字的任意组合。

也可以使用 `PSM_LOG_POOL` 环境变量来更改您环境（例如，单个会话）的池的名称。

### 7.1.4 事件警报配置参数

当您设置用于 `gbackuprestore` 和 `gtape` 实用程序的配置参数时，还要确定是否需要调整 `ALARMPROGRAM` 和 `ALRM_ALL_EVENTS` 配置参数。

使用 `ALARMPROGRAM` 配置参数可将 `log_full.sh` 脚本设置为在日志文件变满时自动对其进行备份。

使用 `ALRM_ALL_EVENTS` 配置参数可使 `ALARMPROGRAM` 在每次调用警报事件时执行。

## 8 附录

### 8.1 对一些备份与恢复错误进行故障诊断

该附录列出备份或恢复期间您可能接收到的一些错误和参考消息，描述在什么情况下可能会发生这些错误或者可能会出现这些消息，并提供了可能的解决方案或变通方法。

通过查看 `gbackuprestore` 活动日志来查找错误

如果使用 `gbackuprestore -b` 备份存储空间或使用 `gbackuprestore -r` 恢复存储空间时发生错误，那么数据库服务器不会在标准输出 (`stdout`) 中显示错误。因此，使用 `gbackuprestore -b` 或 `gbackuprestore -r` 时，您必须检查 `gbackuprestore` 活动日志 (`bar_act_log`) 中的信息。`gbackuprestore` 备份与恢复数据时，它会将进度消息、警告和错误消息写入到 `bar_act.log`。

#### 8.1.1 归档期间页面毁坏

消息 `Archive detects that page is corrupt` 指示页面验证失败。如果收到该消息，您可以确定具有毁坏页面的表。

在归档期间，数据库服务器将每页写入归档设备之前将对其进行验证。该验证将检查页上的元素是否与期望值一致。对某页的这种验证失败时，与以下类似的消息将写入 `online.log` 文件：

```
16:27:49 提出警告：归档检测到页面 1:10164 已毁坏。
16:27:49 人员：会话（5、gbasedbt@cronus、23467、10a921048）
线程（40、arcbackup1、10a8e8ae8、1）
文件：rsarcbu.c 行：2915
16:27:49 pid 23358 的堆栈跟踪已写入 /tmp/af.41043f4
16:27:49 另请参阅：/tmp/af.41043f4
16:27:49 归档检测到页面 1:10164 已毁坏。
16:27:50 rootdbs 的归档已完成，同时检测到 1 个受损页面。
```

检测到 10 个毁坏的页面后，归档停止。`online.log` 文件将显示详尽的错误消息，其中包括前 10 个错误的页地址。随后，只将毁坏页的计数放入 `online.log` 中。

您接收到此消息后，通过检查 `gcheck -pe` 命令的输出来确定毁坏页属于哪个表。要确定毁坏的范围，对该表执行 `gcheck -cid` 命令。

毁坏页被保存到备份介质上。在恢复期间，毁坏页以其毁坏的格式返回。毁坏页恢复时不会向 `online.log` 写入错误消息，只有当归档时才写入。

### 8.1.2 已在运行的日志备份

使用 `gbackuprestore` 创建备份时，`bar_act.log` 文件中的参考消息 `log backup is already running` 以及 `online.log` 文件中的参考消息 `Process exited with return code 152` 可能会在某些情况下出现。

这些消息可能会在以下情况下出现：

- 将 `ALARMPROGRAM` 配置参数设置为 `log_full.sh` 时。  
事件将定期使 `log_full.sh` 触发 `gbackuprestore -b -l` 命令。如果在运行 `gbackuprestore -b -l` 命令时某个日志填满，那么 `gbackuprestore` 还将备份该日志。如果直到下个事件触发时为止备份还未完成，将在 `bar_act.log` 文件中生成一个警告。下个事件触发时，日志备份可以继续进行。
- 自动启动 `gbackuprestore -b -l` 命令时。  
0 级归档（特别是使用 `-w` 选项启动时）将首先归档数据库，接着自动启动 `gbackuprestore -b -l` 命令来备份当前已填满但还未备份的所有逻辑日志。在 `online.log` 中可能没有 `log_full.sh` 消息，这是因为 `gbackuprestore -b -l` 命令是直接启动的。
- 当前一个磁带填满后安装新磁带时，将调度 `log_full.sh` 事件但不触发该事件。  
一旦下个日志填满并在 `log_full.sh` 文件中生成事件触发器时，将归档所有可用日志。

以通过运行 `gbackuprestore -b -l` 来强制执行归档，或通过运行 `gadmin -l` 强制触发 `log_full.sh`。

### 8.1.3 恢复期间没有服务器连接

使用 `gbackuprestore` 进行整个系统恢复期间，`bar_act.log` 文件中可能出现错误 `archive api error: no server connection`。接着 `gbackuprestore` 成功连接到存储管理器，但最终失败，错误为 `archive api error: not yet open`。如果收到这些消息，可以执行相应步骤来解决问题。

`bar_act.log` 文件中包含与以下消息相似的信息：

```
2000-03-09 10:51:06 19304 19303 /usr/gbasedbt/bin/gbackuprestore_d -r -w
2000-03-09 10:51:09 19304 19303 错误：无法启动物理恢复：
归档 API 错误：无服务器连接。
2000-03-09 10:51:09 19304 19303 与存储管理器连接成功。
2000-03-09 10:51:36 19304 19303 进程 19304 收到信号 3。进程将
在清除后退出。
2000-03-09 10:59:13 19811 19810 /usr/gbasedbt/bin/gbackuprestore_d -r -w
2000-03-09 10:59:16 19811 19810 错误：无法启动物理恢复：
归档 API 错误：无服务器连接。
2000-03-09 10:59:16 19811 19810 与存储管理器连接成功。
2000-03-09 11:01:12 19811 19810 开始对 llog1 的 0 级冷恢复。
2000-03-09 11:01:12 19811 19810 错误：无法将恢复数据写入数据库服务器：
归档 API 错误：尚未打开。
```

要解决该问题，请检查数据库服务器是否还在运行。如果还在运行，那么关闭数据库服务器并再次运行该命令。

### 8.1.4 恢复之前删除数据库

如果使用 `gbackuprestore` 和存储管理器执行 0 级归档，接着删除数据库并随后使用 `gbackuprestore -r` 命令执行恢复，那么数据库将仍然处于已删除的状态。恢复将回收日志并且日志包含 `DROP DATABASE` 语句。回收或重放日志时，将删除数据库。如果收到这些消息，可以执行相应步骤来解决问题。

要防止这种情况的发生，请使用 `gbackuprestore -r -p` 命令执行物理恢复，接着使用 `gbackuprestore -r -l` 命令执行逻辑恢复。该操作序列不回收日志却可以恢复数据库。

### 8.1.5 备份或恢复期间没有数据库空间或 Blob 空间



用 `gbackuprestore -r` 或 `gbackuprestore -r -w` 命令启动恢复期间，如果紧急引导文件 `ixbar.servernum` 没有备份中对象的正确条目，`There are no DB/BLOBspaces to backup/restore` 消息将出现在 `bar_act.log` 文件中。

该错误可能在以下情况中出现：

- 在外部恢复中，如果未从源系统中复制紧急引导文件。
- 如果归档备份完成后重新创建了紧急引导文件。先前的文件以格式 `ixbar.xx.xxxx` 保存。
- 试图对不是整个系统备份的备份执行 `gbackuprestore -r -w` 命令。

### 恢复 Blob 空间 BLOB

您可以使用表级别恢复来恢复存储在表中的 BLOB。但不能恢复存储在 blob 空间中的 BLOB。如果尝试恢复 blob 空间 BLOB，那么该列将被设置为 NULL。

## 8.1.6 在备份系统上更改系统时间

在某些情况下，如果系统时间有问题，`gbackuprestore` 会失败，并出现以下消息：`There are no storage spaces or logical logs to backup or restore`。如果发生此情况，可以执行相应步骤来解决问题。

时间线使用 UNIX™ 时间作为数据库空间的归档检查点时间和逻辑日志的结束时间。如果没有自动备份日志并且更改了系统时钟，那么可能会破坏时间线。

例如，如果在归档检查点时间前结束了逻辑日志记录，就会有一个高于归档检查点时间的戳记。数据库空间不需要日志，并且 `gbackuprestore` 将尝试立即恢复备份。如果找不到日志，那么 `gbackuprestore` 失败，并出现以下消息：`There are no storage spaces or logical logs to backup or restore`。

要存储存储空间和逻辑日志：

1. 将时钟改回其原有值。
2. 从备份恢复系统。
3. 将时钟更改成新的时间。

## 8.2 迁移数据、服务器和工具

### 8.2.1 数据库服务器或存储管理器升级之前进行备份

升级到新版本的数据库服务器前，必须执行完整备份。

**重要：** 当您把数据库服务器升级到最新版本时，数据库服务器转换软件将自动重新创建 `sysutils` 数据库。旧版本数据库服务器的所有备份与恢复信息都将丢失。在较旧版本的数据库服务器下进行的备份与新版本的数据库服务器不兼容。

要准备进行升级：

1. 在升级数据库服务器、Storage Manager 或更改存储管理器之前，请先使用 `gbackuprestore` 对所有数据执行 0 级备份。
2. 保存这些备份，万一需要还原到旧数据库服务器版本时，您可以恢复数据。
3. 升级前请备份管理文件。
4. 升级数据库服务器后，备份所有存储空间和逻辑日志。

有关数据库服务器迁移的完整信息，请参阅《GBase 8s 迁移指南》。

如果更改存储管理器供应商，那么在确认新存储管理器对备份与恢复操作均适用之前，请不要除去旧存储管理器。

## 8.2.2 升级第三方存储管理器

如果升级第三方存储管理器供应商，那么在确认新存储管理器对备份与恢复操作均适用之前，请不要除去旧存储管理器。

**先决条件：** 升级之前，执行数据库服务器的完整备份。

要将第三方存储管理器的新版本与数据库服务器一起使用：

1. 先安装新存储管理器，再启动数据库服务器。
2. 使用新的存储管理器定义更新 `sm_versions` 文件。

确认新存储管理器正确运行，再将其用于生产环境：

- 确存储管理器可以找到 `gbackuprestore` 请求的对象。
- 确保新的存储管理器版本能够读取使用旧版本写入的介质。

如果在数据库服务器上设置了连续逻辑日志备份，那么 `gbackuprestore` 将在数据库服务器联机后不久开始备份逻辑日志。

使用 `onsmsync` 实用程序可以让 `sysutils` 数据库和紧急引导文件中旧的备份历史记录到期。

## 8.2.3 更改存储管理器供应商

如果更改存储管理器供应商，那么在有证据表明新存储管理器对备份与恢复操作均适用之前，请不要除去旧存储管理器。您可以使用旧存储管理器作为备份存储管理器，以在新存储管理器无法满足您的需要时进行使用。

`gbackuprestore` 支持同时使用多个存储管理器。要设置为测试一个存储管理器并将另一个作为备份存储管理器，请在 `BAR_BSALIB_PATH` 配置参数中以及 `$GBS_HOME/etc/sm_versions` 文件中指定这两个存储管理器的信息。

如果无法同时使用新旧存储管理器，请使用 `gbackuprestore` 和 GBase 8s 主存储管理器 `Storage Manager` 或 `gtape` 作为在检查备份和恢复操作是否能正确用于新存储管理器时的备份的备用方法。仅当您确认新存储管理器能够正确工作后，才将所有备份作为整个系统的 0 级备份来执行 (`gbackuprestore -b-L 0 -w`)。

如果更改物理连接（例如，将存储设备从本地连接移动到网络服务器），请确保新存储管理器可以在网络中移动数据。另请确保新存储管理器可以向存储设备发送多个数据流。它也可以使用不同版本的 XBSA。

### 8.2.4 从 `gtape` 切换到 `gbackuprestore`

不能使用 `gtape` 备份数据并使用 `gbackuprestore` 恢复这些数据，反之亦然，因为数据存储格式和备份能力是不同的。但是，您可以使用 `gtape` 备份数据，准备使用 `gbackuprestore`，然后使用 `gbackuprestore` 备份。

要从 `gtape` 切换到 `gbackuprestore`：

1. 使用 `gtape` 执行完全备份。
2. 让备份介质处于脱机状态，以防止可能的重用或擦除。
3. 将存储管理器配置为与 `gbackuprestore` 一起使用。
4. 配置环境：
  - a. 设置用于 `gbackuprestore` 和存储管理器的配置参数。
  - b. 使用存储管理器定义创建 `sm_versions` 文件。如果使用 GBase 8s 主存储管理器以外的存储管理器，请使用存储管理器定义创建 `sm_versions` 文件。GBase 8s Primary Storage Manager 不使用 `sm_versions.std` 文件。
5. 使用 `gbackuprestore` (`gbackuprestore -b` 或 `gbackuprestore -b -w`) 执行完全备份。
6. 使用 `gbackuprestore -v` 命令验证备份。

## 8.3 GLS 支持

本附录包含有关将 Global Language Support (GLS) 与 `gbackuprestore` 一起使用的信息。

### 8.3.1 将 GLS 与 `gbackuprestore` 实用程序一起使用

`gbackuprestore` 实用程序支持 Global Language Support (GLS), 它允许用户使用其母语工作。客户机应用程序使用的语言称为客户机语言环境。数据库为特定于其服务器的文件使用的语言称为服务器语言环境。

`gbackuprestore` 必须与数据库服务器在同一台计算机上运行。不过, 您可以在任何具有支持消息和本地化文件的语言环境中运行 `gbackuprestore`。例如: 如果服务器语言环境是英语而客户机语言环境是法语, 那么可以用法语发出 `gbackuprestore` 命令。

以下命令为文件 `tomb` 中指定的数据库空间执行 0 级备份: `gbackuprestore -b -L 0 -f tomb`

`sysutils` 数据库、紧急引导文件和存储管理器引导文件是在 `en_us.8859-1` (缺省为英语) 语言环境中创建的。`sysutils` 数据库中的 `gbackuprestore` 目录表使用的是英语。尝试使用 `DB-Access` 或第三方实用程序连接到 `sysutils` 数据库前, 请先将客户机和数据库语言环境更改为 `en_us.8859-1`。

### 支持非 ASCII 字符的标识

您可以将非 ASCII 字符用于 `gbackuprestore` 和 `gdblogmode` 命令中的数据库名称和文件名, 以及用于 `onconfig` 文件中的文件名。

GBase 8s GLS 用户指南 描述了支持非 ASCII 字符的 SQL 标识。非 ASCII 字符同时包括 8 位和多字节字符。

例如, 您可以为 `gbackuprestore` 活动登录 `BAR_ACT_LOG` 指定非 ASCII 文件名, 并为 `BAR_BSALIB_PATH` 中的存储管理器库指定非 ASCII 路径名。

### 需要 7 位 ASCII 字符的标识

对于存储空间名称和数据库服务器名称, 必须使用 7 位 ASCII 字符。

### `gbackuprestore` 消息的语言环境

除了数据库服务器发出的消息以外, 其他所有 `gbackuprestore` 消息都以客户机语言环境出现在活动日志中。

例如: 表明数据库服务器出错的部分消息以客户机语言环境显示, 服务器生成的部分以服务器语言环境显示。

## 8.3.2 将 `GL_DATETIME` 环境变量与 `gbackuprestore` 一起使用

当客户机应用程序发送日期或时间数据时, 数据库服务器必须知道如何解释和转换这种以日期或时间数据显示的最终用户格式。您可以使用 `GL_DATE` 和 `GL_DATETIME` 环境变量来指定可以替换的日期和时间格式。

如果没有设置这些环境变量, `gbackuprestore` 将使用客户机语言环境的日期和时间格式。

如果执行时间点恢复, 请按照 `GL_DATETIME` 环境变量 (如果已设置) 指定的格式输入日期和时间。

### 8.3.3 将 GLS 与 gtape 实用程序一起使用

gtape 实用程序与 gbackupstore 一样支持 GLS。您可以指定本地语言环境的数据库名称。

The logo for GBASE, featuring the word "GBASE" in a bold, red, sans-serif font with a registered trademark symbol (®) to its upper right. To the left of the text is a vertical bar with a red top half and a grey bottom half.

南大通用数据技术股份有限公司  
General Data Technology Co., Ltd.



微信二维码

■ ■ 技术支持热线：400-013-9696

