

VERITAS Volume Manager™ 3.1

管理员指南

Solaris

2001 年 2 月
30-000116-011


VERITAS

免责声明

本出版物包含的信息若有变动，恕不另行通知。VERITAS Software Corporation 对本手册没有任何形式的保证，包括但不限于出于特定目的的适销性和适合性的暗示保证。VERITAS Software Corporation 不对这里包含的错误负责，也不对与提供、执行或使用本手册相关的偶然或必然损坏负责。

版权信息

版权 © 1999-2000 VERITAS Software Corporation。版权所有。VERITAS 是 VERITAS Software Corporation 在美国和其它国家 / 地区的注册商标。VERITAS 徽标和 VERITAS Volume Manager 是 VERITAS Software Corporation 的商标。所有其它商标或注册商标是其各自所有者的财产。

2001 年 2 月在美国印刷。

VERITAS Software Corporation

1600 Plymouth St.

Mountain View, CA 94043

电话：650-335-8000

传真：650-335-8050

www.veritas.com



目录

前言	xiii
读者	xiii
范围	xiii
结构	xiii
使用本指南	xiv
相关文档	xiv
惯例	xv
第 1 章：了解 Volume Manager	1
介绍	1
如何存储数据	2
Volume Manager 概述	2
物理对象	2
物理磁盘和磁盘命名	2
分区	3
卷和虚拟对象	3
Volume Manager 磁盘	4
磁盘组	4
子磁盘	4
Plex	5
卷	6
Volume Manager 的虚拟对象间联系	8
卷布局	9
连续	10



条带化 (RAID-0)	11
RAID-5	14
镜像 (RAID-1)	15
镜像加条带化 (RAID-1 + RAID-0)	15
条带化加镜像 (RAID-0 + RAID-1)	16
Volume Manager 和 RAID-5	17
传统的 RAID-5 阵列	17
Volume Manager RAID-5 阵列	18
日志	20
分层卷	21
Volume Manager 和操作系统	22
Volume Manager 布局	22
Volume Manager 用户界面	23
用户界面概述	23
为什么要使用 Volume Manager	24
第 2 章：入门指南	27
介绍	27
Volume Manager 初始化	27
Volume Manager 后台驻留程序	28
配置后台驻留程序 vxconfigd	28
卷 I/O 后台驻留程序 vxiod	29
系统设置	30
系统设置顺序实例	30
系统设置原则	31
热重定位原则	31
条带化原则	32
镜像原则	32
脏区日志 (DRL) 原则	33
镜像和条带化原则	33



条带化和镜像原则	34
RAID-5 原则	34
系统保护	34
第 3 章：Volume Manager 操作	37
介绍	37
联机重新布局	38
存储布局	38
联机重新布局的工作机制	38
转换类型	39
转换特性	41
转换和卷长	42
热重定位	42
热重定位的工作机制	42
如何选择重定位的空间	43
解除重定位实用程序	44
卷再同步	44
脏区日志	45
快速镜像再同步 (FMR)	46
FMR 组件	46
FMR 对 VxVM 快照功能性的增强	47
Volume Manager 根目录可置性	49
使用根卷引导	50
启动时卷的限制	50
动态多路径 (DMP)	51
路径故障切换机制	52
负载均衡	52
从 DMP 设备启用	52
启用和禁用控制器	52
显示 DMP 数据库信息	53



VxSmartSync 恢复加速器	53
数据卷配置	53
重做日志卷配置	54
Volume Manager 任务监视器	54
Volume Manager 集群功能性	54
第 4 章：磁盘任务	55
介绍	55
标准磁盘设备	56
磁盘组	57
磁盘和磁盘组命令	58
初始化和添加磁盘	59
将磁盘添加到 Volume Manager	61
向磁盘组添加磁盘	63
从磁盘组中删除磁盘	65
移动磁盘	66
重命名磁盘	66
保留磁盘	67
使磁盘脱机	68
镜像磁盘	68
删除磁盘	69
显示磁盘信息	70
检测和更换故障磁盘	71
热重定位	71
使用 vxunrelocate	75
检测故障磁盘	78
更换磁盘	80
创建磁盘组	81
升级磁盘组	83
删除磁盘组	84



在系统间移动磁盘组	85
破坏磁盘组	88
使用特殊设备	89
使用 vxdisk 实现特殊封装	89
使用 vxdisk 实现 RAM 磁盘	90
使用 vxdisk 显示多路径信息	90
vxdiskadm 菜单界面任务	91
启动 vxdiskadm	92
vxdiskadm 主菜单	92
将物理磁盘置于 Volume Manager 控制之下	96
添加磁盘供以后使用	105
重新初始化磁盘供 Volume Manager 使用	105
将 VM 磁盘添加到热重定位池	108
将 VM 磁盘从热重定位池中删除	108
将热重定位的子磁盘移回磁盘	109
从热重定位使用对象中排除磁盘	110
使磁盘可供热重定位使用	110
显示物理磁盘信息	111
删除物理磁盘	112
禁用物理磁盘（使物理磁盘脱机）	112
启用物理磁盘	113
更换物理磁盘	114
更换故障磁盘或已删除的磁盘	115
向磁盘组添加磁盘	116
创建磁盘组	116
逐出磁盘组	118
导入磁盘组	119
退出 vxdiskadm	120

第 5 章：卷任务 121



介绍	121
卷、Plex 及子磁盘任务和联机备份	121
创建卷	123
创建连续卷	124
创建条带卷	125
创建 RAID-5 卷	126
创建镜像卷	126
重定卷的大小	126
用 vxassist 重定卷的大小	127
用 vxresize 重定卷的大小	128
删除卷	128
镜像卷	128
创建镜像卷	128
镜像一个现有的卷	129
创建启用脏区日志的卷	129
镜像全部卷	129
删除镜像	130
添加 RAID-5 日志	130
添加 DRL 日志	130
删除 DRL 日志	131
删除 RAID-5 日志	131
停止卷	131
启动卷	132
列出无法启动的卷	132
更改卷的读策略	132
准备卷从备份恢复	133
恢复卷	134
显示卷的信息	134
Plex 任务	135
创建 Plex	135



关联 Plex	136
分离和删除 Plex	136
使用镜像备份	137
显示 Plex 信息	138
更改 Plex 属性	138
更改 Plex 状态：分离和挂接 Plex	139
移动 Plex	141
复制 Plex	142
子磁盘任务	142
创建子磁盘	142
删除子磁盘	142
显示子磁盘信息	143
关联子磁盘	143
分离子磁盘	145
更改子磁盘属性	145
移动子磁盘	146
分割子磁盘	146
结合子磁盘	146
执行联机备份	147
快速镜像再同步	148
镜像 VM 磁盘上的卷	150
移动 VM 磁盘上的卷	151
第 6 章：Volume Manager 集群功能性	153
介绍	153
集群功能性概述	153
共享的 Volume Manager 对象	154
集群卷管理的工作机制	155
VxVM 集群中的磁盘	160
磁盘分离策略	160



磁盘组激活	160
脏区日志和集群环境	161
日志格式和大小	161
兼容性	162
DRL 在集群环境中的工作机制	162
升级 Volume Manager 集群功能	163
集群相关的 Volume Manager 实用程序和后台驻留程序	163
vxclust	164
vxclustadm	164
vxconfigd	165
vxconfigd 恢复	165
vxdg	166
vxdisk	168
vxrecover	169
vxdctl	169
vxstat	170
集群和快速镜像再同步 (FMR)	171
第 7 章：恢复	173
介绍	173
UNIX 引导进程	173
发生故障后引导	174
可能的根 (/)、swap 和 usr 配置	175
修复卷上的根 (/) 或 /usr 文件系统	175
备份和恢复根文件系统	178
故障和恢复过程	178
UNIX 分区时的故障	178
存取引导设备的故障	179
因 /etc/vfstab 中的不正确项而引起的故障	179
因缺少或损坏的 /etc/system 而引起的故障	180



从不可用或失效的 plex 引导系统引起的故障	182
热重定位和引导磁盘故障	183
重新添加和替换引导磁盘	183
重新添加故障引导磁盘	184
解除重定位封装子磁盘到新磁盘	185
替换故障引导磁盘	185
重新挂接磁盘	186
重新安装恢复	186
一般重新安装信息	186
重新安装和重新配置过程	187
Plex 和卷状态	195
Plex 状态	195
Plex 状态循环	197
Plex 内核状态	198
卷状态	198
卷内核状态	199
RAID-5 卷恢复	199
RAID-5 卷布局	199
创建 RAID-5 卷	201
初始化 RAID-5 卷	201
故障和 RAID-5 卷	202
RAID-5 恢复	203
杂项 RAID-5 操作	205
处理 RAID-5 日志	205
处理 RAID-5 子磁盘	206
启动 RAID-5 卷	207
不可启动的 RAID-5 卷	207
强制启动 RAID-5 卷	208
在启动 RAID-5 卷时恢复	209
更改 RAID-5 卷的属性	210



向 RAID-5 阵列写入	210
第 8 章：VxVM 性能监视	215
介绍	215
性能准则	215
数据分配	215
条带化	216
镜像	216
镜像和条带化	217
条带化和镜像	218
RAID-5	218
性能监视	218
性能优先级	218
获取性能数据	219
使用性能数据	220
调整 Volume Manager	222
一般调整原则	222
可调值	223
调整大型系统	227
词汇表	229
索引	239



前言

《VERITAS Volume Manager™ 管理员指南》提供有关如何使用 Volume Manager 的信息。

读者

本指南适用于负责安装、配置和维护受 VERITAS Volume Manager 控制的系统的系统管理员。

本指南假定用户已经：

- ◆ 具备 UNIX 操作系统的操作知识
- ◆ 基本了解系统管理
- ◆ 基本了解卷管理

范围

本指南的目的是为系统管理员提供在使用 Volume Manager 进行卷管理和系统管理时所涉及的过程及概念的详细知识。本指南包括如何利用各种 Volume Manager 功能的原则，如何使用 Volume Manager 命令创建和操作对象的说明，以及如何从磁盘故障中恢复的信息。

结构

本指南包括以下部分：

- ◆ 了解 Volume Manager
- ◆ 入门指南
- ◆ Volume Manager 操作
- ◆ 磁盘任务



- ◆ 卷任务
- ◆ Volume Manager 集群功能性
- ◆ 恢复
- ◆ VxVM 性能监视

使用本指南

本指南包含执行 Volume Manager 系统管理任务的指导说明。通过以下一个或多个界面可以执行 Volume Manager 管理功能：

- ◆ 一套复杂命令
- ◆ 单个自动命令 (vxassist)
- ◆ 一个由菜单驱动界面 (vxdiskadm)
- ◆ Storage Administrator (图形用户界面)

本指南描述如何使用各种 Volume Manager 命令行界面进行 Volume Manager 管理。有关如何使用 Storage Administrator 图形用户界面的详细资料，请参见 《VERITAS Volume Manager Storage Administrator 管理员指南》。有关 Volume Manager 实用程序的详细描述、每个实用程序的选项以及如何使用实用程序的详细资料，请参见 Volume Manager 帮助手册。

注意 大多数的 Volume Manager 命令需要超级用户特权或其它适当的特权。

相关文档

下列文档提供与 Volume Manager 相关的信息：

- ◆ VERITAS Volume Manager *安装指南*
- ◆ VERITAS Volume Manager *发行说明*
- ◆ VERITAS Volume Manager *硬件说明*
- ◆ VERITAS Volume Manager *参考指南*
- ◆ VERITAS Volume Manager Storage Administrator *管理员指南*
- ◆ VERITAS Volume Manager *帮助手册*



惯例

下表描述本指南中使用的印刷惯例。

字体	用法	实例
等宽	计算机输出、文件、目录、软件元素（如命令选项）、函数名和参数	从 <code>/etc/vx/tunefstab</code> 文件中读取可调值。 有关更多信息，请参见 <code>ls(1)</code> 帮助手册。
等宽（黑体）	用户输入	<code>#mount -F vxfs /h/filesys</code>
斜体	新术语、书名、强调的内容、用名称或值替换的变量	有关详细信息，请参见《 <i>用户指南</i> 》。 变量 <code>ncsize</code> 确定值为：

符号	用法	实例
%	C shell 提示符	
\$	Bourne/Korn shell 提示符	
#	超级用户提示符（所有 shell）	
\	在下一行继续输入；不用键入此字符	<code>#mount -F vxfs \ /h/filesys</code>
[]	在命令对照表中，方括号表示可选参数	<code>ls [-a]</code>
	在命令对照表中，竖线将互斥的参数分开	<code>mount [suid nosuid]</code>





了解 Volume Manager

介绍

本指南概括描述 VERITAS Volume Manager (VxVM[®]), 介绍什么是 Volume Manager, 它是如何工作的, 如何通过用户界面与之进行通信以及 Volume Manager 概念。“前言”中列出了提供有关 Volume Manager 的具体信息的相关文档。

VERITAS Volume Manager 为计算环境提供容易使用的联机磁盘存储管理。传统的磁盘存储管理经常要求计算机在脱机状态下工作, 这给用户带来了极大的不便。在分布式客户机 / 服务器环境下, 必须保持数据库和其它资源高度可用、易于存取, 并且免受由硬件故障造成的损坏。

VERITAS Volume Manager 提供可提高性能和确保数据的可用性和完整性的工具。Volume Manager 还在系统处于活动状态时动态配置磁盘存储。

本章介绍 VERITAS Volume Manager 概念并描述 Volume Manager 用以执行存储管理的工具。

本章讲述以下主题:

- ◆ [如何存储数据](#)
- ◆ [Volume Manager 概述](#)
- ◆ [物理对象](#)
- ◆ [卷和虚拟对象](#)
- ◆ [卷布局](#)
- ◆ [Volume Manager 和 RAID-5](#)
- ◆ [分层卷](#)
- ◆ [Volume Manager 用户界面](#)



如何存储数据

有许多方法可用于在物理磁盘上存储数据。这些方法在磁盘上组织数据，使数据可以有效地被存储和检索。基本的磁盘组织方法称为 *格式化*。格式化准备硬盘，以便可以使用预先安排的存储方式将文件写入磁盘和从磁盘中检索文件。

格式化硬盘和存储信息有两种方法：物理存储布局和逻辑存储布局。Volume Manager 使用 *逻辑存储布局* 方法。本章介绍 Volume Manager 支持的存储布局类型。

Volume Manager 概述

Volume Manager 使用 *对象* 进行存储管理。Volume Manager 使用的两种对象为 *物理对象* 和 *虚拟对象*。

- ◆ 物理对象

Volume Manager 使用物理磁盘存储数据。

- ◆ 虚拟对象

Volume Manager 在物理磁盘上创建称为 *卷* 的虚拟对象。每个卷都在一个或多个物理磁盘上记录数据或从中检索数据。文件系统、数据库或其它应用程序用访问物理磁盘的方法访问卷。卷也包括用于更改卷配置的其它虚拟对象。卷及其虚拟组件称为 *虚拟对象* 或 *Volume Manager 对象*。Volume Manager 对象可用于在磁盘上执行管理任务，并且不需中断应用程序和用户操作。

物理对象

本节描述 Volume Manager 所使用的物理对象（物理磁盘）。

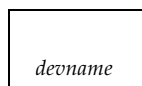
物理磁盘和磁盘命名

物理磁盘 是数据最终存储到的基本存储设备（介质）。通过使用 *设备名* 定位磁盘，可以在物理磁盘上存取数据。物理磁盘的设备名随所使用的计算机系统而异。并非所有的系统都需要使用全部参数。典型的设备名可以包含：`c#t#d#`，其中：

- ◆ `c#` 是控制器
- ◆ `t#` 是目标 ID
- ◆ `d#` 是磁盘号

图 1 “物理磁盘实例”显示了本文档中图示物理磁盘和设备名的方式 (*devname*)。例如，设备 `c0t0d0` 与系统中的控制器 0 相连，其目标 ID 为 0，物理磁盘号为 0。

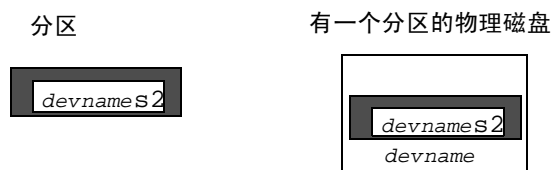
图 1. 物理磁盘实例



分区

在某些计算机系统中，物理磁盘可以分成一个或多个分区。分区号或 s# 添加在设备名称的最后。注意，一个分区可以是整个物理磁盘，如图 2 “分区实例” 中显示的分区。

图 2. 分区实例



卷和虚拟对象

当将物理磁盘置于 Volume Manager 控制之下时，就在物理对象和 Volume Manager 对象之间建立了联系。

Volume Manager 创建虚拟对象（或 Volume Manager 对象），并在对象之间建立逻辑连接。然后，Volume Manager 使用虚拟对象执行存储管理任务。Volume Manager 对象包括：

- ◆ 磁盘组
- ◆ VM 磁盘
- ◆ 卷
- ◆ plex（镜像）
- ◆ 子磁盘

这些对象将在后面的章节中描述。

卷是一种虚拟磁盘设备，在应用程序、数据库和文件系统看来如同物理磁盘。但是，卷没有物理磁盘的限制。当使用 Volume Manager 时，应用程序访问的是在 Volume Manager 磁盘（VM 磁盘）上创建的卷而非物理磁盘。

卷包含其它可用于处理卷中数据的虚拟对象。包含在卷中的虚拟对象是子磁盘和 plex。后面的章节将详细描述这些其它虚拟对象。虚拟对象的组合及卷处理虚拟对象的方式将在第 6 页的“卷”中加以介绍。



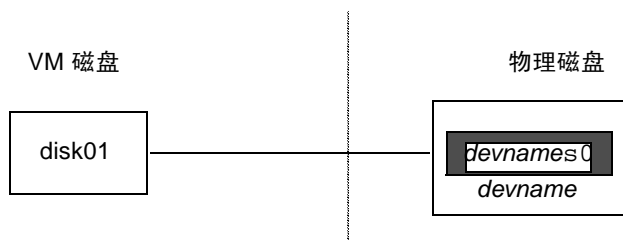
Volume Manager 磁盘

当将物理磁盘置于 Volume Manager 控制之下时，Volume Manager 磁盘（或 VM 磁盘）就被分配给物理磁盘。VM 磁盘受 Volume Manager 控制，并且通常在一个磁盘组中。每个 VM 磁盘至少对应一个物理磁盘。

VM 磁盘通常包括一个公共区域（已分配的存储）和一个存储 Volume Manager 内部配置信息的专用区域。

每个 VM 磁盘有一个唯一的 *磁盘介质名*（虚拟磁盘名）。可以提供磁盘名或允许 Volume Manager 以常用格式 `disk##` 分配默认名。图 3 “VM 磁盘实例”显示介质名为 `disk01` 的 VM 磁盘，该磁盘分配给物理磁盘 `devname`。

图 3. VM 磁盘实例



磁盘组

磁盘组 是共享公用配置的 VM 磁盘的集合。磁盘组配置是包含有关相关的 Volume Manager 对象及其属性和联系的详细信息的记录集。默认磁盘组为 `rootdg`（根磁盘组）。

可以根据需要创建另外的磁盘组。磁盘组允许将磁盘按逻辑集合分组。磁盘组及其组件可以作为一个单元从一台主机移到另一台主机。

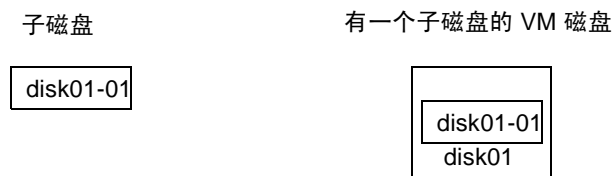
可以在磁盘组内创建卷。必须从同一磁盘组中的磁盘配置特定的卷。

子磁盘

子磁盘 是一组相邻的磁盘块。块是磁盘上的空间单元。Volume Manager 使用子磁盘分配磁盘空间。一个 VM 磁盘可以分成一个或多个子磁盘。一个子磁盘代表一个 VM 磁盘的特定部分，而 VM 磁盘映射到物理磁盘的特定区域。

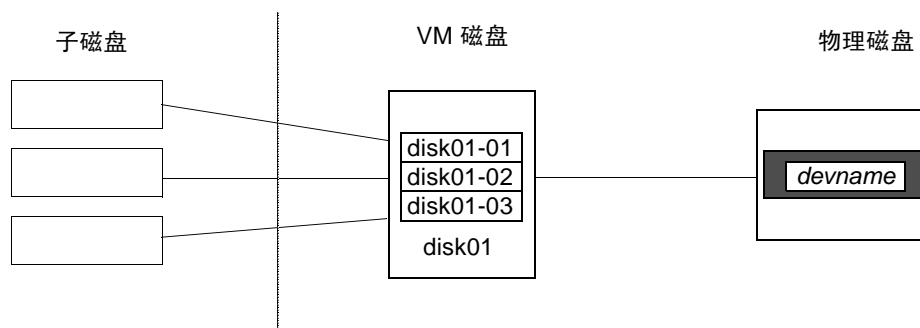
VM 磁盘的默认名称是 `disk##`（如 `disk01`），而子磁盘的默认名称是 `disk##-##`。图 4 “子磁盘实例”中，`disk01-01` 是名为 `disk01` 的 VM 磁盘上第一个子磁盘的名称。

图 4. 子磁盘实例



每个 VM 磁盘都可以包含多个子磁盘，但子磁盘不能重叠或共享 VM 磁盘的同一部分。[图 5 “分配给一个 VM 磁盘的三个子磁盘实例”](#) 显示了有三个子磁盘的 VM 磁盘。VM 磁盘分配给一个物理磁盘。

图 5. 分配给一个 VM 磁盘的三个子磁盘实例



任何不属于子磁盘的 VM 磁盘空间都是空闲空间。可以使用空闲空间创建新的子磁盘。

Volume Manager 3.0 发行版或更高版本允许子磁盘包含卷。在 Volume Manager 的早期版本中，子磁盘不能包含卷。有关更多信息，请参阅第 21 页的[“分层卷”](#)。

Plex

Volume Manager 使用子磁盘生成称为 *plex*（或镜像）的虚拟对象。plex 由位于一个或多个物理磁盘上的一个或多个子磁盘组成。若要在子磁盘上组织数据以构成 plex，请使用下列方法：

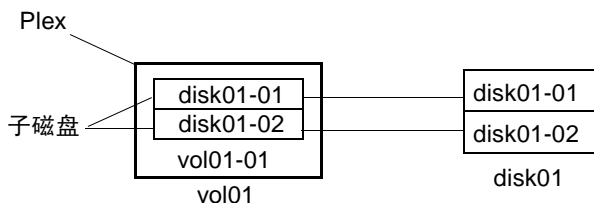
- ◆ 连续
- ◆ 条带化 (RAID-0)
- ◆ 带奇偶校验的条带化 (RAID-5)
- ◆ 镜像 (RAID-1)

注意 独立磁盘冗余阵列 (RAID) 是使用部分组合的存储容量存储阵列内数据复制信息的磁盘阵列，使您得以在磁盘发生故障时重新生成数据。

图 6 “有两个子磁盘的 Plex 实例”显示了有两个子磁盘的 plex。

连续、条带化 (RAID-0)、RAID-5 以及镜像 (RAID-1) 将在第 9 页的“卷布局”中加以介绍。

图 6. 有两个子磁盘的 Plex 实例



卷

卷由一个或多个 plex 组成，每个 plex 都包含卷数据的一个副本。由于卷的虚拟本性，卷并不受具体某个磁盘或磁盘中特定区域的限制。可以通过使用 Volume Manager 用户界面更改卷的配置。更改配置时不用中断正在使用卷的应用程序或文件系统。例如，卷可以镜像在单独的磁盘上或改为使用不同的磁盘存储。

卷最多可以包括 32 个 plex，每个 plex 包含一个或多个子磁盘。每个卷必须至少有一个关联的 plex，该 plex 必须包含卷数据的一个完整拷贝且必须至少有一个关联的子磁盘。注意，卷中的所有子磁盘必须属于同一个磁盘组。

有两个或更多个数据 plex 的卷为“镜像卷”，并且包含数据的镜像映像。每个 plex 包含相同的卷数据拷贝。有关镜像卷的更多信息，请参阅第 15 页的“镜像 (RAID-1)”。

Volume Manager 将 vol## 的默认命名惯例用于卷，而将 vol##-## 的默认命名惯例用于卷中的 plex。应当为卷选择有意义的名称。图 7 “有一个 Plex 的卷实例”显示了有一个 plex 的卷。

图 7. 有一个 Plex 的卷实例

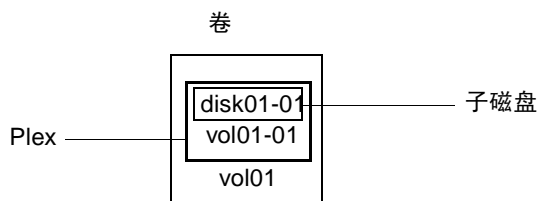


图 7 中的卷 vol01 具有如下特性：

- ◆ 包含一个名为 vol01-01 的 plex
- ◆ 该 plex 包含一个名为 disk01-01 的子磁盘
- ◆ 从 VM 磁盘 disk01 中分配子磁盘 disk01-01

图 8 “有两个 Plex 的卷实例” 显示了有两个 plex 的镜像卷。

图 8. 有两个 Plex 的卷实例

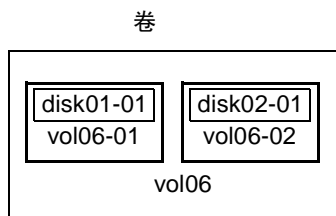


图 8 中的卷 vol06 具有如下特性：

- ◆ 包含两个名为 vol06-01 和 vol06-02 的 plex
- ◆ 每个 plex 包含一个子磁盘
- ◆ 从不同的 VM 磁盘（disk01 和 disk02）中分配每个子磁盘



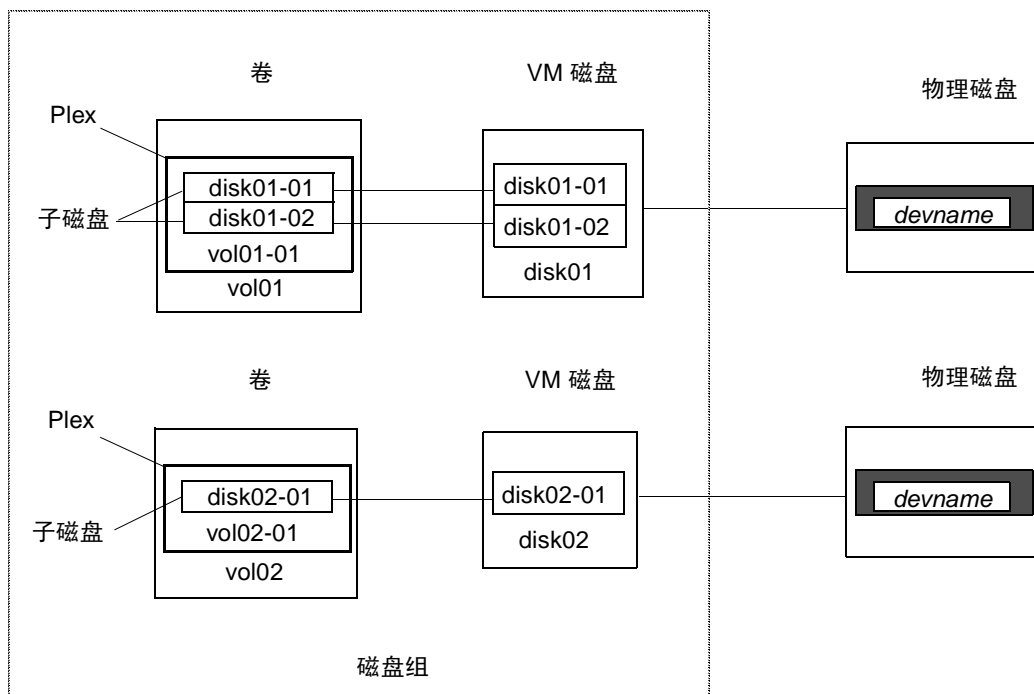
Volume Manager 的虚拟对象间联系

Volume Manager 的虚拟对象组合构成卷。卷中包含的虚拟对象包括：VM 磁盘、磁盘组、子磁盘和 plex。Volume Manager 的对象间联系如下：

- ◆ 若干个 Volume Manager 磁盘分组为磁盘组
- ◆ 一个或多个子磁盘（每个子磁盘代表磁盘的特定区域）组合成 plex
- ◆ 卷由一个或多个 plex 组成

图 9 “Volume Manager 的对象间联系”中的实例显示了 Volume Manager 虚拟对象间的联系以及虚拟对象与物理磁盘的关联方式。图 9 显示了有两个 VM 磁盘（disk01 和 disk02）的磁盘组。disk01 有一个具有一个 plex 和两个子磁盘的卷。disk02 有一个具有一个 plex 和一个子磁盘的卷。

图 9. Volume Manager 的对象间联系



卷布局

可以使用下列布局方法，以虚拟对象的形式组织数据来创建卷：

- ◆ 连续
- ◆ 条带化 (RAID-0)
- ◆ RAID-5（带奇偶校验的条带化）
- ◆ 镜像 (RAID-1)
- ◆ 镜像加条带化
- ◆ 条带化加镜像

下列章节分别介绍每种布局方法。

一个卷定义一个 **Volume Manager** 虚拟设备。卷的布局由卷与一个或多个 **plex** 的关联定义，而每个 **plex** 反过来映射到子磁盘。然后卷为 **Volume Manager** 客户机提供一个虚拟设备接口供其存取数据。这些逻辑生成块重新映射卷地址空间，而这些空间用于在运行时重定向 I/O。

不同的卷布局分别提供不同级别的存储服务。可以配置和重新配置卷布局，使之与特定级别的所需存储服务相匹配。

在 **Volume Manager** 的早期版本中，子磁盘仅限于直接映射到 **VM** 磁盘。这使子磁盘可以定义受 **VM** 磁盘公共区域支持的相邻范围的存储空间。**VM** 磁盘在活动时与基础物理磁盘关联（这便是 **Volume Manager** 逻辑对象映射到物理对象的方式）并在稳定的存储空间中存储数据。

卷布局和提供后备存储的物理磁盘相结合，并由此确定了特定虚拟设备可提供的存储服务。

在 **Volume Manager 3.0** 版或更高版本中可以构造“分层卷”，允许子磁盘和以前一样映射到 **VM** 磁盘，或者映射到称为 *存储卷* 的新逻辑对象。存储卷为递归级映射提供与顶层卷相似的布局。最终，映射的“底层”要求与 **VM** 磁盘关联，并由此与挂接的物理存储关联。

分层卷允许逻辑复合有多种组合，而其中的一些对配置虚拟设备可能很合适。由于允许在整个命令级随意使用分层卷会导致难于管理，因此，**Volume Manager 3.0** 版中已设计了一些现成的分层卷配置。

这些现成的配置与内置规则一起工作，自动在指定的约束范围内匹配所需级别的服务。对于在当前配置上工作的当前命令调用，**Volume Manager** 将“尽其所能”完成此自动配置。

若要从一套虚拟设备中获得所需的存储服务，可能需要在磁盘组中包括一组适当的 **VM** 磁盘，并且还要执行多个配置命令。

Volume Manager 3.0 版在力所能及的范围内，用自己的一组布局和管理接口处理初始配置和联机重新配置，以使此项工作更容易且更具有确定性。



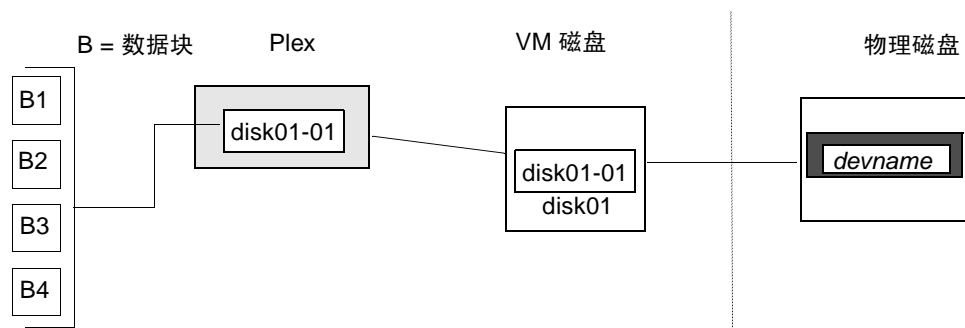
连续

连续以线性方式将数据映射到 plex 中的一个或多个子磁盘上。若要依次存取连续 plex 中的所有数据，首先从头到尾存取第一个子磁盘中的数据。然后从头到尾存取其余子磁盘中的数据，直到最后一个子磁盘的末尾为止。

连续 plex 中的子磁盘不必在物理上相邻，它们可以属于多个 VM 磁盘。使用驻留在多个 VM 磁盘上的子磁盘的连续称为跨越。

图 10 “连续实例”显示了有一个子磁盘的连续。

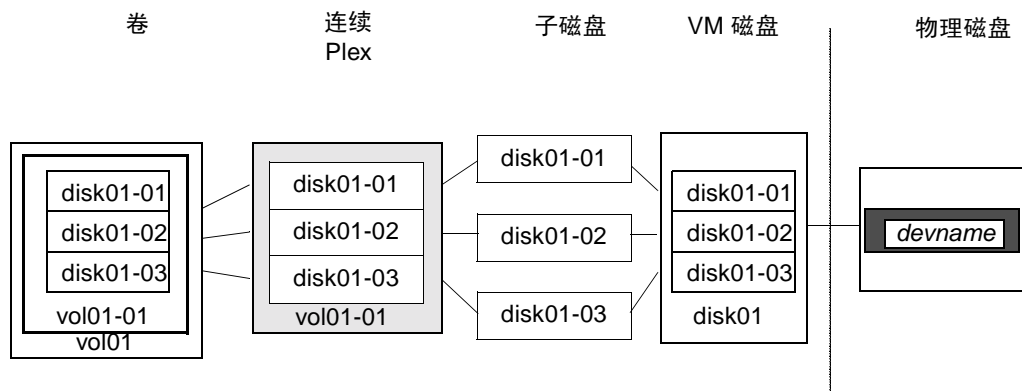
图 10. 连续实例



当任何一个磁盘上的 plex 相邻空间不足时，可以使用有多个子磁盘的连续。这种形式的连续可用于磁盘间的负载平衡，也可用于具体某个磁盘上的磁头运动优化。

图 11 “连续配置中的卷实例”显示了连续配置中的卷。

图 11. 连续配置中的卷实例

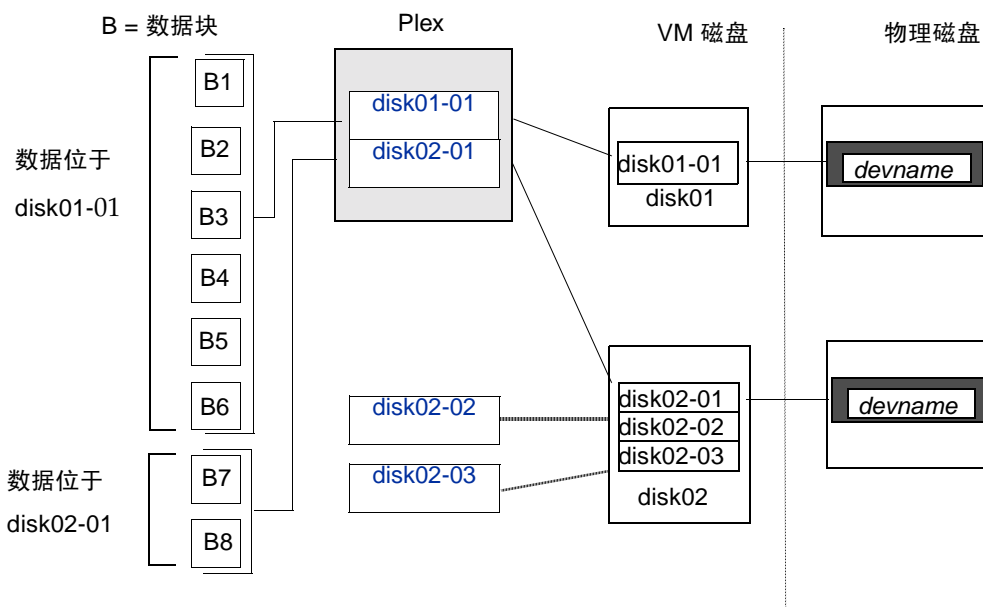


在图 12 所示的实例中，头六个数据块（从 B1 到 B6）使用磁盘上分配给 VM 磁盘 disk01 的大多数空间。这只需要 VM 磁盘 disk01 上的子磁盘 disk01-01 上的空间。但是，最后两个数据块 B7 和 B8 只使用磁盘上分配给 VM 磁盘 disk02 的部分空间。

VM 磁盘 disk02 上的剩余空闲空间可用于其它用途。在该实例中，子磁盘 disk02-02 和 disk02-03 都可用于其它磁盘管理任务。

图 12 “跨越实例”显示了分布在跨越 plex 中的两个子磁盘上的数据。

图 12. 跨越实例



警告 在多个磁盘上跨越 plex 增加了因磁盘故障导致分配的卷发生故障的可能性。使用镜像或 RAID-5（将在后面对它们加以介绍）减小了因单个磁盘故障导致卷故障的风险。

条带化 (RAID-0)

条带化 (RAID-0) 映射数据，使数据可以在两个或更多的物理磁盘间交叉存取。一个条带 plex 包含两个或更多的子磁盘，分布在两个或更多的物理磁盘上。数据交替且均匀地分配给条带 plex 的子磁盘。

子磁盘按组分不同的“列”，每个物理磁盘限于一列。每列包含一个或多个子磁盘，并且可以从一个或多个物理磁盘派生而来。每列的子磁盘的数量和大小可以不同。如果需要，可以向列中添加其它的子磁盘。

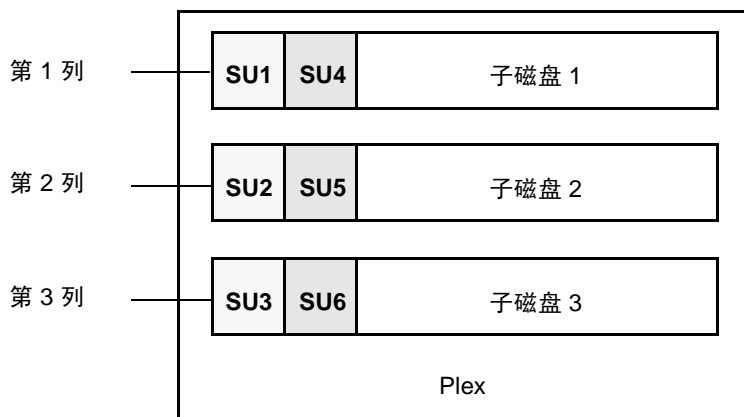
警告 在多个磁盘上对卷进行分条或拆分，将增加因磁盘故障而导致该卷发生故障的可能性。例如，如果五个卷在相同的五个磁盘上进行分条，那么，只要五个磁盘中的任何一个发生故障，都要求从备份中恢复所有五个卷。如果每个卷都在单独的磁盘上，则只需恢复一个卷。使用镜像或 RAID-5 可显著减少因单个磁盘故障导致大量的卷发生故障的可能性。

按同等大小的单元（条带单元称为 *stripe units*）分配数据，这些单元在列间交叉存取。每个条带单元是磁盘上的一组相邻块。条带单元的默认大小为 64 kilobytes。

例如，如果在一个条带 plex 中有三列和六个条带单元，则数据将在三个物理磁盘上分条，如图 13 “跨三个磁盘（列）的条带化” 中所示：

- ◆ 第一个和第四个条带单元分配在第 1 列中
- ◆ 第二个和第五个条带单元分配在第 2 列中
- ◆ 第三个和第六个条带单元分配在第 3 列中

图 13. 跨三个磁盘（列）的条带化



SU = 条带单元

条带由一组位于所有列上的相同位置的条带单元组成。在图 13 中，条带单元 1、2 和 3 构成一个条带。

按顺序查看，第一个条带由下列条带单元组成：

- ◆ 第 1 列中的条带单元 1
- ◆ 第 2 列中的条带单元 2
- ◆ 第 3 列中的条带单元 3

第二个条带由下列条带单元组成：

- ◆ 第 1 列中的条带单元 4
- ◆ 第 2 列中的条带单元 5
- ◆ 第 3 列中的条带单元 6

条带化一直延续到列的长度（如果所有列的长度都相同），或者到达最短列的末尾为止。较长列中的子磁盘结尾处所剩的空间成为未使用空间。

如果需要通过使用并行数据传输到多个磁盘的方式，将大量数据快速写入物理磁盘或从中读取数据，则条带化非常有用。条带化在平衡跨多个磁盘的多用户应用程序的 I/O 负载方面也很有用。

图 14 “每列有一个子磁盘的条带 Plex 实例”显示了带三个大小相等、只包含单个子磁盘的列的条带 plex。每个物理磁盘有一列。

图 14. 每列有一个子磁盘的条带 Plex 实例

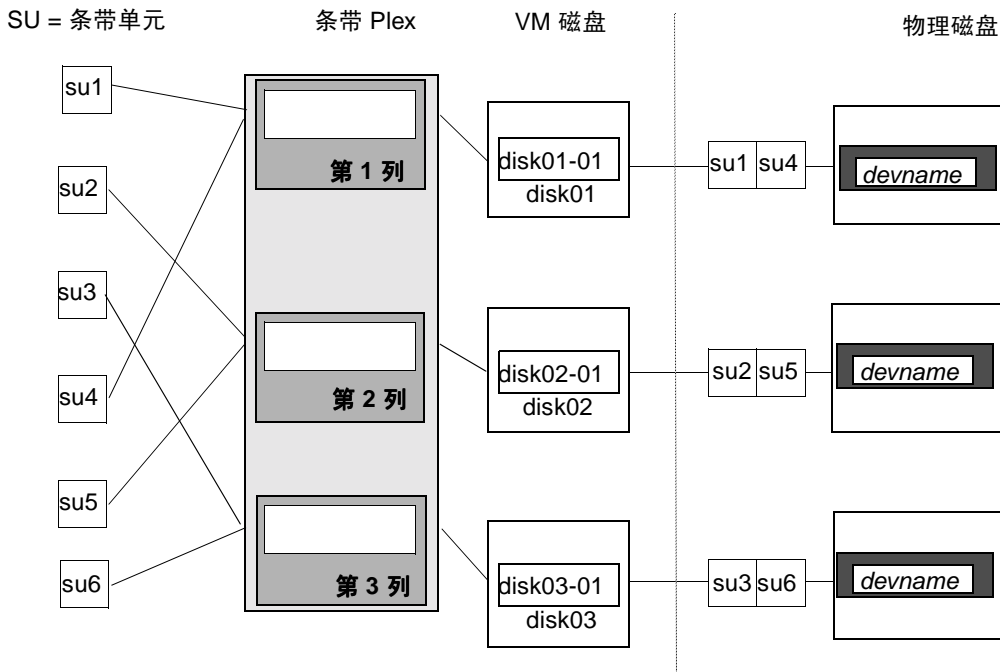
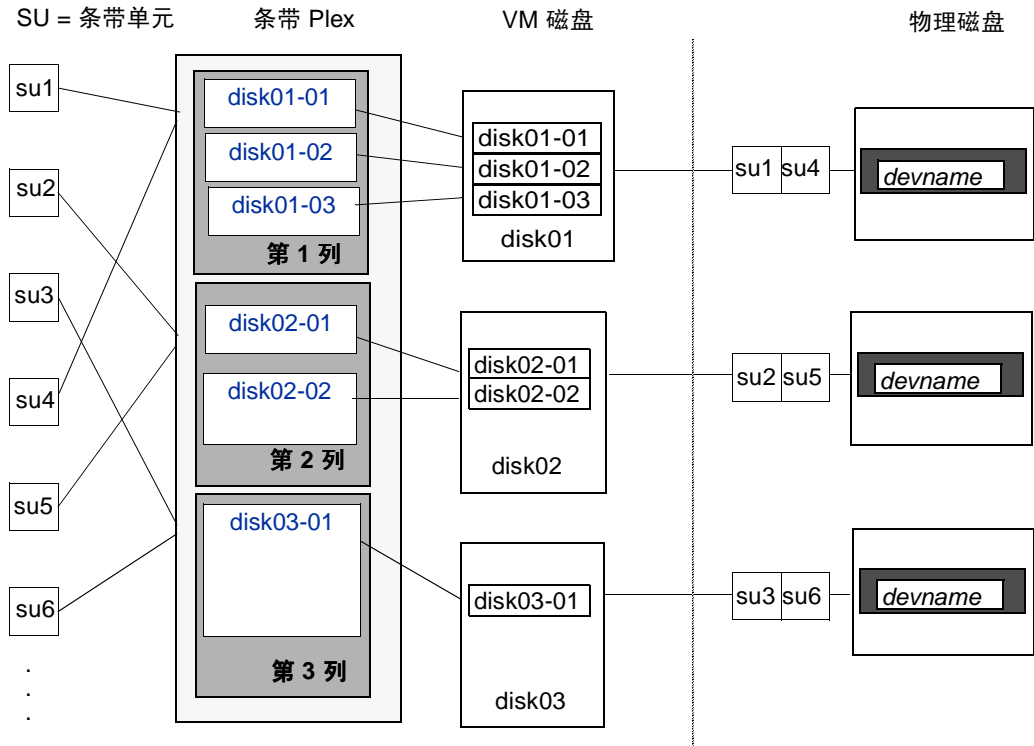


图 14 中的实例显示占用了 VM 磁盘上所有空间的三个子磁盘。一个条带 plex 中的每个子磁盘也有可能只占用 VM 磁盘的一部分，留下空闲空间用于其它磁盘管理任务。

图 15 “每列有连续子磁盘的条带 Plex 实例”显示了带三个包含不同大小子磁盘的列的条带 plex。每列都包含不同数目的子磁盘。每个物理磁盘有一列。通过使用每个条带化 VM 磁盘中的单个磁盘，可以创建条带 plex。也可以从同一磁盘的不同区域或其它磁盘分配空间（例如，如果 plex 增长）。列也可以包含不同 VM 磁盘中的子磁盘。

图 15. 每列有连续子磁盘的条带 Plex 实例



RAID-5

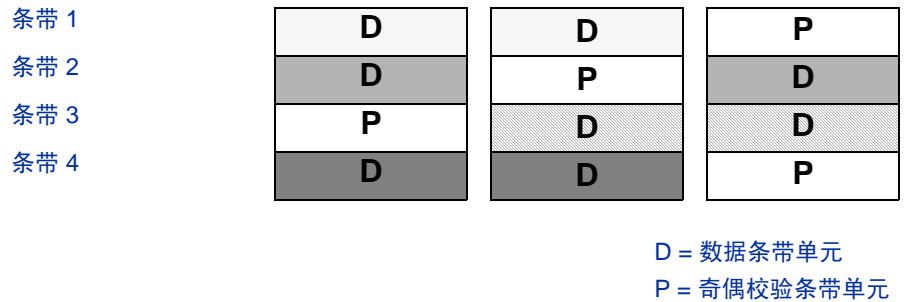
RAID-5 通过使用 *奇偶校验* 提供数据冗余。奇偶校验是一个计算出的值，用于在失败后重建数据。在数据写入 RAID-5 卷的同时，在数据上执行 *异或 (XOR)* 过程来计算奇偶校验。计算出的奇偶校验值随后写入卷。如果 RAID-5 卷的某一部分发生故障，可以从其余的数据和奇偶校验信息中重建发生故障的那部分卷上的数据。

RAID-5 卷维护卷数据的冗余。RAID-5 卷在跨多个磁盘“条带化”的 plex 中保存数据和计算出的奇偶校验值的副本。一旦发生磁盘故障，RAID-5 卷使用奇偶校验重建数据。可以在布局中同时使用连续和条带化。

RAID-5 卷可以做日志以使恢复时间减到最少。RAID-5 卷使用 RAID-5 日志保存当前写入的数据和奇偶校验的副本。RAID-5 日志是可选的，可以与 RAID-5 卷一道创建或以后再添加。

图 16 “RAID-5 模型中的奇偶校验位置”显示了 RAID-5 阵列配置中的奇偶校验位置。每个条带有一个包含奇偶校验条带单元的列和多个包含数据的列。奇偶校验分布在阵列中的所有磁盘上，从而减少了大量独立写操作的写入时间，因为这些写入不必等到单个奇偶校验磁盘可以接受数据时才进行。

图 16. RAID-5 模型中的奇偶校验位置



有关更多信息，请参见第 17 页的“[Volume Manager 和 RAID-5](#)”。

镜像 (RAID-1)

镜像使用多个镜像 (plex) 复制卷中包含的信息。一旦物理磁盘发生故障，故障磁盘上的 plex 变为不可用，但系统将使用未受影响的镜像继续运行。虽然一个卷可以只有一个 plex，但提供数据冗余至少需要两个 plex。为获得冗余，这些 plex 中的每一个都必须包含不同磁盘的磁盘空间。

当在大量的磁盘上进行条带化或跨越时，任何一个磁盘发生故障都会使整个 plex 变为不可用。如果几个磁盘中有任何一个可能会发生故障，则有必要考虑利用镜像提高条带卷或跨越卷的可靠性和可用性。

镜像加条带化 (RAID-1 + RAID-0)

Volume Manager 支持镜像与条带化的组合。此组合布局称为镜像 - 条带布局。当在同一卷上同时使用时，镜像和条带化在提供数据冗余（镜像）的同时还带来跨多个磁盘（条带化）分布数据的好处。

为使镜像和条带化在同时使用时有效，必须从单独的磁盘中分配镜像及其条带 plex。镜像的布局类型可以为连续的或条带的。



条带化加镜像 (RAID-0 + RAID-1)

Volume Manager 支持条带化与镜像的组合。此组合布局称为条带 - 镜像布局。在以前的版本中，镜像必须在条带化上发生。现在，镜像既可以在条带化上发生，也可以在条带化下发生。

镜像在条带化下发生时，镜像条带的每一列。如果条带足够大，每列可以有多个子磁盘，则可以分别镜像每个子磁盘。此布局增强了冗余并减少了发生错误时的恢复时间。

在镜像 - 条带布局中，如果一个磁盘发生故障，需分离整个 plex，从而丢失整个卷上的冗余。更换了磁盘后，必须使整个 plex 的内容是最新的。而恢复整个 plex 可能会花大量的时间。在条带 - 镜像布局中，如果一个磁盘发生故障，则只需分离发生故障的子磁盘，并且卷上只有发生故障的那部分才丢失冗余。更换了磁盘后，只需恢复卷的一部分。

与镜像加条带化相比，条带化加镜像为磁盘故障提供了更多的卷容错。发生磁盘故障时，条带化加镜像的恢复时间也更短。有关更多信息，请参见第 21 页的“[分层卷](#)”。



Volume Manager 和 RAID-5

本节描述 Volume Manager 实现 RAID-5 的方法。

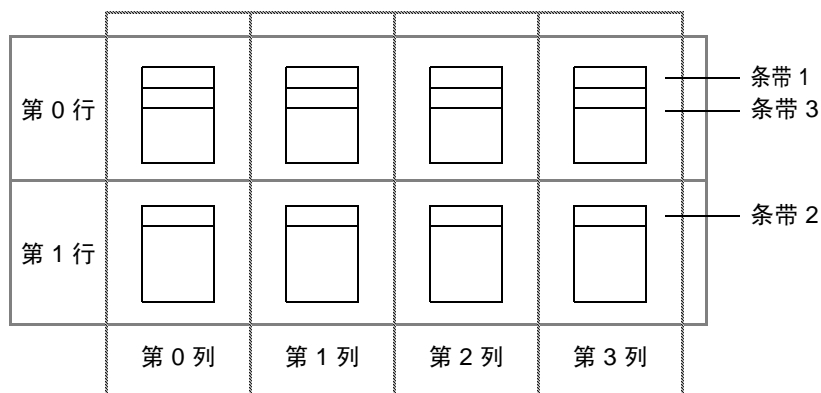
虽然镜像 (RAID-1) 和 RAID-5 都提供数据冗余，但它们使用的方法不同。镜像通过维护卷数据的多个完整副本提供数据冗余。写入镜像卷的数据反映在所有副本中。如果镜像卷的某一部分发生故障，系统将继续使用数据的其它副本。

RAID-5 通过使用奇偶校验提供数据冗余。奇偶校验是一个由计算得出的值，用于在发生故障后重建数据。在数据写入 RAID-5 卷的同时，在数据上执行异或 (XOR) 过程来计算奇偶校验值。计算出的奇偶校验值随后写入卷。如果 RAID-5 卷的某一部分发生故障，可以从其余的数据和奇偶校验信息中重建故障卷上的那部分数据。

传统的 RAID-5 阵列

传统的 RAID-5 阵列是按行和列组织的多个磁盘。列是许多位于阵列中同一有序位置的磁盘。行是支持全宽奇偶校验条带至少需要的磁盘数。图 17 “传统的 RAID-5 阵列” 显示了传统 RAID-5 阵列的行列排列。

图 17. 传统的 RAID-5 阵列



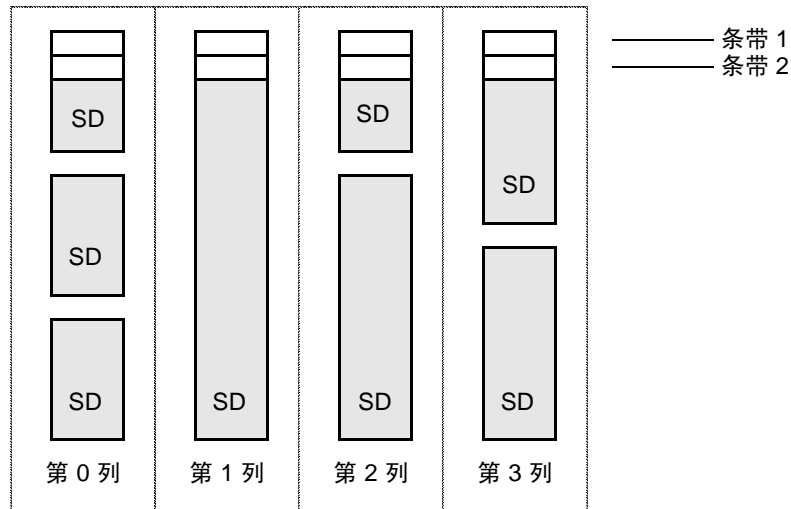
这种传统的阵列结构通过在每列添加更多的行来支持增长。条带化的实现方式是：将第一个条带应用于第 0 行中的所有磁盘、第二个条带应用于第 1 行中的所有磁盘、第三个条带又应用于第 0 行中的所有磁盘上，依此类推。这种阵列要求磁盘的所有列和行大小相等。



Volume Manager RAID-5 阵列

Volume Manager RAID-5 阵列结构不同于传统的结构。由于它的磁盘和其它对象的虚拟本性，Volume Manager 不使用行，而是使用由长度可变的子磁盘组成的列（如图 18 “Volume Manager RAID-5 阵列” 中所示）。每个子磁盘代表磁盘的一个特定区域。

图 18. Volume Manager RAID-5 阵列



SD = 子磁盘

在 Volume Manager RAID-5 阵列结构中，每一列可以由不同数目的子磁盘组成。特定列中的子磁盘可以从不同的物理磁盘派生而来。可以根据需要向列中添加其它的子磁盘。条带化（参见第 11 页的“条带化 (RAID-0)”）通过将第一个条带应用于每一列顶端的每一个子磁盘、另一个条带应用于接下来的每一个子磁盘，依此类推直到列的最后。对于每一个条带，在每列中都放入一个大小相等的条带单元。对于 RAID-5，默认条带单元大小为 16 kilobytes。

注意 当前不支持 RAID-5 卷的镜像。

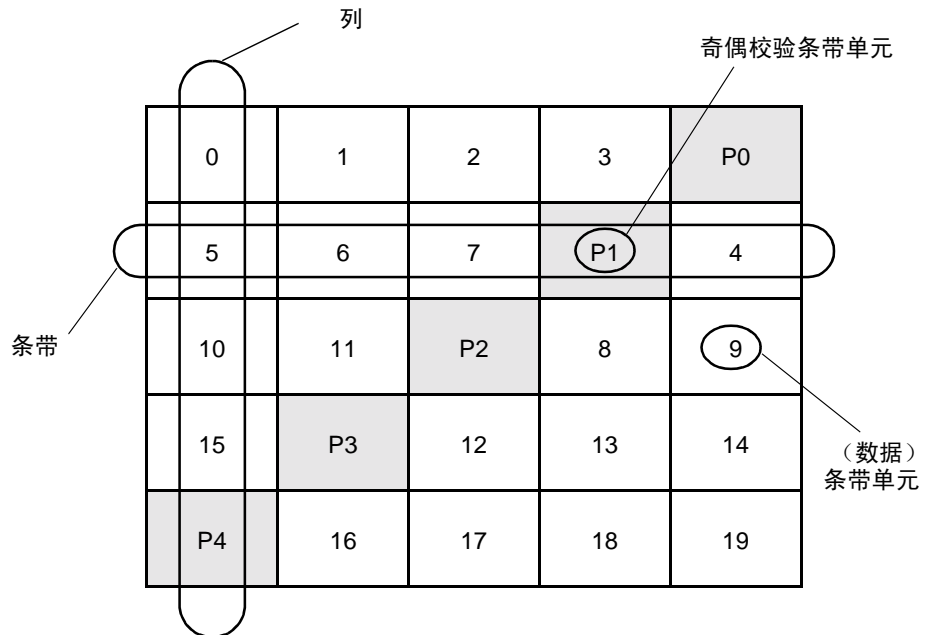
左对称布局

有多种数据和奇偶校验的布局可用于设置 RAID-5 阵列。Volume Manager 实现的 RAID-5 为左对称布局。左对称奇偶校验布局为随机 I/O 操作和大量有序 I/O 操作以便提供最佳性能。在性能方面，布局选择并不象列数选择和条带单元大小选择那样重要。

左对称布局在不同的列上条带化数据和奇偶校验，将奇偶校验值放在每个数据条带的不同列中。第一个奇偶校验条带单元位于第一个条带最右边的列中。每个连续的奇偶校验条带单元位于下一个条带，从前一个奇偶校验条带单元位置左移一列。如果条带比列多，则奇偶校验条带单元位置重新从最右边的列开始。

图 19 “左对称布局” 显示了有五个磁盘（每列一个）的左对称奇偶校验布局。

图 19. 左对称布局



对于每一个条带，都是从奇偶校验条带单元的右侧开始组织数据。在图 19 中，第一个条带的组织从 P0 开始，一直继续到条带单元 0 到 3。第二个条带的组织从 P1 开始，然后继续到条带单元 4，并且一直继续到条带单元 5 到 7。其余条带的组织继续以这种方式进行。

每一个奇偶校验条带单元都包含对同一条带中的数据条带单元数据执行异或 (XOR) 过程的结果。如果与列对应的磁盘上的数据因硬件或软件故障而无法存取时，可以恢复数据。恢复数据时，根据其余列的数据条带单元各自的奇偶校验条带单元（对每个条带而言），对其内容执行异或 (XOR) 过程。

例如，如果对应图 19 中最左边的列的磁盘发生故障，则该卷处于性能降低模式。处于性能降低模式时，可以根据奇偶校验条带单元 P0 对条带单元 1 到 3 执行异或过程来重建条带单元 0，然后根据奇偶校验条带单元 P1 对条带单元 4、6 和 7 执行异或过程来重建条带单元 5，依此类推直到重建故障列上的数据。

注意 在具有 RAID-5 布局的 plex 中，当多个列发生故障时需分离卷。因此不再允许卷满足读写请求。恢复发生故障的卷后，可能必须从备份中恢复用户数据。

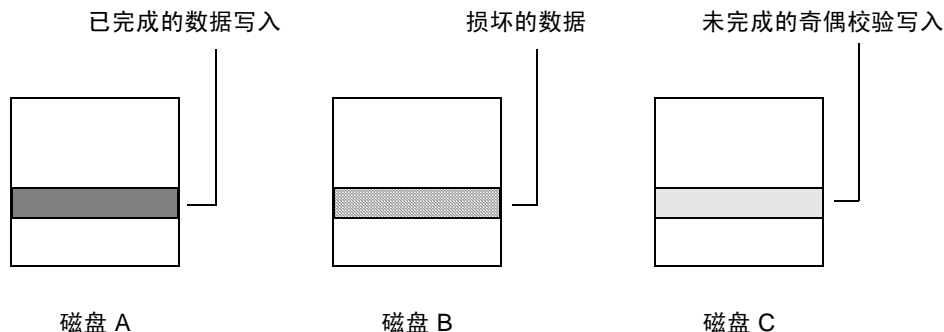
日志

日志（记录）用于防止恢复数据受到损坏。新数据和奇偶校验的日志记录在永久性设备（如驻留在磁盘上的卷或非易失性 RAM）上。新数据和奇偶校验然后写入磁盘。

如果不做日志，则当磁盘和系统同时发生故障时，任何活动的写操作所未涉及的数据可能会丢失或无声无息地损坏。如果发生这样的双重故障，没有办法知道正在写入磁盘数据分区的数据或正在写入奇偶校验分区的奇偶校验是否已真的写入。因此，恢复损坏的磁盘本身可能会造成损坏。

在图 20 “未完成的数据写入”中，恢复磁盘 B 依赖于已经完成的磁盘 A 上的数据和磁盘 C 上的奇偶校验。下图显示已完成的数据写入和未完成的奇偶校验写入导致不正确地重建磁盘 B 上的数据。

图 20. 未完成的数据写入



如果在将数据提交给阵列前记录所有写入的数据，则可以避免这种错误情况。用这种方法可以重播日志，从而在重建故障驱动器之前完成数据和奇偶校验的更新。

日志通过作为日志 plex 挂载的方式与 RAID-5 卷关联。每个 RAID-5 卷可以有多个日志 plex，在这种情况下日志区域是镜像的。

分层卷

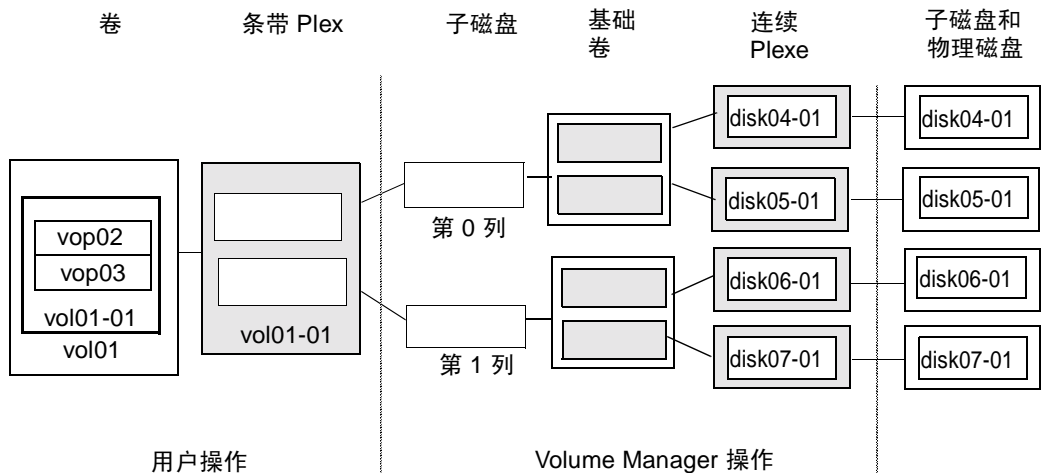
分层卷是在卷的顶部生成虚拟 Volume Manager 对象。分层卷结构的容错能力更强，而且冗余度比标准卷结构大。例如，在条带分层卷和镜像分层卷中，每个镜像 (plex) 都覆盖一个更小的存储空间区域，因此恢复速度比标准镜像卷快。图 21 “条带 - 镜像分层卷实例”显示了一个分层卷实例。

注意 Volume Manager 3.0 和更高版本支持分层卷，早期的版本则不支持。Volume Manager 3.0 或更高版本允许在卷（存储卷）上生成子磁盘，但 Volume Manager 的早期版本不支持子磁盘。

用户任务只能在分层卷的顶层卷上执行。不能通过处理内部结构的方法分离分层卷或在基础卷上执行任何其它操作。可以从包括卷和条带 plex 的用户操作区执行所有必需的操作。在图 21 中，“用户操作”区中的卷和条带 plex 允许执行正常的 Volume Manager 任务。

图 21 中的“Volume Manager 操作”区显示有两列子磁盘在基础卷上生成，而基础卷的每列都已在内部镜像。分层卷是 Volume Manager 中的基础结构，它们允许向 Volume Manager 中添加特定的功能。基础卷由 Volume Manager 专用，不用于用户操作。在这里介绍基础卷结构是为了帮助您了解分层卷的工作方式以及 Volume Manager 使用它们的原因。

图 21. 条带 - 镜像分层卷实例



为排除故障或执行其它操作（如将数据放入特定的磁盘），系统管理员可能需要处理分层卷结构。Volume Manager 使用分层卷执行下列任务和操作：

- ◆ 条带镜像（参见 vxassist 帮助手册）
- ◆ 连续镜像（参见 vxassist 帮助手册）
- ◆ 联机重新布局（参见 vxrelayout 和 vxassist 帮助手册）
- ◆ RAID-5 子磁盘移动（参见 vxsd 帮助手册）
- ◆ RAID-5 快照（参见 vxassist 帮助手册）

Volume Manager 和操作系统

Volume Manager 作为操作系统和数据管理系统（如文件系统和数据库管理系统）间的一个子系统运行。

在磁盘可以受 Volume Manager 控制之前，该磁盘必须可通过操作系统设备接口进行存取。Volume Manager 是在操作系统接口服务之上分层的子系统。因此，Volume Manager 依赖于操作系统存取物理磁盘的方式。

对于如下设备，Volume Manager 依赖操作系统：

- ◆ 操作系统（磁盘）设备
- ◆ 设备句柄
- ◆ VM 磁盘
- ◆ Volume Manager 动态多路径 (DMP) 设备元

Volume Manager 布局

一个卷定义一个 Volume Manager 虚拟设备。卷的布局由卷与一个或多个 plex 的关联定义，而每个 plex 反过来映射到多个子磁盘。然后卷提供一个表现给 Volume Manager 客户机的虚拟设备接口供其存取数据。这些逻辑生成块重新映射卷地址空间，而这些空间用于在运行时重定向 I/O。

不同的卷布局分别提供不同级别的存储服务。可以配置和重新配置卷布局，使之与特定级别的所需存储服务相匹配。

在 Volume Manager 的早期版本中，子磁盘仅限于直接映射到 VM 磁盘。这使子磁盘可以定义受 VM 磁盘公共区域支持的相邻范围的存储空间。VM 磁盘在活动时与基础物理磁盘关联（这便是 Volume Manager 逻辑对象映射到物理对象的方式）并在稳定的存储空间中存储数据。

卷布局和提供后备存储的物理磁盘相结合，并由此确定了特定虚拟设备可提供的存储服务。

在 Volume Manager 3.0 版或更高版本中可以构造“分层卷”，允许子磁盘和以前一样映射到 VM 磁盘，或者映射到称为 *存储卷* 的新逻辑对象。存储卷为递归级映射提供与顶层卷相似的布局。最终，映射的“底层”要求与 VM 磁盘关联，并由此与挂接的物理存储关联。

分层卷允许逻辑复合有多种组合，而其中的一些对配置虚拟设备可能很合适。由于允许在整个命令级随意使用分层卷会导致难于管理，因此，Volume Manager 3.0 版中已设计了一些现成的分层卷配置。

这些现成的配置与内置规则一起工作，自动在指定的约束范围内匹配所需级别的服务。对于在当前配置上工作的当前命令调用，Volume Manager 将“尽其所能”完成此自动配置。

若要从一套虚拟设备中获得所需的存储服务，可能需要在磁盘组中包括一组适当的 VM 磁盘，并且还要执行多个配置命令。

Volume Manager 3.0 版在力所能及的范围内，用自己的一组布局和管理接口处理初始配置和联机重新配置，以使此项工作更容易且更具有确定性。

Volume Manager 用户界面

本节简要描述 VERITAS Volume Manager 用户界面。

用户界面概述

Volume Manager 支持下列用户界面：

- ◆ Volume Manager Storage Administrator (VMSA)
- ◆ Storage Administrator 是 Volume Manager 的一个图形用户界面。Storage Administrator 提供可视化元素（如图标、菜单和对话框）来处理 Volume Manager 对象。Storage Administrator 也充当某些常用文件系统操作的界面。《*VERITAS Volume Manager Storage Administrator 管理员指南*》对 Storage Administrator 进行了描述。
- ◆ 命令行界面

Volume Manager 命令既有简单命令，也有要求详细输入的复杂命令。许多 Volume Manager 命令要求对 Volume Manager 概念有所了解。本章介绍了 Volume Manager 概念。大多数 Volume Manager 命令要求超级用户特权或其它适当的特权。《*VERITAS Volume Manager 参考指南*》对命令行界面进行了描述。
- ◆ Volume Manager Support Operations

Volume Manager Support Operations 界面 (vxdiskadm) 是用于磁盘和卷管理功能的菜单驱动界面。vxdiskadm 使用主菜单，供您选择要执行的存储管理任务。《*VERITAS Volume Manager 管理员指南*》对 vxdiskadm 进行了描述。

由某个界面创建的 Volume Manager 对象与由其它界面创建的对象兼容。



为什么要使用 Volume Manager

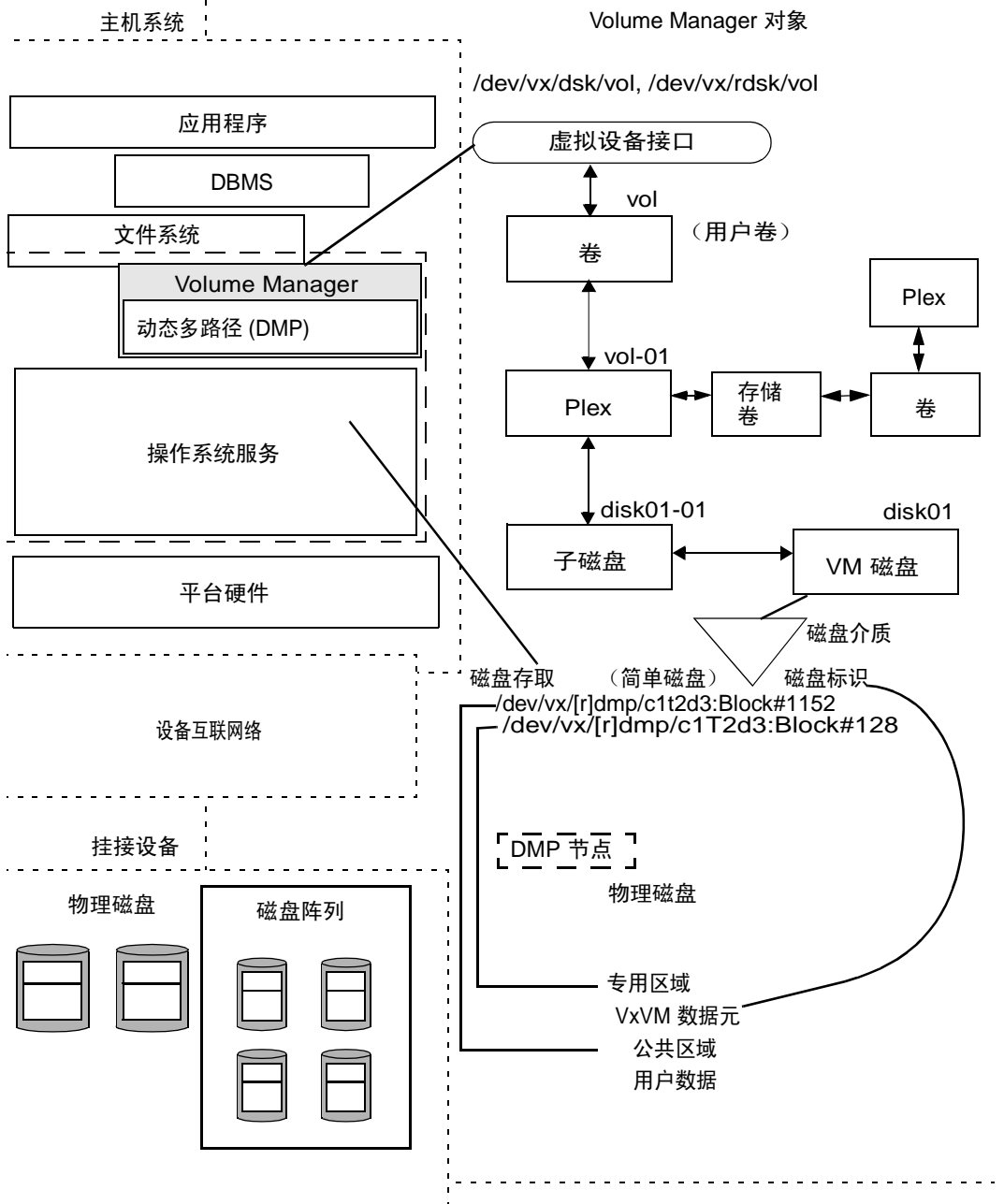
Volume Manager 将数据管理的物理部分与逻辑部分分开，从而提供了增强数据存储服务。Volume Manager 通过控制存储的下列方面来增强数据存储：

- ◆ 空间 —— 分配和使用
- ◆ 性能 —— 增强数据传送
- ◆ 数据可用性 —— 连续操作和多系统存取
- ◆ 设备安装 —— 集中和优化支持
- ◆ 系统 —— 多系统支持和专用 / 共享系统监视

参见第 25 页的图 22 “Volume Manager 系统概念”。



图 22. Volume Manager 系统概念



在主机系统上安装 Volume Manager 后，必须首先执行下列过程才能配置和使用 Volume Manager 对象：

- ◆ 使物理磁盘的内容受 Volume Manager 控制
- ◆ 将 Volume Manager 磁盘集中在磁盘组中
- ◆ 分配创建逻辑卷的磁盘组空间

仅在下列情况下才能使物理磁盘的内容受 Volume Manager 控制：

- ◆ 允许 Volume Manager 控制物理磁盘
- ◆ 磁盘不受其它存储管理器的控制

Volume Manager 写入受 Volume Manager 控制的物理磁盘（要求的磁盘）的标识信息。即使在物理磁盘断开或系统断电后，也可以标识要求的磁盘。Volume Manager 随后可以重新构成磁盘组和逻辑对象，以提供故障检测能力和加快系统恢复速度。

介绍

本节简要描述安装 Volume Manager 和设置 Volume Manager 操作所必须运行的后台驻留程序所需的步骤。本章还提供帮助您您在系统中设置存储管理的原则。

有关如何安装和设置 Volume Manager 和 Storage Administrator 的详细信息，请参阅《*VERITAS Volume Manager 安装指南*》。

本章讲述以下主题：

- ◆ Volume Manager 初始化
- ◆ Volume Manager 后台驻留程序
- ◆ 系统设置
- ◆ 系统设置原则
- ◆ 系统保护

Volume Manager 初始化

使用 `vxinstall` 程序初始化 Volume Manager。 `vxinstall` 将指定的磁盘置于 Volume Manager 控制下。默认情况下，这些磁盘放置在 `rootdg` 磁盘组中。必须使用 `vxinstall` 将至少一个磁盘初始化为 `rootdg`。然后可以使用 `vxdiskadm` 或者 Storage Administrator 将其它磁盘初始化或者封装到磁盘组中。

完成了软件包安装后，请按照以下步骤初始化 Volume Manager。

1. 以超级用户身份登录。
2. 如果想使磁盘不受 Volume Manager 的控制，则创建一个 `disks.exclude` 文件。 `vxinstall` 将忽略该文件中列出的任何磁盘。将该文件放在 `/etc/vx/disks.exclude` 中。
3. 如果想使某个控制器上的所有磁盘不受 Volume Manager 的控制，则创建一个 `cntrls.exclude` 文件。将该文件放在 `/etc/vx/cntrls.exclude` 中。



4. 启动 `vxinstall`，方法是输入命令：`vxinstall`。

`vxinstall` 于是执行下列操作：

- ◆ 运行和显示许可证信息并提示输入密钥
- ◆ 检查并列出挂载在系统上的所有控制器
- ◆ 允许您选择初始化进程：快速安装或者自定义安装

快速安装给您提供选择初始化或封装所有磁盘的选项。如果要封装特定控制器上的某些磁盘而初始化其它磁盘，则使用自定义安装进程。

自定义安装允许您控制向 Volume Manager 控制中添加的磁盘及其添加方式。可以初始化或封装一个控制器上的所有磁盘，或者初始化某些磁盘而封装其它磁盘。

有关如何使用快速安装或自定义安装选项的详细信息，请参阅《*VERITAS Volume Manager 安装指南*》。

您选择的系统配置方式决定了是否需要停机并重新引导。如果选择封装所有磁盘，则必须重新引导。如果必须重新引导，`vxinstall` 将通知您。

安装和初始化 VM 后，可以使用如下命令确认主要的 Volume Manager 进程是否正在运行（`vxconfigd`、`vxnotify` 和 `vxrelocd`）。

```
# ps -ef | grep vx
```

Volume Manager 后台驻留程序

为使 Volume Manager 正常运行，必须运行以下两个后台驻留程序：

- ◆ `vxconfigd`
- ◆ `vxiod`

配置后台驻留程序 `vxconfigd`

Volume Manager 配置后台驻留程序 (`vxconfigd`) 维护 Volume Manager 磁盘和磁盘组配置。`vxconfigd` 向内核传达配置更改并修改存储在磁盘上的配置信息。

启动 Volume Manager 配置后台驻留程序

`vxconfigd` 在引导过程中由启动脚本调用。

若要确定卷后台驻留程序是否已启用，请输入如下命令：

```
# vxdctl mode
```

如果 `vxconfigd` 正在运行并启用，将显示下列信息：

```
方式: enabled
```

如果 `vxconfigd` 正在运行但没有启用，将显示下列信息：

```
方式: disabled
```

若要启用卷后台驻留程序，请输入如下命令：

```
# vxctl enable
```

如果 `vxconfigd` 没有运行，将显示下列信息：

```
方式: not-running
```

若要启动 `vxconfigd`，请输入如下命令：

```
# vxconfigd
```

一旦启动，`vxconfigd` 自动成为一个后台进程。

默认情况下，`vxconfigd` 将错误发送到控制台。不过，可以配置 `vxconfigd` 将错误发送到日志文件。

有关 `vxconfigd` 后台驻留程序的更多信息，请参阅 `vxconfigd(1M)` 和 `vxctl(1M)` 帮助手册。

卷 I/O 后台驻留程序 vxiod

卷扩展 I/O 后台驻留程序允许 (`vxiod`) 在不阻塞调用进程的情况下执行扩展 I/O 操作。

有关 `vxiod` 的更详细信息，请参阅 `vxiod(1M)` 帮助手册。

启动卷 I/O 后台驻留程序

`vxiod` 后台驻留程序在系统引导时启动。通常有几个 `vxiod` 后台驻留程序一直在运行。初始安装后重新引导时 `vxiod` 将启动。

输入如下命令检验 `vxiod` 后台驻留程序是否正在运行：

```
# vxiod
```

由于 `vxiod` 是内核线程，无法通过 `ps` 命令看到，因此这是查看任何 `vxiod` 后台驻留程序是否正在运行的唯一方法。

如果有任何 `vxiod` 后台驻留程序正在运行，将显示下列信息：

```
10 volume I/O daemons running
```

其中，10 是当前正在运行的 `vxiod` 后台驻留程序的数量。

如果当前没有任何 `vxiod` 后台驻留程序正在运行，则输入如下命令启动一些后台驻留程序：

```
# vxiod set 10
```

其中，可以用所需的 `vxiod` 后台驻留程序数代替 10。建议对系统内的每个 CPU，至少存在一个 `vxiod` 后台驻留程序。



系统设置

本节包含可帮助您设置系统以获得高效存储管理的信息。有关特定的设置任务，请参考《*VERITAS Volume Manager 参考指南*》和《*VERITAS Volume Manager Storage Administrator 管理员指南*》。

下列系统设置顺序具有代表性，可作为一个实例。您的系统要求可能与此不同。系统设置原则为特定的设置配置提供了有用的信息。

系统设置顺序实例

下表描述了设置存储管理系统时可以使用的典型步骤。

初始设置

- ◆ 将磁盘置于 Volume Manager 控制下。
- ◆ 创建新的磁盘组（如果不想用 `rootdg` 或者需要其它磁盘组）。
- ◆ 创建卷。
- ◆ 将文件系统放在卷上。

选项

- ◆ 封装引导 / 根磁盘并对其镜像以创建备用引导盘。
- ◆ 指定热重定位备用磁盘。
- ◆ 向卷中添加镜像。

维护

- ◆ 重定卷和文件系统的大小。
- ◆ 添加更多的磁盘 / 磁盘组。
- ◆ 创建快照。

系统设置原则

这些一般原则能够帮助您理解和设计有效的存储管理系统。可以使用每一部分的交叉引用获得有关特定原则的更多信息。

热重定位原则

可以在使用热重定位时遵循下列一般原则。有关更多信息，请参见第 42 页的“[热重定位](#)”。

- ◆ 默认情况下启用热重定位功能。尽管可以禁用热重定位功能，但最好还是启用它。
- ◆ 尽管热重定位不要求指定备用磁盘，但在每个磁盘组内可至少指定一个磁盘作为备用磁盘。这使您可以在一定程度上控制用于重定位的磁盘。如果无备用磁盘，**Volume Manager** 将使用磁盘组内任何可用的空闲空间。当空闲空间用于重定位目的时，则重定位后性能可能会降低。
- ◆ 发生热重定位后，可另外指定一个或更多的磁盘作为备用磁盘以增加备用空间（原来的空闲空间有一部分可能被重定位的子磁盘占用）。
- ◆ 如果特定的磁盘组跨接在多个控制器上且具有不止一个备用磁盘，则可以在不同的控制器上设置备用磁盘以防其中一个控制器出现故障。
- ◆ 对于镜像卷，磁盘组中必须至少有一个磁盘仍未包含卷的镜像。该磁盘必须是有一定可用空间的备用磁盘，或者是有一定空闲空间且可用于热重定位的正规磁盘。
- ◆ 对于镜像卷和条带卷，磁盘组中必须至少有一个磁盘仍未包含卷的镜像或条带 plex 内的其它子磁盘。该磁盘必须是有一定可用空间的备用磁盘，或者是有一定空闲空间且可用于热重定位的正规磁盘。
- ◆ 对于 RAID-5 卷，磁盘组中必须至少有一个磁盘仍未包含卷的 RAID-5 plex（或其中一个日志 plex）。该磁盘必须是有一定可用空间的备用磁盘，或者是有一定空闲空间且可用于热重定位的正规磁盘。
- ◆ 如果镜像卷中的某个 DRL 日志子磁盘是其数据 plex 的一部分，则不能重定位该 plex。可以将日志子磁盘放在不含任何数据的 plex（日志 plex）中。
- ◆ 热重定位不保证能够保持原来的性能特性或数据布局。可检查刚重定位的子磁盘的位置，确定是否应将它们重定位到更合适的磁盘以恢复原来的性能收益。
- ◆ 如果镜像了根磁盘，当它出现故障时，热重定位能够创建一个新的根磁盘镜像。因此，**rootdg** 磁盘组应包含足够的相邻备用空间或空闲空间以容纳根磁盘上的卷（**rootvol** 和 **swapvol** 要求相邻的磁盘空间）。
- ◆ 尽管可以在备用磁盘上生成 VxVM 对象（使用 **vxmake** 或 **Storage Administrator** 界面），但最好是只将备用磁盘用于热重定位。



条带化原则

使用条带化时可遵循下列一般原则。有关更多信息，请参见第 11 页的“[条带化 \(RAID-0\)](#)”。

- ◆ 不要在同一个物理磁盘上放置超过一系列的条带 plex。
- ◆ 认真计算条带单元大小。一般情况下，建议使用适中的条带单元大小（如 64K 字节，这也是 vxassist 使用的默认值）。如果将条带单元大小设置为磁轨大小不可行，并且不知道应用程序的 I/O 模式，则建议使用 64K 字节作为条带单元大小。

注意 许多新式磁盘驱动器具有“可变几何结构”，意思是柱面之间的磁轨大小不同（即外磁轨比内磁轨具有更多的扇区）。因此，将磁轨大小用作条带单元大小并非总是合适的。对于这些驱动器，应使用适中的条带单元大小（如 64K 字节），除非知道应用程序的 I/O 模式。

- ◆ 如果磁盘没有同步主轴，则条带单元小的卷会表现出很差的有序 I/O 等待时间。一般情况下，如果与较大的条带单元和多线程或很大程度上是异步的随机 I/O 流一起使用，非同步主轴磁盘上的条带化性能会更好。
- ◆ 通常情况下，条带内的物理磁盘数越多，I/O 性能的提高越大；但是，这会缩短卷的有效平均故障间隔时间。如果这导致问题，可将条带化与镜像组合使用，为高性能的卷提供改进的可靠性。
- ◆ 如果只条带化一个镜像卷的 plex，务必将条带 plex 的卷策略设置为 prefer。（默认读策略 select 自动完成此设置。）
- ◆ 如果条带化多个镜像卷的 plex，确保每个条带 plex 的条带单元大小都相同。
- ◆ 如果可能，将条带卷的子磁盘分布在不同的控制器和总线相连的驱动器上。
- ◆ 避免使用不支持重叠寻找的控制器（这样的控制器很少见）。

vxassist 命令在为卷内的条带 plex 分配空间时，会自动应用并加强许多这些原则。

镜像原则

使用镜像时可遵循这些一般原则。有关更多信息，请参见第 15 页的“[镜像 \(RAID-1\)](#)”。

- ◆ 不要将镜像卷的不同 plex 中的子磁盘放在同一物理磁盘上。该操作损害镜像适用性的得益并使性能降低。使用 vxassist 可防止发生这种情况。
- ◆ 为了通过使用镜像提供最佳的性能，至少 70% 的物理 I/O 操作必须是读操作。读操作的比例越高，获得的性能收益越大。在写密集工作环境下，镜像并不会带来性能得益，也不会导致性能下降。

注意 UNIX 操作系统操作文件系统的高速缓冲存储器。频繁的阅读请求可通过高速缓冲存储器来满足。这将导致物理 I/O 操作的读出 / 写入比在整个文件系统中以写入操作占多数（相对应用程序的读 / 写比而言的）。

- ◆ 如果可行，在镜像或条带化时使用挂载在不同控制器上的磁盘。大部分磁盘控制器都支持重叠寻找，即允许同时在两个磁盘上开始寻找。不要在挂载不支持重叠寻找的控制器上的磁盘配置同一个卷的两个 plex。这一点对驱动器上没有高速缓冲存储器的旧式控制器或 SCSI 磁盘很重要，但对于当今大部分工作站和服务器的许多新式 SCSI 磁盘和控制器则没有那么重要。在不同的控制器上镜像很有用，因为系统可摆脱控制器故障。其它控制器能够通过其它镜像继续提供数据。
- ◆ 当一个 plex 在多个磁盘上条带化或连续，或者位于非常快的设备上时，会表现出出色的性能。这时可将读策略设置为首选这个“更快的” plex。默认情况下，有一个条带 plex 的卷被配置为优先读取此条带 plex。

脏区日志 (DRL) 原则

使用脏区日志时可以遵循下列一般原则。有关更多信息，请参见第 45 页的“脏区日志”。

脏区日志 (DRL) 功能能够加快系统崩溃后恢复镜像卷的速度。如果启用了 DRL，Volume Manager 将跟踪卷内由于向 plex 写入而更改的区域。Volume Manager 维护一个位图并将这些信息存储在日志子磁盘中。日志子磁盘是为了提供 DRL 而为卷定义并添加到卷中的。日志子磁盘与 plex 无关，被 plex 策略忽略且只用于存放 DRL 信息。

注意 在写密集环境下，使用脏区日志会对系统性能造成负面影响。

使用 DRL 时应遵循以下原则：

- ◆ 为使 DRL 生效，必须镜像卷。
- ◆ 为使 DRL 工作，卷上必须存在至少一个日志子磁盘。但是，每个 plex 只能有一个日志子磁盘。
- ◆ 用作日志子磁盘的子磁盘不能包含必要的数据库。
- ◆ 可以通过使卷包含不止一个日志子磁盘（但每个 plex 只能有一个）来“镜像”日志子磁盘。这将确保即使某个磁盘故障导致一个日志子磁盘不可存取，日志仍能继续。
- ◆ 必须将日志子磁盘配置为具有两个或更多的扇区（最好是一个偶数，因为日志子磁盘若包含奇数个扇区，则不会用到最后一个扇区）。日志子磁盘的大小通常与卷大小成正比。如果卷小于 2 GB，则包含两个扇区的日志子磁盘就足够了。于是，对于每一个额外的 2 GB 卷大小，日志子磁盘大小增加两个扇区。但是，在默认情况下，vxassist 选择合理的大小。一般情况下，建议使用 vxassist 提供的默认日志子磁盘长度。
- ◆ 如有可能，不要将日志子磁盘放在频繁使用的磁盘上。
- ◆ 永久性（非易失性）存储磁盘必须用于日志子磁盘。

镜像和条带化原则

在同时使用镜像和条带化时可以遵循以下一般原则。有关更多信息，请参见第 15 页的“镜像加条带化 (RAID-1 + RAID-0)”。



- ◆ 确保有足够的磁盘可用于条带化和镜像配置。对于条带 plex，需要至少两个磁盘，而对于镜像，需要一个或多个其它磁盘。
- ◆ 决不要将一个 plex 中的子磁盘和其它 plex 中的子磁盘放在同一个物理磁盘上。遵循第 32 页的“条带化原则”中描述的条带化原则。
- ◆ 遵循第 32 页的“镜像原则”中描述的镜像原则。

条带化和镜像原则

在同时使用条带化和镜像时可以遵循以下一般原则。有关更多信息，请参见第 16 页的“条带化加镜像 (RAID-0 + RAID-1)”。

- ◆ 确保有足够的磁盘可用于条带化和镜像配置。对于条带 plex，需要至少两个磁盘，而对于镜像，需要一个或多个其它磁盘。
- ◆ 决不要将一个 plex 中的子磁盘和其它 plex 中的子磁盘放在同一个物理磁盘上。遵循第 32 页的“条带化原则”中描述的条带化原则。
- ◆ 遵循第 32 页的“镜像原则”中描述的镜像原则。

RAID-5 原则

在使用 RAID-5 时可以遵循以下一般原则。有关更多信息，请参见第 14 页的“RAID-5”。

一般情况下，同时使用镜像和条带化的原则也适用于 RAID-5。RAID-5 也应遵守下列原则：

- ◆ 每个 RAID-5 卷只能有一个 RAID-5 plex（但可以存在多个日志 plex）。
- ◆ RAID-5 plex 必须派生自两个或更多个物理磁盘上的至少两个子磁盘。如果存在任何日志 plex，则它们必须属于用于 RAID-5 plex 的磁盘以外的磁盘。
- ◆ 可以镜像和条带化 RAID-5 日志。
- ◆ 如果没有显式指定卷的长度，则卷的长度设置为与卷关联的任意 RAID-5 plex 的长度，否则设置为零。如果显式设置了卷的长度，则肯定是关联的 RAID-5 plex（若有）条带单元大小的倍数。
- ◆ 如果没有显式指定日志的长度，则日志长度设置为关联的最小 RAID-5 日志 plex（若有）的长度。如果不存在任何关联的 RAID-5 日志 plex，则设置为零。
- ◆ 稀疏 RAID-5 的日志 plex 无效。

系统保护

磁盘故障会导致两类问题：故障磁盘上的数据丢失和失去对系统的存取权。失去存取权可能是由主要磁盘（用于系统操作的磁盘）的故障引起的。VERITAS Volume Manager 能够防止系统出现这些问题。



为了维护系统可用性，必须镜像用于运行和引导系统的重要数据。必须保留数据，以便在发生故障时使用。

下面对保护系统和数据提出一些建议：

- ◆ 将包含根文件系统的磁盘（根或引导磁盘）通过封装置于 Volume Manager 控制之下。这将使根设备和交换设备转为卷（`rootvol` 和 `swapvol`）。然后镜像根磁盘，获得一个引导用的备用根磁盘。通过镜像用于引导的关键磁盘，确保一旦有任何磁盘发生故障，不会使系统不可引导和不可用。

为使系统可用性达到最大，为 `rootvol`、`swapvol`、`usr` 和 `var` 卷创建镜像。有关更多信息，请参见第 173 页的“恢复”一章。

- ◆ 使用镜像保护数据。通过镜像数据，可防止因磁盘故障丢失数据。为了保留数据，创建并使用包含至少两个数据 `plex` 的镜像卷。这些 `plex` 必须位于不同的磁盘上。如果因磁盘故障导致一个 `plex` 失败，镜像卷中的数据仍存在于其它磁盘上。

如果使用 `vxassist` 镜像创建镜像，它将定位镜像，使一个磁盘的损失不会造成数据损失。默认情况下，`vxassist` 不创建镜像卷；可以编辑文件 `/etc/default/vxassist` 将默认布局设置为镜像。

- ◆ 保留 Volume Manager 热重定位功能的启用状态，以便能够自动检测故障、通知故障特性、尝试重定位任何受影响的冗余子磁盘以及启动恢复过程。对于每个磁盘组至少提供一个热重定位备用磁盘，这样一旦发生故障，有足够的空间可用重定位。

如果已镜像根磁盘，当原来的根磁盘发生故障时，热重定位能够自动创建根磁盘的另一个镜像。`rootdg` 磁盘组应包含足够的相邻备用空间或空闲空间用于根磁盘上的卷（`rootvol` 和 `swapvol` 卷要求相邻的磁盘空间）。

- ◆ 对于镜像卷，利用脏区日志功能加快系统崩溃后镜像卷的恢复速度。确保每个镜像卷至少有一个日志子磁盘。（注意，`rootvol`、`swapvol` 和 `usr` 卷不能为 DRL 卷。）
- ◆ 对于 RAID-5 卷，利用日志来防止恢复数据的损坏。确保每个 RAID-5 卷至少有一个日志 `plex`。

执行常规备份以保护数据。如果卷的所有副本会以某种方式丢失或损坏，就需要备份。例如，电冲击会损坏系统上的数个（或全部）磁盘。另外，错误地键入命令会删除重要文件或者直接损坏文件系统。



介绍

本章提供关于 VERITAS Volume Manager 功能的详细信息。

本章讲述以下主题：

- ◆ 联机重新布局
- ◆ 热重定位
- ◆ 卷再同步
- ◆ 脏区日志
- ◆ 快速镜像再同步 (FMR)
- ◆ Volume Manager 根目录可置性
- ◆ 动态多路径 (DMP)
- ◆ VxSmartSync 恢复加速器
- ◆ Volume Manager 任务监视器
- ◆ Volume Manager 集群功能性



联机重新布局

联机重新布局允许在不中断数据存取的情况下，在 Volume Manager 支持的任何存储布局间转换存储布局。通常情况下，通过更改 Volume Manager 中的存储布局来更改存储的冗余或性能特点。Volume Manager 通过复制地址空间（镜像）或通过添加奇偶校验 (RAID-5)，将冗余添加到存储中。可以通过更改列数和条带宽度等条带化参数来更改 Volume Manager 中的存储性能特点。

可以将布局更改分为以下几种类型：

- ◆ RAID-5 到镜像
- ◆ 镜像到 RAID-5
- ◆ 添加或删除奇偶校验
- ◆ 添加或删除列
- ◆ 更改条带宽度

存储布局

联机重新布局当前支持下列存储布局：

- ◆ 连续
- ◆ 条带
- ◆ RAID-5
- ◆ 镜像
- ◆ 条带镜像
- ◆ 连续镜像

注意 当使用 VERITAS Volume Manager Storage Administrator 时，条带增强是用于条带镜像的 GUI 术语，而连续增强是用于连续镜像的 GUI 术语。

联机重新布局的工作机制

VERITAS 联机重新布局功能允许在不干扰数据存取的情况下，在原处更改已创建的存储布局。为适应更改的要求，可以更改具体某个布局的性能特点。可以通过调用单个命令将一种布局转换为另一种。

条带镜像 plex 是在镜像卷的基础之上的条带 plex，形成一个既是镜像又是条带的 plex。这种组合形成了一种称为条带镜像 plex 的 plex。可以使用同样的方法镜像连续 plex。联机重新布局支持条带镜像和连续镜像 plex 之间的转换。

注意 目前不支持在转换过程中更改镜像的数目。

例如，可能有一个条带布局的条带单元大小是 128K，而该大小不能提供最佳性能。可以使用重新布局功能更改此布局的条带单元大小。

只要安装在卷上的文件系统提供联机收缩和增长操作，实现这种转换时就不需要卸载文件系统。VFS 提供这些功能。

联机重新布局重用现有的存储空间，并具有满足新布局需要的空间分配策略。布局转换进程使用最少的临时空间，将特定的卷转换为目标布局。

从源布局到目标布局的转换在每次移动一部分数据的基础上完成。源卷中的数据被复制到临时空间。源卷存储区中的数据被分批删除。然后源卷存储区转换成新布局，保存在临时空间中的数据则写回新布局。此操作不断重复，直到源卷中的全部存储和数据都已转换成新布局。

可以使用联机重新布局更改列数、更改条带宽度、删除和添加奇偶校验，以及将 RAID-5 更改成镜像。

转换类型

要使联机重新布局操作生效，必须至少满足下列条件之一。必须执行下面的一项或多项操作：

- ◆ 将 RAID-5 更改为镜像
- ◆ 将镜像更改为 RAID-5
- ◆ 更改列数
- ◆ 更改条带宽度
- ◆ 删除或添加奇偶校验

要符合布局转换条件，镜像卷 plex 的布局必须相同，具有相同的条带宽度和列数。请参见表 1 “支持的布局转换”

表 1. 支持的布局转换

从 / 到	条带镜像	连续镜像	正规镜像	RAID-5	连续	条带
条带镜像	是 1	是 2	否 3	是 4	是 5	是 6
连续镜像	是 7	否 8	否 9	是 10	否 11	是 12
正规镜像	是 13	是 14	否 15	是 16	否 17	否 18
RAID-5	是 4	是 10	否 19	是 20	是 21	是 22



表 1. 支持的布局转换

从 / 到	条带镜像	连续镜像	正规镜像	RAID-5	连续	条带
连续	是 5	否 11	否 17	是 21	否 23	是 24
条带	是 6	是 12	否 18	是 22	是 24	是 25

表 1 中的项包括：

- ◆ 是表示联机重新布局操作是可能的。
- ◆ 否表示操作*也许*是可能的，但不能使用重新布局。
- ◆ 数字表示该特定布局转换中的可能更改的简要描述。参见下面的“[数字描述](#)”。
- ◆ 操作可以在两个方向上执行。

数字描述

表 1 中的数字描述重新布局操作的方式如下：

1. 更改条带宽度或列数。
2. 删除全部列。
3. 不是重新布局，而是转换操作。
4. 将镜像更改为 RAID-5 和 / 或条带宽度 / 列发生更改。
5. 将镜像更改为 RAID-5 和 / 或条带宽度 / 列发生更改。
6. 更改条带宽度 / 列并删除镜像。
7. 添加列。
8. 不是重新布局操作。
9. 转换操作。
10. 将镜像更改为 RAID-5。请参见 `vxconvert` 过程。
11. 删除镜像，不是重新布局操作。
12. 删除镜像并添加条带化。



13. 将旧的镜像卷更改为条带化镜像。只有当列 / 条带宽度发生更改时，才是有效的重新布局，否则是转换操作。请参见 `vxconvert` 过程。
14. 将旧的镜像卷更改为连续镜像。只有当列发生更改时，才是有效的重新布局，否则是转换操作。
15. 没有更改，不是重新布局操作。
16. 将旧的镜像卷更改为 RAID-5。在旧镜像卷中选择一个 plex 以使用重新布局。在重新布局操作结束时删除另一个 plex。
17. 只有选择了镜像卷中的一个 plex 并更改列 / 条带宽度，这才是重新布局操作。
18. 只有选择了镜像卷中的一个 plex 并更改列 / 条带宽度，这才是重新布局操作。
19. 不是重新布局操作。
20. 更改条带宽度 / 列。
21. 删除奇偶校验和全部列。
22. 删除奇偶校验。
23. 没有更改，不是重新布局操作。
24. 删除列。
25. 更改条带宽度 / 列数。

转换特性

从一种布局到另一种布局的数据转换涉及将现有布局中的数据重新组织成新布局。在转换过程中，联机重新布局通过镜像任何使用的临时空间来保持数据冗余。在转换过程中不中断对数据的读 / 写存取。

如果系统在转换过程中失败，则数据不会被破坏。系统恢复之后继续进行转换，并维护读 / 写存取。

可以随时反转布局转换进程，但是数据可能无法回到上一次的确切存储位置。在反转之前，应停止卷中的所有现有转换。

可以使用 `vxrelayout status` 命令确定转换的方向。

只要有足够的冗余和空间移动数据，这些转换就会消除 I/O 故障。



转换和卷长

一些布局转换可能会引起卷长增加或减小。如果布局转换引起卷长增加或减小，联机重新布局会使用 `vxresize` 收缩或增长文件系统。

联机重新布局不转换稀疏的 `plex`，也不会使 `plex` 变稀疏。

注意 联机重新布局只能在用 `vxassist` 命令或 Storage Administrator 创建的卷中使用。

不支持下列转换：

- ◆ 日志 `plex` 的转换
- ◆ 当联机重新布局操作在卷上运行时，此卷的快照

热重定位

热重定位允许系统对冗余 Volume Manager 对象（镜像或 RAID-5）的 I/O 故障自动做出反应，并恢复冗余和对那些对象的存取。Volume Manager 检测对象的 I/O 故障并重定位受影响的子磁盘。子磁盘被重定位到指定为备用磁盘的磁盘和 / 或磁盘组中的空闲空间。然后 Volume Manager 重建在故障发生前存在的对象，并使它们重新冗余和可存取。

当部分磁盘发生故障时（即只影响磁盘上某些子磁盘的故障），磁盘上发生故障的那部分的冗余数据被重定位。磁盘上未受影响的那部分现有卷仍然可存取。

注意 只在故障磁盘的冗余（镜像或 RAID-5）子磁盘上执行热重定位。故障磁盘上的非冗余子磁盘不被重定位，不过系统管理员会收到关于该故障的通知。

热重定位的工作机制

默认情况下启用热重定位功能。当发生故障时，不需要系统管理员的操作即可启动热重定位。

热重定位后台驻留程序 `vxrelocd` 在 Volume Manager 中监视影响冗余的事件并执行热重定位以恢复冗余。`vxrelocd` 也（通过电子邮件）通知系统管理员任何故障和重定位及恢复操作。有关 `vxrelocd` 的更多信息，请参见 `vxrelocd(1M)` 帮助手册。

`vxrelocd` 后台驻留程序在系统启动时启动，监视 Volume Manager 中涉及磁盘、`plex` 或 RAID-5 子磁盘的故障。它在故障发生时触发热重定位功能。

成功的热重定位进程涉及：

1. 检测由磁盘、`plex` 或 RAID-5 子磁盘故障导致的 Volume Manager 事件。
2. 通知系统管理员（和其它指定的用户）有关故障，并标识受影响的 Volume Manager 对象。这项工作通过电子邮件完成。

3. 确定哪些子磁盘可以重定位，在磁盘组中为这些子磁盘查找空间，并重定位这些子磁盘。通知系统管理员这些操作及其成功与否。
4. 启动恢复卷和数据所需的任何恢复过程。通知系统管理员恢复尝试的结果。

注意 热重定位不保证重定位后的数据布局或性能与重定位之前相同。在热重定位后，系统管理员可能更改一些配置。

如何选择重定位的空间

备用磁盘在可以用于替换之前，必须初始化并作为备件放在磁盘组中。如果在故障发生时没有任何已指定为备件的磁盘，Volume Manager 会自动使用发生故障的磁盘组中的任何可用空闲空间。如果没有足够的备用磁盘空间，则使用备用空间和空闲空间的组合。

热重定位中提到的空闲空间是始终可以用于热重定位的空闲空间。使用下列 Storage Administrator 接口可以使磁盘不能用于热重定位：`vxdiskadm` 或 `vxedit`。

在每个磁盘组中，系统管理员可以指定一个或多个磁盘作为热重定位备件。可以使用 Storage Administrator 接口 `vxdiskadm` 或 `vxedit` 将磁盘指定为备件。被指定为备件的磁盘不参与空闲空间模型，也不应有分配给它的存储空间。

在选择重定位空间时，热重定位保留重定位的子磁盘所属的 Volume Manager 对象的冗余特性。例如，热重定位确保不会将故障 plex 中的子磁盘重定位到包含故障 plex 的镜像的磁盘上。如果使用任何可用的备用磁盘或空闲空间都不能保留冗余，则不会发生热重定位。如果不能重定位，将通知系统管理员并且不采取进一步的操作。

为确定在合格的磁盘中具体使用哪一个，热重定位尝试使用距故障磁盘“最近”的磁盘。“靠近”的值取决于控制器、目标和故障磁盘的磁盘号。与故障磁盘在同一个控制器上的磁盘比在不同的控制器上的磁盘近，与故障磁盘在同一目标的磁盘比在不同目标的磁盘近。

如果有可能，热重定位会尝试将故障驱动器上的所有子磁盘移到同一个目标磁盘上。

如果故障磁盘是根磁盘，则只有在能将所有文件系统重定位到同一磁盘上时，热重定位才有效。如果没有这样的磁盘，则会通过电子邮件通知系统管理员。

当发生热重定位时，故障子磁盘从配置数据库中被删除，Volume Manager 确保不将故障子磁盘所使用的磁盘空间作为空闲空间回收。

有关如何禁止热重定位的信息，请参阅《*VERITAS Volume Manager 安装指南*》。



解除重定位实用程序

VxVM 热重定位允许系统对子磁盘级的冗余 VxVM 对象上的 I/O 故障自动做出反应，并采取必要的操作使该对象再次可用。此机制检测子磁盘中的 I/O 故障，定位子磁盘，并恢复与子磁盘关联的 plex。更换了故障磁盘后，Volume Manager 提供 vxunreloc 实用程序，此实用程序可用于将系统恢复到与磁盘故障发生前相同的配置。vxunreloc 允许将热重定位的子磁盘移回因磁盘故障而被更换的磁盘上。

卷再同步

当使用镜像卷或 RAID-5 卷冗余存储数据时，Volume Manager 确保所有数据拷贝完全相同。但是，在某些情况下（通常是由于整个系统发生故障），卷中的一些冗余数据可能变得不一致或不同步。镜像数据与原始数据不完全相同。除正常配置更改外（如分离和重新挂接 plex），只有当正在向卷写入数据时系统崩溃，才会发生这种情况。

数据并行写入卷的镜像中，与 RAID-5 卷的数据和奇偶校验一样。如果系统在全部单个写入完成之间崩溃，则可能一些写入完成，而另一些没完成。这会导致数据变得不同步。对于镜像卷，如果使用不同的镜像满足读请求，数据不同步会导致两次读取卷的同一区域却返回不同的结果。对于 RAID-5 卷，则会导致奇偶校验破坏和不正确的数据重建。

Volume Manager 需要确保所有镜像都包含完全相同的数据，且 RAID-5 卷中的数据 and 奇偶校验一致。此进程称为卷再同步。如果卷属于引导时自动导入的磁盘组（如 rootdg），则该卷的再同步进程将在系统重新引导时发生。

并不是所有的卷在系统发生故障后都需要再同步。从未被写入的卷或在系统发生故障时静止的卷（即没有活动的 I/O）不可能有未完成的写操作，因此不需要再同步。

当卷第一次被写入时，Volume Manager 会记录此操作并将该卷标记为脏。当卷被所有进程关闭或由系统管理员清洁停止时，所有的写都已完成，Volume Manager 会删除该卷的“dirty”标志。只有在系统重新引导时，标记为“dirty”的卷才需要再同步。

再同步的进程取决于卷的类型。包含 RAID-5 日志的 RAID-5 卷可以“重播”那些日志。如果没有可用的日志，则将卷置于重建 - 恢复模式并且重新生成所有的奇偶校验。对于镜像卷，通过将卷置于恢复模式（也称为读-回写恢复模式）来完成再同步。卷数据的再同步在后台完成。这使得卷在恢复过程中也可用。

再同步进程的花费会很昂贵，并且会对系统性能造成不利影响。恢复进程通过分散恢复避免对某个特定的磁盘或控制器造成压力，从而部分减少了上述不利影响。

对于大卷或大量的卷，再同步进程可能要花费一些时间。为消除这种影响，可以对镜像卷使用脏区日志，或确保 RAID-5 卷具有有效的 RAID-5 日志。对由数据库应用程序使用的卷，可以使用 VxSmartSync™ 恢复加速器（参见第 53 页的“VxSmartSync 恢复加速器”）。



脏区日志

脏区日志 (DRL) 是卷的可选特性，用于在系统发生故障后快速恢复镜像卷。DRL 跟踪由于 I/O 写入镜像卷而发生更改的区域。使用此信息，DRL 只恢复卷中需要恢复的部分。

如果没有使用 DRL，则在发生系统故障时，必须将卷的所有镜像都恢复成一致的状态。恢复时需卷的镜像之间复制卷的全部内容。此进程可能要花很长时间并且占用大量 I/O。卷中已经一致的区域可能也需要恢复。

DRL 在逻辑上将卷分成一套连续区域。它跟踪被写入的卷区域。维护脏区日志，用来包含代表每个卷区域的状态位。对于任何写入卷的操作，写入数据前在日志中将被写入的区域标记为“dirty”。如果写操作导致以前是干净的日志区域变脏，则日志同步写入磁盘后才能发生写操作。在系统重新引导时，Volume Manager 只恢复那些在脏区日志中标记为“dirty”的卷区域。

日志子磁盘 用于存储启用 DRL 的卷的脏区日志。有 DRL 的卷至少有一个日志子磁盘，可以使用多个日志子磁盘镜像脏区日志。每个日志子磁盘与卷的一个 plex 关联。每个 plex 只能有一个日志子磁盘。如果 plex 只包含一个日志子磁盘而且不包含数据子磁盘，则此 plex 可以称为 *日志 plex*。

日志子磁盘也可以与包含数据子磁盘的正规 plex 关联。那种情况下，如果由于其中某个数据子磁盘发生故障而必须分离 plex，则日志子磁盘就有变为不可用的危险。

如果使用 vxassist 命令创建脏区日志，默认情况下，该命令创建包含单个日志子磁盘的日志 plex。也可以手工创建脏区日志，方法是创建日志子磁盘并将其与一个 plex 关联。于是，这个 plex 可以同时包含日志子磁盘和数据子磁盘。

任何时候都只能将日志中有限个数的位标记为“dirty”。在数据写入一个区域后并不立即清除此区域的脏位。相反，它将保持为脏，直到相应的卷区域变为最近最少使用的区域。如果特定区域的位已被标记为脏，当再次向该区域写入时，不必将日志写入磁盘，写操作即可发生。

某些卷（如用于 Oracle 重播日志的卷）是按顺序写入的，不会从 DRL 位的惰性清除中获益。对于这些卷，可使用 *有序 DRL* 进一步限制脏位的数目并加快恢复速度。有序 DRL 允许的脏位数由可调的 voldr1_max_dirty *限制*。在按顺序写入的卷上使用有序 DRL 可能会严重影响 I/O 的吞吐量。

注意 对于大多数写访问模式，DRL 添加了少量的 I/O 开销。



快速镜像再同步 (FMR)

快速镜像再同步 (FMR) 功能通过提高 VxVM 快照机制的效率，更好地支持备份和决策支持等操作，从而快速有效地执行无效镜像的再同步。通常，这些操作要求数据存储卷是静止的，并且 / 或者对存储的辅助存取不影响或妨碍主存取（如吞吐量、更新、一致性等）。

为达到这些目标，VxVM 提供了 *快照* 机制，用以创建主卷在具体某个时点的精确副本。创建了快照后，可以独立于创建快照的卷来存取快照。在共享 / 集群 VxVM 环境中，通过从不同的机器上访问快照，可以消除资源争用并消除使用快照的开销。

快速镜像再同步克服了存在于快照机制早期版本中的某些缺点。这些缺点是：

- ◆ 创建了快照后，主卷与快照可能有偏差。它们不再一致。结果，当快照不再有用时必须废弃它，并且必须创建新快照以便获得主数据的当前副本。
- ◆ 创建快照所需的时间会限制快照功能的有效性。这是因为创建快照所需的时间与卷的大小成正比。对于大的企业类卷，这段时间可能要求采取脱机策略，而这在某些情况下是无法接受的。

FMR 组件

FMR 为 VxVM 提供两种基本增强功能。第一种是优化镜像再同步进程（*快速镜像再同步*），第二种是扩展快照模型（*快速镜像重连接*），它提供一种方法，使快照能被刷新和重用，而不是被废弃。

快速镜像再同步组件

FMR 要求跟踪更新应用时因不可用的镜像所遗漏的数据存储更新。当镜像重新开始提供服务时，它可以只重新应用被那个镜像遗漏的更新。有了 FMR，此进程所需的恢复量远远少于当前使用的将全部数据存储复制到返回镜像的方法。

不可用的镜像是已经自动地（因错误而由 VxVM）或直接由系统管理员（通过 VxVM 实用程序，如 `vxplex` 或 `vxassist`）同它所在的卷分离的镜像。*返回镜像*是先前分离，现在正在由 `vxrecover` 或 `vxplex att` 操作重新挂接到其原来所在的卷上的镜像。

FMR 不会更改当前镜像故障和修复管理模型。唯一的可视效果是典型镜像修复操作的完成速度加快。

再同步增强功能允许管理员基于每个卷启用 / 禁用 FMR 并检查 FMR 状态。

快速镜像重连接组件

快速镜像重连接 扩展现有的快照使用模型。若不使用快速镜像再同步，通过快照机制创建的卷副本是独立的。原始卷与副本卷彼此完全独立，它们的数据可能有偏差。

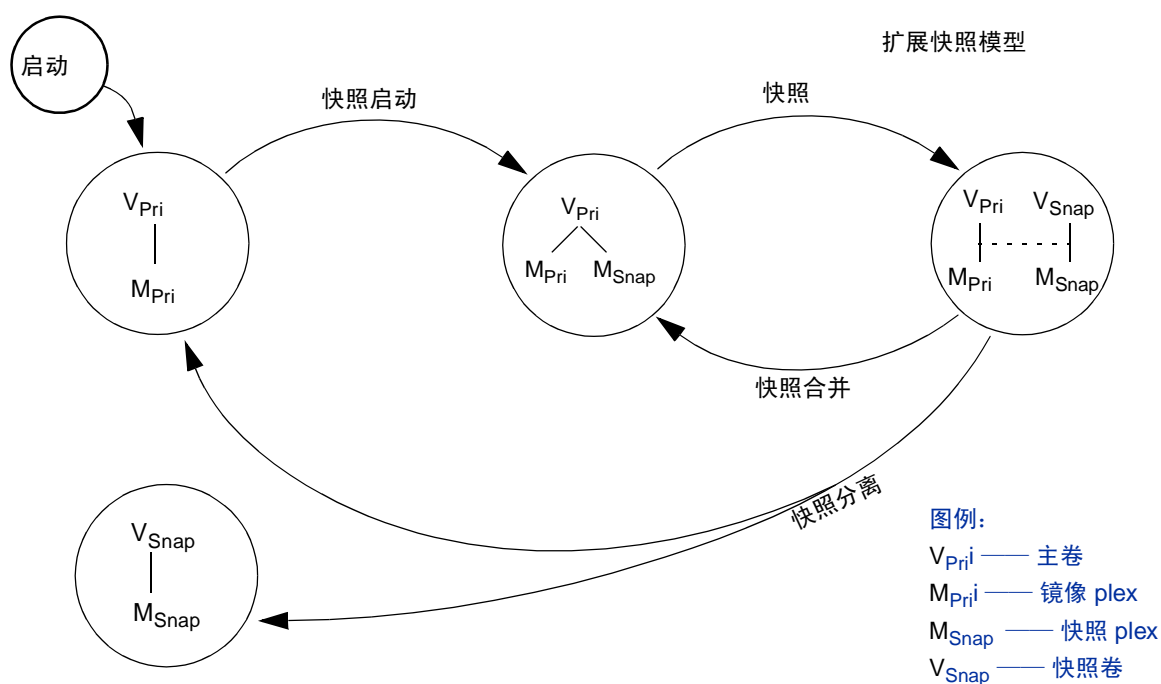
快速镜像重连接 快照增强功能可以将快照卷和它的原始卷重新关联，从而达到减少执行循环操作所需的工作负荷的特殊目的，因为循环操作在很大程度上依赖 VxVM 快照功能性。

FMR 对 VxVM 快照功能性的增强

3.1 发行版的 FMR 快照功能增强扩展了页 47 上的图 23 “FMR 增强快照”中所示的快照模型。从 3.1 发行版开始，`snapshot` 命令创建原始卷与快照卷之间的关联，其它性能与以前一样。新命令 `vxassist snapback` 利用这种关联，方便地将快照 plex (M_{Snap}) 返回到创建快照的卷中（在本例中是 V_{Pri} ）。

图 23 “FMR 增强快照”描述了 `snapback` 和 `snapclear` 命令所引入的扩展快照模型转变。

图 23. FMR 增强快照



另外，新命令 `vxassist snapclear` 通过永久性地销毁 `snapshot` 命令创建的关联，缓解了卷跟踪快照的管理开销。此功能在知道快照永远不会返回创建它的卷的情况中很有用。

操作原理

FMR 的基础在于更改跟踪。跟踪在镜像脱机 / 分离 / 创建快照时遗漏的更新，然后在镜像返回时只应用那些更新，大大减少了再同步卷所需的时间。

此更改跟踪的基础是使用位图。位图中的每个位代表卷地址空间的一个相邻区域（一个范围）。此相邻区域称为区域大小。通常情况下，此区域大小是一个块；卷的每个块由位图的一个位表示。但是，提供了称为 `vol_fmr_logsz` 的可调值，可以用它来限制 FMR 位图的最大大小（以块为单位）。当计算位图的大小时，算法从区域大小 1 开始，如果结果位图大小比 `vol_fmr_logsz` 小，则计算的值成为位图的大小。如果大小比 `vol_fmr_logsz` 大，则会尝试为 `vol_fmr_logsz` 提供区域大小 2，如此下去，直到位图的大小比 `vol_fmr_logsz` 可调值小。

例如：

卷大小 = 1G

`vol_fmr_logsz = 4`

在块的大小是 512 字节（即 $4 \times 512 = 2048$ 字节或 $2048 \times 8 = 16384$ 位）的系统上。

因此，对于 1G 的卷，其区域大小是 2048 位，比四个块小，所以位图的大小是 2048 位或 256 字节。

注意，如果卷的大小增加，将重新计算以确保位图的大小不超过 `vol_fmr_logsz` 可调值。

持久性跟踪与非持久性跟踪

对于 VxVM 3.1，在内存中分配 FMR 位图。与 DRL 不同，它们不驻留在磁盘或永久性存储中。因此，如果系统崩溃，此信息将丢失，必须同步整个卷长。这种方法的优点是 FMR 更新（对此位图的更新）不会造成任何性能方面的损失，因为不必进行磁盘更新。但是，如果系统崩溃，此信息将丢失，必须重新完全再同步镜像。

快照与 FMR

若要在使用快照时利用 FMR 增量跟踪，请使用新的快照选项。创建快照后，快照选项将用于重新挂接快照 `plex`。如果在创建快照前启用了 FMR，而且在快照完成前的任何时候都不禁用它，则 FMR 位图中反映的 FMR 增量更改将在快照合并过程中用于再同步卷。为了使同时创建多个卷的快照变得更容易，快照选项已得到增强，可以接受多个卷而且添加了命名方案。默认情况下，每个副本都称为 `SNAP-<original volume>`。可以使用命令行上的选项覆盖该默认值。只有同一磁盘组中的卷可以一次创建多个卷的快照。

为了使创建单个磁盘组中所有卷的快照变得更容易，已向 `vxassist` 中添加了新的选项，但如果磁盘组中有任何卷没有完整的快照 `plex`，则此选项无效。可以创建同一个卷的多个快照。为每个创建的快照都生成了一个新的 FMR 位图，从而使每个快照的再同步时间减少到最少。

可以将快照 `plex` 选作执行快照合并时的一套首选数据。向快照合并选项添加 `-o resyncfromreplica` 后，快照（副本）`plex` 上的数据将复制到挂载在原始卷上的所有镜像。默认情况下，原始卷上的数据为首选，将被复制到快照 `plex`。

可以增长副本卷或原始卷，同时仍使用 FMR。增长卷将扩展位图 FMR 的使用，以跟踪增量更改。这可能更改位图的大小或其区域大小。不管哪种情况，位图中对应于卷的增长区域的那部分将被标记为 “dirty”，这样该区域将会再同步。如果快照试图创建不完整的快照 `plex`，则快照操作将失败。在这种情况下，在运行快照合并选项前，必须增长副本卷或原始卷。单独增长两个卷将导致快照与卷中的其它镜像共享物理磁盘。为了防止发生这种情况，在完成 `snappoint` 命令后增长卷。

任何更改副本卷布局的操作都会将快照的 FMR 位图标记为 “dirty”，并且需要在快照合并时进行完全再同步。这样的操作包含子磁盘分割、子磁盘移动和副本的联机重新布局。完成快照后执行这些操作很安全。有关更多信息，请参见 `vxvol (1M)`、`vxassist (1M)` 和 `vxplex (1M)` 帮助手册。

FMR 与可写快照

这两个选项中的其中一个用于跟踪可写快照的更改，如下所示：

- ◆ 创建单独的位图跟踪快照卷的更改
- ◆ 更改快照卷的父卷位图。只有当快照卷有很少的更改时才使用此快捷方法，如在备份和 DSS（决策支持系统）应用程序中

VxVM 3.1 中实现的是后一种方法；即，快照卷的父卷位图在写入快照时更新。

说明与限制

RAID-5 卷不支持 FMR。

重定位子磁盘时，整个 `plex` 被标记为 “dirty”，而且必须完全再同步。

Volume Manager 根目录可置性

Volume Manager 可以将来自不同系统（例如，根文件系统、交换设备、`usr` 文件系统和 `stand` 文件系统）的各种文件置于 Volume Manager 控制之下。这称为 *根目录可置性*。可以通过 *封装* 进程将 *根磁盘*（即含根文件系统的磁盘）置于 Volume Manager 控制之下。

封装将该磁盘上的现有分区转换成卷。一旦置于 Volume Manager 控制之下，根和交换设备便显示为卷，并提供与其它 Volume Manager 卷相同的特性。配置为交换区的卷称为 *交换卷*；含根文件系统的卷称为 *根卷*；含 `stand` 文件系统的卷称为 *stand 卷*。

可以镜像 `rootvol`、`swapvol` 和 `standvol` 卷，以及根磁盘上成功引导系统所需的其它部分（例如，`/usr`）。这提供了磁盘发生故障时的完全冗余和恢复能力。若没有 Volume Manager 的根目录可置性，一旦根、交换、`usr` 根或 `stand` 分区丢失，系统将无法从未受故障影响的磁盘引导。



对启动至关重要的磁盘驱动器镜像以便确保单个磁盘故障不会导致系统不能使用。建议的配置是将至关重要的磁盘镜像到其它可用的磁盘上（使用 `vxdiskadm` 命令）。如果含根、stand 和交换分区的磁盘发生故障，系统将从含根镜像的磁盘重新引导。有关镜像引导（根）磁盘和系统恢复过程的更多信息，请参见第 7 章“恢复”。

使用根卷引导

通常情况下，当启用操作系统时，根文件系统、stand 文件系统和交换区在启动过程的开始阶段使用。这发生在可运行用户进程以装入 Volume Manager 配置并启动卷之前。必须在启动 Volume Manager 前完成根、stand 和交换设备的配置。若将 Volume Manager `vxconfigd` 后台驻留程序作为 `init` 进程的一部分启动，则来不及将卷配置成根或交换设备。

为了避免此限制，启动过程中系统将存取 `rootvol`、`standvol` 和 `swapvol` 卷的镜像。在启动过程中，系统将 `rootvol`、`standvol` 和 `swapvol` 卷视作正规分区并使用标准分区编号存取它们。`rootvol`、`standvol` 和 `swapvol` 卷从相邻的磁盘空间中创建，而这些磁盘空间还由每个卷的单个分区映射。由于此限制，不可能条带化或跨接 `rootvol`、`standvol` 或 `swapvol` 卷的主 plex（即用于引导的 plex）。引导所需的这些卷的任何镜像都不能条带化或跨接。

启动时卷的限制

`rootvol`、`standvol`、`swapvol` 和 `usr` 卷与其它卷不同，因为它们在卷配置上有很具体的限制：

- ◆ 根卷 (`rootvol`) 必须存在于默认磁盘组 `rootdg` 中。虽然可以在 `rootdg` 以外的磁盘组中创建名为 `rootvol` 的其它卷，但是只有 `rootdg` 中的 `rootvol` 可用于引导系统。
- ◆ `rootvol` 卷有特定的次要设备编号：次要设备 0。另外，`swapvol` 有次要设备编号 1。`usr` 卷没有特定的次要设备编号。参见第 7 章“恢复”。
- ◆ `rootvol`、`var`、`usrrootvol`、`standvol` 和 `swapvol` 设备的有限个镜像为它们创建了“覆盖”分区。覆盖分区是完全包含由有限个镜像占用的磁盘空间的分区。在引导过程中，在充分配置 `rootvol`、`var`、`usrrootvol`、`standvol` 和 `swapvol` 卷之前，默认卷配置使用覆盖分区存取磁盘上的数据。
- ◆ 虽然可以出于性能考虑将条带镜像添加到 `rootvol` 设备，但是如果主 plex 发生故障，则无法对系统恢复或引导所需的 `rootvol` 的主 plex 或任何镜像进行条带化。
- ◆ `rootvol`、`standvol` 和 `swapvol` 都不能跨接，也不能包含具有多个非邻接子磁盘的主 plex。不能增长或收缩与封装的引导磁盘（`rootvol`、`usr`、`var`、`opt`、`swapvol`，等等）关联的任何卷，因为这些卷映射到磁盘的物理基础分区上，且必须是邻接的。解决方法是解除引导磁盘的封装，根据需要对引导磁盘进行重新分区（根据需要增长或收缩分区），然后重新封装。

- ◆ 当镜像引导磁盘的某些部分时，镜像到的磁盘必须足够大，可以容纳原始 plex 上的数据，否则镜像将不起作用。
- ◆ rootvol、standvol、swapvol 和 usr 不能是脏区日志卷。

除了这些要求外，使根、usr、var、opt、varadm、usrkvm 和交换的每个卷都至少有一个邻接的（如果适当可以柱面对齐）镜像。这使将这些卷从卷转换回正规磁盘分区变得更容易（如在操作系统升级过程中）。

动态多路径 (DMP)

在某些系统上，Volume Manager 支持多路径的磁盘阵列。它自动重新组织到磁盘阵列内具体某个磁盘设备的多个 I/O 路径。Volume Manager 的动态多路径功能通过提供路径故障切换机制，提供更大的可靠性。一旦到磁盘的某个连接丢失，系统将通过到该磁盘的其它可靠的连接继续存取重要的数据。DMP 还通过将 I/O 负载平均分布在多条到磁盘设备的 I/O 路径上，提供更大的 I/O 吞吐量。

在 Volume Manager 中，所有与系统相连的物理磁盘都表示成具有一条或多条物理存取路径的元设备。单个与系统相连的物理磁盘由具有一条路径的元设备表示。作为磁盘阵列一部分的磁盘由具有两个物理存取路径的元设备表示。可以使用象 vxdisk 这样的 Volume Manager 管理实用程序来显示元设备的所有路径以及各种路径的状态信息。

当可以通过多个操作系统设备句柄存取某个物理磁盘时，便存在多路径条件。每个多路径操作系统设备句柄都允许通过替换的主机到设备路径存取和控制数据。

可以配置 Volume Manager 用自己的 DMP 系统组织对多路径设备的存取。Volume Manager 通过使用通用全球设备标识符 (WWID ID) 检测多路径系统。物理磁盘必须通过 WWID ID 提供明确的标识以便由 DMP 存取该设备。

如果 DMP 无法通过 WWID ID 识别物理磁盘，则将通过 Volume Manager 设备检测方法找到标识。设备检测依赖于 Volume Manager 重新组织磁盘上的元数据标识符。

Volume Manager DMP 为每个检测到的多路径目标创建代表元设备的元节点。每个元节点都映射到一套操作系统设备句柄，并使用适当的多路径策略进行配置。Volume Manager DMP 为可通过操作系统设备句柄存取的所有挂接的物理磁盘创建元节点。

Volume Manager DMP 管理多路径目标（如磁盘阵列），多路径目标定义使用多条路径的策略。某些磁盘阵列允许多条路径同时保持活动状态（活动 / 活动）。某些磁盘阵列只允许一条路径保持活动状态，将替换路径用作备件以防现有路径发生故障（主动 / 被动）。某些磁盘阵列具有设计更精细的策略。

一般情况下，Volume Manager 被设计成将 VM 磁盘映射到一个 Volume Manager DMP 元节点。为简化 VxVM 逻辑操作，每个 VM 磁盘都映射到唯一一个 Volume Manager DMP 元节点。不管在多路径配置中是否连接了物理磁盘设备，此映射都将发生。



路径故障切换机制

DMP 与多端口磁盘阵列一起使用时可增强系统可靠性。一旦到磁盘阵列的某个连接丢失，DMP 将自动为 I/O 请求动态选择下一个 I/O 路径，而无需系统管理员的任何操作。

DMP 允许系统管理员向 Volume Manager 中的 DMP 子系统指出是修复还是恢复连接。这称为 DMP 重新配置。重新配置过程还允许检测最新添加的设备，以及完全引导系统后删除的设备（只有当操作系统完全看到它们时）。

负载平衡

为了提供路径间的负载平衡，DMP 遵循活动 / 活动磁盘阵列的平衡路径机制。负载平衡通过使用所有路径的最大带宽，确保最大限度地提高 I/O 吞吐量。不过，到磁盘的有序 I/O 将沿同一条路径发送以优化 I/O 吞吐量。这样做是为了使用磁盘跟踪高速缓冲存储器的效果。

对于主动 / 被动磁盘阵列，I/O 将沿主路径发送，直到主路径失败。一旦主路径失败，I/O 将切换到其它可用的主路径或辅助路径。为了避免 LUN 的拥有权不断地从一个控制器转让给另一个控制器（这将导致 I/O 严重减慢速度），对于主动 / 被动磁盘阵列将不执行路径间的负载平衡。

从 DMP 设备启用

当根磁盘受 Volume Manager 的控制时，如果它是单个磁盘，则将作为 DMP 设备自动存取，如果磁盘是多端口磁盘阵列的一部分，则有多条路径。通过对根磁盘进行封装，将增强系统防备到磁盘的一条或多条现有物理路径丢失的可靠性。

启用和禁用控制器

DMP 允许系统管理员关闭到主机 I/O 控制器的 I/O，以便执行管理操作。它可用于维护挂接在主机上的控制器或由 Volume Manager 支持的磁盘阵列。完成维护任务后，可以启用到主机 I/O 控制器的 I/O 操作。可以使用由 Volume Manager 提供的 `vxddmpadm` 命令完成此操作。

例如，如果系统有 StorEdge A5000™ 阵列，当用户需要更改与此磁盘阵列相连的 A5000 接口板时，应使用 `vxddmpadm` 命令获得连接在此 A5000 接口板上的主机 I/O 控制器列表并应禁用这些控制器。这些控制器一旦被禁用，通过这些控制器对磁盘的进一步 I/O 存取将停止。

然后，用户就可可在不中断对该磁盘阵列中的现有磁盘进行 I/O 存取的情况下来更换接口板。这样要求是因为在正常情况下，对于活动 / 活动类型的磁盘阵列（与本例中的一样），Volume Manager 都使用平衡路径机制来调度具有多条路径的磁盘的 I/O，从而使 I/O 可以在任意时刻经过任何路径。

对于主动 / 被动类型的磁盘阵列，I/O 由 Volume Manager 调度到主路径，直到主路径发生故障。因此，若要更改磁盘阵列上的接口卡或更改与磁盘阵列相连的主机（可能发生）上的卡，则应禁止到主机 I/O 控制器的 I/O 操作。这允许在更改硬件前，使所有的 I/O 都转移到另一个 I/O 控制器上的主动辅助路径或主动主路径。

此操作结束后，可使用 `vxdmpadm` 命令的启用选项，使这些控制器的路径重新发挥作用。

Volume Manager 不允许禁用到根磁盘的最后一条活动路径。

显示 DMP 数据库信息

`vxdmpadm` 命令可用于列出 DMP 数据库信息并执行其它管理任务。此命令使您得以列出系统上的所有控制器（与磁盘相连）和其它存储在 DMP 数据库中的相关信息。该信息可用于定位系统硬件并决定要启用 / 禁用的控制器。

`vxdmpadm` 还提供其它有用的信息，如磁盘阵列序列号和与磁盘阵列相连的 DMP 设备（磁盘）列表，具体某个控制器的路径列表，等等。

VxSmartSync 恢复加速器

VxSmartSync™ 恢复加速器可用于某些系统。用于镜像 Oracle 数据库的 VxSmartSync 是一个功能集，能加快用于 Oracle 通用数据库的卷的再同步进程（称为 *resilvering*）。这些功能使用 Volume Manager 卷和数据库软件之间的扩展接口，可以避免镜像再同步过程中不必要的工作。这些扩展接口使卷恢复所用的时间有不同程度的改善。

Oracle 在 SmartSync 可用时自动利用它。

系统管理员必须正确配置卷使用 VxSmartSync。对于 Volume Manager，有两种类型的卷用于数据库：

- ◆ *重做日志卷* 包含数据库的重做日志。
- ◆ *数据卷* 是所有其它用于数据库的卷（控制文件和表空间文件）。

VxSmartSync 处理这两类卷的方式不同，必须正确配置它们以充分利用扩展接口。这两类卷之间的唯一差别是重做日志卷有脏区日志，而数据卷没有。

数据卷配置

通过让数据库软件确定卷的哪些部分需要恢复，使数据卷的恢复时间得到改善。数据库保留数据库数据的更改记录，并确定卷中的哪些部分需要恢复。通过减少需要恢复的空间量并允许数据库控制恢复进程，可减少总的恢复时间。

另外，恢复发生在数据库软件启动时，而不是发生在系统启动时。这减少了系统重新启动时恢复所造成的总体负面影响。因为恢复由数据库控制，所以卷的恢复时间就是数据库的 *resilvering* 时间（即，重播重做日志所需的时间）。



因为数据库保留了自己的日志，所以 Volume Manager 不必做日志。这样应将数据卷配置成不带脏区日志的镜像卷。除了改善恢复时间外，这样做还避免了由 DRL 导致的任何运行时 I/O 开销，从而改善正常的数据库写访问。

重做日志卷配置

重做日志 是对数据库数据更改的日志记录。数据库不保留重做日志更改的任何日志记录，所以数据库自身不能提供有关哪些部分要求 **resilvering** 的信息。重做日志也按顺序写入，并且由于传统的脏区日志在随机写入的数据中用得最多，因此它们最少用于减少重做日志的恢复时间。不过，Volume Manager 可通过将其脏区日志功能的行为修改成利用有序存取模式来减少脏区的数目。这将减少需要恢复的数据量并减小恢复时间对系统的负面影响。

重做日志的增强接口使数据库软件得以通知 Volume Manager 何时卷可用作重做日志。这使 Volume Manager 得以将卷的 DRL 行为修改成利用存取模式。由于改善后的恢复时间依赖于脏区日志，因此应将重做日志卷配置为具有脏区日志的镜像卷。

Volume Manager 任务监视器

Volume Manager 任务监视器通过监视任务的创建、维护和完成，跟踪系统恢复的进度。任务监视器使您得以监视任务进度并修改任务的特性，如暂停和恢复率（例如，减少对系统性能的负面影响）。还可以监视和修改联机重新布局功能的进度。有关更多信息，请参见“[联机重新布局](#)”。

Volume Manager 集群功能性

Volume Manager 包含 *可选的* 集群功能，使 VxVM 可用于集群环境中。Volume Manager 中的集群功能性允许多个主机同时存取和管理受 Volume Manager 控制的特定磁盘集（VM 磁盘）。

集群 是一组共享磁盘集的主机，在集群中每个主机都称为 *节点*。节点跨整个网络连接在一起。如果某个节点失败，其它节点仍然可以存取磁盘。Volume Manager 集群功能提供所有节点上的磁盘配置（包括更改）的相同逻辑视图。当启用集群功能时，Volume Manager 对象可由集群中的所有节点共享。

有关 Volume Manager 集群功能性的更多信息，请参阅本手册中有关集群功能性的那一章。

注意 Volume Manager 中的集群功能性是需单独获得许可证的功能。

磁盘任务

介绍

本章描述 Volume Manager 的磁盘管理操作。也提供有关磁盘组操作的信息。

注意 大多数 Volume Manager 命令要求超级用户特权或其它适当的特权。

本章讲述以下主题：

- ◆ 标准磁盘设备
- ◆ 磁盘组
- ◆ 磁盘和磁盘组命令
- ◆ 初始化和添加磁盘
- ◆ 将磁盘添加到 Volume Manager
- ◆ 向磁盘组添加磁盘
- ◆ 从磁盘组中删除磁盘
- ◆ 移动磁盘
- ◆ 重命名磁盘
- ◆ 保留磁盘
- ◆ 使磁盘脱机
- ◆ 镜像磁盘
- ◆ 删除磁盘
- ◆ 显示磁盘信息
- ◆ 检测和更换故障磁盘
- ◆ 创建磁盘组



- ◆ 升级磁盘组
- ◆ 删除磁盘组
- ◆ 在系统间移动磁盘组
- ◆ 破坏磁盘组
- ◆ 使用特殊设备
- ◆ vxdiskadm 菜单界面任务

标准磁盘设备

有两类可与 Volume Manager 一起使用的磁盘设备：标准设备和特殊设备。本章稍后将描述特殊设备。

Volume Manager 至多支持物理磁盘上的 8 个分区（也称为片）。这些分区按顺序进行编号（从 0 到 7），其中的分区 2 被保留用于表示整个磁盘。

注意 在某些系统上，Volume Manager 支持多达 16 个分区。在这些系统上，分区从 0 到 15 进行编号，其中的分区 0 被保留用于表示整个磁盘。

当某分区置于 Volume Manager 的控制之下时，将给此分区分配一个 VM 磁盘。可以使用符号名称（*磁盘名称* 或 *磁盘介质名称*），如 disk0 来表示 VM 磁盘。

注意 您的系统使用的 *设备名称* 可能不同于这些实例。

分区的寻址是通过 `c#b#t#d#s#` 形式的物理地址（通常称为 *设备名称* 或 *磁盘存取名称*）来实现的，此形式含有下列元素：

- ◆ `c#` —— 磁盘驱动器所挂接的控制器编号
- ◆ `b#` —— 相应的总线（如果系统使用总线）
- ◆ `t#` 和 `d#` —— 构成控制器上磁盘驱动器地址的目标 ID 和设备编号
- ◆ `s#` —— 磁盘驱动器上的分区号

设备名称的一个实例是 `c0t0d0s2`。按照惯例，`s2` 指 Volume Manager 所使用的标准分区方法。在某些系统中，Volume Manager 使用 `s0` 作为标准分区方法。Volume Manager 将物理磁盘标识为 `c#b#t#d#s#`（`b#` 用于使用总线的系统）。

虽然显示命令报告带 `s#` 后缀的设备，但后缀 `s#` 对于很多命令是可选的。Volume Manager 的 `vxdiskadm` 和 `vxdiskadd` 实用程序采用的设备名称不带 `s2`（或 `s0`）后缀。例如，若要将挂接在第一个控制器上的第二个磁盘指定给 `vxdiskadd`，请使用名称 `c0t1d0`。

在 Volume Manager 中通常使用设备名称 `c0t0d0` 标识引导磁盘（此磁盘包含根文件系统并用于引导系统）。

一个 VM 磁盘有两个区域：

- ◆ **专用区域** —— 存储配置信息的小区域。该区域存储磁盘标号和配置记录。
- ◆ **公共区域** —— 包括磁盘的其余部分并用于存储子磁盘（和分配存储空间）的区域。

Volume Manager 使用 3 种基本磁盘类型：

- ◆ *sliced* —— 公共区域和专用区域在不同的磁盘分区中。
- ◆ *simple* —— 公共区域和专用区域在同一磁盘分区中（公共区域接在专用区域之后）。
- ◆ *nopriv* —— 没有专用区域（只有用于分配子磁盘的公共区域）。

Volume Manager 用尽可能少的分区初始化每个新磁盘（通常每个物理磁盘有 2 个分区）。对于以 `s2`（或 `s0`）结尾的磁盘存取名称，默认类型是 `sliced`。

磁盘组

Volume Manager 将磁盘组织成磁盘组。*磁盘组* 是共享公用配置的命名磁盘集合。卷创建于磁盘组中，并且仅限于使用该磁盘组中的磁盘。

安装了 Volume Manager 的系统具有默认的磁盘组 `rootdg`。默认情况下，操作被定向到 `rootdg` 磁盘组。系统管理员可根据需要创建其它磁盘组。很多系统最多使用一个磁盘组，除非它们有大量磁盘。直到需要用磁盘创建 Volume Manager 对象时，才将磁盘添加到磁盘组。可以先初始化并保留磁盘，稍后再添加到磁盘组中。但是，为完成 Volume Manager 安装过程，必须将至少一个磁盘（分区）添加到 `rootdg`。

将磁盘添加到磁盘组时，系统会给它指定一个名称（例如，`disk02`）。此名称标识磁盘以用于卷操作：卷创建或镜像。此名称直接与物理磁盘相关。如果将物理磁盘移动到不同的目标地址或不同的控制器，该磁盘仍由名称 `disk02` 表示。可以替换磁盘，方法是先将一个不同的物理磁盘与将要替换的磁盘名称关联，然后（从镜像或备份副本）恢复存储在原始磁盘上的全部卷数据。

拥有大磁盘组可能导致填满专用区域。对于更大的磁盘组，其中的磁盘应设置有更大的可记录专用区域。专用区域的主要部分是用于磁盘组配置数据库的空间，而此数据库包含磁盘组中每个 Volume Manager 对象的记录。因为每个配置记录占用 256 个字节（或半个块），所以在磁盘组中可创建的记录数是配置数据库副本大小的两倍。可从命令 `vxvg list diskgroupname` 的输出中获取副本大小。



磁盘和磁盘组命令

Volume Manager 提供多个可用于管理磁盘的界面：

- ◆ 图形用户界面
- ◆ 一套命令行命令
- ◆ 基于菜单的 `vxdiskadm` 界面
- ◆ 解除重定位命令

本章讨论的命令包括：

- ◆ `vxdiskadm` —— 这是“Volume Manager 支持操作”菜单界面。此命令提供磁盘操作菜单。主菜单中的每一项通过向您提供信息和提示，帮助您完成特定的任务。它为很多问题提供默认应答，因此您可以很容易地选择常用应答。有关如何使用 `vxdiskadm` 的信息，请参见 `vxdiskadm(1M)` 帮助手册。
- ◆ `vxdiskadd` —— 此命令用于向 Volume Manager 添加标准磁盘。`vxdiskadd` 通过显示信息和提示，帮助您完成初始化新磁盘的过程。有关如何使用 `vxdiskadd` 的信息，请参见 `vxdiskadd(1M)` 帮助手册。
- ◆ `vxdisk` —— 此命令管理受 VxVM 控制的磁盘。`vxdisk` 定义特殊磁盘设备，初始化存储在磁盘上的信息（Volume Manager 用此信息标识和管理磁盘），并执行其它的特殊操作。有关如何使用 `vxdisk` 的信息，请参见 `vxdisk(1M)` 帮助手册。
- ◆ `vxdbg` —— 此命令在磁盘组上操作。`vxdbg` 创建新磁盘组并管理现有的磁盘组。有关如何使用 `vxdbg` 的信息，请参见 `vxdbg(1M)` 帮助手册。
- ◆ `vxunrelocate` —— 此命令将已由热重定位功能重定位的子磁盘移回原始磁盘。

`vxdiskadd` 实用程序和大多数 `vxdiskadm` 操作仅适用于标准磁盘设备。

大多数 Volume Manager 命令允许使用 `-g` 选项指定磁盘组。例如，若要在磁盘组 `mkt dg` 中创建一个卷，请使用下列命令：

```
# vxassist -g mkt dg make mktvol 50m
```

此卷的（块）卷设备是：

```
/dev/vx/dsk/mkt dg/mktvol
```

如果对象名唯一，则无需指定磁盘组。大多数 Volume Manager 命令使用命令行中指定的对象名确定用于操作的磁盘组。例如，若要在磁盘 `mkt dg01` 上创建一个卷，并且无需指定磁盘组名称，请使用下列命令：

```
# vxassist make mktvol 50m mkt dg01
```

只要两个磁盘组中的对象名不同，很多命令都能以此方式工作。例如，Volume Manager 允许在 `root dg` 和 `mkt dg` 中同时创建名为 `mktvol` 的卷。如果要这样做，必须将 `-g mkt dg` 添加到任何用于处理 `mkt dg` 磁盘组中的卷的命令。



初始化和添加磁盘

磁盘添加到 VxVM 后将被初始化或封装。封装可保留磁盘上的现有数据，但初始化将破坏现有数据。

Volume Manager 的磁盘初始化有两个级别：

1. 格式化磁盘介质本身。这必须在 Volume Manager 外部完成。
2. 在磁盘上存储供 Volume Manager 使用的标识和配置信息。提供 Volume Manager 界面分步完成此级别的磁盘初始化。

可将完全初始化的磁盘添加到磁盘组，替换以前出现故障的磁盘或创建新磁盘组。这些主题将在本章后面部分讨论。

格式化磁盘介质

若要执行第一个初始化阶段，请使用交互式 `format`（在某些系统中为 `diskadd`）命令完成任何磁盘的介质格式化。

注意 SCSI 磁盘通常已预先格式化。通常情况下，仅当格式遭到严重损坏时，才需要 `format`（或 `diskadd`）命令。

Volume Manager 磁盘安装

可以使用 `vxdiskadm` 菜单或 `vxdiskadd` 完成磁盘初始化阶段。本节描述如何使用 `vxdiskadd`。有关如何使用 `vxdiskadm` 初始化控制器上的单个磁盘或所有磁盘的信息，请参阅第 91 页的“[vxdiskadm 菜单界面任务](#)”。

可以使用 `vxdiskadd` 初始化特定的磁盘。例如，若要初始化第一个控制器上的第二个磁盘，请使用下列命令：

```
# vxdiskadd c0t1d2
```

`vxdiskadd` 检查磁盘以确定它是否已初始化，并根据检查结果显示提示。`vxdiskadd` 检查可以封装的磁盘（请参见第 89 页的“[使用 vxdisk 实现特殊封装](#)”）、已添加到 Volume Manager 的磁盘以及其它情况。

注意 如果正在添加的是未初始化的磁盘，则在使用 `vxdiskadd` 的过程中，控制台上会显示警告和出错讯息。忽略这些信息。在磁盘完全初始化后，这些信息将不再出现；当初始化完成时，`vxdiskadd` 将显示成功信息。



在下列提示下，输入 **y**（或按回车键）继续：

添加或初始化磁盘

菜单：VolumeManager/Disk/AddDisks

这里是所选磁盘。输出格式：[设备名]

c0t1d0

继续进行操作吗？[y,n,q,?]（默认：y）**y**

如果磁盘未初始化，或者如果您选择重新初始化磁盘，将显示下列提示信息：

您可以选择将此磁盘添加至现有磁盘组、新磁盘组或留作将来用于添加或替换操作。若要创建新磁盘组，需选择一个尚未存在的磁盘组名称。若要留作将来使用，指定“none”磁盘组名称。

哪一个磁盘组 [<group>,none,list,q,?]（默认：rootdg）

若要将磁盘添加到默认磁盘组 rootdg，请按回车键。若要使磁盘保持空闲以用作更换磁盘（尚未添加到任何磁盘组），请输入 **none**。之后，系统将提示您为磁盘组中的磁盘选择一个名称：

将默认磁盘名用于此磁盘吗？[y,n,q,?]（默认：y）**y**

正常情况下，应接受默认磁盘名称（除非愿意输入特殊磁盘名称）。

在下列提示下，输入 **n** 表示该磁盘不用作热重定位备用磁盘：

添加磁盘作为 rootdg 的备用磁盘吗？[y,n,q,?]（默认：n）**n**

在下列提示下，输入 **n** 表示该磁盘可用于热重定位：

从热重定位使用对象中排除另外的磁盘吗？[y,n,q,?]
（默认：n）**n**

显示以下信息后，输入 **y** 或按回车键以继续操作：

使用默认磁盘名将选定的磁盘添加至磁盘组 rootdg。

c0t1d0

继续进行操作吗？[y,n,q,?]（默认：y）**y**

如果确定在该磁盘上没有需要保存的数据，请在下列提示下输入 **n**：

以下磁盘设备具有有效的 VTOC，但是尚未进行用于 Volume Manager 的初始化。如果该磁盘上存在不应破坏的数据，则应将现有磁盘分区封装为卷而不是将该磁盘作为新磁盘添加。输出格式：[设备名]

c0t1d0

封装此设备吗？[y,n,q,?]（默认：y）

当 `vxdiskadm` 提示您初始化该磁盘时，输入 **y**：

```
不执行封装而是执行初始化吗? [y,n,q,?] (默认: n) y
```

现在出现类似于下面的信息，确认磁盘 `c1t0d1` 是否正受 Volume Manager 的控制。在某些系统上可能还为您提供执行表面分析的选项。

```
初始化设备 c0t1d0。
```

```
执行表面分析 (极力推荐) [y,n,q,?] (默认: y) n
用磁盘名 disk33 将磁盘设备 c0t1d0 添加至磁盘组 rootdg。
```

将磁盘添加到 Volume Manager

必须使磁盘受 Volume Manager 的控制，并将其添加到磁盘组，才能将该磁盘的空间用于卷。如果以前使用过该磁盘，但它未受 Volume Manager 的控制，则可以在保留磁盘上现有数据的同时，仍然让 Volume Manager 控制该磁盘。可以使用 Volume Manager 的封装功能完成此操作。封装可保留磁盘上的任何数据。如果是新磁盘，必须将其初始化。初始化将破坏磁盘上的全部现有数据。如果该磁盘以前不受 Volume Manager 的控制，且不需保留任何数据，则应初始化。

若要添加磁盘，请使用下列命令：

```
# vxdiskadd devname
```

这里的 `devname` 是将要添加的磁盘的设备名称。

若要将设备 `c1t0d0` 添加到 Volume Manager 的控制中，请完成下列操作：

1. 输入下列命令启动 `vxdiskadd`：

```
# vxdiskadd c1t0d0
```

2. 若要继续此任务，请在下列提示下输入 **y**（或按回车键）：

```
添加或初始化磁盘
```

```
菜单: VolumeManager/Disk/AddDisks
```

```
这里是所选磁盘。输出格式: [设备名]
```

```
c1t0d0
```

```
继续进行操作吗? [y,n,q,?] (默认: y) y
```



3. 在下列提示下，指定将向其中添加磁盘的磁盘组，或按回车键接受 `rootdg`：（有关更多信息，请参见第 68 页的“[使磁盘脱机](#)”。）

您可以选择将此磁盘添加至现有磁盘组、新磁盘组或留作将来用于添加或替换操作。若要创建新磁盘组，需选择一个尚未存在的磁盘组名称。要留作将来使用，指定“`none`”磁盘组名称。

哪一个磁盘组 [`<group>,none,list,q,?`]（默认：`rootdg`）

4. 在下列提示下，既可以按回车键接受默认磁盘名称，也可以输入新的磁盘名称：
将默认磁盘名用于此磁盘吗？ [`y,n,q,?`]（默认：`y`）

5. 当提示您是否将该磁盘用作热重定位备用磁盘时，输入 `n`（或按回车键）：
添加磁盘作为 `rootdg` 的备用磁盘吗？ [`y,n,q,?`]（默认：`n`） `n`
（有关更多信息，请参见第 91 页的“[vxdiskadm 菜单界面任务](#)”。）

6. 在下列提示下，输入 `n` 表示该磁盘可用于热重定位：
从热重定位使用对象中排除另外的磁盘吗？ [`y,n,q,?`]
[默认：`n`] `n`

7. 若要继续此任务，请在下列提示下输入 `y`（或按回车键）：
使用默认磁盘名将选定的磁盘添加至磁盘组 `rootdg`。

`c1t0d0`

继续进行操作吗？ [`y,n,q,?`]（默认：`y`） `y`

8. 如果磁盘上有需要保留的数据，请输入 `y` 选择封装：

以下磁盘设备具有有效的 VTOC，但是尚未进行用于 Volume Manager 的初始化。如果该磁盘上存在不应破坏的数据，则应将现有磁盘分区封装为卷而不是将该磁盘作为新磁盘添加。输出格式：[设备名]

`c1t0d0`

封装此设备吗？ [`y,n,q,?`]（默认：`y`） `y`

9. 若要继续此任务，请在下列提示下按回车键：

以下磁盘已被选定用于封装。
输出格式：[设备名]

`c1t0d0`

继续进行封装吗？ [`y,n,q,?`]（默认：`y`）

出现类似于下面这样的信息，指出正在封装磁盘供 Volume Manager 使用：

磁盘设备 `c1t0d0` 将被封装并用磁盘名 `disk01` 添加至磁盘组 `rootdg`。

`c1t0d0` 磁盘已经被配置用于封装操作。

封装的第一阶段已成功完成。

现在应尽可能早地重新启动系统。

封装将需要两至三次重新启动，这将在下一次重新启动后自动发生。若要重新启动，执行命令：

```
shutdown -g0 -y -i6
```

这将更新 `/etc/vfstab` 文件，以便将卷设备用于安装该磁盘设备上的文件系统。您将需要更新其它参考文件，如备份脚本、数据库或手动创建的交换设备。

再见！

记住，只要方便时就执行关闭和重新启动。

向磁盘组添加磁盘

可以向已建立的磁盘组中添加新的磁盘。例如，当前磁盘空间不足以满足应用程序或工作组的要求，特别是如果这些要求发生了变化。

若要向磁盘组添加已初始化的磁盘，请使用下列命令：

```
# vxdiskadd devname
```

若要将设备 `c1t1d0` 添加到 `rootdg`，请使用下列步骤：

1. 输入下列命令启动 `vxdiskadd`：

```
# vxdiskadd c1t1d0
```

`vxdiskadd` 显示下列信息：

添加或初始化磁盘

菜单：VolumeManager/Disk/AddDisks

这里是所选磁盘。输出格式：[设备名]

`c1t1d0`

继续进行操作吗？[y,n,q,?]（默认：y） **y**

2. 在下列提示下，指定将向其中添加磁盘的磁盘组，或按回车键接受 `rootdg`：

您可以选择将此磁盘添加至现有磁盘组、新磁盘组或留作将来用于添加或替换操作。若要创建新磁盘组，需选择一个尚未存在的磁盘组名称。若要留作将来使用，指定 “none” 磁盘组名称。

Which disk group [<group>,none,list,q,?]（默认：rootdg）



3. 在下列提示下，既可以按回车键接受默认磁盘名称，也可以输入新的磁盘名称：
将默认磁盘名用于此磁盘吗？ [y,n,q,?] (默认: y)
 4. 当 `vxdiskadd` 询问是否将该磁盘用作热重定位备用磁盘时，输入 **n**（或按回车键）：
添加磁盘作为 `rootdg` 的备用磁盘吗？ [y,n,q,?] (默认: n) **n**
 5. 在下列提示下，输入 **n** 表示该磁盘可用于热重定位：
从热重定位使用对象中排除另外的磁盘吗？ [y,n,q,?]
[默认: n] **n**
 6. 若要继续此任务，请在下列提示下输入 **y**（或按回车键）：
使用默认磁盘名将选定的磁盘添加至磁盘组 `rootdg`。
`c1t1d0`
继续进行操作吗？ [y,n,q,?] (默认: y) **y**
 7. 下列提示表明以前曾初始化该磁盘供 `Volume Manager` 以后使用；请输入 **y** 确认现在想使用该磁盘：
以下磁盘设备似乎已初始化。该磁盘当前可用作替换磁盘。
输出格式: [设备名]
`c1t1d0`
使用此设备吗？ [y,n,q,?] (默认: y) **y**
 8. 若要重新初始化该磁盘，请在下列提示下输入 **y**（或按回车键）：
下列您所选用的磁盘看来已经初始化供 `Volume Manager` 使用。如果您确信该磁盘已经初始化供 `Volume Manager` 使用，则不需要重新初始化该磁盘设备。
输出格式: [设备名]
`c1t1d0`
重新初始化该设备吗？ [y,n,q,?] (默认: y) **y**
- 现在出现类似于下面这样的信息，确认该磁盘正在重新初始化供 `Volume Manager` 使用。在某些系统上可能还为您提供执行表面分析的选项。
- 初始化设备 `c1t1d0`。
- 执行表面分析（极力推荐） [y,n,q,?] (默认: y) **n**
用磁盘名 `disk03` 将磁盘设备 `c1t1d0` 添加至磁盘组 `rootdg`。
- 若要确认该磁盘是否已添加到磁盘组，请输入：

```
# vxdisk list
```


Volume Manager 返回一个类似于下面这样的列表：

设备	类型	磁盘	组	状态
c0t0d0s2	sliced	disk04	rootdg	联机
c1t0d0s2	sliced	disk01	rootdg	联机
c1t1d0s2	sliced	disk03	rootdg	联机

从磁盘组中删除磁盘

可用此命令将不包含子磁盘的磁盘从磁盘组中删除：

```
# vxdg [-g groupname] rmdisk diskname
```

这里，只有默认磁盘组 rootdg 以外的磁盘组需指定名称。

例如，若要将 disk02 从 rootdg 中删除，请使用此命令：

```
# vxdg rmdisk disk02
```

如果尝试删除的磁盘包含子磁盘，将显示下列出错讯息：

```
vxdg: 磁盘 diskname 被一个或多个子磁盘使用
```

对 vxdg 使用 -k 选项删除设备设定。-k 选项允许删除包含子磁盘的磁盘。有关更多信息，请参见 vxdg(1M) 帮助手册。

注意 对 vxdg 使用 -k 选项会导致数据丢失。

一旦将磁盘从其磁盘组中删除，可以（可选）将它从 Volume Manager 的控制中完全删除，如下所示：

```
# vxdisk rm devicename
```

例如，若要从 Volume Manager 的控制中删除 c1t0d0（或 c1b0t0d0），请使用这些命令：

```
# vxdisk rm c1t0d0s2
```

或在使用总线的系统上：

```
# vxdisk rm c1b0t0d0s0
```

可以删除定义了子磁盘的磁盘。例如，可以将所有卷合并到一个磁盘上。如果使用 vxdiskadm 删除磁盘，可以选择将卷移出该磁盘。为此，请运行 vxdiskadm，并从主菜单中选择项 3（“删除磁盘”）。

如果该磁盘正由某些子磁盘使用，通常会显示下列信息：



下列子磁盘当前使用磁盘 `disk02` 的一部分：

```
home usrvol
```

在可以删除 `disk02` 前必须将子磁盘从 `disk02` 中移出。

将子磁盘移动到其它磁盘吗？ [`y,n,q,?`]（默认：`n`）

如果选择 **y**，则只要可能，所有子磁盘都将被移出该磁盘。有些子磁盘可能无法移动。子磁盘无法移动的最常见原因是：

- ◆ 其余磁盘上的空间不足。
- ◆ 不能在不同于卷中现有 Plex 或条带子磁盘的磁盘上分配 Plex 或条带子磁盘。

如果 `vxdiskadm` 无法移动某些子磁盘，则在继续磁盘删除操作前，可能需要从磁盘中删除一些 `plex` 以释放更多空间。

移动磁盘

若要在磁盘组之间移动磁盘，请将磁盘从一个磁盘组中删除，然后将其添加到另一个组中。例如，若要从磁盘组 `rootdg` 中移动物理磁盘 `c0t3d0`（带有磁盘名称 `disk04`），并将其添加到磁盘组 `mkt dg`，可以使用下列命令：

```
# vxdbg rmdisk disk04
# vxdbg -g mkt dg adddisk mkt dg02=c0t3d0
```

注意 此过程并不保存磁盘上的配置信息或数据。

也可用命令 `vxdiskadm` 移动磁盘。在主菜单中选择项 3（“删除磁盘”），然后选择项 1（“添加或初始化磁盘”）。

重命名磁盘

如果您没有为磁盘指定一个 Volume Manager 名称，当您将该磁盘置于 Volume Manager 控制之下时，Volume Manager 将为该磁盘提供一个默认名称。Volume Manager 用 Volume Manager 名称标识磁盘位置或磁盘类型。可以输入下列命令更改磁盘名称以反映拥有权或使用的变化：

```
# vxedit rename old_diskname new_diskname
```

若要将 `disk01` 重命名为 `disk03`，请输入：

```
# vxedit rename disk01 disk03
```

若要确认名称是否已发生更改，请输入：

```
# vxdisk list
```

Volume Manager 返回下列内容：

设备	类型	磁盘	组	状态
c0t0d0s2	sliced	disk04	rootdg	联机
c1t0d0s2	sliced	disk03	rootdg	联机
c1t1d0s2	sliced	-	-	联机

注意 默认情况下，Volume Manager 在子磁盘对象所在的 VM 磁盘之后对子磁盘对象进行命名。重命名 VM 磁盘并不自动重命名它所包含的子磁盘。

保留磁盘

默认情况下，vxassist 分配任何有空闲空间的磁盘中的空间。可以保留一套磁盘用于特殊目的，如为了避免将特别慢或特别快的磁盘用于常规操作。

使用下列命令保留用于特殊目的的磁盘：

```
# vxedit set reserve=on diskname
```

在输入此命令后，vxassist 并不分配选定磁盘中的空间，除非在 vxassist 命令行中特别提到该磁盘。例如，如果磁盘 disk03 是保留的，命令：

```
# vxassist make vol03 20m disk03
```

将使保留无效，并在 disk03 上创建一个 20 兆字节的卷。而命令：

```
# vxassist make vol04 20m
```

则不使用 disk03，即使其它任何磁盘上都没有空闲空间。

若要关闭磁盘保留功能，请输入：

```
# vxedit set reserve=no diskname
```



使磁盘脱机

可能需要使物理磁盘脱机。如果磁盘被破坏，需要禁用并移去该磁盘。在将物理磁盘设备移动到与其它系统连接的另一位置之前，也必须禁用该磁盘。

若要使物理磁盘脱机，首先将该磁盘从其磁盘组中删除。有关更多信息，请参见第 69 页的“[删除磁盘](#)”。然后用下列命令将磁盘置于“脱机”状态：

```
# vxdisk offline devname
```

注意 这里使用设备名称是因为该磁盘不再属于磁盘组，因此没有管理名称。

镜像磁盘

通过镜像根磁盘，可以将根卷和引导所需的其它区域镜像到另一个磁盘上。

若要将引导（根）磁盘镜像到其它磁盘上，请使用这里概述的步骤或者使用 `vxdiskadm`。这使得有可能通过用引导磁盘镜像来替换引导磁盘的方式将其从故障中恢复。

如果您希望在需要替换时利用 Volume Manager 引导磁盘别名标识引导磁盘镜像，则在镜像引导磁盘前，必须将 EEPROM 变量 `use-nvramrc?` 设置为 `true`。如果将该变量设置为 `false`，您需要自己确定可引导的磁盘。可以按照如下方式将该变量设置为 `true`：

```
eprom use-nvramrc?=true
```

若要镜像引导磁盘，请完成下列操作：

1. 选择一个至少与引导磁盘一样大的磁盘。
2. 使用 `vxdiskadd` 命令，将选定的磁盘添加为新磁盘（如果还未添加该磁盘）。
3. 执行下列命令：

```
# /etc/vx/bin/vxrootmir alternate_disk
```

这里的 `alternate_disk` 是分配给另一磁盘的磁盘名称。

`vxrootmir` 为 `rootvol`（替换磁盘上包含根文件系统的卷）创建镜像。

替换引导磁盘被配置为在主引导磁盘失败时从替换引导磁盘引导。

引导磁盘上可能还有其它卷，如包含 `/home` 或 `/tmp` 文件系统的卷。可以使用 `vxassist` 实用程序分别镜像这些卷。例如，如果在卷 `homevol` 上有 `/home` 文件系统，可以使用下列命令将该卷镜像到 `alternate_disk`：

```
# vxassist mirror homevol alternate_disk
```

如果替换引导磁盘上没有空间容纳部分文件系统的副本，可将它们镜像到其它磁盘上。也可以在挂载在系统上的其它磁盘间跨接或条带化这些卷。

若要列出主引导磁盘上的所有卷，请使用下列命令：

```
# vxprint -t -v -e\'aslist.aslist.sd_disk="boot_disk"\'
```

若要将该磁盘上的所有连续卷镜像到替换引导磁盘，请使用下列命令：

```
# /etc/vx/bin/vxmirror boot_disk alternate_disk
```

删除磁盘

如果磁盘将要发生故障或已发生故障，可以将磁盘从系统中删除并移动到其它系统。在将磁盘从当前系统中删除前，必须：

1. 卸载卷上的全部文件系统。
2. 停止磁盘上的卷。
3. 将卷移动到其它磁盘或备份这些卷。若要移动卷，应先将卷镜像到一个或多个其它磁盘上，然后再删除卷的原始副本。如果不再需要这些卷，可将其删除而非移动。

若要删除磁盘，请使用下列步骤：

4. 用此命令将磁盘从其磁盘组中删除：

```
# vxdg [-g groupname] rmdisk diskname
```

这里的 *groupname* 是磁盘所属的磁盘组的名称，*diskname* 是将要删除的磁盘的名称。

例如，若要从 rootdg 中删除 disk01，请输入：

```
# vxdg rmdisk disk01
```

由于 rootdg 是默认磁盘组，因此不必指定它。

5. 用下列命令将磁盘从 Volume Manager 的控制中删除：

```
# vxdisk rm devicename
```

例如，若要从 Volume Manager 的控制中删除 c1t0d0，请输入：

```
# vxdisk rm c1t0d0s2
```



显示磁盘信息

在使用磁盘前，需要知道该磁盘是否已初始化以及是否受 Volume Manager 的控制。还需要知道该磁盘是否是磁盘组的一部分，因为不能在不属于磁盘组的磁盘上创建卷。vxdisk list 命令显示所有可识别的磁盘的设备名称、磁盘名称、与每个磁盘关联的磁盘组名称和每个磁盘的状态。

可以用此命令显示在 Volume Manager 中定义的所有磁盘的信息：

```
# vxdisk list
```

Volume Manager 返回下列显示信息：

设备	类型	磁盘	组	状态
c0t0d0s2	sliced	disk04	rootdg	联机
c1t0d0s2	sliced	disk01	rootdg	联机
c1t1d0s2	sliced	-	-	联机

若要显示在 Volume Manager 中定义的具体某个磁盘的详细资料，请输入：

```
# vxdisk list disk01
```

vxdisk 实用程序用于显示具体某个元设备的动态多路径信息。元设备是表示具体某个物理磁盘的设备，该物理磁盘有多条与系统 I/O 控制器相连的物理路径。在 Volume Manager 中，系统中的所有物理磁盘都表示为有一条或多条物理路径的元设备。

可用此命令查看具体某个元设备的多路径信息：

```
# vxdisk list device
```

device 是由 Volume Manager 的 DMP 子系统构成的元设备。

该输出显示由元设备 c2t0d0s2 表示的物理设备的两条路径。c2t0d0s2 为活动状态 (state=enabled)，另一条路径 c1t0d0s2 为无效状态 (state=disabled)。

Volume Manager 返回下列显示信息：

```
Device      device
devicetag  c2t0d0
type       sliced
hostid     aparajita
disk       name=disk01 id=861086917.1052.aparajita
group      name=rootdg id=861086912.1025.aparajita
flags      online ready autoconfig autoimport imported
pubpaths   block=/dev/vx/dmp/c2t0d0s4 char=/dev/vx/rdmp/c2t0d0s4
privpaths  block=/dev/vx/dmp/c2t0d0s3 char=/dev/vx/rdmp/c2t0d0s3
version    2.1
iosize     min=512 (bytes) max=2048 (blocks)
public     slice=4 offset=0 len=1043840
private    slice=3 offset=1 len=1119
update     time=861801175 seqno=0.48
```



```

headers      0 248
configs      count=1 len=795
logs         count=1 len=120
Defined regions
config       priv 000017-000247[000231]:copy=01 offset=000000 enabled
config       priv 000249-000812[000564]:copy=01 offset=000231 enabled
log          priv 000813-000932[000120]:copy=01 offset=000000 enabled
Multipathing information:
numpaths:    2
c2t0d0s2    state=enabled          type=primary
c1t0d0s2    state=disabled         type=secondary

```

以 *type* 的形式显示活动 / 被动类型磁盘阵列上的磁盘的附加信息。此信息表明到磁盘的主和辅助路径。例如，DG Clariion、Hitachi DF350 等。

不为活动 / 活动类型磁盘阵列上的磁盘显示此类信息，因为对于这些磁盘阵列上的磁盘而言，不存在主路径和辅助路径的概念。例如，StorEdge A5000 和 Sparc Storage Array (SSA) 磁盘阵列。

检测和更换故障磁盘

本节描述如何检测磁盘故障和更换故障磁盘。首先介绍热重定位功能，此功能在发生故障时自动尝试恢复冗余的 Volume Manager 对象。

热重定位

热重定位自动对冗余（镜像或 RAID-5）Volume Manager 对象的 I/O 故障做出反应，并恢复冗余和对这些对象的访问。Volume Manager 检测对象的 I/O 故障并将受影响的子磁盘重定位到被指定为备用磁盘的磁盘和 / 或磁盘组内的空闲空间。然后 Volume Manager 重建故障发生前已存在的对象，并使它们冗余和重新可存取。有关热重定位的描述，请参阅“[Volume Manager 操作](#)”。

注意 只对故障磁盘上的冗余（镜像或 RAID-5）子磁盘执行热重定位。不对故障磁盘的非冗余子磁盘进行重定位，但通知系统管理员所发生的故障。

当发生故障时，热重定位在默认情况下启用并生效，无需系统管理员的干预。热重定位后台驻留程序 `vxrelocd` 检测并对表示下列故障类型的 Volume Manager 事件做出反应：

- ◆ 磁盘故障——当 Volume Manager 对象发生 I/O 故障时，通常会检测到这类故障。Volume Manager 将尝试纠正此错误。如果无法纠正此错误，Volume Manager 将尝试存取磁盘专用区域中的配置信息。如果不能存取专用区域，便认为该磁盘发生故障。
- ◆ plex 故障——当 plex 中存在无法纠正的 I/O 错误时（此错误将影响 plex 中的子磁盘），通常会检测到这类故障。对于镜像卷，plex 被分离。
- ◆ RAID-5 子磁盘故障——当存在无法纠正的 I/O 错误时，通常会检测到这类故障。子磁盘被分离。



当检测到这类故障时，`vxrelocd` 通过电子邮件通知系统管理员所发生的故障和受到影响的 Volume Manager 对象。然后，`vxrelocd` 确定可以重定位哪些子磁盘（如果有）。如果能够重定位，`vxrelocd` 查找合适的重定位空间并重定位子磁盘。

在发生故障的磁盘组中，从预留热重定位的磁盘中选择热重定位空间。如果没有备用磁盘或者需要额外的空间，则使用同一磁盘组中的空闲空间，标记为 `nohotuse` 的磁盘（不可用于热重定位）除外。一旦重定位子磁盘，每个重定位的子磁盘都重新挂接到其 `plex` 上。

最后，`vxrelocd` 启动适当的恢复过程。例如，对于镜像卷，恢复涉及镜像再同步，对于 RAID-5 卷则涉及数据恢复。将通知系统管理员所采取的热重定位和恢复操作。

如果不能重定位，也将通知系统管理员，并且不采取进一步的操作。在下列情况下不能重定位：

- ◆ 如果子磁盘非冗余（即，它们不属于镜像或 RAID-5 卷），则不能重定位。
- ◆ 如果磁盘组中没有足够的可用空间（备用磁盘或空闲空间，并且未用 `nohotuse` 标志从热重定位使用对象中排除），则不能重定位故障子磁盘。
- ◆ 如果唯一可用的空间所在的磁盘已包含故障 `plex` 的镜像，则不能重定位该 `plex` 中的子磁盘。
- ◆ 如果唯一可用的空间所在的磁盘已包含 RAID-5 卷日志 `plex` 或它的一个完好子磁盘，则不能重定位该 RAID-5 `plex` 中的故障子磁盘。
- ◆ 如果镜像卷将 Dirty 区日志子磁盘作为其数据 `plex` 的组成部分，则不能重定位属于该 `plex` 的子磁盘。
- ◆ 如果 RAID-5 卷日志 `plex` 或镜像卷 DRL 日志 `plex` 发生故障，则将在别处创建新的日志 `plex`（所以实际上并没有重定位该日志 `plex`）。

通过在每个磁盘组指定一个或多个磁盘作为热重定位备用磁盘，可以为热重定位做准备。有关如何将磁盘指定为备用磁盘的信息，请参见“[Volume Manager 操作](#)”一章。如果发生故障时没有可用的备用磁盘，或者备用磁盘上的空间不足，则将自动使用空闲空间（如果该空间未用 `nohotuse` 标志从热重定位使用对象中排除）。

通过指定备用磁盘，可以控制在发生故障时用于重定位的空间。如果空闲空间和备用磁盘空间合起来仍不够用或者不符合冗余限制，则不重定位子磁盘。

通过指定 `nohotuse` 磁盘，可以控制不能用作替换磁盘的空闲磁盘。

有两种方法使磁盘不可用于热重定位。例如：

```
# vxedit -g rootdg set nohotuse=on disk01
# vxdiskadm
```

从 `vxdiskadm` 主菜单中选择选项 15（“从热重定位使用对象中排除磁盘”）

有两种方法使磁盘可用于热重定位。例如：

```
# vxedit -g rootdg set nohotuse=off disk01
# vxdiskadm
```

从 vxdiskadm 主菜单中选择选项 16 (“使磁盘可供热重定位使用”)

有两种方法查找备用磁盘或不能用于热重定位的磁盘。例如：

```
# vxdisl list
# vxprint
```

在成功重定位后，需要删除和更换故障磁盘（参见“[更换磁盘](#)”）。根据重定位的子磁盘的位置，可以选择在重定位后将子磁盘移动到别处（参见第 74 页的“[移动重定位的子磁盘](#)”）。

修改 vxrelocd

只要 vxrelocd 在运行，热重定位就是打开的。保留热重定位的打开状态，以便在发生故障时可以利用此功能。但是，如果选择禁止此功能（不希望磁盘上的某些空闲空间用于重定位），则将阻止 vxrelocd 在系统启动时启动。

有关如何在系统启动时禁止热重定位的信息，请参阅《*VERITAS Volume Manager 安装指南*》。通过中止 vxrelocd 进程（当正在进行热重定位尝试时不要这样做），可随时停止热重定位。

通过在调用 vxrelocd(/etc/rc2.d/S95vxvm-recover) 的启动文件中编辑 vxrelocd 行，或中止现有 vxrelocd 进程并用不同的选项重新启动它，可以稍微更改 vxrelocd 的工作方式。在启动文件中更改了 vxrelocd 的调用方式后，需要重新启动系统以使更改生效。相反，如果选择中止并重新启动后台驻留程序，请确保在中止 vxrelocd 进程时，没有进行热重定位。应立即重新启动后台驻留程序，以便发生故障时热重定位可生效。

可按如下方式改变 vxrelocd 的行为：

- ◆ 默认情况下，当检测到故障和执行重定位操作时，vxrelocd 向 root 发送电子邮件。通过按如下方式添加适当的用户名并调用 vxrelocd，可以指示 vxrelocd 通知其他用户：

```
# vxrelocd root user_name1 user_name2 &
```

- ◆ 若要减小恢复对系统性能造成的影响，可以指示 vxrelocd 增加卷中每个区域恢复之间的延迟时间，如下所示：

```
# vxrelocd -o slow[=IOdelay] root &
```

这里的可选 *IOdelay* 表示所需的延迟（以毫秒为单位）。延迟的默认值是 250 毫秒。有关更多信息，请参见 vxrelocd(1M) 帮助手册。



-o 此选项用于回复到早期版本。通过指定 `VxVM_version -o`，可以指示 `vxrelocd` 使用那个版本中的重定位方案。

-s 在 `vxrelocd` 试图重定位之前，将当前配置的快照保存在 `/etc/vx/saveconfig.d` 中。此选项指定每个磁盘组所保留配置的最多数量。默认值是 32。

显示备用磁盘信息

使用命令 `vx dg spare` 显示有关可用于重定位的所有备用磁盘的信息。输出显示下列信息：

GROUP	DISK	DEVICE	TAG	OFFSET	LENGTH	FLAGS
rootdg	disk02	c0t2d0s2	c0t2d0	0	658007	s

在此实例中，`disk02` 是唯一被指定为备用磁盘的磁盘。`LENGTH` 字段表示磁盘上当前可用于重定位的备用空间量。

可用下列命令显示有关当前被指定为备用磁盘的磁盘的信息：

- ◆ `vx disk list` —— 列出磁盘信息并显示带 `spare` 标志的备用磁盘。
- ◆ `vx print` —— 列出磁盘和其它信息，并显示带 `SPARE` 标志的备用磁盘。

移动重定位的子磁盘

当发生热重定位时，子磁盘重定位到磁盘组中的备用磁盘和 / 或可用的空闲空间。子磁盘的新位置所提供的性能或数据布局与进行热重定位之前可能有所不同。可以移动重定位的子磁盘（在完成热重定位后）以提高性能。

也可将重定位的子磁盘移出备用磁盘，以保留备用磁盘空间空闲供热重定位使用。移动子磁盘的另一个原因是重建热重定位之前的配置。

在热重定位期间，发送给 `root` 的电子邮件信息如下例所示：

```

收件人: root
主题: 主机 teal 上的 Volume Manager 故障

尝试从 plex home-02 重定位子磁盘 disk02-03。
Dev_offset 0 length 1164 dm_name disk02 da_name c0t5d0s2。
可用的 plex home-01 将被用于修复数据。

```

此邮件包含重定位前子磁盘的信息，可用于确定重定位后将子磁盘移向何处。

下面的实例信息显示子磁盘重定位后的新位置：

```

收件人: root
主题: 在主机 teal 上尝试 VxVM 重定位

卷 home 的子磁盘 disk02-03 重定位至 disk05-01，
但尚未被恢复。

```

在移动任何重定位的子磁盘前，先修复或更换故障磁盘（如上一节所述）。一旦完成此项工作，即可将重定位的子磁盘移回原始磁盘。例如，可用此命令将重定位的子磁盘 disk05-01 移回 disk02：

```
# vxassist -g rootdg move home !disk05 disk02
```

注意 在移动子磁盘的操作中，RAID-5 卷是非冗余的。

使用 vxunrelocate

VxVM 热重定位允许系统自动对子磁盘级别上冗余 VxVM 对象的 I/O 故障做出反应，然后采取必要的操作使该对象重新可用。此机制检测子磁盘上的 I/O 故障，重定位子磁盘，并恢复与子磁盘关联的 plex。在更换磁盘后，vxunreloc 允许将系统恢复到磁盘故障发生前的配置。vxunreloc 允许将热重定位的子磁盘移回由于故障而更换的磁盘上。

当调用 vxunreloc 时，必须指定磁盘介质名称，这是热重定位的子磁盘最初所在的位置。当 vxunreloc 移动子磁盘时，它将磁盘移动到原始偏移。如果试图解除重定位到比原始故障磁盘小的磁盘上，vxunreloc 只会返回出错讯息。

vxunreloc 提供一个选项，可将重定位的子磁盘移动到不同于它们最初所在的磁盘上。vxunreloc 还提供一个选项，可将子磁盘解除重定位到不同的偏移，只要目标子磁盘足以容纳所有的子磁盘。

如果 vxunreloc 不能将子磁盘替换回与原来一样的偏移，则有一个强制选项可供您将子磁盘移动到指定的磁盘，但不移回到原始偏移。有关更多信息，请参阅 vxunreloc(1M) 帮助手册。

以下实例说明了 vxunreloc 的用法。

实例 1：

假设 disk01 发生故障，并且重定位了所有子磁盘。在更换 disk01 后，vxunreloc 可用于将所有热重定位的子磁盘移回 disk01。

```
vxunreloc -g newdg disk01
```



实例 2:

vxunreloc 实用程序提供的 `-n` 选项可将重定位的子磁盘移动到不同于它们最初所在的磁盘上。

假设 `disk01` 发生故障，并且其上驻留的所有子磁盘都已热重定位到其它磁盘上。vxunreloc 提供的一个选项可将重定位的子磁盘移动到不同于它们最初所在的磁盘上。磁盘修复后，可使用不同的名称（例如，`disk05`）重新添加到磁盘组。如果想将所有热重定位的子磁盘移回新磁盘，可使用下列命令：

```
vxunreloc -g newdg -n disk05 disk01
```

实例 3:

假设 `disk01` 发生故障，并且重定位了子磁盘，而您希望将热重定位的子磁盘移动到某些子磁盘现在所驻留的磁盘 `disk05` 上。可以使用强制选项将热重定位的子磁盘移动到 `disk05`，但不移到与原来完全一样的偏移：

```
vxunreloc -g newdg -f -n disk05 disk01
```

实例 4:

如果子磁盘由于多次发生故障而被多次热重定位，仍然可解除该子磁盘的重定位并使它回到原始位置。例如，如果 `disk01` 发生故障并将一个名为 `disk01-01` 的子磁盘移到 `disk02`，然后 `disk02` 也发生磁盘故障，其上驻留的所有子磁盘（包括热重定位到该磁盘上的那一个子磁盘）都将被再次移动。更换了 `disk02` 后，对 `disk02` 执行 vxunreloc 操作丝毫不会影响热重定位的子磁盘 `disk01-01`。但是，如果在更换了 `disk01` 后紧接着执行 vxunreloc，`disk01-01` 将移回 `disk01`。

在修复或更换了发生故障的磁盘后，可以使用 vxunreloc 将所有的热重定位子磁盘移回磁盘。当热重定位子磁盘后，它的原始磁盘介质名称和在该磁盘内的偏移保存在配置数据库。当使用 vxunreloc 将子磁盘移回原始磁盘或新磁盘上时，这些信息将被删除。原始磁盘介质名称和原始偏移将保存在子磁盘记录中。若要打印 rootdg 磁盘组中 `disk01` 磁盘上的所有热重定位子磁盘，请使用下列命令：

```
# vxprint -g rootdg -se 'sd_orig_dmname="disk01"'
```

若要将磁盘 `disk01` 上的所有热重定位子磁盘移回原始磁盘，请键入：

```
# vxunreloc -g rootdg disk01
```

vxunreloc 实用程序提供的 `-n` 选项可将子磁盘移动到不同于它们最初重定位到的磁盘上。例如，当 `disk01` 发生故障时，其上驻留的所有子磁盘都热重定位到其它磁盘。该磁盘修复后，使用不同的名称，例如，`disk05` 重新添加到磁盘组中。如果希望将所有热重定位的子磁盘移动到新磁盘，可使用下列命令：

```
# vxunreloc -g rootdg -n disk05 disk01
```



目标磁盘上的存储容量至少应等于原始磁盘上处于使用状态的空间大小。如果没有足够的空间，解除重定位操作将失败，并且不会移动任何子磁盘。

当 `vxunreloc` 移动热重定位的子磁盘时，它将磁盘移到原始偏移。但是，如果现有的某些子磁盘占用目标磁盘的部分或全部区域，`vxunreloc` 将失败。基本上有两个选择：(1) 将现有子磁盘移动到别处，并重新执行命令 `vxunreloc`，或 (2) 使用 `vxunreloc` 提供的 `-f` 选项，将子磁盘移动到目标磁盘，并且让 `vxunreloc` 负责在目标磁盘上寻找空间。只要目标磁盘足够大，使磁盘上用于存储子磁盘的区域能容纳所有子磁盘，则所有热重定位的子磁盘都将“解除重定位”，但不移回原始偏移。

在发生错误后重新启动 `vxunreloc`

在内部，`vxunreloc` 分 3 个阶段移动子磁盘。第一个阶段，在指定的目标磁盘上创建与将要解除重定位的子磁盘数量相同的子磁盘。创建了子磁盘后，`vxunreloc` 用字符串“UNRELOC”填充子磁盘记录中的注释字段，以用作标识。第二个阶段是实际的数据移动。如果成功完成所有的子磁盘移动，第三阶段将清除子磁盘记录的注释字段。

创建子磁盘的操作遵从“全或无”原则。如果 `vxunreloc` 无法成功地创建全部子磁盘，则不创建任何子磁盘，`vxunreloc` 随即退出。移动磁盘的操作不遵从“全或无”原则。子磁盘的移动相互独立，因此，如果一个子磁盘移动失败，`vxunreloc` 实用程序将打印出错信息然后退出。但是，所有后续子磁盘仍驻留在热重定位到的磁盘上，并且不会移回。对于已移回的子磁盘，其子磁盘记录中的注释字段仍标记为“UNRELOC”，这是由于还未执行清除阶段。

如果当在目标磁盘上创建新的子磁盘后、但在移回子磁盘前系统关闭，可以在系统恢复后再次运行 `unrelocate` 实用程序。如上所述，在创建新的子磁盘后，`vxunreloc` 将子磁盘的注释字段设置为“UNRELOC”。当重新执行 `vxunreloc` 时，它检查目标磁盘上现有子磁盘的 `offset`、`len` 和 `comment` 字段，确定上一次执行 `vxunreloc` 时子磁盘是否留在了目标磁盘上，然后 `vxunreloc` 将在认为合适的时候使用子磁盘。

不要手工修改注释字段中的字符串“UNRELOC”。

如果一系列子磁盘移动中有任何移动失败，`vxunreloc` 将退出。在此情况下，应检查导致子磁盘移动失败的错误，并确定是否可进行解除重定位。当重新执行 `vxunreloc` 恢复子磁盘移动时，它使用在上一次运行中创建的子磁盘。

用一个事务就可完成清除阶段。对于驻留在目标磁盘上并标记为“UNRELOC”的所有子磁盘，`vxunreloc` 将它们的注释字段重置成 `NULL` 字符串。此操作包括清除以前未成功完成的任何 `vxunreloc` 调用中解除重定位的子磁盘。



检测故障磁盘

注意 Volume Manager 热重定位功能自动检测磁盘故障，并通过电子邮件通知系统管理员所发生的故障。

如果热重定位功能是禁止的，或者错过了电子邮件，可通过 `vxprint` 命令的输出查看磁盘故障，或使用图形用户界面查看磁盘状态。也可在控制台或系统信息文件中查看驱动程序出错讯息。

如果卷发生 I/O 故障（例如，由于磁盘发生无法纠正的错误），Volume Manager 可以分离故障所涉及的 plex。

如果已分离 plex，该 plex 上的 I/O 将停止，但卷的其余 plex 上仍将继续 I/O。如果磁盘彻底坏了，Volume Manager 可将该磁盘从其磁盘组中分离出来。

如果分离了磁盘，该磁盘上的所有 plex 都将被禁用。如果分离的磁盘上有任何未镜像的卷，这些卷同样将被禁用。

部分磁盘故障

如果启用了热重定位，当由于故障分离 plex 或磁盘时，系统将给 root 发送指出故障对象的邮件。如果发生部分磁盘故障，邮件将指出故障 plex。例如，如果包含镜像卷的磁盘发生故障，您会收到如下例所示的邮件信息：

```
收件人: root
主题: 主机 teal 上的 Volume Manager 故障

VERITAS Volume Manager 检测到了故障:

出错的子磁盘:
  home-02
  src-02
```

有关如何向 root 以外的用户发送邮件的信息，请参见第 73 页的“[修改 vxrelocd](#)”。

用下列命令可以确定在上例信息中导致故障的磁盘：

```
vxstat -s -ff home-02 src-02
```



下列信息是典型的输出显示：

类型名称	失败的	
	读取	写入
sd disk01-04	0	0
sd disk01-06	0	0
sd disk02-03	1	0
sd disk02-04	1	0

此显示表明故障发生在 disk02（子磁盘 disk02-03 和 disk02-04 受影响）上。

热重定位操作自动重定位受影响的子磁盘，并启动任何必要的恢复过程。但是，如果无法进行重定位，或热重定位功能是禁止的，必须调查问题并尝试恢复 plex。电缆故障可能导致这些错误，所以请检查与系统上的磁盘连接的电缆。如果电缆有明显的问题，纠正问题并用下列命令恢复 plex：

```
# vxrecover -b home src
```

此命令在后台启动对故障 plex 的恢复（此命令在操作完成前返回）。如果稍后出现出错信息，或者 plex 再次分离并且没有明显的电缆故障，请更换磁盘（参见第 80 页的“[更换磁盘](#)”）。

完全磁盘故障

如果磁盘彻底坏了，并且热重定位是启用的，邮件信息将列出发生故障的磁盘和使用该磁盘的所有 plex。例如，您会收到如下例所示的邮件：

```
收件人: root
主题: 主机 teal 上的 Volume Manager 故障

VERITAS Volume Manager 检测到了故障:

出错的磁盘:
  disk02

出错的 plex:
  home-02
  src-02
  mktng-01

出错磁盘:
  disk02
```

此信息表明，由于故障已分离 disk02。分离磁盘后，不能在其上进行 I/O 操作。plex home-02、src-02 和 mktng-01 也被分离（也许是因为磁盘故障）。

问题可能又出在电缆上。如果不是电缆错误，请更换磁盘（参见第 80 页的“[更换磁盘](#)”）。



更换磁盘

可更换彻底坏了的磁盘（由于故障已分离），方法是运行 `vxdiskadm` 并从主菜单中选择项 5（“替换一个出错的或已删除的磁盘”）。如果有任何已初始化但未添加的磁盘，可从这些磁盘中选择一个作为更换用的磁盘。

注意 即使原磁盘驱动器出现在选择列表中，也不要选择它作为更换用的磁盘。如果没有合适的初始化磁盘，可以选择初始化新的磁盘。

如果磁盘故障导致卷被禁用，则在替换磁盘后必须从备份恢复该卷。若要识别完全驻留在由于故障而被禁止的磁盘上的卷，请使用下列命令：

```
# vxinfo
```

任何作为 `Unstartable` 列出的卷必须从备份恢复。下面是实例 `vxinfo` 显示：

```
home      fsgen      Started
mkting    fsgen      Unstartable
src        fsgen      Started
standvol  gen        Started (用于某些系统)
rootvol   root       Started
swapvol   swap       Started
```

若要重新启动卷 `mkting` 使它能从备份恢复，请使用下列命令：

```
# vxvol -o bg -f start mkting
```

`-o bg` 选项组合以后台任务的形式再同步 `plex`。

如果某磁盘开始发生故障，但尚未彻底坏了，更换该磁盘。这涉及两个步骤：

1. 将磁盘从磁盘组中分离。
2. 用新磁盘更换该磁盘。

若要分离磁盘，请运行 `vxdiskadm` 并从主菜单中选择项 4（“为替换而删除磁盘”）。如果有已初始化的磁盘可用作更换磁盘，可以将该磁盘指定为操作的一部分。否则，必须稍后通过从主菜单中选择项 5（“替换一个出错的或已删除的磁盘”）来指定更换磁盘。

当选择要删除的磁盘以用于更换时，将显示所有可能受此操作影响的卷。下面是一个实例显示：

作为此操作的结果，下列卷将丢失镜像：

```
home src
```

这些卷上的数据将不会丢失。

下列卷正在使用中，作为此操作的结果，它们将被禁用：

```
mkting
```


任何正在使用这些卷的应用程序将无法进行未来的访问。这些卷需要从备份中恢复。

确定要进行此操作吗？ [y,n,q,?] (默认: n)

如果有任何卷有可能被禁止，请从 `vxdiskadm` 中退出并保存该卷。备份该卷或将该卷移出磁盘。若要将卷 `mkting` 移动到 `disk02` 以外的磁盘上，请使用下列命令：

```
# vxassist move mkting !disk02
```

在备份或移动卷后，再次运行 `vxdiskadm` 并继续删除磁盘以用于更换。

删除了磁盘以用于更换后，可从 `vxdiskadm` 主菜单中选择项 5 (“替换一个出错的或已删除的磁盘”) 来指定更换磁盘。

创建磁盘组

通常为 一组特定的用户或应用程序创建磁盘组。磁盘必须位于磁盘组中才能由 Volume Manager 用于卷。Volume Manager 始终有默认磁盘组 `rootdg`，但如果需要，可以添加其它磁盘组。

注意 如果未做出进一步的指定，则所有卷都创建在 `rootdg` 中。所有命令也默认为 `rootdg`。

若要创建磁盘组 `newdg`，请遵照下列步骤：

1. 输入下列命令启动 `vxdiskadd`：

```
# vxdiskadd c1t1d0
```

2. 在下列提示下，按回车键继续：

添加或初始化磁盘

菜单: VolumeManager/Disk/AddDisks

这里是所选的磁盘。输出格式: [设备名]

c1t1d0

继续进行操作吗？ [y,n,q,?] (默认: y)

3. 在下列提示下，指定将向其中添加磁盘的磁盘组 (本例中为 `newdg`)：

您可以选择将此磁盘添加至现有磁盘组、新磁盘组或留作将来用于添加或替换操作。若要创建新磁盘组，需选择一个尚未存在的磁盘组名称。若要留作将来使用，指定 “none” 磁盘组名称。

哪一个磁盘组 [<group>,none,list,q,?] (默认: rootdg) **newdg**

4. 当 `vxdiskadd` 确认当前不存在同名的活动磁盘组时，将提示您确认是否希望创建此名称的新磁盘组。输入 **y** 继续：



不存在名为 `newdg` 的活动磁盘组。

创建一个名为 `newdg` 的新磁盘组吗? [`y,n,q,?`] (默认: `y`) **y**

5. 在下列提示下, 既可按回车键接受默认磁盘名称, 也可输入新的磁盘名称:

将默认磁盘名用于此磁盘吗? [`y,n,q,?`] (默认: `y`)

6. 当 `vxdiskadd` 询问此磁盘是否用作热重定位备用磁盘时, 输入 **n** (或按回车键):

添加磁盘作为 `rootdg` (或 `newdg`) 的备用磁盘吗? [`y,n,q,?`] (默认: `n`) **n**

在下列提示下, 输入 **n** 表示该磁盘可用于热重定位:

从热重定位使用对象中排除另外的磁盘吗? [`y,n,q,?`]

[默认: `n`] **n**

7. 若要继续此任务, 请在下列提示下输入 **y** (或按回车键):

将创建一个名为 `newdg` 的新磁盘组, 并使用默认磁盘名将选定的磁盘添加至该磁盘组。

`c1t1d0`

继续进行操作吗? [`y,n,q,?`] (默认: `y`) **y**

现在出现类似于下列这样的信息, 确认正在初始化该磁盘供 Volume Manager 使用:

初始化设备 `c1t1d0`。

创建一个名为 `newdg` 的新磁盘组, 它包含具有 `newdg01` 名称的磁盘设备 `c1t1d0`。

8. 若要检验是否已创建了该磁盘组, 请使用下列命令:

```
# vxdisk list
```

Volume Manager 返回下列显示信息:

设备	类型	磁盘	组	状态
<code>c0t0d0s2</code>	<code>sliced</code>	<code>disk04</code>	<code>rootdg</code>	联机
<code>c1t0d0s2</code>	<code>sliced</code>	<code>disk03</code>	<code>rootdg</code>	联机
<code>c1t1d0s2</code>	<code>sliced</code>	<code>newdg01</code>	<code>newdg</code>	联机

也可使用操作 `vx dg init` 创建磁盘组。若要使用 `vx dg` 实用程序创建磁盘组, 请使用下列命令:

```
# vx dg init diskgroup diskname=devicename
```

例如, 若要在设备 `c1t0d0s2` 上创建名为 `mktdg` 的磁盘组, 请使用下列命令:

```
# vx dg init mktdg mktdg01=c1t0d0
```



为 `vxdg` 指定的磁盘设备名称必须已由 `vxdiskadd` 初始化。此磁盘不能属于磁盘组。

升级磁盘组

在早于 Volume Manager 3.0 版的版本中，当导入磁盘组时，磁盘组版本自动升级（如果需要）。

Volume Manager 磁盘组升级功能将导入磁盘组和升级版本这两个操作分开。可以导入低版本的磁盘组，并在不升级的情况下使用它。

当希望使用新功能时，可将磁盘组升级。升级是显式操作。一旦升级，磁盘组将不再与不支持新版本的 VxVM 早期版本兼容。

在升级导入的磁盘组之前，不会对磁盘组做任何更改，即不会阻止磁盘组在导入版本上使用，直到管理员显式地将它升级到当前版本。

在升级完成之前，可“按原样”使用磁盘组，只要不尝试使用当前版本的功能。如果尝试使用当前版本的功能，而此功能不是磁盘组的导入版本的功能，则将导致类似于下面这样的出错讯息：

```
vxvm:vxedit: 错误: 磁盘组版本不支持的功能
```

若要使用任何一个新功能，管理员需要执行相应的命令将磁盘组显式升级到支持这些功能的版本。

所有磁盘组都有关联的版本号。Volume Manager 发行版支持一套特定的磁盘组版本。Volume Manager 能导入特定版本的磁盘组并在这些磁盘组上执行操作。操作受限于磁盘组版本所支持的功能和操作。

表 2 总结了引入并支持特定磁盘组版本的 VxVM 发行版，如下所示：

表 2. 磁盘组版本指定

VxVM 发行版	引入版本	支持版本
1.2	10	10
1.3	15	15
2.0	20	20
2.2	30	30
2.3	40	40
3.0	60	20-60
3.1	70	20-70



注意 除 VxVM 3.0 外，其它所有 VxVM 发行版在导入时，都执行所支持的磁盘组版本的升级。

在 VxVM 3.0 系统上导入先前版本的磁盘组将阻止使用自那个版本发行以来所引入的功能。表 3 总结了特定的磁盘组版本不支持的功能，如下所示：

表 3. 不支持的磁盘组功能

磁盘组版本	不支持的功能
40	
30	热重定位
20	VxSmartSync Recovery Accelerator
10, 15	RAID-5 卷、新式条带、恢复检查点、磁盘组配置复制限制、Dirty 区日志、镜像卷日志

通过使用下列命令指定磁盘组名称，可获得磁盘组版本列表：

```
# vxdg list dgname
```

通过使用带 -l 格式选项的 vxprint(1M) 命令，可确定磁盘组版本。

VxVM 将磁盘组升级到当前正在运行的 VxVM 发行版所支持的最高版本。若要升级磁盘组，请使用下列命令：

```
# vxdg upgrade dgname
```

默认情况下，VxVM 创建 VxVM 发行版支持的最高版本的磁盘组。例如，VxVM 3.0 创建 60 版的磁盘组。

可能有必要创建较低版本的磁盘组。在运行 VxVM 3.0 版的系统上，以默认方式创建的磁盘组是 60 版的磁盘组。这在运行 VxVM 2.5 版的系统上是不可导入的，因为此系统只支持到版本 50。因此，若要在运行 VxVM 3.0 版的系统上创建可由运行 VxVM 2.5 版的系统导入的磁盘组，必须用 40 版或更低版本创建该磁盘组。

若要创建这种磁盘组，应在 vxdg init 命令中指定 -T *version* 选项。*version* 应为磁盘组所需的磁盘组版本。例如，若要创建可由运行 VxVM 2.5 的系统导入的磁盘组，请使用下列命令：

```
# vxdg -T 40 init newdg newdg01=c0t3d0s2
```

创建磁盘组版本为 40 的磁盘组 newdg。该磁盘组可由 VxVM 2.5 版导入。注意，尽管可在 VxVM 2.5 系统上导入该磁盘组，但使用 VxVM 3.0 功能的尝试将失败。

删除磁盘组

若要删除磁盘组，应卸载并停止磁盘组中的所有卷，然后使用下列命令：



```
# vxdg deport diskgroup
```

逐出磁盘组实际上并未将其删除，而只是禁止系统使用该磁盘组。但是，被逐出的磁盘组中的磁盘仍可以重用、重新初始化或添加到其它磁盘组中。

在系统间移动磁盘组

能在系统间移动磁盘组是磁盘组的一个重要特性。如果将磁盘组中的所有磁盘从一个系统移动到另一个系统，则该磁盘组可由第二个系统使用。无需重新指定配置。

使用下列步骤可以在系统间移动磁盘组：

1. 在第一个系统上，停止磁盘组中的所有卷，然后用下列命令逐出（禁止本地存取）该磁盘组：

```
# vxdg deport diskgroup
```

2. 将所有磁盘移动到第二个系统中，并执行必要（与系统相关）的步骤，使第二个系统和 Volume Manager 识别新的磁盘。

该过程可能要求重新启动，重新启动时，vxconfigd 后台驻留程序重新启动并识别新磁盘。如果不重新启动，请使用命令 vxctl enable 重新启动 vxconfigd，这样 Volume Manager 也能识别新磁盘。

3. 用下列命令在第二个系统上导入（启用本地存取）磁盘组：

```
# vxdg import diskgroup
```

4. 在导入磁盘组后，用下列命令启动磁盘组上的所有卷：

```
# vxrecover -g diskgroup -sb
```

可以从已崩溃的系统中移出磁盘。这种情况下，不能将磁盘组从第一个系统中逐出。当在系统上创建或导入磁盘组时，系统会锁定磁盘组中的所有磁盘。

注意 锁定的目的是确保两个系统不会同时使用 *双端口磁盘*（两个系统可同时存取的磁盘）。如果两个系统试图同时管理相同的磁盘，将破坏存储在磁盘上的配置信息。磁盘及其所包含的数据将不可用。

如果从系统移出磁盘时，系统已崩溃或在磁盘移动前未能检测到磁盘组，那么存储在磁盘上的锁仍保留着，必须清除这些锁。系统返回下列出错讯息：

```
vxdg: 磁盘组 groupname: 导入失败: 磁盘正被另一主机使用
```

若要清除一套指定设备上的锁，请使用下列命令：

```
# vxdisk clearimport devicename ...
```



使用下列命令可以在导入期间清除锁：

```
# vxdg -C import diskgroup
```

注意 如果系统上确实包含双端口磁盘，在使用命令 `vxdisk clearimport` 或 `vxdg -C import` 时必须特别小心。清除锁将使多个主机可以同时存取这些磁盘，并可能导致破坏数据。

有些时候，可能需要在某些磁盘不可用时导入磁盘组。如果在挂接到系统上的磁盘驱动器中找不到磁盘组的某些磁盘，则导入操作通常会失败。如果导入操作失败，将显示下列任一出错讯息：

```
vxdg: 磁盘组 groupname: 导入失败：磁盘组没有有效的配置副本
```

此信息表明有致命错误发生，需要修复硬件或创建新磁盘组。

```
vxdg: 磁盘组 groupname: 导入失败：未找到用于磁盘组的磁盘
```

此信息表明发生了无法恢复的错误。

如果磁盘组中有部分磁盘发生故障，可以使用下列命令强制导入该磁盘组：

```
# vxdg -f import diskgroup
```

注意 在使用 `-f` 选项时必须小心。否则会导致从不同的磁盘集中两次导入同一个磁盘组，由此使该磁盘组前后不一致。

可使用 `vxdiskadm` 执行这些操作。若要使用 `vxdiskadm` 逐出磁盘组，请选择菜单项 9（“禁止访问（逐出）磁盘组”）。若要导入磁盘组，请选择项 8（“允许访问（导入）磁盘组”）。`vxdiskadm` 导入操作检查主机导入锁并提示您查看是否有要清除的查找结果。它还启动磁盘组中的卷。

重命名磁盘组

每个系统只能有一个特定名称的磁盘组。不能导入或逐出在目标系统中已经有同名磁盘组的磁盘组。为避免发生此问题，Volume Manager 允许在导入或逐出过程中重命名磁盘组。

例如，由于每个运行 Volume Manager 的系统都只能有一个 `rootdg` 默认磁盘组，如何在系统上导入或逐出 `rootdg` 是一个值得注意的问题。同一个系统上不能有两个 `rootdg` 磁盘组。为避免此问题，可以在导入或逐出过程中重命名 `rootdg` 磁盘组。

若要在导入过程中给磁盘组起一个新名称，请使用下列命令：

```
# vxdg [-t] -n newdg_name import diskgroup
```



如果包括 `-t` 选项，导入将是临时的，经过重新启动后即不存在。这种情况下，磁盘组的存储名称在其原始主机上保留不变，但导入主机将该磁盘组认作 `newdg_name`。如果未使用 `-t` 选项，则名称更改将是永久性的。

也可以使用下列命令在逐出过程中重命名磁盘组：

```
# vxdbg [-h hostname] -n newdg_name deport diskgroup
```

当在逐出过程中进行重命名时，可以指定 `-h hostname` 选项将锁分配给替换主机。这样可以确保当替换主机重新启动时自动导入磁盘组。

若要临时将 `rootdg` 磁盘组从一个主机移动到另一个主机上（例如，用于根卷上的修复工作），然后再将其移回，请使用下列过程：

1. 在原始主机上，使用下列命令标识将导入到其它主机的 `rootdg` 磁盘组的磁盘组 ID：

```
# vxdisk -s list
```

输出显示的磁盘组信息与以下实例类似：

```
dgname:rootdg
dgid:774226267.1025.tweety
```

2. 在导入主机上，使用下列命令导入并重命名 `rootdg` 磁盘组：

```
# vxdbg -tC -n newdg_name import diskgroup
```

这里，`-t` 表示临时导入名称；`-C` 用于清除导入锁；`-n` 为将要导入的 `rootdg` 指定临时名称（以便不与现有的 `rootdg` 冲突）；`diskgroup` 是将要导入的磁盘组的磁盘组 ID（例如，`774226267.1025.tweety`）。

如果此时发生重新启动或系统崩溃，临时导入的磁盘组将变为未导入，需要重新导入。

3. 如果已在导入的 `rootdg` 上完成必要的工作，请使用下列命令将它逐回原始主机：

```
# vxdbg -h hostname deport diskgroup
```

这里，`hostname` 是返回其 `rootdg` 的系统的名称（可用命令 `uname -n` 确认系统名称）。

此命令将导入的 `rootdg` 从导入主机中删除，并将锁返回其原始主机。于是，原始主机自动在下次重新启动时导入它的 `rootdg`。



保留磁盘组的次要编号

Volume Manager 允许为指定的磁盘组选择某个范围内的次要编号。在创建卷的过程中使用这个范围的次要编号。这样将确保每个卷在经过重新启动或重新配置后具有相同的次要编号。如果两个磁盘组有重叠区域，将检测到导入冲突，因此需要有避免这种情况或重编号的机制。

如果为每个磁盘组分配不同范围的卷设备编号，则可以移动一组机器中的所有磁盘组，而不会导致设备编号冲突。

若要为磁盘组设置基卷设备次要编号，请使用下列命令：

```
# vxdg init diskgroup minor=base_minor devicename
```

在为磁盘组选择卷设备编号时，次要编号从 *base_minor* 数开始。次要编号（在大多数系统上）可一直延伸到 131071。最后将保留一个合理大小的范围用于临时设备编号重映射（以防两个设备编号仍有冲突）。

如果不在 `vxdg init` 命令行中指定 `minor` 操作数，Volume Manager 将选择随机数。选择的数最小为 1000，或是 1000 的倍数，并生成由 1000 个设备编号组成的可用范围。选择的默认数在由任何当前导入的 1000 个磁盘组组成的范围内不重叠。也不与任何当前已分配的卷设备编号重叠。

注意 默认策略确保少量磁盘组能在一组机器间成功合并。但是，对于使用故障转移机制自动合并磁盘组的情况，应选择避免重叠的范围。

有关保留次要编号的进一步信息，请参见 `vxdg(1M)` 帮助手册。

破坏磁盘组

`vxdg` 命令提供一个破坏选项，可以将磁盘组从系统中删除，并释放磁盘组中的磁盘供其它磁盘组使用。使用下列命令可删除不再需要的磁盘组，以便其它磁盘组可使用这些磁盘：

```
# vxdg destroy diskgroup
```

还可以使用 `vxdg deport` 命令使磁盘不可存取。Volume Manager 禁止其它磁盘组使用从磁盘组中逐出的磁盘。

使用特殊设备

本节描述 Volume Manager 完成管理任务所使用的特殊设备。

使用 vxdisk 实现特殊封装

*封装*是指将指定磁盘上的现有分区转换为卷的过程。如果任何分区中包含文件系统，则将修改 `/etc/vfstab` 项，使文件系统改为装载到卷上。

磁盘封装要求磁盘上有用于存储 Volume Manager 标识和配置信息的空闲空间。此空闲空间不能包括在其它任何分区中。（有关更多信息，请参见《*VERITAS Volume Manager 安装指南*》和 `vxencap(1M)` 帮助手册。）

可以封装那些没有可用于 Volume Manager 专用区域分区的空间的磁盘。 `vxdisk` 实用程序封装没有可用空间的磁盘。通过使用特殊类型的磁盘设备完成封装，这种设备称为 `nopriv` 设备，没有专用区域。

若要使用 `vxdisk`，请在磁盘设备上创建一个分区，而该设备映射您想存取的磁盘的所有部分。然后使用下列命令添加该分区的分区设备：

```
# vxdisk define partition-device type=nopriv
```

这里， `partition-device` 是 `/dev/dsk` 目录中的设备基名。例如，若要使用磁盘设备 `c0t4d0` 的分区 3，请使用下列命令：

```
# vxdisk define c0t4d0s3 type=nopriv
```

通过以下步骤为磁盘驱动器上的其它分区创建卷：

- ◆ 将设备添加到磁盘组
- ◆ 确定那些分区在封装分区内的位置
- ◆ 使用 `vxassist` 创建具有所需偏移和长度的卷

默认情况下， `vxassist` 重新初始化所创建的卷上的数据区。如果此分区上有要保留的数据，请不要使用 `vxassist`。用 `vxmake` 创建卷，并用 `vxvol init active` 启动卷。

使用 `nopriv` 设备的一个缺点是 Volume Manager 无法跟踪地址或磁盘控制器上的更改。正常情况下， Volume Manager 使用存储在物理磁盘上专用区域中的标识信息跟踪物理磁盘位置的更改。因为 `nopriv` 设备没有专用区域，并且物理磁盘上没有存储标识信息，所以不能跟踪更改。

特殊封装分区设备的最佳用途是封装磁盘，以便可以使用 Volume Manager 将空间移出磁盘。当磁盘上有可用空间时，可删除特殊分区设备，然后将磁盘封装为标准磁盘设备。

磁盘组不能完全由 `nopriv` 设备构成。这是因为 `nopriv` 设备不提供存储磁盘组配置信息的空间。配置信息必须存储在磁盘组中的至少一个磁盘上。



使用 vxdisk 实现 RAM 磁盘

注意 本节只适用于装有 RAM 磁盘的系统。

一些系统支持 RAM 磁盘的创建。RAM 磁盘是创建在系统 RAM 中的设备，就像一个小磁盘设备。通常，当系统重新启动时，将删除 RAM 磁盘上的内容。在重新启动中被删除内容的 RAM 磁盘会妨碍 Volume Manager 标识物理磁盘。这是因为存储在物理磁盘上的信息（在重新启动中被删除）用于标识磁盘。

nopriv 设备有一个支持 RAM 磁盘的特殊功能：*volatile* 选项向 Volume Manager 指出设备中的内容在重新启动时不再存在。易失性设备在系统启动时将接受特殊处理。如果卷已镜像，始终可通过从非易失性 plex 中复制数据来恢复从易失性设备创建的 plex。

若要使用 RAM 磁盘，必须为 `/dev/dsk` 和 `/dev/rdisk` 目录中的磁盘创建设备节点（例如，`/dev/dsk/ramd0` 和 `/dev/rdisk/ramd0`）。若要定义 Volume Manager 的 RAM 磁盘设备，请使用下列命令：

```
# vxdisk define ramd0 type=nopriv volatile
```

正常情况下，Volume Manager 不启动完全由包含易失性子磁盘的 plex 构成的卷。这是因为任何 plex 都不能确保包含最新的卷内容。

如果要在重新启动后重新创建所有卷的内容，可以使用一些 RAM 磁盘。在这些情况中，通过使用下列命令，可以强制在重新启动时启动从 RAM 磁盘形成的卷：

```
# vxvol set startopts=norecov volume_name
```

此选项只能用于 gen 类型的卷。有关 vxvol set 操作和 norecov 选项的更多信息，请参见 vxvol(1M)。

使用 vxdisk 显示多路径信息

注意 本节只适用于支持动态多路径 (DMP) 功能的系统。

在 Volume Manager 中，与系统相连的物理磁盘表示为有一条或多条物理存取路径的元设备。存取路径取决于磁盘是单个磁盘还是与系统相连的多端口磁盘阵列的组成部分。使用 vxdisk 实用程序显示元设备路径，并显示每条路径的状态（例如，启用或禁止）。例如，若要显示名为 disk01 磁盘的详细资料，请使用下列命令：

```
# vxdisk list disk01
```

Volume Manager 返回下列显示：

```
Device      c2t0d0s2
devicetag   c2t0d0
type        sliced
hostid      aparajita
disk        name=disk01 id=861086917.1052.aparajita
```



```

group      name=rootdg id=861086912.1025.aparajita
flags      online ready autoconfig autoimport imported
pubpaths   block=/dev/vx/dmp/c2t0d0s4 char=/dev/vx/rdmp/c2t0d0s4
privpaths  block=/dev/vx/dmp/c2t0d0s3 char=/dev/vx/rdmp/c2t0d0s3
version    2.1
iosize     min=512 (bytes) max=2048 (blocks)
public     slice=4 offset=0 len=1043840
private    slice=3 offset=1 len=1119
update     time=861801175 seqno=0.48
headers    0 248
configs    count=1 len=795
logs       count=1 len=120
Defined regions
config     priv 000017-000247[000231]:copy=01 offset=000000 enabled
config     priv 000249-000812[000564]:copy=01 offset=000231 enabled
log        priv 000813-000932[000120]:copy=01 offset=000000 enabled
Multipathing information:
numpaths:      2
c2t0d0s2       active
c1t0d0s2       failed

```

以上信息显示由元设备 `c2t0d0s2` 表示的物理设备的两条路径。路径 `c2t0d0s2` 为活动状态，而另一条路径 `c1t0d0s2` 为失败状态。

vxdiskadm 菜单界面任务

本节描述由菜单驱动界面，即所谓的 Volume Manager 支持操作（或 `vxdiskadm`）。`vxdiskadm` 用于执行物理和逻辑设备管理。

本节提供有关如何使用 `vxdiskadm` 执行磁盘和卷任务的下列信息：

- ◆ 将物理磁盘置于 Volume Manager 控制之下
- ◆ 添加磁盘供以后使用
- ◆ 重新初始化磁盘供 Volume Manager 使用
- ◆ 将 VM 磁盘添加到热重定位池
- ◆ 将 VM 磁盘从热重定位池中删除
- ◆ 显示物理磁盘信息
- ◆ 删除物理磁盘
- ◆ 禁用物理磁盘（使物理磁盘脱机）
- ◆ 启用物理磁盘
- ◆ 更换物理磁盘



- ◆ 更换故障磁盘或已删除的磁盘
- ◆ 向磁盘组添加磁盘
- ◆ 创建磁盘组
- ◆ 逐出磁盘组
- ◆ 导入磁盘组

启动 vxdiskadm

若要启动 vxdiskadm，请使用下列命令：

```
# vxdiskadm
```

将显示 vxdiskadm 主菜单。

vxdiskadm 主菜单

vxdiskadm 主菜单如下所示：

```
Volume Manager Support Operations  
菜单: VolumeManager/Disk
```

```
1      添加或初始化一个或多个磁盘  
2      封装一个或多个磁盘  
3      删除磁盘  
4      为替换而删除磁盘  
5      替换一个出错的或已删除的磁盘  
6      镜像磁盘上的卷  
7      从磁盘中移动卷  
8      允许访问（导入）磁盘组  
9      禁止访问（逐出）磁盘组  
10     启用（联机）磁盘设备  
11     禁用（脱机）磁盘设备  
12     将磁盘标记为磁盘组的备件  
13     关闭磁盘上的备用标志  
14     将子磁盘解除重定位至原磁盘  
15     从热重定位使用对象中排除磁盘  
16     使磁盘可供热重定位使用  
list   磁盘信息列表  
  
?      显示关于菜单的帮助信息  
??     显示关于菜单系统的帮助信息  
q      退出菜单
```

选择要执行的操作：

- ◆ 可以随时输入 ? 以获得有关使用菜单的帮助。? 的输出列出操作及每个操作的定义。
- ◆ ?? 列出可在任何提示下使用的输入。
- ◆ 如果需要重新启动进程，q 将返回主菜单；但是在主菜单级使用 q 将退出 Volume Manager 支持操作。

vxdiskadm 菜单提供对下列任务的访问。下面的编号与主菜单中列出的项相对应：

1. 添加或初始化一个或多个磁盘。

可以向系统添加格式化磁盘。SCSI 磁盘已经格式化。有关其它磁盘的信息，请参见生产商文档中关于格式化的说明。系统将提示您输入磁盘设备。可以指定向其中添加磁盘的磁盘组；如果选定 none，则磁盘保留为备用磁盘，将来执行操作或更换磁盘时不必初始化即可使用该磁盘。也可以指定将选定的磁盘标记为磁盘组的热重定位备用磁盘。如果该磁盘尚未初始化，则将分区并初始化供 Volume Manager 使用。

2. 封装一个或多个磁盘。

可以在安装 Volume Manager 之前就已添加到系统中的磁盘置于 Volume Manager 控制之下。系统提示您输入磁盘设备、磁盘组和磁盘名称；将磁盘添加到指定的磁盘组；检查磁盘，查找用于文件系统或其它目的的分區；创建卷以替换磁盘分区，以此作为存取现有数据的方式。如果封装不能自动确定分区的作用，将询问您使用此分区的目的。您可以选择用卷替换分区，保留分区不变，或删除分区。

如果已装载的文件系统或正在运行的应用程序正在使用任何分区，则必须重新启动系统。可能必须修改应用程序配置文件，以使用卷（而不是直接使用磁盘设备）存取磁盘分区。自动调整文件系统装载信息。

3. 删除磁盘。

可以从磁盘组中删除磁盘。系统将提示您要删除的磁盘名称。如果有任何卷使用磁盘上的存储空间，则不能删除该磁盘。如果有任何卷正在使用磁盘上的存储空间，可以选择让 Volume Manager 将此存储空间移到磁盘组中的其它磁盘上。

注意 不能使用此任务删除磁盘组中的最后一个磁盘。如果希望使用磁盘组中的所有剩余磁盘完成某项任务，应禁止（逐出）该磁盘组。然后就可以重新使用这些磁盘。但是，无法逐出 rootdg。

4. 为替换而删除磁盘。

可以在保留磁盘名称的同时将物理磁盘从磁盘组中删除。这将使命名磁盘的状态改为 removed。如果任何初始化磁盘不是磁盘组的组成部分，则可以在其中选择一个磁盘作为更换磁盘。

5. 替换一个出错的或已删除的磁盘。



对于使用“为替换而删除磁盘”菜单项删除的磁盘，可以为其指定一个更换磁盘，也可以为在使用过程中发生故障的磁盘指定更换磁盘。系统将提示您要更换的磁盘名称和用作更换的磁盘设备。可以选择未初始化的磁盘，这种情况下将初始化该磁盘，也可以选择已用“添加或初始化磁盘”菜单操作初始化的磁盘。

6. 镜像磁盘上的卷

可以镜像磁盘上的卷。可将这些卷镜像到具有可用空间的其它磁盘上。用这种方式创建卷的镜像副本可以在磁盘发生故障时防止数据丢失。不能使用此任务镜像已镜像的卷，或由多个子磁盘组成的卷。通过从引导磁盘镜像卷，可以生成用作替换引导磁盘的磁盘。

7. 从磁盘中移动卷。

可以将使用磁盘的任何卷（或卷的一部分）移动到其它磁盘上。在使用该菜单任务后，紧接着永久性地删除磁盘或为了更换而删除磁盘。

注意 如果只将卷移出磁盘，而没有删除磁盘，则不能防止以后的操作将其它卷移动到该磁盘上。

8. 允许访问（导入）磁盘组。

可以启用该系统对磁盘组的存取。如果希望将磁盘组从一个系统移动到另一个，必须首先在原系统上禁止（逐出）该磁盘组。然后，将被逐出的磁盘组中的磁盘移动到另一个系统，并在该系统上启用（导入）该磁盘组。系统将提示您输入磁盘组名称。

9. 禁止访问（逐出）磁盘组。

可以禁止对当前由该系统启用（导入）的磁盘组的存取。如果想将磁盘组中的磁盘移动到其它系统上，需逐出该磁盘组。同样，如果想将磁盘组中所有剩余的磁盘用于某种新目的，也需逐出该磁盘组。

系统将提示您磁盘组的名称。还会询问您是否要禁用这些磁盘（使它们脱机）。对于某些系统上的可移动的磁盘设备，应在移动磁盘前禁止对磁盘的所有存取，这一点很重要。

10. 启用（联机）磁盘设备。

如果在正常系统操作过程中，将磁盘从一个系统移动到另一个，Volume Manager 不会自动识别该磁盘。使用此菜单任务让 Volume Manager 扫描该磁盘以识别它，并确定该磁盘是否是磁盘组的一部分。而且，对于由磁盘组逐出任务或磁盘设备禁用（脱机）操作禁用的磁盘，可以使用此任务重新启用对磁盘的存取。

11. 禁用（脱机）磁盘设备。

可以通过 **Volume Manager** 禁止对磁盘设备的所有存取。此任务只能应用于当前不在磁盘组中的磁盘。如果想从系统中移走磁盘但不重新启动，则使用此任务。

注意，某些系统不支持可在正常操作中从系统中移走的磁盘。在这些系统上，脱机操作没有什么用。

12. 将磁盘标记为磁盘组的备件。

可以将一个磁盘保留为自动更换磁盘（用于热重定位），以防磁盘组中的其它磁盘发生故障。

13. 关闭磁盘上的备用标志

可以将磁盘从用作备件的磁盘列表中删除，并将其空间返回到常规的可用空间池中。

14. 将子磁盘解除重定位至原磁盘。

VxVM 热重定位允许系统自动对子磁盘级别上冗余 VxVM 对象的 I/O 故障做出反应，并采取必要的操作使对象重新可用。此机制可以检测子磁盘上的 I/O 故障、重定位子磁盘并恢复与子磁盘关联的 **plex**。更换了磁盘后，**Volume Manager** 提供 **vxunreloc** 实用程序，该程序可用于将系统恢复到与发生磁盘故障前相同的配置。**vxunreloc** 允许将热重定位的子磁盘移回由于磁盘故障而被更换的磁盘上。

15. 从热重定位使用对象中排除磁盘。

使空闲池（非备用磁盘）中的磁盘不用于热重定位。

16. 使磁盘可供热重定位使用。

撤消第 15 步操作，并使空闲池（非备用磁盘）中的磁盘可用于热重定位。

当执行磁盘管理时，应识别 **设备名称** 和 **磁盘名称** 之间的差别，这一点很重要。

注意 您的系统所使用的 **设备名称** 可能与实例不同。有关设备名称的更多信息，请参见第 1 页的“[了解 Volume Manager](#)”一章。

设备名称（有时也称为 *devname* 或 **磁盘存取名称**）是磁盘的位置。设备名称的语法是 **c#b#t#d#s#**，这里：

- ◆ **c#** 是磁盘驱动器所挂接的控制器的编号。
- ◆ **b#** 是相应的总线编号（如果系统使用总线）。
- ◆ **t#** 是控制器上目标磁盘的编号。
- ◆ **d#** 是磁盘的编号。
- ◆ **s#** 是磁盘片的编号。



设备的全路径名是 `/dev/vx/dmp/devicename`。在本文档中，只列出设备名称，并采用 `/dev/vx/dmp`。设备名称的实例是 `c0t0d0s2`。

磁盘名称（有时也称为 *磁盘介质名称*）是磁盘的管理名称，如 `disk01`。如果没有指定磁盘名称，并且如果将磁盘添加到 `rootdg`，则该磁盘的名称默认为 `disk##`（这里的 `##` 是序列号）。否则，默认磁盘名称为 `groupname##`，这里的 *groupname* 是指向其中添加磁盘的磁盘组的名称。

将物理磁盘置于 Volume Manager 控制之下

当向运行 Volume Manager 的系统中添加磁盘时，需要将该磁盘置于 Volume Manager 控制之下，以便 Volume Manager 能控制该磁盘上的空间分配。

除非指定了其它磁盘组，否则 Volume Manager 将新磁盘放入默认磁盘组 `rootdg`。本章稍后将提供关于创建其它磁盘组的说明。

采用何种方法将磁盘置于 Volume Manager 控制下取决于具体的情况：

- ◆ 如果是新磁盘，需要先将其初始化，再置于 Volume Manager 控制之下（参见第 97 页的“[将一个磁盘置于 Volume Manager 控制之下](#)”）。
- ◆ 如果不是立即需要该磁盘，可以先将其初始化（但不添加到磁盘组中），然后保留起来供以后使用（参见第 105 页的“[添加磁盘供以后使用](#)”）。
- ◆ 如果以前已初始化该磁盘供 Volume Manager 使用，可以将其重新初始化并置于 Volume Manager 控制之下（参见第 105 页的“[重新初始化磁盘供 Volume Manager 使用](#)”）。
- ◆ 如果以前使用过该磁盘，但当时未受 Volume Manager 的控制，可能希望保留该磁盘上的现有数据，同时仍然让 Volume Manager 控制该磁盘。使用封装可实现此目的（参见第 103 页的“[封装磁盘供 Volume Manager 使用](#)”）。
- ◆ 可以同时将一个或多个控制器上的多个磁盘置于 Volume Manager 控制之下。根据具体的情况，不一定以同一种方式处理所有的磁盘（参见第 99 页的“[将多个磁盘置于 Volume Manager 控制之下](#)”）。

当同时初始化或封装多个磁盘时，可以排除特定的磁盘或控制器。若要排除磁盘，在初始化或封装之前，在文件 `/etc/vx/disks.exclude` 中列出要排除的磁盘名称。同样，通过在文件 `/etc/vx/cntrls.exclude` 中列出要排除的控制器，可以将那些控制器上的所有磁盘排除在初始化或封装操作外。

后面的章节提供的详细实例说明如何使用 `vxdiskadm` 以各种方式和在各种情况下将磁盘置于 Volume Manager 控制之下。

注意 磁盘必须格式化（例如，使用 `format` 命令）或添加到系统（使用 `diskadd`）后才能置于 Volume Manager 控制之下。如果尝试通过 `vxdiskadm` 命令将未格式化的磁盘置于 Volume Manager 控制之下，初始化过程仍会照常开始，但不久就会退出，同时显示一条信息，告诉您该磁盘似乎无效，可能未格式化。如果发生这种情况，需要正确地格式化磁盘，然后尝试将该磁盘重新置于 Volume Manager 控制之下。

将一个磁盘置于 Volume Manager 控制之下

本节描述如何将格式化的磁盘置于 Volume Manager 控制之下。磁盘既可以是新的，也可以是以前在不受 Volume Manager 控制的情况下使用过的磁盘。

注意 初始化不保留磁盘上的数据。

按如下所示，初始化一个磁盘供 Volume Manager 使用：

1. 从 vxdiskadm 主菜单中选择菜单项 1 (“添加或初始化一个或多个磁盘”)。
2. 在下列提示下，输入将添加到 Volume Manager 控制中的磁盘的磁盘设备名称（或输入 **list** 以显示磁盘列表）：

添加或初始化磁盘

菜单：VolumeManager/Disk/AddDisks

使用此操作添加一个或多个磁盘至磁盘组。您可以将选定的磁盘添加至现有磁盘组或新磁盘组，创建新磁盘组将作为此操作的一部分。选定的磁盘也可以添加至磁盘组作为备件。也可以不将选定的磁盘添加至磁盘组而进行初始化，以留作替换磁盘使用。

在此提示下可输入多个磁盘或磁盘组合。这里是某些磁盘选择的示例：

```
all:                全部磁盘
c3 c4t2:           控制器 3 及 4 的目标 2 上的全部磁盘
c3t4d0:           单个磁盘
```

选择要添加的磁盘设备：

```
[<pattern-list>,all,list,q,?] list
```

<pattern-list> 既可以是单个磁盘，也可以是一系列磁盘和 / 或控制器（带可选目标）。如果 <pattern-list> 由多个项组成，必须用空格将这些项彼此分开。

如果在提示下输入 **list**，则 vxdiskadm 将显示可由系统使用的磁盘列表，应在紧接着的提示下输入将要添加的磁盘的设备名称：

设备	磁盘	组	状态
c0t0d0	disk01	rootdg	联机
c0t1d0	disk02	rootdg	联机
c0t2d0	disk03	rootdg	联机
c0t3d0	-	-	联机
c1t0d0	disk10	rootdg	联机
c1t0d1	-	-	错误
.			
.			
.			
c3t0d0	-	-	错误
c3t1d0	disk33	rootdg	联机
c3t2d0	disk34	rootdg	联机
c3t3d0	disk35	rootdg	联机



选择要添加的磁盘设备:

```
[<pattern-list>,all,list,q,?] c1t0d1
```

所有挂接在系统上的磁盘都可由 Volume Manager 识别并且显示在这里。

“状态”列中的词“错误”告诉您还有磁盘未添加到 Volume Manager 的控制中。这些磁盘以前可能已初始化，也可能未初始化。与磁盘名称和磁盘组一起列出的磁盘不能用于此任务，因为它们已经受 Volume Manager 的控制。

- 若要继续此操作，请在下列提示下输入 **y**（或按回车键）:

这里是所选磁盘。输出格式: [设备名]

```
c1t2d0
```

继续进行操作吗? [y,n,q,?] (默认: y) **y**

- 在下列提示下，指定将向其中添加磁盘的磁盘组或按回车键接受 rootdg:

您可以选择将此磁盘添加至现有磁盘组、新磁盘组或留作将来用于添加或替换操作。若要创建新磁盘组，需选择一个尚未存在的磁盘组名称。若要留作将来使用，指定“none”磁盘组名称。

哪一个磁盘组 [<group>,none,list,q,?] (默认: rootdg)

- 在下列提示下，按回车键接受默认磁盘名称或输入磁盘名称:

将默认磁盘名用于此磁盘吗? [y,n,q,?] (默认: y)

- 当 vxdiskadm 询问该磁盘是否用作热重定位备用磁盘时，输入 **n**（或按回车键）:

添加磁盘作为 rootdg 的备用磁盘吗? [y,n,q,?] (默认: n) **n**

- 当 vxdiskadm 询问该磁盘是否不用于热重定位时，输入 **n**（或按回车键）:

从热重定位使用对象中排除磁盘吗? [y,n,q,?] (默认: n)

- 若要继续此操作，请在下列提示下输入 **y**（或按回车键）:

使用默认磁盘名将选定的磁盘添加至磁盘组 rootdg。

```
c1t2d0
```

继续进行操作吗? [y,n,q,?] (默认: y) **y**

9. 如果确信磁盘上没有数据，输入 **n** 免去封装。当 `vxdiskadm` 询问是否希望初始化该磁盘时，输入 **y**：

下列磁盘设备具有有效的 vTOC，但似乎并未初始化供 Volume Manager 使用。如果磁盘上有“不”能破坏的数据，则应将现有磁盘分区封装为卷，而不是将其添加为新磁盘。

输出格式：[设备名]

```
c1t2d0
```

```
封装该设备吗? [y,n,q,?] (默认: y) n
```

```
c1t2d0
```

```
进行初始化, 而不是封装? [y,n,q,?] (默认: n) y
```

现在出现类似于下面这样的信息，确认正在将磁盘 `c1t2d0` 置于 Volume Manager 的控制之下。在某些系统上，还提供了执行表面分析的选项。

```
初始化设备 c1t2d0。
```

```
执行表面分析 (极力推荐) [y,n,q,?] (默认: y) n
```

```
用磁盘名 disk39 将磁盘设备 c1t2d0 添加至磁盘组 rootdg。
```

10. 在下列提示下，指出是希望继续初始化更多磁盘 (**y**) 还是返回 `vxdiskadm` 主菜单 (**n**)：
- ```
添加或初始化其它磁盘吗? [y,n,q,?] (默认: n)
```

### 将多个磁盘置于 Volume Manager 控制之下

本节描述如何将多个磁盘同时置于 Volume Manager 控制之下。磁盘集可以包括系统上的所有磁盘、控制器上的所有磁盘、选定的磁盘或前述这些磁盘的组合。

根据具体的情况，处理所有磁盘的方式可能不同。例如，可以初始化某些磁盘，而封装其它磁盘。

---

**注意** 初始化不保留磁盘上的数据。

---

当同时初始化或封装多个磁盘时，可以排除特定的磁盘或控制器。若要排除磁盘，在初始化或封装之前，在文件 `/etc/vx/disks.exclude` 中列出要排除的磁盘名称。通过在文件 `/etc/vx/cntrls.exclude` 中列出要排除的控制器，可以将那些控制器上的所有磁盘排除在初始化或封装操作外。

按如下操作，将多个磁盘同时置于 Volume Manager 控制之下：

1. 从主菜单 `vxdiskadm` 中选择菜单项 1 (“添加或初始化一个或多个磁盘”)。



- 在下列提示下，输入要添加到 Volume Manager 控制中的磁盘的 *pattern-list*。在本例中，输入 **c3** 表示控制器 3 上的所有磁盘：

添加或初始化磁盘

Menu:VolumeManager/Disk/AddDisks

使用此操作添加一个或多个磁盘至磁盘组。您可以将选定的磁盘添加至现有磁盘组或新磁盘组，创建新磁盘组将作为此操作的一部分。选定的磁盘也可以添加至磁盘组作为备件。也可以不将选定的磁盘添加至磁盘组而进行初始化，以留作替换磁盘使用。

在此提示下可输入多个磁盘或磁盘组合。这里是某些磁盘选择的示例：

```
all: 全部磁盘
c3 c4t2: 控制器 3 及 4 的目标 2 上的全部磁盘
c3t4d0: 单个磁盘
```

选择要添加的磁盘设备：

```
[<pattern-list>,all,list,q,?] c3
```

这里的 *<pattern-list>* 既可以是单个磁盘，也可以是一系列磁盘和 / 或控制器（带可选目标）。如果 *<pattern-list>* 由多项组成，必须用空格将这些项彼此分开。

如果不知道要添加的磁盘的地址（设备名称），在提示处输入 **list** 以获得可用磁盘的完整列表。

- 若要继续此操作，请在下列提示下输入 **y**（或按回车键）：

这里是所选磁盘。输出格式：[设备名]

```
c3t0d0 c3t1d0 c3t2d0 c3t3d0
```

```
继续进行操作吗？[y,n,q,?]（默认：y） y
```

- 若要将这些磁盘添加到默认磁盘组 **rootdg**，请在下列提示下输入 **y**（或按回车键）：

您可以选择将此磁盘添加至现有磁盘组、新磁盘组或留作将来用于添加或替换操作。若要创建新磁盘组，需选择一个尚未存在的磁盘组名称。若要留作将来使用，指定“none”磁盘组名称。

```
哪一个磁盘组 [<group>,none,list,q,?]（默认：rootdg） y
```

- 若要允许 vxdiskadm 对每个磁盘都使用默认磁盘名称，请在下列提示下输入 **y**（或按回车键）：

```
将默认磁盘名用于磁盘吗？[y,n,q,?]（默认：y） y
```

- 在下列提示下，输入 **n** 表示这些磁盘不用作热重定位备用磁盘：

```
添加磁盘作为 rootdg 的备用磁盘吗？[y,n,q,?]（默认：n） n
```



7. 当 vxdiskadm 询问该磁盘是否不用于热重定位时，输入 **n**（或按回车键）：

从热重定位使用对象中排除磁盘吗？ [y,n,q,?]（默认：n）

8. 若要继续此操作，请在下列提示下输入 **y**（或按回车键）：

使用默认磁盘名将选定的磁盘添加至磁盘组 rootdg。

c3t0d0 c3t1d0 c3t2d0 c3t3d0

继续进行操作吗？ [y,n,q,?]（默认：y） **y**

9. 下列提示列出了任何已初始化可供 Volume Manager 使用的磁盘；输入 **y** 以表示现在将使用所有这些磁盘：

下列磁盘设备似乎已经初始化。

这些磁盘当前可用作替换磁盘。

输出格式： [ 设备名 ]

c3t1d0 c3t2d0 c3t3d0

使用这些设备吗？ [Y,N,S(elect),q,?]（默认：Y） **y**

注意，该提示允许您就 *所有这些* 磁盘表明“是”或“否”（**y** 或 **n**），或选择如何基于每个磁盘进行处理（**s**）。

如果确认希望重新初始化所有磁盘，请在下列提示下输入 **y**：

下列您所选用的磁盘看来已经初始化供 Volume Manager 使用。如果您确信这些磁盘已经初始化供 Volume Manager 使用，则不需要重新初始化这些磁盘设备。

输出格式： [ 设备名 ]

c3t1d0 c3t2d0 c3t3d0

重新初始化这些设备吗？ [Y,N,S(elect),q,?]（默认：Y） **y**

10. vxdiskadm 现在指出控制器 3 上的一个磁盘是用于封装的磁盘；输入 **y**（或按回车键）封装该磁盘：

以下磁盘设备具有有效的 vTOC，但是尚未进行用于 Volume Manager 的初始化。如果该磁盘上存在不应破坏的数据，则应将现有磁盘分区封装为卷而不是将磁盘作为新磁盘添加。

输出格式： [ 设备名 ]

c3t0d0

封装该磁盘吗？ [y,n,q,?]（默认：y） **y**

封装允许将活动磁盘添加到 Volume Manager 控制中，并保留该磁盘上的数据。



**注意** 磁盘封装要求重新启动系统，并且可能要求几次后续重新启动。如果需要，将提示您进行这些操作。

vxdiskadm 现在显示出类似于下面这样的信息，确认这些磁盘正在初始化和添加到 Volume Manager 的控制中：

初始化设备 c3t1d0。

初始化设备 c3t2d0。

初始化设备 c3t3d0。

用磁盘名 disk33 将磁盘设备 c3t1d0 添加至磁盘组 rootdg。

用磁盘名 disk34 将磁盘设备 c3t2d0 添加至磁盘组 rootdg。

用磁盘名 disk35 将磁盘设备 c3t3d0 添加至磁盘组 rootdg。

除以上显示的输出外，您可能还会看见一些提示，使您可以选择执行表面分析。

**11.** vxdiskadm 然后确认任何被选定用于封装的磁盘，并提示您允许进行封装：输入 **y**（或按回车键）继续封装：

以下磁盘已被选定用于封装。

输出格式：[ 设备名 ]

c3t0d0

继续进行封装吗？[y,n,q,?]（默认：y） **y**

vxdiskadm 现在显示出封装状态，并告诉您必须尽快执行一次关闭和重新启动：

磁盘设备 c3t0d0 将被封装并用磁盘名 disk38 添加至磁盘组 rootdg。

封装的第一阶段已成功完成。

现在应尽可能早地重新启动系统。

封装将需要两至三次重新启动，这将在下一次重新启动后自动发生。若要重新启动，执行命令：

```
shutdown -g0 -y -i6
```

这将更新 /etc/vfstab 文件，以便将卷设备用于安装该磁盘设备上的文件系统。您将需要更新其它参考文件，如备份脚本、数据库或手动创建的交换设备。

**12.** 在下列提示下，指出是希望继续初始化更多磁盘 (**y**) 还是返回 vxdiskadm 主菜单 (**n**):

添加或初始化其它磁盘吗？[y,n,q,?]（默认：n）

## 封装磁盘供 Volume Manager 使用

本节描述如何封装磁盘供 Volume Manager 使用。当磁盘受 Volume Manager 控制时，封装可保留磁盘上的任何现有数据。

若要减小封装失败的可能性，请确保磁盘：

- ◆ 包含少量不属于任何分区的空闲空间（在磁盘的起始处或末尾）
- ◆ 有两个空闲分区
- ◆ 有一个表示整个磁盘的 s2（在某些系统上为 s0）片

在一些系统上，当封装引导（根）磁盘时，应将交换分区标记为 swap，以便稍后能转储到该分区中。

在封装引导磁盘前，如果想在需要更换时利用 Volume Manager 引导磁盘别名标识引导磁盘镜像，则必须将 EEPROM 变量 use-nvramrc? 设置为 true。如果将此变量设置为 false，您需要自己确定可引导的磁盘。可以按如下方式将此变量设置为 true：

```
eprom use-nvramrc?=true
```

若要封装磁盘供 Volume Manager 使用，请使用下列过程：

1. 从 vxdiskadm 主菜单中选择菜单项 2（“封装一个或多个磁盘”）。

---

**注意** 您的系统所使用的设备名称可能与实例不同。有关设备名称的更多信息，请参见第 1 页的“了解 Volume Manager”一章。

---

2. 在下列提示下，输入要封装的磁盘的磁盘设备名称：

封装一个或多个磁盘

菜单：VolumeManager/Disk/Encapsulate

使用此操作将一个或多个磁盘转换为使用 Volume Manager。它将磁盘添加至磁盘组并用卷替换现有分区。磁盘封装要求重新启动以使更改生效。

在此提示下可输入多个磁盘或磁盘组合。

这里是某些磁盘选择的示例：

```
all: 全部磁盘
c3 c4t2: 控制器 3 及 4 的目标 2 上的全部磁盘
c3t4d0: 单个磁盘
```

选择要封装的磁盘设备：

```
[<pattern-list>,all,list,q,?] c2t5d0
```

这里的 **<pattern-list>** 既可以是单个磁盘，也可以是一系列磁盘和 / 或控制器（带可选目标）。如果 **<pattern-list>** 由多项组成，必须用空格将这些项彼此分开。

如果不知道要封装的磁盘的地址（设备名称），在提示下输入 **l** 或 **list** 以获得可用磁盘的完整列表。



3. 若要继续此操作，请在下列提示下输入 **y**（或按回车键）：

这里是所选磁盘。输出格式：[ 设备名 ]

```
c2t5d0
```

继续进行操作吗？[y,n,q,?]（默认：y） **y**

4. 若要将磁盘添加到默认磁盘组 `rootdg`，请在下列提示下按回车键：

您可以选择将此磁盘添加至现有磁盘组或新磁盘组。若要创建新磁盘组，需选择一个尚未存在的磁盘名称。

哪一个磁盘组 [<group>,list,q,?]（默认：rootdg）

5. 在下列提示下，按回车键接受默认磁盘名称，或输入新的磁盘名称：

将默认磁盘名用于此磁盘吗？[y,n,q,?]（默认：y）

6. 若要继续此操作，请在下列提示下输入 **y**（或按回车键）：

使用默认磁盘名将选定的磁盘封装并添加至磁盘组 `rootdg`。

```
c2t5d0
```

继续进行操作吗？[y,n,q,?]（默认：y） **y**

7. 若要确认继续封装，请在下列提示下输入 **y**（或按回车键）：

以下磁盘已被选定用于封装。输出格式：[ 设备名 ]

```
c2t5d0
```

继续进行封装吗？[y,n,q,?]（默认：y） **y**

出现类似于下面这样的信息，确认正在封装磁盘供 Volume Manager 使用，并告诉您需要重新启动：

```
磁盘设备 c2t5d0 将被封装并用磁盘名 disk01 添加至磁盘组 rootdg。
```

```
c2t5d0 磁盘已经被配置用于封装操作。
```

封装的第一阶段已成功完成。现在应尽可能早地重新启动系统。

封装将需要两至三次重新启动，这将在下次重新启动后自动发生。若要重新启动，执行命令：

```
shutdown -g0 -y -i6
```

这将更新 `/etc/vfstab` 文件，以便将卷设备用于安装该磁盘设备上的文件系统。您将需要更新其它参考文件，如备份脚本、数据库或手动创建的交换设备。



8. 在下列提示下，指出是希望封装更多磁盘 (y) 还是返回 vxdiskadm 主菜单 (n):

封装其它磁盘吗? [y,n,q,?] (默认: n) **n**

在某些情况下，封装磁盘会失败。封装失败通常是因为磁盘上没有足够的可用空闲空间容纳 Volume Manager。如果发生这种情况，上述过程会突然终止，并显示一条类似于下面这样的出错讯息：

磁盘设备 c2t5d0 将被封装并用磁盘名 disk01 添加至磁盘组 rootdg。

封装操作因以下错误而失败：

由于下列原因，不可能封装 c2t5d0：

<vxvm:vxslicer: 错误: 不支持的磁盘布局。 >

击 RETURN 键继续。

## 添加磁盘供以后使用

如果希望将磁盘添加到 Volume Manager 控制中供以后使用，请遵循第 96 页的“[将物理磁盘置于 Volume Manager 控制之下](#)”中概述的步骤。但是，当您命名磁盘组时，输入 **none**，而不要选择 rootdg 或键入新的磁盘组名称。磁盘于是像以前那样初始化，但保留起来供以后使用。磁盘只有添加到磁盘组后才能使用。

---

**注意** 不要将这类“备用磁盘”同“热重定位备用磁盘”混淆。

---

## 重新初始化磁盘供 Volume Manager 使用

本节描述如何将以前初始化过的磁盘重新初始化供 Volume Manager 使用。

如果希望添加的磁盘以前使用过，但当时不受 Volume Manager 的控制，则使用下列任一过程：

- ◆ 封装磁盘并保留该磁盘上的信息（参见第 103 页的“[封装磁盘供 Volume Manager 使用](#)”）。
- ◆ 重新初始化磁盘使 Volume Manager 得以配置磁盘供 Volume Manager 使用。注意重新初始化不保留磁盘上的数据。如果希望重新初始化磁盘，确保该磁盘上不包含应保留的数据。



若要重新初始化磁盘供 Volume Manager 使用，请使用下列过程：

1. 从 vxdiskadm 主菜单中选择菜单项 1 (“添加或初始化一个或多个磁盘”)。
2. 在下列提示下，输入要添加到 Volume Manager 控制中的磁盘的磁盘设备名称：

添加或初始化磁盘

菜单：VolumeManager/Disk/AddDisks

使用此操作添加一个或多个磁盘至磁盘组。您可以将选定的磁盘添加至现有磁盘组或新磁盘组，创建新磁盘组将作为此操作的一部分。选定的磁盘也可以添加至磁盘组作为备件。也可以不将选定的磁盘添加至磁盘组而进行初始化，以留作替换磁盘使用。

在此提示下可输入多个磁盘或磁盘组合。这里是某些磁盘选择的示例：

```
all: 全部磁盘
c3 c4t2: 控制器 3 及 4 的目标 2 上的全部磁盘
c3t4d0: 单个磁盘
```

选择要添加的磁盘设备：

```
[<pattern-list>,all,list,q,?] c1t3d0
```

这里的 *<pattern-list>* 既可以是单个磁盘，也可以是一系列磁盘和 / 或控制器（带可选目标）。如果 *<pattern-list>* 由多项组成，必须用空格将这些项彼此分开。

如果不知道要添加的磁盘的地址（设备名称），在下列提示下输入 **l** 或 **list** 以获得可用磁盘的完整列表。

3. 若要继续此操作，请在下列提示下输入 **y**（或按回车键）：

这里是所选磁盘。输出格式：[ 设备名 ]

```
c1t3d0
```

```
继续进行操作吗？[y,n,q,?]（默认：y） y
```

4. 在下列提示下，指定将向其中添加磁盘的磁盘组，或按回车键接受 rootdg：

您可以选择将此磁盘添加至现有磁盘组、新磁盘组或留作将来用于添加或替换操作。若要创建新磁盘组，需选择一个尚未存在的磁盘组名称。若要留作将来使用，指定“none”磁盘组名称。

```
哪一个磁盘组 [<group>,none,list,q,?]（默认：rootdg）
```

5. 在下列提示下，按回车键接受默认磁盘名称，或输入新的磁盘名称：

```
将默认磁盘名用于此磁盘吗？[y,n,q,?]（默认：y）
```

6. 当 vxdiskadm 询问是否将该磁盘用作热重定位备用磁盘时，输入 **n**（或按回车键）：  
添加磁盘作为 rootdg 的备用磁盘吗？ [y,n,q,?]（默认：n） **n**
  
7. 当 vxdiskadm 询问是否将该磁盘从热重定位使用对象中排除时，输入 **n**（或按回车键）：  
从热重定位使用对象中排除磁盘吗？ [y,n,q,?]（默认：n）
  
8. 若要继续此操作，请在下列提示处输入 **y**（或按回车键）：  
使用默认磁盘名将选定的磁盘添加至磁盘组 rootdg。  
c1t3d0  
继续进行操作吗？ [y,n,q,?]（默认：y） **y**
  
9. 下列提示指出该磁盘以前曾初始化供 Volume Manager 使用；输入 **y** 确认现在希望使用该磁盘：  
以下设备似乎已初始化。该磁盘当前可用作替换磁盘。输出格式： [ 设备名 ]  
c1t3d0  
使用该设备吗？ [y,n,q,?]（默认：y） **y**
  
10. 若要重新初始化该磁盘，请在下列提示下输入 **y**（或按回车键）：  
下列您所选用的磁盘看来已经初始化供 Volume Manager 使用。如果您确信该磁盘已经初始化供 Volume Manager 使用，则不需要重新初始化该磁盘设备。  
输出格式： [ 设备名 ]  
c1t3d0  
重新初始化该设备吗？ [y,n,q,?]（默认：y） **y**  
现在出现类似于下面这样的信息，确认正在重新初始化该磁盘供 Volume Manager 使用：  
初始化设备 c1t3d0。  
用磁盘名 disk40 将磁盘设备 c1t3d0 添加至磁盘组 rootdg。
  
11. 在下列提示下，指出是希望继续初始化更多磁盘 (**y**) 还是返回 vxdiskadm 主菜单 (**n**)：  
添加或初始化其它磁盘吗？ [y,n,q,?]（默认：n）



## 将 VM 磁盘添加到热重定位池

通过将冗余子磁盘重定位到其它磁盘，热重定位使系统得以自动对 I/O 故障做出反应。之后，热重定位还原受影响的 Volume Manager 对象和数据。如果已将某磁盘指定为磁盘组中的备用磁盘，则将故障磁盘中的子磁盘重定位到该备用磁盘。否则，将使用磁盘组中任何合适的空闲空间，磁盘上以前排除在热重定位使用外的空闲空间除外。

若要将磁盘指定为备用磁盘，请完成下列操作：

1. 从主菜单 vxdiskadm 中选择菜单项 12 (“将磁盘标记为磁盘组的备件”)。

2. 在下列提示下，输入磁盘名称 (如 disk01)：

将磁盘标记为磁盘组的备件

菜单：VolumeManager/Disk/MarkSpareDisk

使用此操作将磁盘标记为磁盘组的备件。此操作取磁盘名作为输入。这是当添加该磁盘至此磁盘组时给予磁盘的名称。

输入磁盘名称 [`<disk>,list,q,?`] disk01

vxdiskadm 显示下列确认信息：

完成标记 rootdg 中的 disk01 为备用磁盘。

3. 在下列提示下，指出是希望添加更多的磁盘作为备用磁盘 (y) 还是返回 vxdiskadm 主菜单 (n)：

将另外的磁盘标记为备件吗？ [`y,n,q,?`] (默认：n)

当发生故障时，磁盘组中的任何 VM 磁盘现在都可以将该磁盘用作备用磁盘。如果磁盘发生故障，将自动进行热重定位 (如果可能)。系统将用电子邮件通知您所发生的故障和重定位。重定位成功后，可能需要更换故障磁盘。

当备用空间不足以重定位故障子磁盘时，热重定位自动使用空闲空间。通过指定禁止热重定位使用的空闲磁盘，用户可以限制由热重定位使用的空闲空间。

## 将 VM 磁盘从热重定位池中删除

尽管某磁盘被指定为备用磁盘，但该磁盘上的空间并不用作在磁盘组中创建 Volume Manager 对象的空闲空间。如果需要，可以通过将备用磁盘从热重定位磁盘池中删除的方式，释放备用磁盘供常规使用。

若要检验磁盘当前是否被指定为备用磁盘，请从 vxdiskadm 主菜单中选择菜单项 list。列出的备用磁盘都带 spare 标记。

若要将磁盘从热重定位池中删除，请完成下列操作：

1. 从 vxdiskadm 主菜单中选择菜单项 13 (“关闭磁盘上的备用标志”)。
2. 在下列提示下，输入备用磁盘名称 (如 disk01)：
 

关闭磁盘上的备用标志  
菜单：VolumeManager/Disk/UnmarkSpareDisk

使用此操作关闭磁盘上的备用标志。  
此操作取磁盘名作为输入。这是当添加该磁盘至此磁盘组时给予磁盘的名称。

输入磁盘名称 [`<disk>,list,q,?`] disk01

vxdiskadm 显示下列信息：

rootdg 中的磁盘 disk01 不再标记为备用磁盘。
3. 在下列提示下，指出是希望禁用更多的备用磁盘 (y) 还是返回 vxdiskadm 主菜单 (n)：
 

关闭另一磁盘的备用标志吗？ [`y,n,q,?`] (默认：n)

## 将热重定位的子磁盘移回磁盘

由于磁盘故障更换磁盘后，可以使用该选项将热重定位的子磁盘移回它们最初所驻留的磁盘上。

1. 从 vxdiskadm 主菜单中选择菜单项 14 (“将子磁盘解除重定位至原磁盘”)。
2. 该选项首先提示原始磁盘介质名称。
 

在下列提示下，输入热重定位的子磁盘最初所驻留的磁盘介质名称：

输入原始磁盘名称 [`<disk>,list,q,?`]

但是，如果系统中没有热重定位的子磁盘，vxdiskadm 将打印 “当前不存在热重定位的子磁盘” 并让您按回车键继续。
3. 然后该选项提示新的目标磁盘。
 

在解除子磁盘重定位时，可以选择将子磁盘移动到不同于原始磁盘的磁盘上。  
解除重定位至新磁盘 [`y,n,q,?`] (默认：n)
4. 如果不可能将子磁盘移动到原始偏移，也可以选择 “强制选项” 将子磁盘解除重定位到指定的磁盘，但不一定移动到与原来完全一样的偏移。
 

如果移动到准确偏移失败，使用 `-f` 选项解除子磁盘重定位吗？ [`y,n,q,?`] (默认：n)



5. 然后该选项确认所请求的操作，并询问是否希望继续此操作。

所请求的操作是将所有从 disk10 热重定位的子磁盘移回磁盘组 rootdg 的 disk10 磁盘。

继续进行操作吗？[y,n,q,?]（默认：y）

6. 最后，该选项告诉用户操作状态。

完成至磁盘 disk10 的解除重定位。

## 从热重定位使用对象中排除磁盘

若要从热重定位使用对象中排除磁盘：

1. 从 vxdiskadm 主菜单中选择菜单项 15（“从热重定位使用对象中排除磁盘”）。

2. 在下列提示下，输入磁盘名称（如 disk01）：

从热重定位使用对象中排除磁盘

菜单：VolumeManager/Disk/MarkNoHotUse

使用该操作将磁盘从热重定位使用对象中排除。该操作取磁盘名作为输入。这是当添加该磁盘到磁盘组时给予磁盘的名称。

输入磁盘名称 [<disk>,list,q,?] **disk01**

vxdiskadm 显示下列信息：

从热重定位使用对象中排除 rootdg 中的 disk01 已完成。

3. 在下列提示下，指出是希望使更多的磁盘不可用于热重定位 (y) 还是按回车键返回 vxdiskadm 主菜单 (n)：

从热重定位使用对象中排除另外的磁盘吗？[y,n,q,?]（默认：n）

## 使磁盘可供热重定位使用

如果以前磁盘从热重定位使用对象中排除，下列操作将撤消该设置并将磁盘重新添加到热重定位池：

若要使磁盘可供热重定位使用：

1. 从 vxdiskadm 中选择菜单项 16（“使磁盘可供热重定位使用”）。

使磁盘可供热重定位使用

2. 在下列提示下，输入磁盘名称（如 disk01）：

使磁盘可供热重定位使用

菜单：VolumeManager/Disk/UnmarkNoHotUse

使用此操作使磁盘可用于热重定位。这只适用于那些先前从热重定位使用对象中排除的磁盘。该操作取磁盘名称作为输入。这是当添加磁盘到磁盘组时给予磁盘的名称。

输入磁盘名称 [`<disk>,list,q,?`] **disk01**

vxdiskadm 显示下列信息:

完成使 rootdg 中的磁盘 disk01 可供热重定位使用的操作。

3. 在下列提示下, 指出是希望使更多的磁盘可用于热重定位 (**y**) 还是返回 vxdiskadm 主菜单 (**n**):

使另一磁盘可供热重定位使用吗? use?[`y,n,q,?`] (默认: n)

## 显示物理磁盘信息

通过显示磁盘信息, 可以提供已初始化的磁盘、磁盘所属的磁盘组以及磁盘状态的信息。list 命令显示所有识别的磁盘的设备名称、磁盘名称、与每个磁盘关联的磁盘组名称和每个磁盘的状态。

若要显示磁盘信息, 请完成下列操作:

1. 从 vxdiskadm 主菜单中选择 list。
2. 在下列显示中, 输入想查看的磁盘的地址, 或输入 **all** 以获得所有磁盘的列表:

磁盘信息列表

菜单: VolumeManager/Disk/ListDisk

使用此菜单操作显示磁盘列表。也可以选择列出指定磁盘设备地址上的磁盘的详细信息。

输入磁盘设备或 "all" [`<address>,all,q,?`] (默认: all)

如果您输入 **all**, Volume Manager 将显示下列信息 (某些系统可能显示 rootdisk 而不是 c0t0d0s2):

| 设备     | 磁盘       | 组      | 状态     |
|--------|----------|--------|--------|
| c0t0d0 | c0t0d0s2 | rootdg | online |
| c1t0d0 | disk01   | rootdg | online |
| c1t1d0 | -        | -      | online |

要详细列表的设备 [`<address>,none,q,?`] (默认: none)

- ◆ 如果输入希望获取其信息的设备的地址, 将显示完整的磁盘信息 (包括设备名称、磁盘类型以及磁盘的公共分区和专用分区信息)。

检查完此信息后, 按回车键返回主菜单。



## 删除物理磁盘

在删除磁盘前，确定它不包含任何数据，不再需要其上的数据，或者可以将数据移动到其它磁盘上。按如下方式删除磁盘：

1. 从 vxdiskadm 主菜单中选择菜单项 3 (“删除磁盘”)。

---

**注意** 必须禁用磁盘组后才能删除磁盘组中的最后一个磁盘。

---

2. 在下列提示下，输入要删除的磁盘的磁盘名称：

删除磁盘

菜单：VolumeManager/Disk/RemoveDisk

使用此操作从磁盘组中删除磁盘。此操作取磁盘名作为输入。这是当添加该磁盘至此磁盘组时给予磁盘的名称。

输入磁盘名称 [`<disk>,list,q,?`] **disk01**

3. 如果磁盘上存在卷，Volume Manager 将询问您是否将卷从磁盘中迁移出。如果希望保留卷，回答 **y**。否则，回答 **n**。

4. 在下列确认提示下，按回车键继续：

所请求的操作是将磁盘 `disk01` 从磁盘组 `rootdg` 中删除。

继续进行操作吗？ [`y,n,q,?`] (默认：`y`)

vxdiskadm 实用程序将磁盘从磁盘组中删除，并显示下列成功信息：

完成磁盘 `disk01` 的删除。

现在可以移走磁盘或将该磁盘保留在系统上作为更换磁盘。

5. 在下列提示下，指出是希望删除其它磁盘 (**y**) 还是返回 vxdiskadm 主菜单 (**n**)：

删除另外的磁盘吗？ [`y,n,q,?`] (默认：`n`)

## 禁用物理磁盘（使物理磁盘脱机）

如果磁盘被破坏，必须使其脱机，然后将其移走。可能要物理磁盘设备移动到与其它系统相连的其它位置。若要使磁盘脱机，首先将其从磁盘组中删除，然后完成下列操作：

1. 从 vxdiskadm 主菜单中选择菜单项 11 (“禁用（脱机）磁盘设备”)。



2. 在下列提示下，输入希望禁用的磁盘的地址：

禁用（脱机）磁盘设备

菜单：VolumeManager/Disk/OfflineDisk

使用此菜单操作禁止所有通过 Volume Manager 对磁盘设备的访问。此操作只适用于那些当前不在磁盘组中的磁盘。如果想要从系统移走磁盘而不重新启动，则使用此操作。

注意：许多系统不支持那些可在正常操作期间从系统中移走的磁盘。在这样的系统上，脱机操作几乎无用。

选择要禁用的磁盘设备 [`<address>,list,q,?`] **clt1d0**

vxdiskadm 程序禁用指定的磁盘。

3. 在下列提示下，指出是希望禁用其它设备 (y) 还是返回 vxdiskadm 主菜单 (n)：

禁用另外的设备吗？ [`y,n,q,?`]（默认：n）

## 启用物理磁盘

如果在正常系统操作过程中，将磁盘从一个系统移动到另一个，则 Volume Manager 不会自动识别该磁盘。通过使用启用磁盘任务，使 Volume Manager 得以扫描该磁盘以识别它，并确定该磁盘是否是磁盘组的一部分。而且，可以使用启用磁盘任务重新启用由磁盘组逐出任务或磁盘设备禁用（脱机）任务禁用的磁盘。若要启用磁盘，请完成下列操作：

1. 从 vxdiskadm 主菜单中选择菜单项 10（“启用（联机）磁盘设备”）。
2. 在下列提示下，输入要启用的磁盘的设备名称（或输入 **list** 以获得设备列表）：

启用（联机）磁盘设备

菜单：VolumeManager/Disk/OnlineDisk

使用此操作恢复对采用“禁用（脱机）磁盘设备”操作而禁用的磁盘的访问。

也可以使用此操作重新扫描可能已被外部的 Volume Manager 更改的磁盘。例如，如果磁盘在两个系统间共享，运行在另一系统上的 Volume Manager 可能更改了该磁盘。如果是这样，您可以使用此操作重新扫描该磁盘。

注意：许多 vxdiskadm 操作重新扫描磁盘而无需用户干预。这将消除直接联机一个磁盘的需要，除非该磁盘已被直接脱机。

选择要启用的磁盘设备 [`<address>,list,q,?`] **clt1d0**

vxdiskadm 启用指定的设备。

3. 在下列提示下，指出是希望启用其它设备 (y) 还是返回 vxdiskadm 主菜单 (n)：

启用另外的设备吗？ [`y,n,q,?`]（默认：n）

---

**注意** 在热磁盘交换后应发出 `vxctl enable` 命令，使 VxVM 能够识别磁盘和路径。

---



## 更换物理磁盘

如果磁盘发生故障，需要用其它磁盘更换它。此任务要求禁用并移走故障磁盘，然后在它的位置安装新的磁盘。若要更换磁盘，请完成下列操作：

1. 从 vxdiskadm 主菜单中选择菜单项 4 (“为替换而删除磁盘”)。
2. 在下列提示下，输入要替换的磁盘的名称（或输入 **list** 以获得磁盘列表）：

为替换而删除磁盘

```
Menu:VolumeManager/Disk/RemoveForReplace
```

使用此菜单操作将一个物理磁盘从磁盘组中删除而保留磁盘名。这将使该磁盘名的状态更改为“已删除磁盘”。如果存在任何不属于磁盘组的已初始化磁盘，您将有权选择使用这些磁盘中的一个作为替换磁盘。

```
输入磁盘名称 [<disk>,list,q,?] disk02
```

其它显示信息说明了与希望删除的磁盘关联的任何卷。必须决定是使数据保持与卷关联，还是允许数据在更换磁盘时丢失。据此回答任何提示。

3. 在下列提示下，（从提供的列表中）选择更换磁盘的设备名称，或者按回车键选择默认磁盘：

下列设备可用于作为替换设备：

```
c1t1d0
```

您可以从这些磁盘中选择一个来替换 **disk02**。

如果您不想选择替换磁盘，则选“none”。

选择一个设备，或者选择“none”

```
[<device>,none,q,?]（默认：c1t1d0）
```

4. 在下列提示下，按回车键继续：

所请求的操作是从磁盘组 **rootdg** 中删除磁盘 **disk02**。

删除的磁盘将用磁盘设备 **c1t1d0** 替换。

```
继续进行操作吗？[y,n,q,?]（默认：y）
```

vxdiskadm 显示下列成功讯息：

磁盘 **disk02** 的删除已成功地完成。

继续进行以用设备 **c1t1d0** 替换 **disk02**。

成功地完成了磁盘替换。

5. 在下列提示下，指出是希望删除其它磁盘 (**y**) 还是返回 vxdiskadm 主菜单 (**n**)：  
删除另外的磁盘吗？[y,n,q,?]（默认：n）

## 更换故障磁盘或已删除的磁盘

可以删除磁盘，并在稍后进行更换。为此，请使用菜单项 4（“为替换而删除磁盘”）删除磁盘，然后完成下列操作：

1. 从 vxdiskadm 主菜单中选择菜单项 5（“替换一个出错的或已删除的磁盘”）。

2. 选择要更换的磁盘的名称：

替换一个出错的或已删除的磁盘

菜单：VolumeManager/Disk/ReplaceDisk

使用此菜单操作，给使用“为替换而删除磁盘”菜单操作的删除的磁盘或使用期间出错的磁盘指定一个替换磁盘。系统将提示您提供需要替换的磁盘名称和要用作替换的磁盘设备。可以选择一个未初始化磁盘，该磁盘将被初始化；或者可以选择一个已使用添加或初始化磁盘菜单操作初始化的磁盘。

选择一个已删除或出错的磁盘 [`<disk>,list,q,?`] **disk02**

3. vxdiskadm 显示可用作更换磁盘的磁盘设备的设备名称；输入选择的设备的设备名称或按回车键选择默认设备：

---

**注意** 您的系统所使用的设备名称可能与实例不同。有关设备名称的更多信息，请参见第 1 页的“了解 Volume Manager”一章。

---

下列设备可用于作为替换设备：

`c1t0d0s2 c1t1d0s2`

您可以从这些磁盘中选择一个来替换 `disk02`。

选择 `"none"` 则初始化其它磁盘来替换 `disk02`。

选取一个设备，或者选择 `"none"`

[`<device>,none,q,?`]（默认：`c1t0d0s2`）

4. 在下列提示下，按回车键更换磁盘：

所请求的操作是使用已初始化的设备 `c1t0d0s2` 替换磁盘组 `rootdg` 中已删除或出错的磁盘 `disk02`。

继续进行操作吗？ [`y,n,q,?`]（默认：`y`）

vxdiskadm 程序显示下列成功讯息：

用磁盘设备 `c1t0d0s2` 对磁盘组 `rootdg` 中磁盘 `disk02` 的替换成功完成。

5. 在下列提示下，指出是希望更换其它磁盘 (**y**) 还是返回 vxdiskadm 主菜单 (**n**)：

替换另外的磁盘吗？ [`y,n,q,?`]（默认：`n`）



## 向磁盘组添加磁盘

您可能希望向已经建立的磁盘组添加新磁盘。例如，当前磁盘上的空间可能不足以满足项目或工作组的要求，特别是当这些要求发生变化时。按照添加磁盘所需的步骤进行操作，可以向磁盘组添加磁盘。参见第 96 页的“[将物理磁盘置于 Volume Manager 控制之下](#)”。

## 创建磁盘组

您可能希望与一组特定用户使用的一套特定的应用程序相关的所有数据在其它系统上也可以被存取。具体实例包括：

- ◆ 系统发生故障，必须将该系统中的数据移到其它系统。
- ◆ 必须在许多系统之间协调工作负荷。

在这些情况下，与特定应用程序或用户关联的数据应位于一套可标识的磁盘驱动器上，这一点很重要。当移动这些磁盘时，只移动磁盘上属于特定应用程序或用户的所有数据，其它数据不动。

---

**注意** Volume Manager 支持默认磁盘组 rootdg，如果不做进一步的指定，所有卷都创建在该磁盘组中。所有命令也默认为 rootdg。

---

磁盘组只能与磁盘一同创建。磁盘组必须至少有一个关联的磁盘。

如果除 rootdg 外，还需要创建其它磁盘组，请完成下列操作：

1. 从 vxdiskadm 主菜单中选择菜单项 1（“添加或初始化一个或多个磁盘”）。
2. 在下列提示下，输入要添加到 Volume Manager 控制中的磁盘的磁盘设备名称：

添加或初始化磁盘

菜单：VolumeManager/Disk/AddDisks

使用此操作添加一个或多个磁盘至磁盘组。您可以将选定的磁盘添加至现有磁盘组或新磁盘组，创建新磁盘组将作为此操作的一部分。选定的磁盘也可添加至磁盘组作为备件。也可以不将选定的磁盘添加至磁盘组而进行初始化，以留作替换磁盘使用。

在此提示下可输入多个磁盘或磁盘组合。

这里是某些磁盘选择的示例：

```
all: 全部磁盘
c3 c4t2: 控制器 3 及 4 的目标 2 上的全部磁盘
c3t4d0: 单个磁盘
```

选择要添加的磁盘设备：

```
[<pattern-list>,all,list,q,?] c1t2d0
```



这里的 `<pattern-list>` 既可以是单个磁盘，也可以是一系列磁盘和 / 或控制器（带可选目标）。如果 `<pattern-list>` 由多项组成，必须用空格将这些项彼此分开。

如果不知道要添加的磁盘的地址（设备名称），在提示下输入 `l` 或 `list` 以获得所有磁盘的列表。

- 若要继续此操作，请在下列提示下输入 `y`（或按回车键）：

这里是所选磁盘。输出格式：[ 设备名 ]

```
c1t2d0
```

继续进行吗？[y,n,q,?]（默认：y） **y**

- 在下列提示下，指定将向其中添加磁盘的磁盘组（本例中为 `anotherdg`）：

您可以选择将此磁盘添加至现有磁盘组、新磁盘组或留作将来用于添加或替换操作。若要创建新磁盘组，需选择一个尚未存在的磁盘组名称。若要留作将来使用，指定 “none” 磁盘组名称。

哪一个磁盘组 [ <group>,none,list,q,? ]（默认：rootdg）

```
anotherdg
```

- `vxdiskadm` 确认当前不存在同名的活动磁盘组，并提示您确认是否确实想创建新的磁盘组：

不存在名为 `anotherdg` 的活动磁盘组。

创建一个名为 `anotherdg` 的新磁盘组吗？[y,n,q,?]（默认：y） **y**

- 在下列提示下，既可以按回车键接受默认磁盘名称，也可以输入新的磁盘名称：

将默认磁盘名用于此磁盘吗？[y,n,q,?]（默认：y）

- 当 `vxdiskadm` 询问是否使该磁盘不可用于热重定位时，输入 `n`（或按回车键）：

从热重定位使用对象中排除磁盘吗？[y,n,q,?]（默认：n）

- 当 `vxdiskadm` 询问该磁盘是否可用作热重定位备用磁盘时，输入 `n`（或按回车键）：

添加磁盘作为 `anotherdg` 的备用磁盘吗？[y,n,q,?]（默认：n） **y**

- 若要继续此操作，请在下列提示下输入 `y`（或按回车键）：

将创建一个名为 `anotherdg` 的新磁盘组，并使用默认磁盘名将选定的磁盘添加至该磁盘组。

```
c1t2d0
```

继续进行吗？[y,n,q,?]（默认：y） **y**

- 如果该磁盘以前曾初始化供 Volume Manager 使用，将出现下列提示；输入 `y` 确认现在希望使用该磁盘：



以下磁盘设备似乎已初始化。该磁盘当前可用作替换磁盘。  
输出格式: [ 设备名 ]

c1t2d0

使用该设备吗? [y,n,q,?] (默认: y) **y**

如果确定希望重新初始化该磁盘, 请在下列提示下输入 **y** (或按回车键):

下列您所选用的磁盘看来已经初始化供 Volume Manager 使用。如果您确信该磁盘已经初始化供 Volume Manager 使用, 则不需要重新初始化该磁盘设备。

输出格式: [ 设备名 ]

c1t2d0

重新初始化该设备吗? [y,n,q,?] (默认: y) **y**

出现类似于下面这样的信息, 确认正在重新初始化该磁盘供 Volume Manager 使用:

初始化设备 c1t2d0。

创建一个名为 anotherdg 的新磁盘组, 它包含具有 another01 名称的磁盘设备 c1t2d0。

**11.** 在下列提示下, 指出是希望继续初始化更多磁盘 (**y**) 还是返回 vxdiskadm 主菜单 (**n**):

添加或初始化其它磁盘吗? [y,n,q,?] (默认: n)

## 逐出磁盘组

使用逐出磁盘组任务禁止对当前由该系统启用 (导入) 的磁盘组的存取。如果想将磁盘组中的磁盘移动到另一个系统, 需逐出该磁盘组。同样, 如果想将磁盘组中所有剩余的磁盘用于某种新目的, 也需逐出该磁盘组。

若要逐出磁盘组, 请完成下列操作:

1. 从 vxdiskadm 主菜单中选择菜单项 9 (“禁止访问 (逐出) 磁盘组”)。
2. 在下列提示下, 输入要逐出的磁盘组的名称:

禁止访问 (逐出) 磁盘组

菜单: VolumeManager/Disk/DeportDiskGroup

使用此菜单操作删除对当前系统已启用 (导入) 的磁盘组的存取。如果想要将一个磁盘组中的磁盘移动到另一系统, 则逐出该磁盘组。如果想要将一个磁盘组中剩余的全部磁盘用作新的目的, 也要逐出该磁盘组。

系统将提示您提供磁盘组的名称, 还将要求您选择是否禁用 (脱机) 磁盘。对于某些系统上的可移动磁盘设备, 在移动磁盘之前禁止对该磁盘的所有访问是重要的。

输入磁盘组名称 [<group>,list,q,?] (默认: list)

**newdg**

3. 如果想删除该磁盘组中的磁盘，在下列提示下输入 **y**：

所请求的操作是禁止访问名为 `newdg` 的可删除磁盘组。该磁盘组存储在下列磁盘上：  
`newdg01 on device c1t1d0s2`

您可以选择禁止访问（也称为“脱机”）  
 这些磁盘。如果要从系统中实际移走磁盘，可能有必要这样做以防止出错。

禁用（脱机）所指出的磁盘吗？ [**y,n,q,?**]（默认：**n**）

4. 在下列提示下，按回车键继续此操作：

继续进行操作吗？ [**y,n,q,?**]（默认：**y**）

一旦逐出磁盘组，`vxdiskadm` 将显示下列信息：

成功地删除了磁盘组 `newdg`。

5. 在下列提示下，指出是希望禁止其它磁盘组 (**y**) 还是返回 `vxdiskadm` 主菜单 (**n**)：

禁用另外的磁盘组吗？ [**y,n,q,?**]（默认：**n**）

## 导入磁盘组

使用此菜单任务启用该系统对磁盘组的存取。如果希望将磁盘组从一个系统移到另一个，必须首先在原始系统禁止（逐出）该磁盘组，然后在系统间移动磁盘并启用（导入）磁盘组。

若要导入磁盘组，请完成下列操作：

1. 从 `vxdiskadm` 主菜单中选择菜单项 8（“允许访问（导入）磁盘组”）。
2. 在下列提示下，输入要导入的磁盘组的名称：

允许访问（导入）磁盘组

菜单：`VolumeManager/Disk/EnableDiskGroup`

使用此操作恢复对磁盘组的访问。此操作可作为将磁盘组从一个系统移动到另一个系统的最后操作部分。移动磁盘组的第一部分操作是在原始主机上使用“禁止访问（逐出）磁盘组”操作。

磁盘组可以无需首先逐出而从出错的另一主机导入。必须确保磁盘组中的所有磁盘均已在主机间移动。

如果两个主机共享一条 SCSI 总线，一定要非常注意确保另一主机确实已出错或已将该磁盘组逐出。如果两个活动的主机同时导入一个磁盘组，磁盘组将被损坏并变得不能使用。

选择要导入的磁盘组 [**<group>,list,q,?**]（默认：`list`）

**newdg**

一旦导入完成，`vxdiskadm` 将显示下列成功信息：



newdg 的导入成功。

3. 在下列提示下，指出是希望导入其它磁盘组 (**y**) 还是返回 vxdiskadm 主菜单 (**n**):  
选择另外的磁盘组吗? [y,n,q,?] (默认: n)

## 退出 vxdiskadm

当完成所有磁盘管理活动时，可以从主菜单中选择 **q** 菜单项退出 vxdiskadm。



# 卷任务

---

## 介绍

本章描述如何创建和维护受 Volume Manager 控制的系统配置。包含关于创建、删除以及维护 Volume Manager 对象（卷、plex 和子磁盘）的信息。还包含联机备份信息。

## 卷、Plex 及子磁盘任务和联机备份

本章提供关于下列任务的信息：

### 卷任务

- ◆ 创建卷
- ◆ 重定卷的大小
- ◆ 删除卷
- ◆ 镜像卷
- ◆ 删除镜像
- ◆ 添加 RAID-5 日志
- ◆ 添加 DRL 日志
- ◆ 删除 DRL 日志
- ◆ 删除 RAID-5 日志
- ◆ 停止卷
- ◆ 启动卷
- ◆ 更改卷的读策略
- ◆ 准备卷从备份恢复
- ◆ 恢复卷
- ◆ 显示卷的信息



### **Plex 任务**

- ◆ 创建 Plex
- ◆ 关联 Plex
- ◆ 分离和删除 Plex
- ◆ 使用镜像备份
- ◆ 显示 Plex 信息
- ◆ 更改 Plex 属性
- ◆ 更改 Plex 状态：分离和挂接 Plex
- ◆ 移动 Plex
- ◆ 复制 Plex

### **子磁盘任务**

- ◆ 创建子磁盘
- ◆ 删除子磁盘
- ◆ 显示子磁盘信息
- ◆ 关联子磁盘
- ◆ 分离子磁盘
- ◆ 更改子磁盘属性
- ◆ 移动子磁盘
- ◆ 分割子磁盘
- ◆ 结合子磁盘

### **执行联机备份**

- ◆ 快速镜像再同步
- ◆ 镜像 VM 磁盘上的卷
- ◆ 移动 VM 磁盘上的卷

---

**注意** 一些 Volume Manager 命令要求超级用户特权或其它适当的特权。

---

## 创建卷

创建卷是为了利用 Volume Manager 虚拟磁盘概念。卷一旦存在，就可将文件系统放在卷上以组织包含文件和目录的磁盘空间。同时，也可使用应用程序（例如数据库）来组织卷上的数据。

在能够创建卷之前，必须先为 Volume Manager 初始化和定义磁盘及磁盘组。

可以使用基本或者高级方法创建卷：

- ◆ 基本 —— 基本方法获取您想完成的任务的信息，然后执行必要的基础任务。该方法只需要您最小限度的输入操作，但同时也允许较详细的规定。基本操作主要通过 `vxassist` 命令执行。通过仅将想要的卷的基本属性用做输入，`vxassist` 创建所需的 `plex` 和子磁盘。`vxassist` 也可修改现有的卷。它自动修改任何基础的或关联的对象。对于许多卷属性，`vxassist` 使用默认值，除非您提供了特定的值。
- ◆ 高级 —— 高级方法由若干通常需要您指定详细输入的命令组成。这些命令使用“积木式”方法，要求您非常了解基础结构和组件，从而手工执行必要的命令完成特定的任务。高级操作通过数个 Volume Manager 命令执行。

涉及手工创建卷的步骤包括：

- ◆ 创建子磁盘
- ◆ 创建 `plex`
- ◆ 关联子磁盘和 `plex`
- ◆ 创建卷
- ◆ 关联卷和 `plex`
- ◆ 初始化卷

卷的创建涉及 `plex` 和子磁盘组件的创建。用基本方法创建卷时，您指出想要的卷特性，系统会自动创建基础的 `plex` 和子磁盘。

可将卷创建为具有各种布局类型：

- ◆ 连续 —— 一种卷，它的子磁盘在 `plex` 内顺序地且连续地排列。如果一个磁盘的单个区域没有足够空间容纳整个卷，连续布局允许从一个或更多磁盘的多个区域创建卷。
- ◆ 条带 —— 一种卷，它的数据均匀地分布在多个磁盘上。**条带**是大小相等的段，它们被交替地和均匀地分配给单个 `plex` 的子磁盘。条带 `plex` 中至少要有两个子磁盘，每个子磁盘必须在不同的磁盘上。吞吐量随 `plex` 条带化时跨过的磁盘数增长。条带化帮助平衡 I/O 负载，以防某些子磁盘上存在高通信量的区域。
- ◆ RAID-5 —— 一种卷，它使用条带化将数据和奇偶校验均匀地分布在一个阵列的多个磁盘上。每个条带都包含一个奇偶校验条带单元和多个数据条带单元。如果其中一个磁盘发生故障，奇偶校验可用于重建数据。与条带卷的性能相比较，RAID-5 卷的写吞吐量降低了，因为每次访问数据，都必须更新奇偶校验信息。但是，与镜像相比较，使用奇偶校验减少了所需的空间量。



- ◆ 镜像 —— 一种由多个 plex 组成的卷，其中每个 plex 复制卷中包含的信息。尽管一个卷可以只有单个 plex，但是真正的镜像（数据的冗余）至少需要两个 plex。为了使冗余有用，这些 plex 中的每一个应包含来自不同磁盘的磁盘空间。
- ◆ 分层卷 —— 一种构建在卷之上的卷。可通过将一个子磁盘映像到 VM 磁盘或者存储卷来构造分层卷。存储卷提供了一种与顶层卷相似的递归层次布局。分层卷允许更多的逻辑布局组合。

**注意** 在 VERITAS Volume Manager Storage Administrator 中，条带增强是用于条带镜像的 GUI 术语，连续增强是用于连续镜像的 GUI 术语。

vxassist 命令提供了创建新卷的最简单方法。可以如下使用 vxassist 创建卷：

```
vxassist make volume_name length [attributes]
```

这里，make 是卷创建的关键字，*volume\_name* 是给卷起的名称，*length* 则指定卷中的扇区数（默认情况下）。通过分别使用后缀字符 k、m 或者 g，可以千字节、兆字节或者吉字节为单位指定长度。有关创建卷时指定长度单位的更多信息，请参阅 vxintro (1M) 帮助手册。适当时，也可指定其它属性。

默认情况下，vxassist 在 rootdg 磁盘组内创建卷。通过将 -g *diskgroup* 包含在 vxassist 命令行中，可以指定其它的磁盘组。

在磁盘组 rootdg 内创建一个卷的操作创建了两个可用来存取卷的设备节点文件：

- ◆ /dev/vx/dsk/*volume\_name*（卷的块设备节点）
- ◆ /dev/vx/rdsk/*volume\_name*（卷的原始设备节点）

对于 rootdg 和 rootdg 以外磁盘组内的卷，这些名称包含如下所示的磁盘组名称：

- ◆ /dev/vx/dsk/*diskgroup\_name*/*volume\_name*
- ◆ /dev/vx/rdsk/*diskgroup\_name*/*volume\_name*

第 124 页的“[创建连续卷](#)”描述了创建（默认）卷的最简单方法。稍后将介绍如何创建具有特定属性的卷。

## 创建连续卷

默认情况下，vxassist 创建占用磁盘空间的一个或更多部分的连续卷。在一个分段磁盘上，这允许您组合一个比可用空闲磁盘空间的任何单个部分都要大的卷。

如果单个磁盘上没有足够的空间，vxassist 将创建跨接卷。跨接卷是一个连续卷，包含分散在多个磁盘上的磁盘空间部分。由于跨接卷占用不止一个磁盘的空间，它可能比最大的单个磁盘要大。

## 在任一磁盘上创建连续卷

如果没有指定磁盘，Volume Manager 选择一个磁盘来创建卷。

使用此命令创建连续的默认卷：

```
vxassist make volume_name length
```

这里，*volume\_name* 是卷的名称，*length* 以扇区为单位指定卷的长度（除非用后缀字符指定了另一大小单位）。当创建卷时，可以将扇区、千字节、兆字节或者吉字节指定为新卷的长度单位。将计量单位添加为长度的后缀（s、m、k 或者 g）。如果没有指定单位，将以扇区为单位。

可以使用此命令创建长度为 10 兆字节的卷 *voldefault*：

```
vxassist make voldefault 10m
```

## 在特定的磁盘上创建连续卷

除非您指定了其它磁盘，否则 Volume Manager 将自动选择每个卷驻留的磁盘。如果希望

一个卷驻留在特定的磁盘上，必须为 Volume Manager 指定那个磁盘。可以指定不止一个磁盘。

使用此命令在特定的磁盘上创建卷：

```
vxassist make volume_name length diskname [...]
```

若要在 *disk03* 上创建卷 *volspecific*，请使用此命令：

```
vxassist make volspecific 3m disk03
```

## 创建条带卷

条带卷至少包含一个由位于两个或更多物理磁盘上的两个或更多子磁盘组成的 *plex*。

使用此命令创建条带卷：

```
vxassist make volume_name length layout=stripe
```

若要创建条带卷 *volzebra*，请使用此命令：

```
vxassist make volzebra 10m layout=stripe
```

此命令在默认数目的磁盘上创建一个具有默认条带单元大小的条带卷。

通过在命令行的最后指定磁盘名，可以指出将要创建卷的磁盘。例如，若要在三个特定的磁盘（*disk03*、*disk04* 和 *disk05*）上创建一个 30 兆字节的条带卷，请使用此命令：

```
vxassist make stripevol 30m layout=stripe disk03 disk04 disk05
```



## 创建 RAID-5 卷

RAID-5 卷包含一个由位于两个或更多物理磁盘上的两个或更多子磁盘组成的 RAID-5 plex。每个卷只能有一个 RAID-5 plex。一个 RAID-5 卷也可以包含一个或更多的 RAID-5 日志 plex，它们用来记录写入卷的数据和奇偶校验信息。

使用此命令创建 RAID-5 卷：

```
vxassist make volume_name length layout=raid5
```

若要创建 RAID-5 卷 volraid，请使用此命令：

```
vxassist make volraid 10m layout=raid5
```

此命令在默认数目的磁盘上创建一个具有默认条带单元大小的 RAID-5 卷。默认情况下，它同时创建一个 RAID-5 日志。

## 创建镜像卷

使用此命令创建一个新的镜像卷：

```
vxassist make volume_name length layout=mirror
```

若要创建镜像卷 volmir，请使用此命令：

```
vxassist make volmir 5m layout=mirror
```

## 重定卷的大小

重定卷的大小可以更改卷的大小。使用 `vxassist` 或者 `vxresize` 都可达到此目的。

如果卷不够大，不足以存储需要存储在卷内的大量数据，则需要扩展卷的长度。如果卷的大小增加，则 `vxassist` 自动查找可用的磁盘空间。

当重定卷的大小时，可以将新卷的长度单位指定为扇区、千字节、兆字节或吉字节。将计量单位添加为长度的后缀（`s`、`m`、`k` 或者 `g`）。如果没有指定单位，将以扇区为单位。

---

**警告** 不要将卷缩小到小于文件系统的大小。如果有一个 VxFS 文件系统，则可以先缩小文件系统，然后再缩小卷。如果没有首先缩小文件系统，则有可能导致数据不可恢复。

---

## 用 vxassist 重定卷的大小

vxassist 可以按照下列方法重定卷的大小：

- growto —— 将卷增加到指定的长度
- growby —— 按指定的数量增加卷
- shrinkto —— 将卷减小到指定的长度
- shrinkby —— 按指定的数量减小卷

---

**警告** 不能放大或缩小与封装的引导磁盘（rootvol、usr、var、opt、swapvol 等等）关联的任何卷，因为它们映射到磁盘的物理基础分区上且必须是相邻的。如果试图增加 rootvol、usrvol、varvol 或者 swapvol 的大小，则当需要还原为片进行引导时，系统可能变为不可引导。这还会妨碍 Solaris 成功的升级，而您可能不得不重新进行安装。另外，upgrade\_start 脚本可能会失败。

---

### 扩展到特定的长度

使用此命令将卷扩展 *到* 特定的长度：

```
vxassist growto volume_name length
```

若要将 volcat 扩展到 2000 个扇区，请使用此命令：

```
vxassist growto volcat 2000
```

### 按特定的长度扩展

使用此命令，*按* 特定的长度扩展卷：

```
vxassist growby volume_name length
```

若要按 100 个扇区扩展 volcat，请使用此命令：

```
vxassist growby volcat 100
```

### 缩小到特定的长度

使用此命令将卷缩小 *到* 特定的长度：

```
vxassist shrinkto volume_name length
```

确保不要将卷收缩到低于文件系统的当前大小或者使用卷的数据库的当前大小。在空卷上可以安全地使用此命令。

若要将 volcat 缩小到 1300 个扇区，请使用此命令：

```
vxassist shrinkto volcat 1300
```



### 按特定的长度收缩

使用此命令，*按* 特定的长度收缩卷：

```
vxassist shrinkby volume_name length
```

若要按 300 个扇区收缩 volcat，请使用此命令：

```
vxassist shrinkby volcat 300
```

### 用 vxresize 重定卷的大小

可以使用 vxresize 命令重定包含文件系统的卷的大小。尽管也可以使用其它命令重定包含文件系统的卷的大小，但 vxresize 具有自动重定文件和卷的大小的优点。有关如何使用 vxresize 的详细信息，请参见 vxresize(1M) 帮助手册。注意，用 vxresize 只能重定 vxfs 和 ufs 文件系统的大小。

如果卷比所需的大，可以缩小卷的大小。

## 删除卷

一个卷一旦不再是必需的时（例如，不活动和已存档），可以使用命令 vxassist remove volume 删除卷并释放磁盘空间用于其它用途。

用 vxassist 命令可以删除整个卷。如下例所示，使用关键字 remove 和 volume 并在命令行上提供卷的名称：

```
vxassist remove volume volume_name
```

## 镜像卷

镜像是卷的一个副本，与卷的原始副本存储在不同的磁盘上。镜像卷确保当其中一个磁盘发生故障时，那个卷中的数据不会丢失。

### 创建镜像卷

用 vxassist 创建镜像卷：

```
vxassist make vol_name length layout=mirror
```



## 镜像一个现有的卷

可将镜像 (plex) 添加到现有的卷中。为此可使用 `vxassist` 命令：

```
vxassist mirror volume_name
```

例如：

```
vxassist mirror voltest
```

镜像一个现有卷的另一种方法是，使用以下命令先创建一个 `plex`，然后将其与一个卷关联：

```
vxmake plex plex_name sd=subdisk_name ...
vxplex att volume_name plex_name
```

## 创建启用脏区日志的卷

若要创建启用脏区日志 (DRL) 的镜像卷，请使用此命令创建带日志的镜像卷：

```
vxassist make volume_name length layout=mirror,log
```

默认情况下，`vxassist` 为每个日志子磁盘创建一个日志 `plex`。

## 镜像全部卷

若要将系统上全部现有的卷镜像到可用的磁盘空间，请使用此命令：

```
/etc/vx/bin/vxmirror -g diskgroup -a
```

也可配置 Volume Manager 在默认情况下创建镜像卷。为此请输入此命令：

```
/etc/vx/bin/vxmirror -d yes
```

如果做了此更改，仍能通过将 `nmirror=1` 指定为 `vxassist` 命令的属性来生成未镜像卷。例如，若要创建一个名为 `nomirror` 的未镜像卷，其大小为 20 兆字节，请使用此命令：

```
vxassist make nomirror 20m nmirror=1
```



## 删除镜像

当不再需要镜像时，可以删除它。可以删除镜像以提供空闲磁盘空间。

---

**注意** 不能删除与卷关联的最后一个有效 plex。

---

**警告** 若要保存待删除镜像上的数据，必须知道那个镜像的配置。那个配置（条带单元大小和子磁盘排序）的参数对于创建包含相同数据的新镜像至关重要。在删除此类镜像之前，必须先记录其配置。

---

可以使用此命令将镜像与关联的卷分离并删除：

```
vxplex -o rm dis plex_name
```

若要分离并删除一个名为 vol01-02 的镜像，请使用此命令：

```
vxplex -o rm dis vol01-02
```

此命令删除镜像 vol01-02 和所有关联的子磁盘。

## 添加 RAID-5 日志

每个 RAID-5 卷只能有一个 RAID-5 plex。其余任何 plex 都是 RAID-5 日志 plex，它们用来记录写入卷的数据和奇偶校验信息。当使用 vxassist 创建 RAID-5 卷时，默认情况下，将为那个卷创建一个日志 plex。

用此命令向现有的卷添加 RAID-5 日志：

```
vxassist addlog volume_name
```

若要为 RAID-5 卷 volraid 创建日志，请使用此命令：

```
vxassist addlog volraid
```

## 添加 DRL 日志

要使卷的脏区日志生效，必须将一个日志子磁盘添加到那个卷并镜像该卷。每个 plex 只能有一个日志子磁盘。

用此命令将一个 DRL 日志添加到现有的卷：

```
vxassist addlog volume_name
```

若要为卷 vol03 创建日志，请使用此命令：

```
vxassist addlog vol03
```

当使用 `vxassist` 向卷添加日志子磁盘时，默认情况下也创建一个日志 `plex` 来包含日志子磁盘。

一旦创建，包含日志子磁盘的 `plex` 可被视为正规 `plex`。可向日志 `plex` 添加数据子磁盘。可以使用删除普通 `plex` 和子磁盘的相同步骤删除日志 `plex` 和日志子磁盘。

## 删除 DRL 日志

使用如下 `vxassist` 命令可以删除日志。

```
vxassist remove log volume_name
```

使用属性 `nlog=` 指定要删除的日志数。默认情况下，`vxassist` 删除一个日志。

## 删除 RAID-5 日志

使用如下 `vxassist` 命令可以删除 RAID-5 日志：

```
vxassist remove log volume_name
```

使用属性 `nlog=` 指定要删除的日志数。默认情况下，`vxassist` 删除一个日志。

## 停止卷

停止一个卷使其不可用。停止一个卷将使卷的状态从 `ENABLED` 或 `DETACHED` 改为 `DISABLED`。如果命令不能停止卷，则卷将保持当前状态。若要停止一个卷，请使用此命令：

```
vxvol stop volume_name ...
```

例如，停止名为 `vol01` 的卷的命令是：

```
vxvol stop vol01
```

若要停止所有 `ENABLED` 卷，请使用此命令：

```
vxvol stopall
```



## 启动卷

启动一个卷使其可用。

启动一个卷将使卷的状态从 `DISABLED` 或 `DETACHED` 改为 `ENABLED`。如果无法启用卷，则卷将保持当前状态。若要启动一个卷，请使用此命令：

```
vxrecover -s volume_name ...
```

若要启动所有 `DISABLED` 卷，请使用此命令：

```
vxrecover -s
```

## 列出无法启动的卷

一个无法启动的卷可能是配置不正确，或者有其它妨碍其启动的错误和情况。若要显示无法启动的卷，请使用命令 `vxinfo`。`vxinfo` 显示关于一个或更多卷的可存取性和可用性信息：

```
vxinfo [volume_name]
```

## 更改卷的读策略

Volume Manager 提供下列读策略选项：

- ◆ `round` 对于每个检测到的非顺序 I/O，按“循环”方式依次读取每个 `plex`。顺序存取使得只能存取一个 `plex`。这利用了驱动器或控制器的读前高速缓存策略。
- ◆ `prefer` 首先从被指定为首选 `plex` 的 `plex` 开始读。
- ◆ `select` 根据与卷关联的 `plex` 选择一个默认策略。如果卷具有一个已启用的条带 `plex`，`select` 默认为首选那个 `plex`；否则默认为循环方式。

可将读策略从 `round` 改为 `prefer`（或者反过来），或者改为一个不同的首选 `plex`。`vxvol rdpol` 命令设置卷的读策略。

若要将读策略设置为 `round`，请使用此命令：

```
vxvol rdpol round volume_name
```

例如，将卷 `vol01` 的读策略设置为循环读的命令是：

```
vxvol rdpol round vol01
```

若要将读策略设置为 `prefer`，请使用此命令：

```
vxvol rdpol prefer volume_name preferred_plex_name
```

例如，将 `vol01` 的策略设置为首先从 `plex vol01-02` 读起的命令是：

```
vxvol rdpol prefer vol01 vol01-02
```



若要将读策略设置为 `select`，请使用此命令：

```
vxvol rdpol select volume_name
```

## 准备卷从备份恢复

创建卷的备份副本很重要。这将提供与数据在备份时的状态相同的拷贝。备份拷贝用于恢复由于磁盘故障丢失的卷和由于人为错误被破坏的数据。Volume Manager 允许在最小限度的中断用户的情况下备份卷。

可以按照下列过程，用 `vxassist` 备份卷：

### 1. 创建将要备份的卷的快照镜像。

`vxassist snapstart` 任务创建只写备份镜像，它挂接在将要备份的卷上并与该卷同步。当与卷同步时，备份镜像随时准备用作快照镜像。但是，它一直更新到此过程中实际创建快照的时候，这时它将被分离。分离时可能需要一些时间，具体取决于卷的大小。

用此命令创建卷的快照镜像：

```
vxassist snapstart volume_name
```

若要创建一个称为 `voldef` 的卷的快照镜像，请使用此命令：

```
vxassist snapstart voldef
```

### 2. 选择适当的时间创建快照卷。

尽量安排在用户对卷的存取尽可能少的时候创建快照。

### 3. 创建一个反映快照时的原始卷的快照。

在具有快照镜像的卷上运行 `vxassist snapshot` 命令以完成联机备份过程。该任务分离已完成的快照镜像，创建一个新的标准卷，并将快照镜像挂接在其上。快照随后成为一个只读卷。该步骤应只需几分钟。

用此命令创建快照卷：

```
vxassist snapshot volume_name new_volume_name
```

若要创建 `voldef` 的快照卷，请使用此命令：

```
vxassist snapshot voldef snapvol
```

这时，备份实用程序可以使用快照卷，而原始卷可继续由应用程序和用户使用。

可以用任何喜欢的方法备份快照卷。为避免浪费空间，备份结束时应删除快照卷。快照卷占用的空间与原始卷一样多。



## 恢复卷

系统崩溃或者 I/O 错误会破坏卷的一个或更多的 plex，并使所有卷都不处于 CLEAN 或 ACTIVE 状态。可以将其中一个 plex 标记为 CLEAN 并指示系统将那个 plex 用作恢复其它 plex 的源。

若要将一个 plex 置于 CLEAN 状态，请使用此命令：

```
vxmend fix clean plex_name
```

例如，将名为 vol101-02 的 plex 置于 CLEAN 状态的命令是：

```
vxmend fix clean vol101-02
```

有关如何使用 vxmend 的信息，请参见 vxmend (1M) 帮助手册。

## 显示卷的信息

可以使用 vxprint 命令显示关于如何配置卷的信息。

使用此命令可显示系统中所有卷的卷、plex 和子磁盘记录信息：

```
vxprint -ht
```

这是 vxprint 输出的一个实例：

磁盘组: rootdg

| DG NAME      | NCONFIG  | NLOG    | MINORS   | GROUP-ID       |           |          |      |
|--------------|----------|---------|----------|----------------|-----------|----------|------|
| DM NAME      | DEVICE   | TYPE    | PRIVLEN  | PUBLEN         | STATE     |          |      |
| V NAME       | USETYPE  | KSTATE  | STATE    | LENGTH         | READPOL   | PREFPLEX |      |
| PL NAME      | VOLUME   | KSTATE  | STATE    | LENGTH         | LAYOUT    | NCOL/WID | MODE |
| SD NAME      | PLEX     | DISK    | DISKOFFS | LENGTH         | [COL/]OFF | DEVICE   | MODE |
| dg rootdg    | default  | default | 0        | 759096729.1025 | tweety    |          |      |
| dm disk10    | c1t0d0s2 | sliced  | 559      | 1044400        | -         |          |      |
| dm disk20    | c2t0d0s2 | sliced  | 559      | 1044400        | -         |          |      |
| v pubs       | fsgen    | ENABLED | ACTIVE   | 2288           | SELECT    | -        |      |
| pl pubs-01   | pubs     | ENABLED | ACTIVE   | 2288           | CONCAT    | -        | RW   |
| sd disk10-01 | pubs-01  | disk10  | 0        | 2288           | 0         | c0t0d0   | ENA  |
| v voldef     | sgen     | ENABLED | ACTIVE   | 20480          | SELECT    | -        |      |
| pl voldef-01 | voldef   | ENABLED | ACTIVE   | 20480          | CONCAT    | -        | RW   |
| sd disk10-02 | voldef-0 | disk10  | 2288     | 20480          | 0         | c0t1d0   | ENA  |

这里，dg 是磁盘组，dm 是磁盘，v 是卷，pl 是 plex，sd 是子磁盘。头几行指出同后面的输出行的每个类型匹配的标题。每个卷都与关联的 plex 和子磁盘一起列出。

使用此命令可显示一个特定卷的与卷相关的信息：

```
vxprint -t volume_name
```



若要显示关于 voldef 的信息，请使用此命令：

```
vxprint -t voldef
```

这是 vxprint 输出的一个实例：

磁盘组：rootdg

| V | NAME   | USETYPE | KSTATE  | STATE  | LENGTH | READPOL | PREFPLEX |
|---|--------|---------|---------|--------|--------|---------|----------|
| v | voldef | fsgen   | ENABLED | ACTIVE | 20480  | SELECT  | -        |

## Plex 任务

Plex 是子磁盘的逻辑分组，这些子磁盘创建不受物理磁盘大小或其它限制约束的磁盘空间区。磁盘数据的复制（镜像）是通过为单个卷创建多个 plex 来完成的。每个 plex 包含卷数据的相同副本。由于每个 plex 必须驻留在不同的磁盘上，镜像提供的复制防止在单点磁盘子系统发生故障时丢失数据。多个 plex 也提供了增强的数据完整性和可靠性。

Plex 任务包括：

- ◆ 创建 plex
- ◆ 使用镜像备份
- ◆ 关联 plex
- ◆ 分离和删除 plex
- ◆ 列出所有 plex
- ◆ 显示 plex
- ◆ 更改 plex 属性
- ◆ 更改 plex 状态
- ◆ 移动 plex
- ◆ 复制 plex

## 创建 Plex

vxmake 命令创建 Volume Manager 对象，如 plex。当创建 plex 时，标识子磁盘并将它们与要创建的 plex 关联。

若要从现有子磁盘中创建 plex，请使用此命令：

```
vxmake plex plex_name sd=subdisk_name,...
```

例如，使用两个名为 disk02-01 和 disk02-02 的现有子磁盘创建一个名为 vol01-02 的连续 plex 的命令是：

```
vxmake plex vol01-02 sd=disk02-01,disk02-02
```



## 创建条带 Plex

若要创建条带 plex，需要指定其它的属性。例如，创建一个名为 p1-01、条带宽度为 32 个扇区和 2 列的条带 plex 的命令是：

```
vxmake plex p1-01 layout=stripe stwidth=32 ncolumn=2 \
sd=disk01-01,disk02-01
```

如果想用 plex 生成卷，必须使 plex 与卷关联（有关更多信息，请参见第 136 页的“[关联 Plex](#)”）。

## 关联 Plex

通过将 plex 与卷关联，plex 成为卷的参与 plex。若要关联 plex 和现有的卷，请使用此命令：

```
vxplex att volume_name plex_name
```

例如，将名为 vol01-02 的 plex 与名为 vol01 的卷关联的命令是：

```
vxplex att vol01 vol01-02
```

如果卷尚未创建，作为卷创建命令的一部分，可以将一个 plex（或者多个 plex）与将要创建的卷关联：

```
vxmake -U usetype vol volume_name plex=plex_name1, plex_name2...
```

例如，创建一个名为 home、fsgen 类型的镜像卷并关联两个名为 home-1 与 home-2 的现有 plex 的命令是：

```
vxmake -Ufsgen vol home plex=home-1,home-2
```

---

**注意** 也可在现有卷上使用下列命令添加和关联 plex：

```
vxassist mirror volume_name
```

---

## 分离和删除 Plex

当不再需要 plex 时，可以删除它。下列情况下，可能需要删除 plex：

- ◆ 提供空闲磁盘空间
- ◆ 减少一个卷内的镜像数以便增加另一个镜像及其关联卷的长度。当删除 plex 和子磁盘时，由此产生的空间可添加到其它卷。
- ◆ 删除为备份卷而创建的、不再需要的临时镜像。
- ◆ 更改 plex 的布局



---

**警告** 若要保存将要删除的 plex 上的数据，必须知道那个 plex 的配置。那个配置（条带单元大小和子磁盘排序）的参数对于创建容纳相同数据的新 plex 至关重要。在删除 plex 之前，必须先记录其配置。有关更多信息，请参见第 138 页的“显示 Plex 信息”。

---

可以使用此命令将 plex 与关联的卷分离并删除：

```
vxplex -o rm dis plex_name
```

若要分离并删除一个名为 vol01-02 的 plex，请使用此命令：

```
vxplex -o rm dis vol01-02
```

此命令删除 plex vol01-02 及其关联的所有子磁盘。

可以使用下列命令，先分离 plex 和子磁盘，然后将其删除：

```
vxplex dis plex_name
vxedit -r rm plex_name
```

这些命令合在一起可达到与 vxplex -o rm dis 一样的效果。

## 使用镜像备份

如果已镜像了卷，可以通过将其中一个卷镜像脱机一段时间来完成对那个卷的备份。这样，就不必仅仅为了备份而增加额外的磁盘空间。但是，这同时也去除了进行备份所需的持续时间内的卷冗余。

---

**注意** 本节中的信息不适用于 RAID-5。

---

使用这些步骤，可以在活动系统上执行镜像卷的备份：

1. 使用户活动停止一会儿以改善备份的一致性。（可选）

2. 分离其中一个卷镜像（本实例中指 vol01-01）：

```
vxplex dis vol01-01
```

3. 使用分离的 plex，创建一个新的临时卷：

```
vxmake -U gen vol tempvol plex=vol01-01
```

4. 启动临时卷：

```
vxvol start tempvol
```

5. 使用临时卷，执行适当的备份过程。



**6.** 停止临时卷:

```
vxvol stop tempvol
```

**7.** 将备份 plex 与其临时卷分离:

```
vxplex dis vol01-01
```

**8.** 将备份 plex 与其原始卷重新关联以恢复卷冗余:

```
vxplex att vol01 vol01-01
```

**9.** 删除临时卷:

```
vxedit rm tempvol
```

有关使用 vxassist 命令的另一种联机备份方法的信息, 请参阅第 147 页的“[执行联机备份](#)”。

## 显示 Plex 信息

列出 plex 有助于识别用于生成卷的空闲 plex。使用带 plex (-p) 选项的 vxprint 命令列出关于所有 plex 的信息。

若要显示系统中所有 plex 的详细信息, 请使用此命令:

```
vxprint -lp
```

若要显示特定 plex 的详细信息, 请使用此命令:

```
vxprint -l plex_name
```

-t 选项显示 plex 的单行信息。若要列出空闲 plex, 请使用此命令:

```
vxprint -pt
```

## 更改 Plex 属性

---

**警告** 更改 plex 属性时要非常谨慎, 只有在必要时才应更改。

---

vxedit 命令更改 plex 和其它 Volume Manager 对象的属性。若要更改 plex 属性, 请使用此命令:

```
vxedit set field=value ... plex_name ...
```

Volume Manager 命令在 plex 创建后使用注释域以及 putil 和 tutil 域。putil 属性在重新引导时保持, tutil 域是临时的, 在重新引导时不保留。

putil 和 tutil 都有三个功能，且按照这些功能编号。必要时可以修改这些域。Volume Manager 使用标记为 putil0 和 tutil0 的实用程序域。其它 VERITAS 产品则使用标记为 putil1 和 tutil1 的域。标记为 putil2 和 tutil2 的域是用户域。第 139 页的表 4 列出了 putil 和 tutil 域的功能。

表 4. putil[n] 和 tutil[n] 域

| 域      | 实用程序域描述                           |
|--------|-----------------------------------|
| putil0 | 预留给 Volume Manager 命令使用，在重新引导时保留。 |
| putil1 | 预留给图形用户界面这样的高级实用程序使用。该域在重新引导时保留。  |
| putil2 | 预留给系统管理员或站点特有的应用程序使用。该域在重新引导时保留。  |
| tutil0 | 预留给 Volume Manager 命令使用，在重新引导时清除。 |
| tutil1 | 预留给图形用户界面这样的高级实用程序使用。该域在重新引导时清除。  |
| tutil2 | 预留给系统管理员或站点特有的应用程序使用。该域在重新引导时清除。  |

命令：

```
vxedit set comment="my plex" tutil2="u" user="admin" vol01-02
```

使用 vxedit 设置下列属性：

- ◆ 将注释域（标识 plex 的用途）设置为 my plex
- ◆ 将 tutil2 设置为 u 以表明子磁盘正在使用中
- ◆ 将 user ID 更改为 admin

若要防止特定的 plex 与卷关联，请按照此命令中所指定的，将 putil0 域设置为非空字符串：

```
vxedit set putil0="DO-NOT-USE" vol01-02
```

## 更改 Plex 状态：分离和挂接 Plex

一旦创建卷并将其置于联机 (ENABLED) 状态，Volume Manager 便能够临时将 plex 从卷中分离出来。例如，当 plex 驻留的硬件需要检修时，或当卷处于无法启动状态而必须手工选择恢复卷的源 plex 时，这是个有用的功能。

解决磁盘或系统故障涉及使卷脱机以及挂接和分离其 plex。用于解决磁盘故障的两个命令是：vxmend 和 vxplex。

若要将 plex 置于 OFFLINE 状态，以便在包含那个 plex 的子磁盘的物理磁盘上执行检修或维护任务，请使用此命令：

```
vxmend off plex_name ..
```



如果磁盘的磁头损坏，则应将所有在受影响的磁盘上有关联子磁盘的 plex 置于 OFFLINE 状态。例如，如果 plex vol101-02 和 vol102-02 在要检修的驱动器上有子磁盘，则使用此命令：

```
vxmend off vol101-02 vol102-02
```

此命令将 vol101-02 和 vol102-02 置于 OFFLINE 状态，并且它们保持那种状态直到被更改。

## 分离 Plex

若要临时分离镜像卷的一个 plex，请使用此命令：

```
vxplex det plex_name
```

例如，临时分离名为 vol101-02 的 plex 并将其置于维护方式的命令是：

```
vxplex det vol101-02
```

此命令临时分离此 plex，但仍维护该 plex 和其卷之间的关联。但是，该 plex 不能用于 I/O。用前述命令分离的 plex 在系统重新引导时恢复。该 plex 的状态设置为 STALE，这样，如果 vxvol start 命令在适当的卷上运行（例如，当系统重新引导时），该 plex 的内容将恢复并处于 ACTIVE 状态。

当 plex 即将作为其卷的一个活动部分返回时，遵循下列过程：

- ◆ 如果卷不是 ENABLED，用此命令启动它：

```
vxvol start volume_name
```

如果卷是无法启动的，使用下列命令将其中一个 plex 设置为 CLEAN：

```
vxmend fix clean plex_name
```

然后启动卷。

- ◆ 如果 plex 尚未具有 ENABLED 的 *内核状态*，使用此命令：

```
vxplex att volume_name plex_name ...
```

与将 OFFLINE 的 plex 返回到 ACTIVE 状态一样，此命令恢复 plex 的内容，然后将 plex 的状态设置到 ACTIVE。

## 挂接 Plex

当检修或更换了磁盘并准备重新使用它时，必须使 plex 回到联机状态（将 plex 的状态设置为 ACTIVE）。

如果卷当前是 ENABLED，则使用此命令：

```
vxplex att volume_name plex_name ...
```

例如，针对 vol01 卷上的名为 vol01-02 的 plex 的命令是：

```
vxplex att vol01 vol01-02
```

此命令开始恢复 plex 的内容，并在恢复完成后，将 plex 实用程序状态设置为 ACTIVE。

如果卷当前没有在使用（不是 ENABLED），则使用此命令：

```
vxmend on plex_name
```

例如，针对名为 vol01-02 的 plex 的命令是：

```
vxmend on vol01-02
```

这种情况下，vol01-02 的状态设置为 STALE。当下一次启动卷时，此 plex 上的数据从其它 plex 再生，并以设置为 ACTIVE 的状态合并到卷。

如果需要手工更改 plex 的状态，请参阅“卷任务”中的“恢复卷”。有关这些命令的更多信息，请参见 vxmake(1M) 和 vxmend(1M) 帮助手册。

## 移动 Plex

通过移动 plex，可以将原始 plex 中的数据内容复制到新的 plex 上。将数据从一个 plex 移动到另一个 plex 上的命令是：

```
vxplex mv original_plex new_plex
```

若要成功完成移动任务，必须满足下列条件：

- ◆ 旧的 plex 必须是一个活动 (ENABLED) 卷的活动部分。
- ◆ 新的 plex 应至少与旧的 plex 一样大或者更大。
- ◆ 新的 plex 一定不能与另一个卷关联。

plex 的大小有多层含义：

- ◆ 如果新的 plex 比原始 plex 小或者稀疏，则将导致原始 plex 上的数据拷贝不完整。如果这种情况下想移动 plex，必须使用 `-o force` 选项。
- ◆ 如果新的 plex 比原始 plex 长或者没有其稀疏，则原始 plex 上存在的数据将被复制到新的 plex 上。任何原始 plex 上没有的、但在新的 plex 上存在的区域由其它与同一卷关联的完整 plex 填充。
- ◆ 如果新的 plex 比卷本身长，那么新的 plex 中超过卷大小的剩余区域不初始化且保持未使用状态。



## 复制 Plex

该任务将一个卷的内容复制到指定的 plex 上。不能启用将要复制的卷，plex 不能与任何其他卷关联。若要复制 plex，请使用此命令：

```
vxplex cp volume_name new_plex
```

复制任务完成后，*new\_plex* 不与指定的卷 *volume\_name* 关联。plex 包含卷数据的完整副本。复制的 plex 应与卷大小相同或者比卷大。如果复制的 plex 比卷小，则会导致数据拷贝不完整。同理，*new\_plex* 不能稀疏。

## 子磁盘任务

子磁盘是 Volume Manager 配置中的低层生成块，用于生成卷。以下各节描述可在子磁盘上执行的任务。

### 创建子磁盘

**注意** 当使用 vxassist 或者 Storage Administrator 创建卷时，将自动为您创建子磁盘。

可以使用 vxmake 命令创建 Volume Manager 对象，例如子磁盘。创建子磁盘时，必须指定子磁盘的名称和长度以及磁盘内的子磁盘起始点（偏移）和磁盘介质名称。

若要创建子磁盘，请使用此命令：

```
vxmake sd subdisk_name disk,offset,len
```

例如，创建一个名为 disk02-01、起始于磁盘 disk02 的起点、长度为 8000 个扇区的子磁盘的命令显示如下：

```
vxmake sd disk02-01 disk02,0,8000
```

默认情况下，Volume Manager 命令采用扇区作为大小单位。添加后缀（例如 k、m 或者 g）可更改大小单位。

如果想使用新的子磁盘生成卷，必须关联子磁盘与 plex（参见第 143 页的“[关联子磁盘](#)”）。所有 plex 布局（连续、条带、RAID-5）的子磁盘创建方法与此相同。

### 删除子磁盘

若要删除子磁盘，请使用此命令：

```
vxedit rm subdisk_name
```

例如，可以使用下列命令删除名为 disk02-01 的子磁盘：

```
vxedit rm disk02-01
```



## 显示子磁盘信息

`vxprint` 命令显示 Volume Manager 对象的信息。若要显示所有子磁盘的一般信息，请使用此命令：

```
vxprint -st
```

`-s` 选项指定子磁盘的信息。`-t` 选项显示依赖于列出的对象类型的单行输出记录。

使用下列命令，可以显示一个特定子磁盘的完整信息：

```
vxprint -l subdisk_name
```

例如，下面是获得名为 `disk02-01` 的子磁盘的所有信息的命令：

```
vxprint -l disk02-01
```

此命令提供下列输出：

```
磁盘组 : rootdg
Subdisk: disk02-01
info: disk=disk02 offset=0 len=205632
assoc: vol=mvol plex=mvol-02 (offset=0)
flags: enabled
device: device=c2t0d1s2 path=/dev/vx/dmp/c2t0d1s4 diskdev=32/68
```

## 关联子磁盘

通过关联子磁盘和 plex，可以按 plex 内的特定偏移放置子磁盘定义的磁盘空间量。其它子磁盘的任何部分不能占用该子磁盘填充的整个空间。根据配置的总体状态，有多种关联子磁盘和 plex 的方法。

如果已经创建了一个特定 plex 所需的所有子磁盘，使用下列命令在创建 plex 时关联子磁盘：

```
vxmake plex plex_name sd=subdisk_name,...
```

例如，下列命令创建 plex `home-1` 并使子磁盘 `disk02-01`、`disk02-00` 及 `disk02-02` 与 plex `home-1` 关联。

```
vxmake plex home-1 sd=disk02-01,disk02-00,disk02-02
```

子磁盘按照列出的顺序，从偏移 0 开始关联。定义为 `disk02-01` 的磁盘空间是第一个，磁盘空间 `disk02-00` 是第二个，`disk02-02` 是第三个。如果使用这种命令，则不必象指定多个命令那样，先创建 plex 然后使每个子磁盘与那个 plex 关联。这种关联子磁盘的方法在初始配置过程中很方便。

也可以将子磁盘与已经存在的 plex 关联。用下列命令可以将一个或更多的子磁盘与现有的 plex 关联：

```
vxsd assoc plex_name sd_name [sd_name2 sd_name3 ...]
```



例如，可以使用此命令将名为 `disk02-01`、`disk02-00` 和 `disk02-02` 的子磁盘与名为 `home-1` 的 `plex` 关联：

```
vxsd assoc home-1 disk02-01 disk02-00 disk02-01
```

如果 `plex` 是非空的，则新的子磁盘添加到已经与 `plex` 关联的任何子磁盘后，除非在命令中指定了 `-l` 选项。`-l` 选项按 `plex` 内的特定偏移关联子磁盘。

当为一个特定的卷创建了稀疏 `plex`（即子磁盘之间有缝隙的 `plex`），并想使该 `plex` 完整时，需要使用 `-l` 选项。若要使 `plex` 完整，需要创建一个子磁盘，它的大小正好适合稀疏 `plex` 中的缝隙。然后，通过将 `plex` 中的间隙开始位置指定为偏移，关联这个子磁盘与 `plex`。使用此命令：

```
vxsd -l offset assoc sparse_plex_name exact_size_subdisk
```

---

**注意** 子磁盘的大小必须刚好合适，因为 Volume Manager 不允许两个子磁盘定义的空间在一个 `plex` 内重叠。

---

对于条带子磁盘，可以指定子磁盘的列数和列偏移：

```
vxsd -l column_#/offset assoc plex_name sd_name ...
```

如果 `-l` 选项只指定了条带 `plex` 的一个数字，则将该数字当作列数并在列的末尾关联子磁盘。

## 关联日志子磁盘

**日志子磁盘** 添加到将成为使用脏区日志的卷的一部分的 `plex`。当卷已镜像并且至少有一个日志子磁盘时，将对该卷启用脏区日志。

有关脏区日志的描述，请参阅第 45 页的“脏区日志”。只要涉及到通常的 `plex`，就忽略日志子磁盘，它只用于容纳脏区日志。

---

**注意** 一个 `plex` 只能有一个关联的日志子磁盘。由于要频繁地写入该日志子磁盘，应注意将其放在使用不太多的磁盘上。将日志子磁盘放在大量使用的磁盘上会降低系统性能。

---

可以使用下列命令向现有卷添加日志子磁盘：

```
vxassist addlog volume_name disk
```

此命令自动在日志 `plex` 内为指定的卷创建日志子磁盘。

若要向现有的 `plex` 添加日志子磁盘，请使用此命令：

```
vxsd aslog plex subdisk
```

这里，`subdisk` 是用作日志子磁盘的名称。必须将 `plex` 与镜像卷关联才能使 DRL 生效。

例如，可以使用此命令将名为 `disk02-01` 的子磁盘与名为 `vol01-02` 的 `plex`（它已经与卷 `vol01` 关联）关联：

```
vxsd aslog vol01-02 disk02-01
```



## 分离子磁盘

若要中断子磁盘与其所属的 plex 之间已建立的关系，需将子磁盘与 plex 分离。当删除子磁盘或者在另一个 plex 中使用子磁盘时，子磁盘便被分离。若要分离子磁盘，请使用此命令：

```
vxsd dis subdisk_name
```

例如，使用下列命令，可以将名为 disk02-01 的子磁盘与其当前关联的 plex 分离：

```
vxsd dis disk02-01
```

---

**注意** 用下列命令也可删除子磁盘：

```
vxsd -orm dis subdisk_name
```

---

## 更改子磁盘属性

---

**警告** 更改子磁盘属性时要非常谨慎，应只在必要时才更改。

---

vxedit 命令将子磁盘的属性更改为其它 Volume Manager 对象。若要更改有关子磁盘的信息，请使用此命令：

```
vxedit set field=value ... subdisk_name
```

例如，下面是更改名为 disk02-01 的子磁盘的注释域的命令：

```
vxedit set comment="new_comment" disk02-01
```

使用 vxedit 可更改的子磁盘域有：

- ◆ 名称
- ◆ putil[*n*] 域
- ◆ tutil[*n*] 域
- ◆ len（只有在子磁盘被分离时）
- ◆ 注释

---

**注意** 如果子磁盘已不是 plex 的一部分，在 putil0 域中输入数据可防止将该子磁盘用作 plex 的一部分。

---



## 移动子磁盘

通过移动子磁盘，可以将一个子磁盘的磁盘空间内容复制到另一个子磁盘上。如果移动的子磁盘与 plex 关联，那么原始子磁盘上存储的数据将复制到新的子磁盘上。原始子磁盘与 plex 分离，而新的子磁盘与 plex 关联。该关联处于与源子磁盘相同的 plex 内部偏移。若要移动子磁盘，请使用此命令：

```
vxsd mv old_subdisk_name new_subdisk_name
```

若要使子磁盘移动任务正确执行，必须满足下列条件：

- ◆ 所涉及的子磁盘的大小必须相同。
- ◆ 移动的子磁盘必须是活动 (ENABLED) 卷上的活动 plex 的一部分。
- ◆ 新磁盘不能与任何其它 plex 关联。

## 分割子磁盘

通过分割子磁盘，可以将现有的子磁盘分成两个子磁盘。若要分割子磁盘，请使用此命令：

```
vxsd -s size split subdisk_name newsd1 newsd2
```

这里：

- ◆ *subdisk\_name* 是原始子磁盘的名称
- ◆ *newsd1* 是将要创建的两个子磁盘中第一个子磁盘的名称
- ◆ *newsd2* 是将要创建的第二个子磁盘的名称

-s 选项是指定将要创建的两个子磁盘中**第一个磁盘**大小所必需的。第二个子磁盘占用原始子磁盘所使用的剩余空间。

如果执行任务前原始子磁盘与某个 plex 关联，则所有生成的子磁盘都与同一 plex 关联。

若要将原始子磁盘分割为两个以上的子磁盘，则对所得到的子磁盘按所需次数重复前面的命令。

## 结合子磁盘

通过结合子磁盘，可以将两个或更多现有的子磁盘合并成一个子磁盘。若要结合子磁盘，子磁盘必须在同一磁盘上相邻。如果选定的子磁盘是关联的，它们必须是和同一 plex 关联且在那个 plex 上相邻。若要结合子磁盘，请使用此命令：

```
vxsd join subdisk1 subdisk2 new_subdisk
```

## 执行联机备份

Volume Manager 提供卷设备的快照备份。这是通过 `vxassist` 和其它命令实现的。根据卷内容一致性要求的不同，存在各种执行备份的过程。这些过程具有相同的基本要求：`plex` 足够大，可以存储卷的完整内容。`plex` 可以比所需要的大，但是如果使用的 `plex` 过小，则将导致副本不完整。

推荐的卷备份方法是使用 `vxassist` 命令，它易于使用。`vxassist snapstart`、`snapwait` 及 `snapshot` 任务提供在最小限度地中断用户的情况下，执行卷的联机备份的方法。

`vxassist snapshot` 过程由两个步骤组成：

1. 运行 `vxassist snapstart` 创建快照镜像
2. 运行 `vxassist snapshot` 创建快照卷

---

**注意** 通过使用本节描述的卷备份推荐方法，可以用 `vxassist` 命令创建 RAID-5 卷的快照。

---

`vxassist snapstart` 步骤创建挂载在卷上并与卷同步的只写备份 `plex`。当与卷同步时，备份 `plex` 随时准备用作快照镜像。当新的快照镜像将其状态更改为 `SNAPDONE` 时，表明更新过程已结束。此更改可由 `vxassist snapwait task` 跟踪，它一直等到至少其中一个镜像的状态更改为 `SNAPDONE`。如果挂载进程失败，则删除快照镜像并释放其空间。

一旦快照镜像被同步，它便继续更新直到被分离。之后，可以选择在方便的时间创建一个快照卷作为现有卷的映象。也可以请用户在执行快照所需的短暂时间内（一般少于一分钟）暂时不要使用系统。与创建快照卷所花费的短暂时间相比，创建快照镜像所涉及的时间就显得长了。

通过在具有 `SNAPDONE` 镜像的卷上运行 `vxassist snapshot` 命令完成联机备份过程。该任务分离已完成的快照（它成为一个标准镜像），创建一个新的标准卷并将快照镜像挂接到快照卷。于是，快照成为一个发挥作用的标准镜像，且快照的状态设置为 `ACTIVE`。

如果中断了快照过程，则在启动卷时自动删除快照镜像。

使用下列步骤执行完整的 `vxassist` 备份：

1. 用此命令为卷创建快照镜像：  

```
vxassist snapstart volume_name
```
2. 当完成了 `snapstart` 步骤且镜像处于 `SNAPDONE` 状态时，选择方便的时间完成快照任务。通知用户即将创建快照，请他们保存文件并在那段时间内暂时不要使用系统。



3. 用此命令创建反映原始卷的快照卷：

```
vxassist snapshot volume_name temp_volume_name
```

4. 使用 `fsck`（或某个适合于正在卷上运行的应用程序的实用程序）清除临时卷的内容。例如，可以使用此命令：

```
fsck -y /dev/vx/rdisk/temp_volume_name
```

5. 将临时卷的内容复制到磁带或其它一些适当的备份介质上。

6. 用此命令删除新卷：

```
vxedit -rf rm temp_volume_name
```

## 快速镜像再同步

快速镜像再同步 (FMR) 功能通过提高 VxVM 快照机制的效率，快速有效地执行失效镜像的再同步，从而更好地支持备份和决策支持这样的操作。

### 激活 FMR

当用 `vxassist` 创建新卷时，可以指定用于打开或关闭 FMR 的属性。关键字 `fmr` 和 `fastresync` 都可以用作属性来指定是否在卷上使用 FMR。

若要创建 FMR 激活的卷，请使用如下所示的 `vxassist make` 命令：

```
vxassist make volume_name size fmr=on
```

默认情况下，FMR 设置为 `off`，但是可以在 `vxassist` 默认文件内更改该默认值。

也可以使用 `vxvol` 命令 *打开或关闭* FMR 功能性。若要使用 FMR，必须在创建快照时激活 FMR，并且 FMR 一直保持激活状态直到完成快照。关闭 FMR 将释放指定卷的所有跟踪映像。所有后续的重新挂接将不使用 FMR 工具，而是执行卷的完整再同步。即使稍后打开 FMR，这种情况也会发生。

若要打开 FMR，请使用下列命令：

```
vxvol set fmr=on volume_name
```

若要关闭 FMR，请使用下列命令：

```
vxvol set fmr=off volume_name
```

## 合并快照卷

卷的快照副本可以与原始卷合并。将快照 `plex` 与快照卷分离并将其与原始卷挂接，删除快照卷。该任务再同步卷数据，以使 `plex` 保持一致。

若要将快照与其原始卷合并，请使用此命令：

```
vxassist snapback replica-volume
```

这里，`replica-volume` 是卷的快照副本。

默认情况下，原始 `plex` 内的数据用于合并的卷。相反，若要使用复制卷的数据拷贝，请使用此命令：

```
vxassist -o resyncfromreplica snapback replica-volume
```

## 分离快照卷

可以永久断开快照和其原始卷之间的链接，使快照卷成为一个独立的卷。

若要将快照与其原始卷分离，请使用此命令：

```
vxassist snapclear replica-volume
```

这里，`replica-volume` 是卷的快照副本。

## 显示快照卷的信息

`vxassist snapprint` 命令显示原始卷与它们各自的复制品（快照副本）之间的关联。

`snapprint` 选项的语法是：

```
vxassist snapprint [volume-name]
```

下面显示的是此命令的输出：

| V NAME | USETYPE | LENGTH | RP NAME   | VOLUME    | LENGTH | RRPLEXID  |
|--------|---------|--------|-----------|-----------|--------|-----------|
| v vol  | fsgen   | 2048   | rp vol-05 | SNAP1-vol | 3040   | rp vol-04 |

如果指定了卷，此命令将显示那个卷的输出，如果没有为那个卷启用 `FMR` 映象，则将显示错误信息。否则，将显示磁盘组内所有卷的信息。



## 镜像 VM 磁盘上的卷

镜像 VM 磁盘上的卷使您得以在另一磁盘位置上有一个或更多的卷副本。通过创建卷的镜像副本，可以防止系统在磁盘发生故障时丢失数据。可以在根磁盘上使用该任务，在替换磁盘上创建可使用的根信息的另一个副本。这样，即使根磁盘被破坏，仍能引导系统。

---

**注意** 该任务只镜像连续卷。已镜像的卷或者包含驻留在多个磁盘上的子磁盘的卷被忽略。

---

若要镜像一个磁盘上的卷，确保目标磁盘具有空间与源磁盘相同或比源磁盘多，然后执行下列操作：

1. 从 `vxdiskadm` 主菜单中选择菜单项 6（镜像磁盘上的卷）。

2. 在下列提示中输入希望镜像的磁盘的磁盘名称：

镜像磁盘上的卷  
菜单：VolumeManager/Disk/Mirror

此操作可被用于镜像磁盘上的卷。这些卷可被镜像至另一磁盘或任何可用的磁盘空间上。已镜像的卷将不再镜像，那些由多个子磁盘组成的卷也将不被镜像。

从引导磁盘镜像卷将产生一个备用的引导磁盘。

输入磁盘名称 [`<disk>,list,q,?`] `disk02`

3. 在下列提示中输入目标磁盘名称（该磁盘的大小必须与源磁盘相同或比源磁盘大）：

您可以选择将卷从磁盘 `disk02` 镜像至任何可用磁盘空间，或者镜像至特定磁盘上。若要镜像至特定磁盘，需选择该磁盘的名称。若要镜像至任何可用磁盘空间，选择 "any"。

输入目标磁盘 [`<disk>,list,q,?`]（默认：any）`disk01`

---

**注意** 当创建备用根磁盘时，一定要始终指定目标磁盘。否则，Volume Manager 将选择一个磁盘作为备用根磁盘。但是，系统可能无法从那个磁盘引导。

---

4. 在下列提示下，按回车键创建镜像：

所请求的操作是将磁盘组 `rootdg` 中磁盘 `disk02` 上的全部卷镜像至磁盘 `disk01` 上的可用磁盘空间。

注意：此操作可能需要花费很长时间完成。

继续进行操作吗？ [`y,n,q,?`]（默认：y）

vxdiskadm 显示镜像操作的状态:

```
Mirror volume voltest-bk00 ...
```

磁盘 disk01 的镜像完成。

5. 在下列提示下, 指出是想镜像另一磁盘上的卷 (**y**), 还是返回 vxdiskadm 主菜单 (**n**):

```
镜像另一磁盘上的卷吗? [y,n,q,?] (默认: n)
```

## 移动 VM 磁盘上的卷

在关闭或删除磁盘前, 可能需要将那个磁盘的数据移到系统上的其它磁盘上。为此, 确保目标磁盘有足够空间, 然后执行下列操作:

1. 从 vxdiskadm 主菜单中选择菜单项 7 (从磁盘中移动卷)。
2. 在下列提示下, 输入希望移动的卷所在磁盘的磁盘名称:

从磁盘中移动卷

```
菜单: VolumeManager/Disk/Evacuate
```

使用此菜单操作将任何正在使用一个磁盘的卷移到其它磁盘上。此操作应紧接删除磁盘 (永久地或为替换) 之前使用。可以将卷移动到指定的磁盘列表上, 或者将卷移动到同一磁盘组中任何可用的磁盘空间上。

注意: 简单地将卷移离一个磁盘, 而没有删除该磁盘, 将不能防止在未来的操作中将卷移回到该磁盘上。例如, 连续两次使用移动操作可能将卷从第二个磁盘移回第一个磁盘。

```
输入磁盘名称 [<disk>,list,q,?] disk01
```

在下列显示之后, 可以有选择地指定将卷移动到的磁盘列表。

您现在可以指定一个磁盘列表以将卷移至其上。将指定的所有磁盘介质名 (例如, disk01) 列在同一行上并用空格分开。如果未输入任何磁盘介质名, 则卷将被移到磁盘组中任何可用的磁盘空间上。

在下列提示下, 按**回车键**移动卷:

```
所请求的操作是从磁盘组 rootdg 中的磁盘 disk01 上移走全部卷。
```

注意: 完成此操作可能需要很长时间。

```
继续进行操作吗? [y,n,q,?] (默认: y)
```



当从磁盘中移走卷时，`vxdiskadm` 将显示该操作的状态：

```
Move volume voltest ...
Move volume voltest-bk00 ...
```

在移走全部卷后，`vxdiskadm` 将显示下列成功讯息：

磁盘 `disk01` 的迁移已完成。

3. 在下列提示下，指出是想移动另一磁盘上的卷 (y)，还是返回 `vxdiskadm` 主菜单 (n)：

从另一磁盘中移动卷吗？ [y,n,q,?] (默认：n)



## Volume Manager 集群功能性

### 介绍

本章论述 VERITAS Volume Manager (VxVM) 提供的集群功能性。Volume Manager 包含一个可选的集群功能，它使 VxVM 能够用于集群环境。Volume Manager 中的集群功能性是需要单独获得许可证的功能。

本章讲述以下主题：

- ◆ 集群功能性概述
- ◆ VxVM 集群中的磁盘
- ◆ 脏区日志和集群环境
- ◆ 升级 Volume Manager 集群功能
- ◆ 集群相关的 Volume Manager 实用程序和后台驻留程序
- ◆ 集群和快速镜像再同步 (FMR)

有关与集群相关的出错讯息，请参阅 《VERITAS Volume Manager 参考指南》的错误信息章节。

### 集群功能性概述

Volume Manager 中的集群功能性允许多个主机同时存取和管理受 Volume Manager 控制的特定磁盘集（VM 磁盘）。集群是共享磁盘集的一组主机，集群中的每个主机都被称为一个节点。节点在整个网络中是相连的。如果某个节点发生故障，其它节点仍然可以存取磁盘。Volume Manager 集群功能在所有节点上显示相同的磁盘配置（包括更改）的逻辑视图。

---

**注意** 如果启用了集群支持，Volume Manager 支持每个集群最多四个节点。

---

以下各节提供了有关 Volume Manager 中的集群功能性的更多信息。



## 共享的 Volume Manager 对象

如果启用了集群功能，Volume Manager 对象可由特定集群中的所有节点共享。

Volume Manager 集群功能允许两种类型的磁盘组：

- ◆ **专用磁盘组**，它只属于一个节点。一个专用磁盘组仅由一个系统导入。从物理上讲，可以从一个或多个系统存取专用磁盘组中的磁盘，但实际上仅限于从一个系统存取。
- ◆ **集群共享磁盘组**，它由所有的节点共享。集群共享（或共享）磁盘组由所有的集群节点导入。集群共享磁盘组中的磁盘在物理上必须能从可能加入集群的所有系统存取。

在 Volume Manager 集群中，大多数磁盘组是共享的。但是，根磁盘组 (rootdg) 始终是专用磁盘组。

集群中的所有节点都可以存取共享磁盘组中的磁盘，使得多个集群节点上的应用程序能够同时存取同一个磁盘。取决于许可和磁盘组激活方式的描述，共享磁盘组中的卷可由集群中一个以上的节点同时存取。

节点上的共享磁盘组必须是激活的，来自该节点的应用程序 I/O 才能存取共享磁盘组中的卷。应用程序读取卷或写入卷的能力由磁盘组的激活方式支配。共享磁盘组的有效激活方式是 **排它写**、**共享写**、**只读**、**共享读** 和 **关闭**（或非活动），如表 5 “共享磁盘组的激活方式”。中所示。

表 5. 共享磁盘组的激活方式

|            |                                                                            |
|------------|----------------------------------------------------------------------------|
| <b>排它写</b> | 节点有对磁盘组的排它写存取。任何其它节点都不能激活磁盘组以获得写存取。                                        |
| <b>共享写</b> | 节点有对磁盘组的写存取。                                                               |
| <b>只读</b>  | 节点有对磁盘组的读存取并拒绝集群中所有其它节点的写存取。该节点没有对磁盘组的写存取。在其它节点上激活磁盘组以获得两种写模式中的任何一个的尝试将失败。 |
| <b>共享读</b> | 节点有对磁盘组的读存取。该节点没有对磁盘组的写存取，但其它节点可以获得写存取。                                    |
| <b>关闭</b>  | 节点对磁盘组既没有读存取也没有写存取。但允许在磁盘组上执行查询操作。                                         |

**注意** 磁盘组激活是 VxVM 3.0 版的一项新功能。为了保持与前一版本的兼容性，默认情况下，激活方式对 Volume Manager 实用程序是透明的。在共享写模式中自动激活共享磁盘组。

集群的一些特殊使用，如 HA 应用程序和脱离主机备份，可以利用磁盘组激活直接控制来自集群中不同节点的卷 I/O 能力。第 160 页的“[磁盘组激活](#)”中描述了激活方式的使用。

**说明：**

- ◆ Volume Manager 3.0 中引入的新功能：*条带镜像卷、任务监视器和联机布局*，可用于专用磁盘组，但尚不支持共享磁盘组。
- ◆ 只有原始设备存取是通过 Volume Manager 集群功能执行的。不支持包含文件系统的共享卷。
- ◆ Volume Manager 集群功能目前不支持在集群共享磁盘组中使用 RAID-5 卷。但 RAID-5 卷可用于挂载在集群特定节点上的专用磁盘组中。
- ◆ 如果作为共享导入的磁盘组包含不支持的对象，则逐出该磁盘组。将包含的卷重新组织成支持的布局，然后将该磁盘组作为共享重新导入。

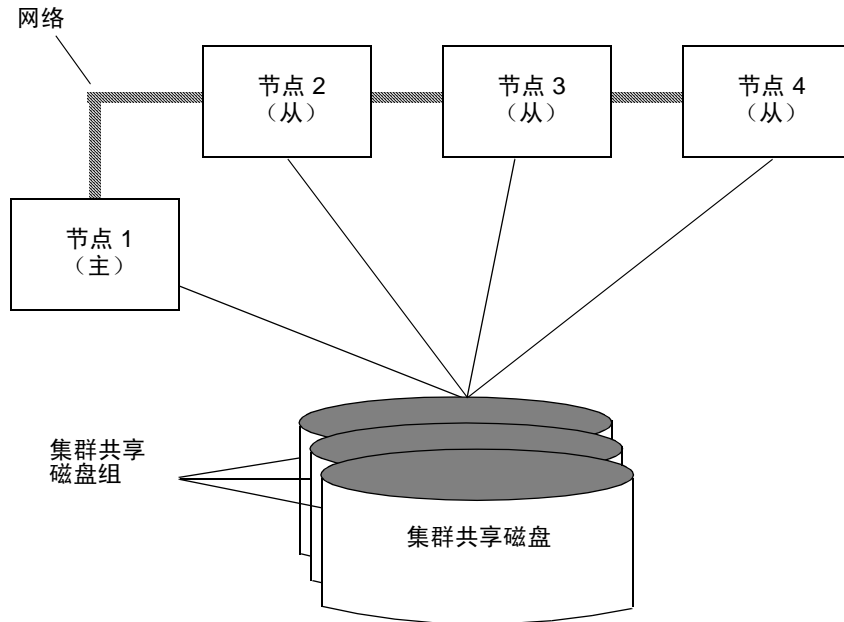
## 集群卷管理的工作机制

Volume Manager 集群功能同外部提供的**集群管理器**一起工作，**集群管理器**是通知 VxVM 所发生的集群成员更改的后台驻留程序。各节点的启动相互独立，并且都有各自的操作系统副本、带集群支持的 VxVM 和集群管理器。当节点**加入**集群时，它就获得了对共享磁盘的存取。当节点**离开**集群时，它就不再具有对那些共享磁盘的存取。系统管理员通过启动节点上的集群管理器将节点加入到集群。

第 156 页的图 24 “[4 节点集群的实例](#)”图解了一个简单的集群排列。所有节点通过网络加入在一起。然后，节点加入到集群共享磁盘组。对于集群管理器，所有的节点都是一样的。然而，Volume Manager 集群功能需要一个节点充当**主节点**，其它节点作为**从属节点**。主节点负责协调某些 Volume Manager 活动。VxVM 软件确定执行主功能的节点（任何一个节点都可以作为主节点），只有当主节点离开集群后此角色才改变。如果主节点离开了集群，从属节点中的一个将成为新的主节点。在图 24 “[4 节点集群的实例](#)”中，节点 1 是主节点，节点 2、节点 3 和节点 4 为从属节点。



图 24.4 节点集群的实例



系统管理员使用 `vxdg` 实用程序将一个磁盘组指定为集群共享磁盘组（有关更多信息，请参见第 166 页的“`vxdg`”）。一旦将某个节点的磁盘组作为集群共享磁盘组导入，其磁盘磁头就标记上了集群 ID。当其它节点加入集群时，它们将识别出该磁盘组是集群共享磁盘组并将其导入。系统管理员可以随时导入或逐出共享磁盘组，该操作以分布式方式在所有节点上发生。

每个物理磁盘都带有一个唯一的磁盘 ID。当集群在主节点上启动时，它导入所有的共享磁盘组（具有 `noautoimport` 属性设置的磁盘组除外）。当从属节点试图加入时，主节点将已导入的磁盘 ID 列表发送给从属节点，从属节点查看是否可以存取所有这些磁盘。如果从属节点不能存取列表上的某个导入磁盘，它就放弃加入集群的尝试。如果它可以存取列表上的所有磁盘，就导入与主节点相同的那套共享磁盘组并加入集群。当一个节点离开集群时，它逐出所导入的全部共享磁盘组，但这些磁盘组仍保留在继续存在的节点上。

对共享磁盘组的任何重配置都通过所有节点的合作来完成。磁盘组的配置更改在所有节点上同时发生并且更改完全相同。这些更改在本质上是原子式的，因此，这些更改要么在所有节点上同时发生，要么根本不发生。

根据激活方式，所有集群成员具有对任何集群共享磁盘组的同时读写存取。（参见第 160 页的“**磁盘组激活**”。）由集群活动节点进行的存取不受任何其它节点的故障影响。只要集群中至少一个节点是活动的，则包含在集群共享磁盘中的数据可用。不管哪个节点存取集群共享磁盘组，该磁盘组的配置看起来都是相同的。运行在每个节点上的应用程序可以同时存取 VM 磁盘上的数据。

---

**注意** VxVM 不对多个节点同时写入共享卷提供保护。这基于任何一致性控制都在应用程序级（例如，使用分布式锁管理器）完成的假设。

---

## 配置和初始化

在任何节点可以第一次加入新集群之前，系统管理员必须提供某些配置信息。此信息是在集群管理器设置期间提供的，正常情况下存储在某种类型的集群管理器配置数据库中。此信息的精确内容和格式取决于集群管理器的特性。VxVM 需要的信息类型如下：

- ◆ 集群 ID
- ◆ 节点 ID
- ◆ 节点的网络地址
- ◆ 端口地址

当节点加入集群时，此信息在节点启动时自动装入该节点上的 VxVM 中。

节点初始化通过集群管理器启动过程生效，该过程在节点上调出各种集群组件（例如，带集群支持的 VxVM、集群管理器和分布式锁管理器）。一旦完成初始化，就可以启动应用程序。系统管理员在将要加入集群的每个节点上调用集群管理器启动过程。

对于集群环境中的 VxVM，初始化操作由装入集群配置信息和在集群中加入节点组成。第一个加入的节点成为主节点，后面的节点（从属节点）加入到主节点。如果两个节点同时加入，则由 VxVM 软件选择主节点。一旦完成特定节点的加入，该节点就具有对共享磁盘的存取。

## 集群重配置

无论何时集群的状态发生更改（节点离开或加入），都将引起集群重配置。每个节点的集群管理器监视集群中的其它节点并在集群成员更改时通知 VxVM。VxVM 于是采取适当的操作。

在集群重配置期间，对共享磁盘的 I/O 操作暂时停止。当重配置完成后，I/O 操作继续进行。因此，应用程序似乎有短暂的停顿。

如果其它操作（例如，Volume Manager 操作或恢复）正在进行，则可能推迟集群重配置，直到其它操作已经完成。卷重配置（稍后将介绍）不与集群重配置同时发生。根据具体的情况，操作可能停顿并在稍后重新启动。在大多数情况下，优先进行集群重配置。然而，如果卷重配置正处于提交阶段，则先完成它。

如果 SunCluster™ 作为集群管理器，则由“vxclust”实用程序协调集群重配置，并提供 VxVM 和 SunCluster™ 之间的通信。SunCluster™ 和 vxclust 一起工作，保证集群重配置中的每一步都按正确的顺序完成。

如果 VCS 作为集群框架，集群重配置将完全在 VxVM 内核中处理。

有关集群重配置的更多信息，请参见第 210 页的“vxclust”和第 164 页的“vxclustadm”。



## 卷重配置

**卷重配置**是在配置中创建、更改和删除 Volume Manager 对象（如磁盘组、卷、镜像等）的过程。在集群中，此过程通过所有节点的合作来完成。卷重配置分布到所有节点，在所有节点上同时发生完全相同的配置更改。

**注意** 卷重配置由主节点启动和协调，因此系统管理员必须运行相应的实用程序，请求更改主节点上的 Volume Manager 对象。

vxconfigd 后台驻留程序在卷重配置中起积极作用。要成功完成重配置，必须在所有节点上运行 vxconfigd。

主节点上的实用程序联系它的本地 vxconfigd 后台驻留程序，该后台驻留程序执行一些本地检查以确保所请求的更改合理。例如，当已经存在同名磁盘组时，创建新磁盘组的尝试将失败。主节点上的 vxconfigd 随后给集群中所有其它节点的 vxconfigd 后台驻留程序发送信息，告诉它们有关更改的详细资料。每个从属节点上的 vxconfigd 于是执行各自的检查。例如，从属节点核实它没有与正在创建的磁盘组同名的专用磁盘组；如果该操作涉及新磁盘，则每个节点都检查它能否存取该磁盘。当所有节点上的全部 vxconfigd 都同意所提出的更改合理时，每个 vxconfigd 就通知它的内核，然后这些内核协同提交或终止事务。在事务可以提交前，所有的内核确保没有任何正在进行中的 I/O。主节点负责启动重配置并协调事务提交。

如果任何节点上的 vxconfigd 在重配置过程中离开，则将通知所有节点，操作将失败。如果有任何节点离开集群，除非主节点已提交操作，否则操作将失败。如果主节点离开集群，新的主节点（原来是从属节点）或者完成操作，或者使操作失败。这取决于它是否从前一个主节点接收到成功完成的通知。如果新的主节点没有接收到此通知，则任何其它从属节点也不会接收到。

如果一个节点在正在执行卷重新配置时尝试加入集群，其结果取决于重配置进行到了哪里。如果还没有涉及内核，则卷重配置暂时停止，并在加入完成后重新启动。如果已经涉及内核，则要等到完成重配置后再进行加入。

当发生错误时（例如从属节点上的检查失败或有节点离开集群），将向实用程序返回该错误，并向主节点上的控制台发出一条信息，指出发生错误的节点。

## 节点关闭

系统管理员可以通过在特定的节点上调用集群管理器的关闭程序在该节点上关闭集群。该操作在集群应用程序停止后终止集群组件。VxVM 支持 *干净节点关闭*，这是在共享卷的所有存取停止之后，节点干净地离开集群的能力。主机仍然运行，但集群应用程序不能在主机上运行。

Volume Manager 集群功能为每个卷维护全局状态信息。这使 VxVM 能够在节点崩溃时精确确定需要恢复的卷。当节点由于崩溃或以其它某种非干净方式离开集群时，VxVM 确定哪些卷可能有未完成的写操作，并由主节点将那些卷再同步。如果所有那些卷的脏区日志都是活动的，则将使用它们的脏区日志。

干净节点关闭操作应与停止全部集群应用程序的过程配合使用，或在该过程之后使用。根据集群应用程序及其关闭过程的特性，成功地关闭可能需要很长一段时间（几分钟到几小时）。例如，很多应用程序都有“排干”的概念，即在退出之前不接受新工作，而是完成所有正在进行的工作。此过程可能会花很长时间，比如在激活了一个长时间运行的事务时。

当调用 VxVM 关闭过程时，在正在关闭的那个节点上检查所有共享磁盘组中的所有卷，然后或者继续关闭或者失败：

- ◆ 如果共享磁盘组中的所有卷都关闭了，则 VxVM 使它们不可由应用程序使用。由于所有节点都知道这些卷在离开的节点上是关闭的，因此不执行再同步。
- ◆ 如果共享磁盘组中有任何卷是打开的，VxVM 关闭过程将返回故障。可以重试关闭过程，直到成功。在此操作中没有限制检查 — 超时检查是用于检验集群应用程序是否不再活动的服务。

**注意** 节点一旦成功关闭，便离开了集群。除非节点再加入集群，否则不可能存取共享卷。

由于关闭可能是一个很长的过程，因此在关闭进行当中可以进行其它的重配置。正常情况下，关闭尝试推迟到其它的重配置完成之后。不过，如果关闭很早以前就已经开始了，也可能首先完成关闭。

### 节点异常终止

如果节点没有干净地离开，这可能是由于主机崩溃，也可能是由于某个集群组件在紧急情况下决定使此节点离开。随后发生的集群重配置将调用 VxVM 异常终止函数。此函数会立即尝试暂停所有对共享卷的存取，但此操作确实要等到该磁盘上的 I/O 完成之后。

还没有启动的 I/O 操作失败，共享卷被删除。正在存取共享卷的应用程序因此失败，并出现错误信息。

在节点异常终止或崩溃后，必须恢复共享卷（通过继续存在的节点或通过随后的集群重新启动），因为很可能有不同步的镜像。

### 集群关闭

当集群中的所有节点都离开时，在下次启动集群时必须决定是否恢复共享卷。如果所有的节点都干净地离开，则不需要恢复。如果最后一个节点干净地离开，且由前面节点的不干净离开引起的再同步已经完成，也不需要恢复。但是，如果最后一个节点没有干净地离开，或者前面的离开引起的再同步没有完成，则必须执行恢复。





## VxVM 集群中的磁盘

集群中的节点必须始终就磁盘的状态达成一致。具体说来，如果一个节点无法写入特定的磁盘，则在写操作的结果返回给调用者之前，所有节点都必须停止对该磁盘的存取。因此，如果一个节点不能联系某个磁盘，则它应接触其它节点以检查该磁盘的状态。如果该磁盘发生故障，则没有节点可以访问它，所有节点会一致同意与该磁盘分离。如果该磁盘没有发生故障，而是来自一些节点的访问路径发生故障，则节点无法就该磁盘的状态达成一致。必须有相应的策略来解决这种矛盾。

### 磁盘分离策略

为了解决上述矛盾，提供了下列策略（为磁盘组设置）。可以通过使用 `vxedit (1M)` 命令来设置这些策略。

在共享磁盘组的全局连通性策略中，如果集群中的任何节点报告磁盘故障，分离就在集群范围内（全局）发生。这是默认策略。

在局部连通性策略中，如果发生磁盘故障，故障仅限于看到故障的特定节点中。注意，系统会尝试与集群中的所有节点通信，以确定磁盘的可用性。如果所有节点都报告磁盘有问题，则会发生集群范围内的分离。

### 磁盘组激活

磁盘组激活控制来自集群中不同节点的卷 I/O 能力。如果已经以冲突方式在集群的另一个节点上激活了磁盘组，则不可能在特定的节点上再激活它。

表 6 “允许的和冲突的激活方式”总结了共享磁盘组的允许的和冲突的激活方式，如下所示：

表 6. 允许的和冲突的激活方式

| 集群中磁盘组的激活方式 | 尝试在另一个节点上将磁盘组激活为 ... |     |    |     |
|-------------|----------------------|-----|----|-----|
|             | 排它写                  | 共享写 | 只读 | 共享读 |
| 排它写         | 失败                   | 失败  | 失败 | 成功  |
| 共享写         | 失败                   | 成功  | 失败 | 成功  |
| 只读          | 失败                   | 失败  | 成功 | 成功  |
| 共享读         | 成功                   | 成功  | 成功 | 成功  |

若要将激活方式置于用户的控制之下，必须创建默认文件 `/etc/default/vxdg`，并包含下列行：

```
default_activation_mode=activation-mode
```

这里的激活方式是：`off`、`shared-write`、`shared-read`、`read-only` 或 `exclusive-write`。





当创建或导入共享磁盘组时，将以指定的方式激活它。当一个节点加入到集群时，将以指定的方式激活全部共享磁盘组。

**说明：**

- ◆ 当使用默认文件启用激活时，建议使集群的所有节点上的默认文件都完全相同。否则，激活的结果将是无法预见的。
- ◆ 如果默认激活模式是除关闭外的其它任何一种，且集群的另一节点已经以冲突方式激活了某个磁盘组，则在集群加入、磁盘组创建或导入之后的激活可能失败。

## 脏区日志和集群环境

*脏区日志 (DRL)* 是卷的一个可选属性，在系统故障后提供镜像卷的快速恢复。在集群共享磁盘组中支持脏区日志。本节简要介绍 DRL，并描述 DRL 在集群环境中的行为方式。

DRL 跟踪由于向镜像卷的 I/O 写入而更改的区域，并使用这一信息，只恢复卷中需要恢复的那部分。DRL 将卷逻辑地划分为一系列连续区域，并维护一个脏区日志，该脏区日志中包含状态位，表示卷的每个区域。*日志子磁盘* 用于存储启用了 DRL 的卷的脏区日志。带 DRL 的卷至少有一个日志子磁盘，该子磁盘与卷的一个 plex 关联。

在向卷中写入任何数据之前，要写入的区域在日志中被标记为 **dirty**。如果先前干净的日志区域由于写操作变 **dirty**，在写操作可以发生之前，日志同步地写入磁盘。在完成向镜像的写操作之后，日志区域又变为干净。在系统重新启动时，Volume Manager 只恢复在脏区日志中标记为 **dirty** 的卷区域。

在集群环境中，Volume Manager 的 DRL 实现与通常的实现稍有不同。下面各节略述一些区别，并讨论集群环境实现的几个方面。

### 日志格式和大小

与非集群环境中的 VxVM 一样，集群脏区日志位于镜像卷的日志子磁盘中。

VxVM 脏区日志有一个恢复映射和一个活动映射。但是，集群脏区日志有一个恢复映射和多个活动映射（每个对应一个集群节点）。与 VxVM 不同，集群功能将恢复映射放在日志的开始。

集群脏区日志的大小通常比 VxVM 脏区日志大，因为它必须容纳集群中所有节点的活动映射外加一个恢复映射。脏区日志中每个映射的大小是一个或多个完整的块。vxassist 自动负责分配足够大的脏区日志。

日志的大小取决于卷的大小和节点数。日志必须足够大，可以容纳全部映射（每个节点一个映射，外加一个恢复映射）。对应于每两千兆字节的卷大小，每个映射应为一个块的长度。在一个含两个节点的集群中，一个两千兆字节的卷需要三块的日志大小（每个映射一个块），这是日志大小的最小值。在一个含四个节点的集群中，一个四千兆字节的卷需要十块的日志大小，依此类推。



当将节点添加到现有集群中时，需要分离并删除现有的 DRL 日志（使用 `vxplex -o rm dis`），然后再重新创建它（使用 `vxassist addlog`）。此操作增加了日志的大小，使它们可以容纳附加节点的映射。

## 兼容性

除了增加集群特有的幻数以外，集群环境中的 DRL 头与非集群中的相同。

可以将 VxVM 磁盘组（和它的卷）作为集群环境中的共享磁盘组导入，反之亦然。不过，导入磁盘组的脏区日志可能被认为无效，并可能导致完全恢复。

如果共享磁盘组由不带集群支持的 VxVM 系统导入，VxVM 将认为共享卷的日志无效，并将进行完全卷恢复。完成此恢复后，Volume Manager 将使用集群功能的脏区日志。

Volume Manager 集群功能可以在非共享的 VxVM 卷上执行 DRL 恢复。但是，如果将 VxVM 卷移到有集群支持的 VxVM 系统中并作为共享导入，脏区日志可能太小，无法容纳集群中的所有节点。集群功能因此会将日志标记为无效，且无论如何都进行完全恢复。同样，将 DRL 卷从两节点集群移到四节点集群可能导致日志太小，集群功能将通过完全卷恢复解决这一问题。在两种情况中，系统管理员都负责分配足够大的新日志。

## DRL 在集群环境中的工作机制

当集群的一个或多个节点崩溃时，DRL 需要能够处理在崩溃发生时由那些节点使用的所有卷的恢复。在初始的集群启动时，全部活动映射都并入恢复映射；这是在 `volume start` 操作过程中完成的。

不允许崩溃的（即“dirty”地离开集群的）节点重新加入集群，除非它们的 DRL 活动映射已并入所有受影响的卷的恢复映射。在恢复实用程序将崩溃节点的活动映射与恢复映射进行比较并进行必要的更新后，节点才能重新加入集群并恢复对卷的 I/O（改写活动映射）。在这段时间内，其它节点可以继续执行 I/O。

VxVM 内核跟踪已崩溃的节点。如果在特定的时间有多个节点恢复操作正在集群中进行，它们各自的恢复和恢复映射更新彼此竞争。因此 VxVM 内核跟踪 DRL 恢复状态的更改并防止 I/O 操作冲突。

主节点为每个卷执行 DRL 恢复映射更新的易失跟踪，并防止多个实用程序同时更改恢复映射。

## 升级 Volume Manager 集群功能

滚动式升级功能允许管理员升级在集群中运行的 Volume Manager 版本，而不必关闭整个集群。要安装在集群中运行的 Volume Manager 的新版本，系统管理员可以从集群中拉出一个节点，将其升级，然后再将此节点加入回集群中。对集群中的每个节点都进行同样的操作。

从 3.1 版开始的每个 VxVM 发行版都有一个关联的 *集群协议版本号*。这与发行版本号不同。集群协议版本存储在 `/etc/vx/volboot` 文件中。在重新安装 VxVM 的过程中，`/etc/vx` 目录中不存在 `volboot` 文件，`vxdctl init` 创建该文件，并将集群协议版本设置为支持的最高版本。

一个新的 VxVM 发行版支持两种版本的集群协议。较低版本对应于现有的 VxVM 发行版。它有一套固定的功能和通信协议。较高的版本对应于新的 VxVM 发行版，包含一套新的上述功能。如果新的 VxVM 发行版在功能或协议上没有任何改动，集群协议版本就保持不变。例如，只有错误修复或很小的改动的情况。在这种情况下，不需要执行 `vxdctl upgrade`。

在滚动式升级过程中，每个节点都必须关闭，且必须安装带最新集群协议版本的 VxVM 发行版。所有具有新的 VxVM 发行版的节点继续使用较低版本。安装了新的集群协议版本的从属节点尝试加入集群。如果主节点上没有使用新的集群协议版本，它将拒绝加入，并将当前的集群协议版本提供给从属节点。从属节点使用主节点提供的集群协议版本重试加入。如果这时加入失败，说明主节点上的集群协议版本超出了正在加入的从属节点支持的协议版本范围。在这种情况下，系统管理员必须通过每个 VxVM 的中间发行版升级集群，以达到支持的最新集群协议版本。

所有节点都升级为最新集群协议版本，并且新功能变为可用。

一旦所有节点都安装了新的发行版，就必须在主节点上运行 `vxdctl upgrade` 命令，切换到较高的集群协议版本。

## 集群相关的 Volume Manager 实用程序和后台驻留程序

创建或修改下列实用程序和 / 或后台驻留程序，使之可用于集群环境中的 Volume Manager:

- ◆ `vxclust`
- ◆ `vxclustadm`
- ◆ `vxconfigd`
- ◆ `vxdg`
- ◆ `vxdisk`
- ◆ `vxrecover`



- ◆ vxctl
- ◆ vxstat

以下各节包含关于在集群环境中如何使用这些实用程序中的每一种的信息。有关这些实用程序中的任何一种的进一步详细资料，请参阅其帮助手册。

## vxclust

每次有集群重配置时，集群中的每个现有节点都按编排好的步骤一步步运行 `vxclust` 实用程序。集群管理器实用程序确保在所有节点上同时执行同一步。只有当上一步在所有节点上都已完成时，一个特定的步骤才会开始。在重配置的每一步，`vxclust` 确定 Volume Manager 集群功能下一步应该做什么。在通知 VxVM 下一步做什么之后，`vxclust` 就等待结果（成功、失败或重试），并将其传达给集群管理器。

### 注意：

- ◆ 大多数 Volume Manager 命令要求超级用户特权。
- ◆ `vxclust` 作为集群管理器与 SunCluster™ 一起工作。

如果节点在特定的超时期内没有对 `vxclust` 请求做出响应，该节点就异常终止。然后 `vxclust` 根据情况确定是重新启动重配置还是放弃。如果导致重配置的是无法修复的本地错误，`vxclust` 就放弃。如果一个节点由于另一个节点已经离开而不能完成操作，则这个继续存在的节点超时。在这种情况下，`vxclust` 请求重配置，期望另一个节点会离开。如果没有其它节点离开，`vxclust` 将使本地节点离开。

如果一个重配置步骤失败，`vxclust` 就将一个错误返回给集群管理器。集群管理器可能会决定异常终止节点并使它立即离开集群。任何正在进行中的对共享磁盘的 I/O 都失败，对共享磁盘的存取也停止。

`vxclust` 在被告知集群中的更改时确定采取什么操作。如果需要新的主节点（由于先前的主节点失败），`vxclust` 就决定哪个节点成为新的主节点。

## vxclustadm

---

**注意** `vxclustadm` 仅适用于 VCS。

---

`vxclustadm` 命令在集群的一个节点上启用或禁用集群的 VxVM 功能性。在 VCS 集群启动和关闭过程中，从联机和脱机脚本中调用此命令。

`startnode` 选项将集群配置信息传递给 VxVM 内核。响应该命令时，内核与配置后台驻留程序 `vxconfigd` 执行初始化。

`stopnode` 选项在一个节点上停止集群功能性。它等到所有未解决的 I/O 完成，所有应用程序关闭共享卷。

`abortnode` 选项在一个节点上异常终止集群活动。这是紧急关闭，异常终止所有未完成的共享卷上的 I/O。

`nodestate` 选项确定集群的节点状态。

有关更多信息，请参阅 `vxclustadm(1M)`。

## vxconfigd

Volume Manager 配置后台驻留程序 `vxconfigd` 维护 VxVM 对象的配置。`vxconfigd` 从基于 SunCluster 的 `vxclust` 实用程序或运行 VCS 时的内核中接收与集群相关的指令。每个节点上保留一份单独的 `vxconfigd` 副本；这些副本通过网络实用程序互相通信。对于集群的每个节点，Volume Manager 实用程序与在那个节点上运行的 `vxconfigd` 通信；实用程序不尝试与其它节点上的 `vxconfigd` 后台驻留程序加入。在集群启动过程中，`vxclust` (SunCluster) 或内核 (VCS) 指示 `vxconfigd` 开始集群操作，并告诉它是作为主节点还是从属节点。

当一个节点为集群操作做好初始化时，系统会通知 `vxconfigd` 此节点将要加入集群，并提供下列信息（从集群管理器配置数据库）：

- ◆ 集群 ID
- ◆ 节点 ID
- ◆ 主节点 ID
- ◆ 节点的角色
- ◆ 每个节点的 `vxconfigd` 网络地址

在主节点上，`vxconfigd` 设置共享配置（例如，导入共享磁盘组）并告诉 `vxclust` (SunCluster) 或内核 (VCS) 从属节点何时准备加入。

在从属节点上，当从属节点可以加入集群时，系统将通知 `vxconfigd`。当从属节点加入集群时，`vxconfigd` 和 Volume Manager 内核与其在主节点上的副本通信，以便设置共享配置。

当节点离开集群时，`vxconfigd` 后台驻留程序通知所有其它节点上的内核。然后主节点执行任何必需的清除。如果主节点离开集群，内核就选择一个新的主节点，且所有节点上的 `vxconfigd` 后台驻留程序都将接收到有关该选择的通知。

`vxconfigd` 也参与卷重配置。有关 `vxconfigd` 在卷重配置中的作用的信息，请参见第 158 页的“[卷重配置](#)”。

## vxconfigd 恢复

可以随时停止和 / 或重新启动 Volume Manager `vxconfigd` 后台驻留程序。如果 `vxconfigd` 停止，直到 `vxconfigd` 重新启动后，卷重配置才能发生，其它节点也才能加入集群。在集群中，从属节点的 `vxconfigd` 后台驻留程序总是加入到主节点的 `vxconfigd` 后台驻留程序。因此，停止任何集群节点上的 `vxconfigd` 后台驻留程序都是



不明智的。

如果由于某个原因停止了 `vxconfigd`，根据停止的后台驻留程序所在的节点采取不同的操作：

- ◆ 如果 `vxconfigd` 在从属节点上停止，则主节点不采取任何操作。当 `vxconfigd` 在从属节点上重新启动时，从属节点的 `vxconfigd` 会尝试重新加入主节点并重新获得关于共享配置的信息。（内核的共享配置视图不受影响，对共享磁盘的存取也不受影响。）在节点 `vxconfigd` 已成功地重新加入主节点之前，几乎没有关于共享配置的信息，任何显示或修改共享配置的尝试都可能失败。具体说来，如果列出了共享磁盘组（使用 `vx dg list`），它们将被标记为 `disabled`；当重新加入成功完成时，它们将被标记为 `enabled`。
- ◆ 如果在主节点上停止 `vxconfigd`，从属节点上的 `vxconfigd` 会定期尝试重新加入主节点。直到在主节点上重新启动 `vxconfigd` 后这种尝试才会成功。在这种情况下，从属节点 `vxconfigd` 关于共享配置的信息没有丢失，因此配置显示是准确的。
- ◆ 如果在主节点和从属节点上都停止了 `vxconfigd`，则直到它们都重新启动了 `vxconfigd`，且都重新加入后，从属节点才会显示准确的配置信息。

当 `vxclust` (SunCluster) 或内核 (VCS) 注意到 `vxconfigd` 在一个节点上停止时，将重新启动 `vxconfigd`。

---

**注意** 使用 VxVM 时，`vxconfigd` 的 `-r reset` 选项重新启动 `vxconfigd` 并从头开始创建全部状态。此选项当节点在集群中时不可用，因为如果在这种情况下使用此选项，将导致集群信息丢失，`vxconfigd` 将无法启动。

---

## vx dg

`vx dg` 实用程序管理 Volume Manager 磁盘组。可以使用 `vx dg` 指定集群可共享的磁盘组。`vx dg` 的 `-s` 选项用于将磁盘组作为“共享”初始化或导入。

如果已经运行集群软件设置了集群，可以使用下列命令创建共享磁盘组：

```
vx dg -s init diskgroup [medianame=]accessname
```

这里 `diskgroup` 是磁盘组名称；`medianame` 是为磁盘选择的管理名称；而 `accessname` 是磁盘存取名（或设备名称）。

可以使用 `vx dg -s import` 将磁盘组作为共享导入。如果在运行集群软件之前设置磁盘组，可以使用下列命令将磁盘组导入到集群排列中：

```
vx dg -s import diskgroup
```

这里 `diskgroup` 是磁盘组的名称或 ID。在随后的集群重启动中，会自动地将该磁盘组作为共享导入。注意，在调用此命令之前，可能必须逐出该磁盘组（使用 `vx dg deport diskgroup`）。

通过 `vx dg deport` 逐出磁盘组，然后使用 `vx dg import diskgroup` 将其导入，可以将磁盘组由共享转换为专用。



**注意** 系统无法断定磁盘是否是共享的。为了在处理可由多个系统存取的磁盘时保护数据的完整性，系统管理员在将磁盘添加到磁盘组时必须小心，一定要使用正确的指定值。当管理员尝试将物理上没有共享的磁盘添加到共享磁盘组时，如果集群只有一个节点，且在该节点上此磁盘是可存取的，则 Volume Manager 允许在该节点上进行这种操作。但是，其它节点将无法加入集群。此外，如果管理员尝试同时将同一磁盘添加到位于两个节点上的不同磁盘组中，结果就无法确定。因此应该只在一个节点上处理全部配置。

`vxdg` 有一个强制选项 (`-f`)，可以使用它强制导入磁盘组，或强制将磁盘添加到磁盘组。

**注意** 使用强制选项 (`-f`) 时应小心，而且只有在系统管理员已完全意识到可能的后果时才使用。

当重新启动集群时，出于下列某种原因，VxVM 可能拒绝自动导入磁盘组：

- ◆ 由于磁盘的硬件错误，磁盘组中的一个磁盘已经不可存取。在这种情况下，系统管理员可以用下面的强制选项重新导入磁盘组：

```
vxdg -s -f import diskgroup
```

- ◆ 磁盘组中的磁盘挂接到的一些节点当前不在集群中，因此磁盘组不能存取它的所有磁盘。在这种情况下，强制导入是不安全的，不应尝试（因为可能导致不一致的镜像）。

如果 VxVM 不将磁盘添加到现有磁盘组（因为该磁盘与磁盘组的其它磁盘未挂接在同一节点上），系统管理员可以按如下所示强制添加该磁盘：

```
vxdg -f adddisk -g diskgroup [medianame=]accessname
```

也可以使用 `vxdg` 列出共享磁盘组。下列命令为每个磁盘组显示一行信息：

```
vxdg list
```

此命令的输出如下：

| 名称     | 状态             | ID                  |
|--------|----------------|---------------------|
| rootdg | enabled        | 774215886.1025.teal |
| group2 | enabled,shared | 774575420.1170.teal |
| group1 | enabled,shared | 774222028.1090.teal |

共享磁盘组由 `shared` 标志表示。

下列命令为每个共享磁盘组显示一行信息：

```
vxdg -s list
```

此命令的输出如下：



| 名称     | 状态             | ID                  |
|--------|----------------|---------------------|
| group2 | enabled,shared | 774575420.1170.teal |
| group1 | enabled,shared | 774222028.1090.teal |

下列命令显示关于一个特定磁盘组的信息，包括该磁盘组是否为共享的：

```
vxdg list diskgroup
```

这里 *diskgroup* 是磁盘组名称。

(磁盘组 group1 的) 主节点上的 vxdg list group1 的输出如下：

```
Group: group1
dgid: 774222028.1090.teal
import-id: 32768.1749
flags: shared
copies: nconfig=default nlog=default
config: seqno=0.1976 permlen=1456 free=1448 templen=6 loglen=220
config disk c1t0d0s2 copy 1 len=1456 state=clean online
config disk c1t1d0s2 copy 1 len=1456 state=clean online
log disk c1t0d0s2 copy 1 len=220
log disk c1t1d0s2 copy 1 len=220
```

注意， flag: 字段设置为 shared。在从属节点上，此命令的输出稍有不同。

## vxdisk

vxdisk 实用程序管理 Volume Manager 磁盘。可以使用 vxdisk 确定一个磁盘是否是集群共享磁盘组的一部分，如下所示：

```
vxdisk list accessname
```

这里 *accessname* 是磁盘存取名（或设备名称）。

此命令（对设备 c1t0d0s2 而言）的输出如下：

```
Device: c1t0d0s2
devicetag: c1t0d0
type: sliced
clusterid: cvm
disk: name=disk01 id=774215890.1035.teal
group: name=group1 id=774222028.1090.teal
flags: online ready autoconfig shared imported
pubpaths: block=/dev/dsk/c1t0d0s4 char=/dev/rdisk/c1t0d0s4
privpaths: block=/dev/dsk/c1t0d0s3 char=/dev/rdisk/c1t0d0s3
version: 3.1
iosize: min=512 (bytes) max=248 (blocks)
public: slice=4 offset=0 len=2050272
private: slice=3 offset=1 len=2015
update: time=778564769 seqno=0.1614
headers: 0 248
configs: count=1 len=1456
```



```

logs: count=1 len=220
Defined regions:
config priv 000017-000247[000231]:copy=01 offset=000000 enabled
config priv 000249-001473[001225]:copy=01 offset=000231 enabled
log priv 001474-001693[000220]:copy=01 offset=000000 enabled

```

注意, `clusterid`: 字段设置为 `cvm` (集群的名称), 而 `flag`: 字段包含一个 `shared` 项。当节点未加入时, `flag`: 字段包含 `autoimport` 标志而非 `imported`。

## vxrecover

`vxrecover` 实用程序在磁盘更换后恢复 `plex` 和卷。

当节点离开集群时, 它可能使一些镜像状态不一致。 `vxrecover` 实用程序在处于此状态的所有卷上执行恢复。 `-c` 选项使 `vxrecover` 对集群共享磁盘组中的所有卷执行恢复。 `vxclust` 在必要时自动调用 `vxrecover -c`。

---

**注意** 当 `vxrecover` 是活动的时, 系统性能可能会有些下降。

---

## vxctl

`vxctl` 实用程序管理卷配置后台驻留程序 `vxconfigd` 的某些方面。可以使用 `-c` 选项请求集群信息。可以使用 `vxctl` 确定 `vxconfigd` 是否启用和 / 或正在运行, 如下所示:

```
vxctl -c mode
```

根据具体的情况, 显示的输出类似于下面这样:

```

方式: 已启用: 集群活动 —— MASTER
方式: 已启用: 集群活动 —— SLAVE
方式: 已启用: 集群不活动
方式: 已启用: 集群活动 —— 未设置规则

```

---

**注意** 如果禁用 `vxconfigd`, 将不显示集群信息。

---

有关 `vxctl` 的完整描述, 请参阅 `vxctl(1M)` 帮助手册。

`vxctl` 列出 集群协议版本和集群协议范围。当集群中的所有节点都用新的集群协议更新后, 用下列命令升级整个集群:

```
vxctl upgrade
```

使用 `vxctl protocolversion` 命令检查现有的集群协议版本。例如:

```
vxctl protocolversion
集群在协议 10 上运行
```

`vxctl protocolrange` 命令显示当前 VxVM 发行版支持的最大和最小集群协议版本, 例如,



```
vxdctl protocolrange
最小协议版本 :10, 最大协议版本 : 20
```

vxdctl list 命令显示在节点上运行的集群协议版本。以下是 vxdctl list 命令输出的实例：

```
Volboot 文件
版本: 3/1
序列号: 0.19
集群协议版本: 20
主机 ID: giga
入口:
```

vxdctl support 命令显示节点支持的最大和最小协议版本及当前协议版本。以下是 vxdctl support 命令输出的实例：

```
支持信息:
vold_vrsn: 11
dg_minimum: 60
dg_maximum: 70
kernel: 10
protocol_minimum: 10
protocol_maximum: 20
protocol_current: 20
```

## vxstat

vxstat 返回指定对象的统计信息。在集群环境中，vxstat 从集群的所有节点中收集统计信息。统计信息提供全部节点对请求对象的总的使用情况。如果指定本地对象，则返回其本地使用情况。

vxstat 允许调用者有选择地指定节点的子集：

```
vxstat -g diskgroup -n node[,node...]
```

这里 *node* 是整数。如果提供了逗号分隔的节点列表，vxstat 显示列表中节点的统计信息的总和。

在下列实例中，vxstat 被指示获得节点 2 卷 vol1 的统计信息：

```
vxstat -g group1 -n 2 vol1
```

产生的输出可能类似于下面这样：

| 类型  | 名称   | 操作   |    | 块      |    | 平均时间 (毫秒) |     |
|-----|------|------|----|--------|----|-----------|-----|
|     |      | 读取   | 写入 | 读取     | 写入 | 读取        | 写入  |
| vol | vol1 | 2421 | 0  | 600000 | 0  | 99.0      | 0.0 |

vxstat 也可以获得并显示整个集群的统计信息，如下所示：

```
vxstat -b
```

所有节点的统计信息加在一起。例如，如果节点 1 做了 100 个 I/O 而节点 2 做了 200 个 I/O，则 vxstat -b 将返回 300。



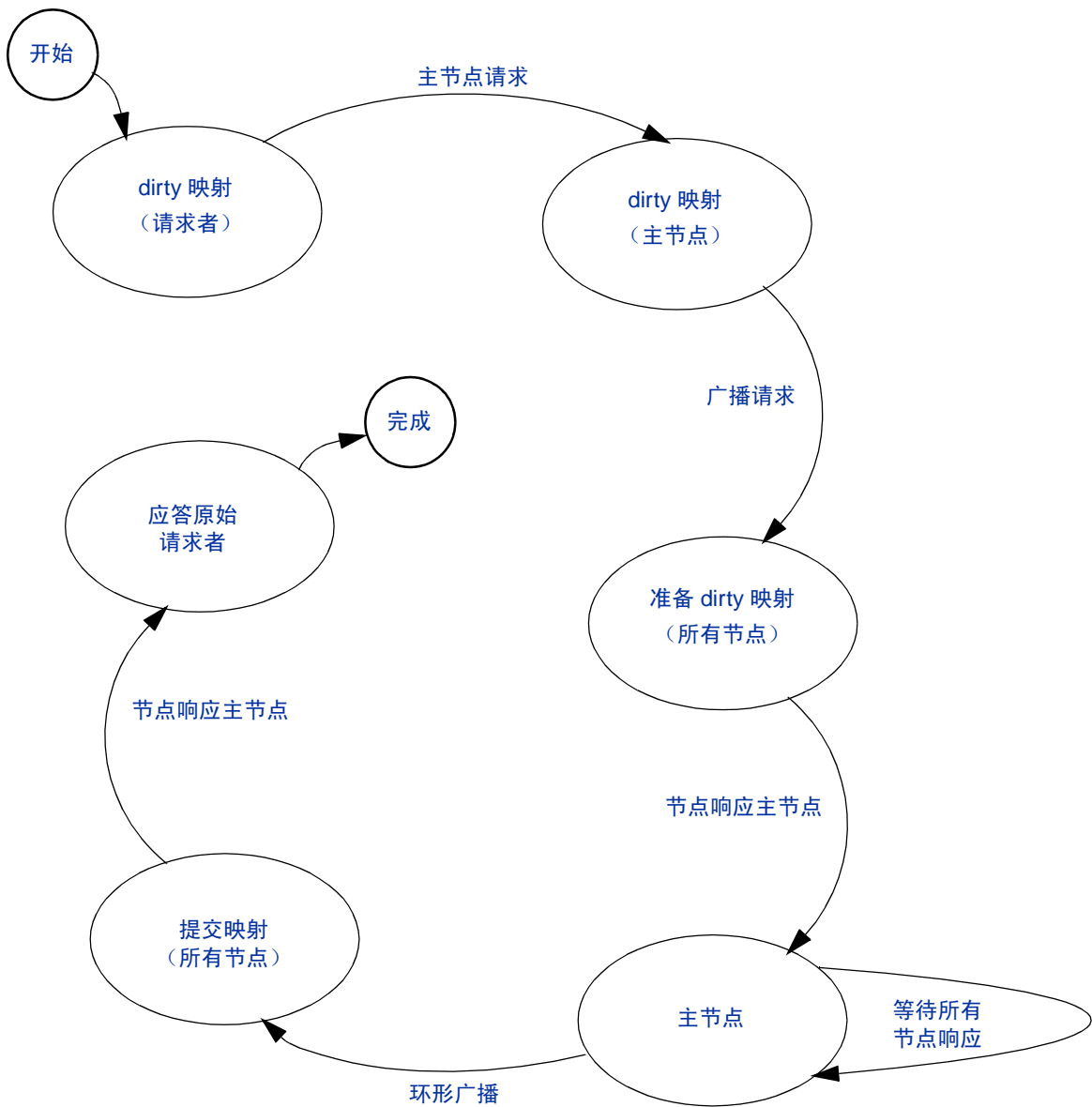
## 集群和快速镜像再同步 (FMR)

支持共享卷的快速镜像再同步 (FMR)。更新映射 (FMR 映射) 分布在整个集群中。在集群环境中，很少讨论非永久性映射的存在，因为集群中只需要有一个节点必须是活动的，FMR 映射就可用。一个节点崩溃不会导致 FMR 映射丢失。

映射更新的应用受主节点的支配。当主节点将更新分布到所有节点时，所有更新或者应用于所有节点，或者哪个节点也不应用。主节点为任何更新的应用编排两阶段提交。请参见第 172 页的中的图 25 “位图集群化”。



图 25. 位图集群化



## 介绍

VERITAS Volume Manager 保护系统免受磁盘故障影响，并帮助您从磁盘故障中恢复。本章将描述恢复过程，并提供信息帮助您防止因磁盘故障而引起的数据或系统存取损失。还将描述可能的 plex 和卷状态。

有关系统保护的更多信息，请参见第 27 页的“[Volume Manager 初始化](#)”和第 30 页的“[系统设置](#)”。

本章讲述以下主题：

- ◆ [UNIX 引导进程](#)
- ◆ [可能的根 \(/\)、swap 和 usr 配置](#)
- ◆ [故障和恢复过程](#)
- ◆ [热重定位和引导磁盘故障](#)
- ◆ [重新添加和替换引导磁盘](#)
- ◆ [重新挂接磁盘](#)
- ◆ [重新安装恢复](#)
- ◆ [Plex 和卷状态](#)
- ◆ [RAID-5 卷恢复](#)
- ◆ [杂项 RAID-5 操作](#)

## UNIX 引导进程

除非 `autoboot` 标志已在由固件使用的非易失性存储区域中设置，否则 Sun SPARC 系统会提示 `boot` 命令。安装有 PROM 早期版本的机器的提示符与 PROM 较新的 V2 和 V3 版的提示符不同。PROM 的新版本也称为 OpenBoot PROM (OBP)。这两种类型的 PROM 的 `boot` 命令的语法不同：

```
ok boot [OBP names] [filename] [boot-flags]
```

*OBP names* 指定开式引导 PROM 指示。例如，在桌面 SPARC 系统中，指示



```
/sbus/esp@0,800000/sd@3,0:a
```

表示 SCSI 磁盘 (sd) 位于目标 3, SCSI 总线上的逻辑单元为 0, 以及 esp 主机适配器插在 0 插槽中。

**注意** 通过 Volume Manager, 可以使用引导磁盘别名。这些别名可以采用 Volume Manager 提供的名称格式 (如: vx-rootdisk 或 vx-disk01) 或操作系统提供的名称格式 (如: disk1)。可以通过在 OpenBoot OK 提示下使用下列命令来查看可能的可引导设备列表: devalias。

*filename* 是 boot 程序的一个独立程序的名称。默认情况下, 从 root 分区引导 /kernel/unix。可以在命令行中指定其它程序 (如: /stand/diag)。固件的某些版本允许将默认的文件名保存在系统的非易失性存储区域中。

boot 程序将 -a 标志解释为 “询问我”, 并提示您输入要引导的独立程序名。然后, -a 标志被传送给独立程序。

**注意** 运行具有根目录可置性的 Volume Manager 的系统不使用 -a 标志提供的默认值进行引导。有关对 boot -a 的正确响应, 请参见第 181 页的 “/etc/system 副本可用”。

boot 程序不解释标志。boot 程序将所有的引导标志传送给 *filename* 标识的文件。引导。有关默认独立程序 /kernel/unix 的可用选项的信息, 请参见 kernel (1) 和 kadb (1M) 帮助手册。

## 发生故障后引导

如果已镜像引导磁盘, 则可以在主引导磁盘发生故障时, 使用替换引导磁盘来引导系统。若要在主引导磁盘发生故障后引导系统, 请按下列步骤进行:

1. 在 OpenBoot 命令提示符下使用 devalias 命令检查有别名的 VM 磁盘。

适合根磁盘镜像的磁盘用下列名称列出: vx *medianame*, 此处的 *medianame* 是带有候选根文件系统的磁盘的磁盘介质名称。

2. 请输入命令:

```
ok boot alias_name
```

此处的 *alias\_name* 是选定磁盘的别名。

如果选定的磁盘包含无效的根镜像, 则 vxconfigd 显示 “镜像不可用” 这样的错误声明, 并列岀所有有效的用于替换的可引导磁盘。

## 可能的根 (/)、swap 和 usr 配置

在安装过程中，根、swap 和 usr 文件系统可能有不同的配置。可能出现下列情况：

- ◆ usr 是 root 下的一个目录，但没有为它分配单独的分区。在这种情况下，当封装根磁盘并使其受 Volume Manager 控制时，usr 成为 rootvol 的一部分。
- ◆ usr 在根磁盘上的一个独立分区中。在这种情况下，为 usr 分区创建一个独立卷，vxmirror 在目标磁盘上镜像 usr 卷。
- ◆ usr 在根磁盘以外的一个独立分区中。在这种情况下，只在用 Volume Manager 封装磁盘时才为 usr 分区创建一个卷。注意，在这种情况下，如果 usr 分区因某种原因而变得无法存取，即使封装根磁盘和拥有根卷的镜像也没有用。建议您封装包含 usr 分区的磁盘和根磁盘，并镜像 usr、rootvol 和 swapvol 卷以获得系统的最大可用性。

rootvol 卷必须在 rootdg 磁盘组中。有关 rootvol 和 usr 卷限制的信息，请参见第 50 页的“启动时卷的限制”。

Volume Manager 允许您将 swap 分区放在任何磁盘上，在引导进程的早期阶段，不需要初始的 swap 区域。在默认情况下，Volume Manager 安装选择选定磁盘上的分区 0 作为根分区，选择分区 1 作为 swap 分区。但是，swap 分区也可以在根磁盘以外的分区上。在这种情况下，建议您封装该磁盘并为 swap 卷创建镜像。如果不这样做，则对 swap 分区的损坏将最终导致系统崩溃。或许还可以引导系统，但具有 swapvol 卷的镜像可以防止系统故障。

## 修复卷上的根 (/) 或 /usr 文件系统

如果根 (/) 或 /usr 文件系统变得无法使用，则建议从网络安装的根文件系统或有效备份引导。备份必须包含根磁盘中所有有关的文件系统分区。此外，在封装根磁盘前，还需打印输出根磁盘分区表。

当根文件系统或 /usr 文件系统定义在镜像卷上时，执行此任务更困难。当 Volume Manager 稍后进行引导并假定镜像被合理同步化时，更改作为镜像的基础之一的一个分区可能会造成破坏。

该问题有下列两种应对方法：

- ◆ 最简单的一种方法是安装根文件系统或 /usr 文件系统的 **plex**，修复它，然后卸载它，最后使用 **dd** 将修复的 **plex** 复制到所有其它的 **plex**。但这容易发生错误。
- ◆ 另一种方法是从有效的备份磁带恢复系统。将在下面说明该过程。该过程不需要从底层 CD-ROM 安装操作系统。

下面描述的过程提供了一种在根磁盘及其镜像都受到损坏时使用的简单、有效和可靠的恢复方法。



## 从备份磁带恢复 Volume Manager 根磁盘（和根镜像）

该过程假定您具有：

- ◆ 原始 Volume Manager 根磁盘上所有文件系统的最近的完整备份。
- ◆ 已安装的用于替换受到物理损坏的初始引导磁盘的新引导磁盘。

该过程要求重新安装 Volume Manager 根磁盘。若要防止磁盘上的数据损失涉及到重新安装中，应当在重新安装过程中只涉及 Volume Manager 根磁盘。

安装的若干自动选项无需管理员确认便可存取除根磁盘外的磁盘。因此，明智的做法是在开始该过程前将所有其它磁盘（包含卷）从系统断开。断开其它磁盘可以确保其不受重新安装的影响。该过程完成后重新连接这些磁盘。

### 恢复根磁盘

在下面的过程中，假定（新）引导磁盘为 `c0t0d0`，并且需要分别恢复 `/` 和 `/usr` 文件系统，`s0` 和 `s6`。

1. 从 CDROM 引导操作系统
2. 使用 `format` 命令在（新）引导磁盘 (`c0t0d0`) 上创建完全相同的分区来包含初始引导磁盘上先前的文件系统。

---

**注意** 由于需要恢复文件系统，因此最多可以有 7 个分区。另外，当需要即刻重新封装该磁盘时，您最多只能有 5 个分区，因为磁盘上的专用区域和公共区域需要 2 个分区。

---

3. 将 `/dev/rdisk/c0t0d0s0` 安装在 `/a/root` 上。从磁带恢复根文件系统。使用 `installboot` 在 `/a/root` 上安装引导块设备。
4. 将 `/dev/rdisk/c0t0d0s6` 安装在 `/a/usr` 上。从磁带恢复 `/usr` 文件系统。
5. 按照如下修改恢复的根文件系统：
  - ◆ `touch /a/root/etc/vx/reconfig.d/state.d/install-db`
  - ◆ 通过删除下列两行修改 `/a/root/etc/system`：

```
rootdev:/pseudo/vxio@0:0
set vxio:vol_rootdev_is_volume=1
```
  - ◆ 通过用标准磁盘设备 `/dev/dsk/c0t0d0s0` 和 `/dev/dsk/c0t0d0s6` 替换 Volume Manager 卷设备项 `/dev/vx/dsk` 来修改 `/a/root/etc/vfstab`。



6. 从（新）引导磁盘重新启动系统。这允许引导磁盘认为尚未安装 Volume Manager。

该过程的下一步取决于在旧的 rootdg 中是否有根磁盘镜像：

- ◆ 如果在旧的 rootdg 中存在未用作根磁盘镜像的其它磁盘，转至步骤 7。
- ◆ 如果在旧的 rootdg 中只有根磁盘镜像，转至步骤 8。

7. 如果在旧 rootdg 中包含没有用作根磁盘镜像的其它磁盘，则按照这些步骤引入旧的 rootdg（减去 Volume Manager 认为已经发生故障的引导磁盘）并建立新引导磁盘。

- a. 删除下列涉及安装但不再需要的文件：

```
rm -r /etc/vx/reconfig.d/state.d/installldb
```

- b. 启动 VxVM I/O 后台驻留程序：

```
vxiod set 10
```

- c. 以禁用模式启用 VxVM 配置后台驻留程序：

```
vxconfigd -m disable
```

- d. 初始化 vxconfigd 后台驻留程序：

```
vxdctl init
```

- e. 启用 vxconfigd：

```
vxdctl enable
```

上述步骤应引入旧的 rootdg（减去 VxVM 认为发生故障的根磁盘）。

- f. 使用 vxedit 命令（或 Volume Manager Storage Administrator）删除旧的根磁盘卷和根磁盘本身。

- g. 使用 vxdiskadm 命令封装（新）引导磁盘并初始化充当根磁盘镜像的任何磁盘。完成所需的重新启动后，将根磁盘镜像到根磁盘镜像。

8. 如果在旧的 rootdg 中只有根磁盘镜像：

运行 vxinstall 命令封装（新）引导磁盘，并初始化根磁盘镜像。

完成所需的重新启动后，将根磁盘镜像到根磁盘镜像。



## 备份和恢复根文件系统

备份根文件系统是一个好主意，这样便可在其受到损坏时将其恢复。

如果您正在使用 `ufs` 文件系统，则可以使用下列命令备份根文件系统：

```
/usr/lib/fs/ufs/ufsdump [dump-options] /dev/vx/rdisk/rootvol
```

在发生故障后可以按如下恢复根文件系统：

1. 从 CD-ROM 或网络安装的根文件系统引导，然后运行 Volume Manager（请参见第 175 页的“[修复卷上的根 \(/\) 或 /usr 文件系统](#)”）。
2. 使用下列命令安装和恢复根文件系统：

```
newfs /dev/vx/rdisk/rootvol
mount /dev/vx/dsk/rootvol /mnt
cd /mnt
/usr/lib/fs/ufs/ufsrestore [restore-options]
```

## 故障和恢复过程

由于有多种类型的故障可以妨碍系统引导，因此可以采用相同的基本过程来启动系统。在系统引导失败后，应首先尝试识别屏幕上留下的故障证据，然后再尝试修复该问题（例如，接通意外断电的驱动器）。如果该问题无法修复（如：引导磁盘上的数据错误），则从替换引导磁盘（包含根卷的镜像）引导系统以便修复损坏或替换故障磁盘。

本节概述一些可能的故障并提供有关校正操作的指令。

### UNIX 分区时的故障

一旦装入了引导程序，便会尝试通过正常的 UNIX 分区信息来存取根磁盘。如果该信息受到损坏，则引导程序将发生故障，并出现下列错误消息：

```
File just loaded does not appear to be executable
```

如果该消息在引导过程中出现，则应当从替换引导磁盘来引导系统。在引导时，大多数磁盘驱动程序在控制台上显示有关故障磁盘的无效 UNIX 分区信息的错误。该消息与下列消息类似：

```
WARNING: unable to read label
WARNING: corrupt label_sdo
```

这表示故障是由无效磁盘分区引起的。可以尝试按第 184 页的“[重新添加故障引导磁盘](#)”中所述的方法重新添加磁盘。但是，如果重新挂接失败，则随后需要按第 185 页的“[替换故障引导磁盘](#)”中所述的方法将磁盘替换。

## 存取引导设备的故障

在引导进程早期，如果立即进行系统初始化，可能会出现与下列消息类似的消息：

```
SCSI device 0,0 is not responding
Can't open boot device
```

这表示系统 PROM 无法从引导驱动器读取引导程序。出现该问题的一般原因可能是：

- ◆ 引导磁盘未通电。
- ◆ SCSI 总线未终止。
- ◆ 存在某种控制器故障。
- ◆ 磁盘正发生故障并锁定了总线，使任何磁盘不能向控制器标识自己，并使控制器以为没有挂接的磁盘。

诊断此问题的第一步是仔细检查 SCSI 总线上的每个部件是否正常。如果磁盘断电或总线未终止，则更正该问题并重新启动系统。如果其中一个磁盘发生故障，则从总线中移去该磁盘并替换该磁盘。

如果没有发现任何硬件问题，则错误可能是由于引导磁盘上的数据错误引起。为了修复该问题，尝试从替换引导磁盘（包含根卷的镜像）引导系统。如果无法从替换引导磁盘引导系统，则可能仍然有某些类型的硬件问题。与此相类似，如果用替换引导磁盘取代故障磁盘后仍无法引导系统，这也表明有硬件问题。

## 因 /etc/vfstab 中的不正确项而引起的故障

当封装根磁盘并使其受 Volume Manager 控制时，作为正常封装进程的一部分，为磁盘上的所有分区创建了卷。Volume Manager 修改 /etc/vfstab 来使用相应的卷而不用磁盘分区。在手动编辑 /etc/vfstab 文件时应小心。最重要的项是那些对应于 / 和 /usr 的项。在 Volume Manager 安装前便已存在的 vfstab 保存在 /etc/vfstab.prevm 中。

### /etc/vfstab 中损坏的 / 项

如果 / 在 /etc/vfstab 中的项丢失或不正确，则系统将以单用户模式引导。显示与下列消息类似的消息：

```
File just loaded does not appear to be executable
```

建议此刻运行 fsck：

```
fsck /dev/vx/rdisk/rootvol
```

在引导进程的此刻，/ 还未安装 read/write。由于 /etc/vfstab 中的项或不正确或已删除，使用下列命令手动将 / 作为 read/write 来安装：

```
mount -o remount /dev/vx/dsk/rootvol
```



将 / 作为 read/write 来安装后，退出 shell。系统提示输入运行级。对于多用户模式，输入运行级 3：

```
ENTER RUN LEVEL (0-6,s 或者 S): 3
```

系统引导后，为 / 恢复 /etc/vfstab 中的项。

### **/etc/vfstab 中损坏的 /usr 项**

只有当 /usr 位于一个独立的磁盘分区上时， /etc/vfstab 才有 /usr 的一个项。封装包含 /usr 分区的磁盘后，Volume Manager 将更改 /etc/vfstab 中的项以使用相应的卷。

在丢失 /etc/vfstab 中 /usr 的项的情况下，系统将无法引导（即使有 /usr 卷的镜像）。在这种情况下，从 CD-ROM 引导系统并恢复 /etc/vfstab。（请参见第 175 页的“修复卷上的根 (/) 或 /usr 文件系统”。）

## **因缺少或损坏的 /etc/system 而引起的故障**

---

**注意** 不要编辑 /etc/system 中由 Volume Manager 添加的任何项。所有 Volume Manager 项都用 \*vxvm\_START 和 \*vxvm\_END 括起。

---

在更改之前，明智的做法是在根文件系统中制作 /etc/system 的副本，然后再对其进行更改。如果对新 /etc/system 文件的更改不正确，则保存的系统文件随后可被指定给 boot 程序。若要将保存的系统文件指定给 boot 程序，请用下列命令引导系统：boot -a。当系统提示输入系统文件的名称时，请输入保存的系统文件的路径。

### **/etc/system 副本不可用**

如果 /etc/system 文件受到损坏，且保存的系统文件副本不可用，则无法用 Volume Manager 的根目录可置性功能来引导系统。如果 /usr 不是一个卷，则可以不用 Volume Manager 根目录可置性（即不用将 rootvol 作为 /）来引导系统。

若要不用 Volume Manager 根目录可置性来引导系统，请按照第 175 页的“修复卷上的根 (/) 或 /usr 文件系统”中所概述的步骤来进行：

1. 从 CD-ROM 启动 Volume Manager。有关更多信息，请参见《VERITAS Volume Manager 安装指南》。
2. 运行 fixmountroot 命令。
3. 建立并安装 /tmp/rootvol。

然后，可以编辑文件 /tmp/rootvol/etc/system 并进行任何其它必需的修复。

引导系统后，建立 /etc/system 中的下列项：

```

* vxvm_START
rootdev:/pseudo/vxio@0:0
set vxio:vol_rootdev_is_volume=1
* vxvm_END

```

还应强制装入根镜像磁盘所需的所有驱动程序。为此，编辑文件 `/etc/system` 以便使其包含每个驱动程序的下列格式行：

```
forceload:drv/driver_name
```

可以通过在 `/dev/dsk/root_device` 上列一个长表来获得这些磁盘的驱动程序名称。驱动程序名称的一个实例为 `io-unit`。

### `/etc/system` 副本可用

如果 `/etc/system` 文件受到损坏，且保存的 `etc/system` 文件副本可用，则可以用 Volume Manager 根目录可置性来引导系统。

若要用 Volume Manager 根目录可置性引导系统，则使用下列命令来引导系统并响应提示问题（按回车键接受所有提示的默认值，根设备名除外）：

```

ok boot -a
.
.
Rebooting with command: -a
Boot device: /iommu/sbus/espdma/esp/sd@5,0 File and args: -a
Enter filename [/kernel/unix]:
Name of system file [/etc/system.sav]:
Name of default directory for modules [/kernel /usr/kernel]:
Enter name of device instance number file [/etc/path_to_inst]:
root file system type [ufs]:
Enter physical name of root device
[/iommu.....]:/pseudo/vxio@0:0

```

### `/etc/system` 不可用且 `/usr` 是卷

如果 `etc/system` 文件受到损坏或丢失，并且其备份副本不可用，而且 `/usr` 是一个卷，必须从 CD-ROM 引导系统（使用第 175 页的“[修复卷上的根 \(/\) 或 /usr 文件系统](#)”中概述的步骤）。一旦完成此工作，安装根卷并编辑其上的 `etc/system` 文件。创建 `etc/system` 文件中的下列项：

```

* vxvm_START
rootdev:/pseudo/vxio@0:0
set vxio:vol_rootdev_is_volume=1
set vxio:vol_swapdev_is_volume=1
* vxvm_END

```

还应强制装入根镜像磁盘所需的所有驱动程序（如前面所述）。更改后，从 `root` 分区（系统文件也在此分区恢复）重新启动系统。



## 从不可用或失效的 plex 引导系统引起的故障

如果磁盘在系统运行时不可用，则驻留在该磁盘上的卷的任何镜像都将失效。这意味着该磁盘上的数据与对应于该卷的其它镜像不一致。在引导进程中，在获得该卷的一个完整配置前，系统只存取根卷的一个副本（引导磁盘上的副本）。

如果结果表明用于引导的该卷的 plex 失效，则必须从包含有效 plex 的替换引导磁盘来引导系统。该问题有可能发生，例如，如果在关闭初始引导磁盘情况下从 Volume Manager 建立的可引导磁盘来引导系统。系统会正常引导，但驻留在关闭的磁盘上的 plex 失效。如果系统在重新打开初始引导磁盘的情况下从该磁盘重新启动，则系统使用失效的 plex 进行引导。

如果引导磁盘上 Volume Manager 磁头中的错误使 Volume Manager 不能正确标识磁盘，则可能会发生另一个问题。在这种情况下，Volume Manager 不知道该磁盘的名称。这个问题是因为 plex 与磁盘名称相关联而引起的，所以在未标识磁盘上的任何 plex 都不可用。

如果根磁盘上的故障影响到根卷 plex，也会发生问题。在下次引导尝试中，系统仍期望使用故障根 plex 来引导。如果在发生故障时已镜像根磁盘，则可以指定引导系统的替换根磁盘（带有有效的根 plex）。

如果发生这些情况中的任何一种，则在作为引导序列的 init 进程的部分进行系统配置时，Volume Manager 实用程序 vxconfigd 将会注意到该情况。vxconfigd 将显示描述错误的消息并显示处理错误的方法。例如，如果磁盘 rootdisk 上根卷 rootvol 的 plex rootvol-01 无效，vxconfigd 将显示下列消息：

```
vxvm:vxconfigd: 警告: 根卷的 Plex rootvol-01 失效或者不可用。
vxvm:vxconfigd: 错误: 系统引导磁盘没有有效的根 plex
请从下列磁盘之一启动:
磁盘: disk01 设备: c0t1d0s2
vxvm:vxconfigd: 错误: 系统启动失败
系统关闭。
```

这将通知管理员：命名为 disk01 的替换引导磁盘包含根 plex 的可用副本，应使用该副本来引导系统。当显示此消息时，应从替换引导磁盘重新启动系统。

一旦引导了系统，则需要确定所发生的问题。如果只是引导磁盘上的 plex 失效，则系统在引导时将自动捕获失效的 plex。另一方面，如果磁盘上的专用区域有问题或磁盘产生故障，则需要重新添加或替换磁盘。

如果引导磁盘上的 plex 不可用，则会收到 Volume Manager 实用程序描述该问题的邮件。确定问题的另一方法是用 vxdisk 实用程序列出磁盘。在上面的实例中，如果问题出在 rootdisk 专用区域的故障（如：由于介质故障或意外地改写了磁盘上 Volume Manager 的专用区域），则 vxdisk list 将显示下列内容：

| 设备       | 类型     | 磁盘       | 组      | 状态              |
|----------|--------|----------|--------|-----------------|
| -        | -      | rootdisk | rootdg | 故障 位于: c0t3d0s2 |
| c0t1d0s2 | sliced | disk01   | rootdg | 联机              |

## 热重定位和引导磁盘故障

如果引导（根）磁盘发生故障，并且引导（根）磁盘已被镜像，则热重定位将自动尝试用新镜像替换产生故障的根磁盘镜像。为此，热重定位使用仍存在的根磁盘镜像在备用磁盘或有足够空闲空间的磁盘上创建新镜像。这可以确保始终至少有可用于引导系统的两个根磁盘镜像。热重定位后台驻留程序还调用 `vxbootsetup` 实用程序，它用新镜像将磁盘配置为可引导磁盘。

如果 `rootdg` 磁盘组没有足够备用或空闲空间来容纳故障根磁盘中的卷，则根磁盘的热重定位可能失败。`rootvol` 和 `swapvol` 卷需要相邻的磁盘空间。如果无法将故障根磁盘上的根卷和其它卷重定位到同一新磁盘，则可将这些卷重定位到不同的磁盘。

`rootvol` 和 `swapvol` 卷的镜像必须是磁柱对准的，这样便只有在有足够空间的磁盘上创建它们的卷镜像，从而使其子磁盘能够在其磁柱边界上开始和结束。如果这些磁盘不可用，则热重定位将失败。

## 重新添加和替换引导磁盘

对引导系统不很关键的数据只能由 `Volume Manager` 在系统完全正常运转后进行存取，所以该数据不必位于特定的区域中。`Volume Manager` 可以找到这些数据。但是，对引导系统关键的数据必须放在可引导磁盘上的特定区域中，以便引导进程查找。

在某些系统上，磁盘控制器在进程中执行控制器特有的操作，并且系统 BIOS 限制此关键数据的位置。

磁盘发生故障时，可以采取下列两种方法来更正问题：

- ◆ 如果是暂态错误或可更正性错误，则可以重新使用同一磁盘；这称为 *重新添加* 磁盘。在某些情况下，重新格式化故障磁盘或进行磁盘表面分析来重建替换扇区映射便足以使磁盘可再次使用及作为候选磁盘重新添加。
- ◆ 如果磁盘确实发生了故障，则应将其替换。

下面章节描述如何重新添加或替换故障引导磁盘。





## 重新添加故障引导磁盘

除使用相同的物理磁盘外，重新添加磁盘与替换磁盘的过程相同。需要重新添加的磁盘通常已经被分离。这意味着 Volume Manager 已经检测到磁盘故障并已停止存取该磁盘。

**注意** 系统可以使用不同于实例的 *设备名或路径*。有关设备名的更多信息，请参见第 1 页的“[了解 Volume Manager](#)”。

例如，假想有 disk01 和 disk02 两个磁盘的系统，这两个磁盘通常在引导中分别作为磁盘 c0t0d0s2 和 c0t1d0s2 映射到系统配置中。某个故障导致了 disk01 分离。这可以通过使用带有下列命令的 vxdisk 实用程序来列出磁盘而得到确认：

```
vxdisk list
```

vxdisk 显示下列（实例）列表：

| 设备       | 类型     | 磁盘     | 组      | 状态                |
|----------|--------|--------|--------|-------------------|
| c0t0d0s2 | sliced | -      | -      | 错误                |
| c0t1d0s2 | sliced | disk02 | rootdg | 联机                |
| -        | -      | disk01 | rootdg | 故障 位于: c0b0t0d0s0 |

注意，磁盘 disk01 没有与其关联的设备，其状态为故障，且指出了与磁盘分离的设备。也有可能根本不列出设备 c0b0t0d0s0。在磁盘完全发生故障，并且磁盘控制器未在总线上（对使用总线的系统而言）检测到该故障时便会发生这种情况。

在某些情况下，vxdisk list 的输出可以不同。例如，如果引导磁盘有与 UNIX 分区表关联且无法更正的故障。则可能有一个丢失的根分区无法更正，但在 Volume Manager 专用区域没有错误。vxdisk list 命令的输出显示下列（实例）列表：

| 设备       | 类型     | 磁盘     | 组      | 状态 |
|----------|--------|--------|--------|----|
| c0t0d0s2 | sliced | disk01 | rootdg | 联机 |
| c0t1d0s2 | sliced | disk02 | rootdg | 联机 |

但是，由于无法用已描述的过程来更正错误，因此磁盘被视为失败。在这种情况下，必需手动将故障磁盘从其设备分离。使用 vxdiskadm 实用程序的“为替换而删除磁盘”功能来完成此工作（有关 vxdiskadm 的更多信息，请参见 vxdiskadm (1M) 帮助手册）。一旦磁盘从设备分离，可以执行更正问题的任何特殊过程（如：重新格式化设备）。

若要重新添加磁盘，使用 vxdiskadm 实用程序的“替换出错的或已删除的磁盘”功能来替换磁盘，并选择 *相同* 设备作为替换磁盘。请使用前述实例，用设备 c0t0d0s2 或 c0b0t0d0s2 替换 disk01（对使用总线的系统而言）。

如果在镜像引导磁盘发生故障时启用热重定位，将尝试创建新镜像并从故障引导磁盘删除故障子磁盘。如果在热重定位成功之后重新添加磁盘也获得成功，那么受磁盘故障影响的根和 / 或其它卷将不再存在于重新添加的磁盘上。此时，可以调用 vxunreloc 以将热重定位子磁盘移回到新替换的磁盘上。



## 解除重定位封装子磁盘到新磁盘

封装引导磁盘时，根文件系统和其它系统区域（如：引导磁盘上的交换设备）被制成卷。由于 Volume Manager 使用部分交换区域创建专用区域，因此它将置于磁盘中央。当将磁盘初始化（而非封装）成 Volume Manager 磁盘时，将在磁盘的开始处创建专用区域。

当封装的引导磁盘发生故障时，所有的子磁盘将被热重定位到其它磁盘。作为热重定位进程的部分，初始磁盘名称和偏移存储在子磁盘记录中。在用有一个有同样存储容量的磁盘替换故障引导磁盘后，它被“初始化”并添加到磁盘组。可以运行 `vxunreloc` 以自动将所有子磁盘移回磁盘。但是，初始化的磁盘和封装的磁盘之间的磁盘布局差异将影响为每个解除重定位的子磁盘计算磁盘中偏移的方式。必须使用 `-f` 选项将子磁盘移到磁盘上，但不能按照精确的偏移量移动。如果替换磁盘比初始的引导磁盘至少大 2M，那么将所有子磁盘按照精确的偏移量移到磁盘将可以获得成功。

`vxunreloc` 在将所有子磁盘移到磁盘上后便使新的磁盘变成可引导磁盘。

注意：通常将转储设备配置成根磁盘的交换分区。无论何时将交换磁盘从一个磁盘移到（通过热重定位、解除重定位或手动）另一磁盘，都必须将转储设备重新配置到新磁盘。

对于 Solaris 2.6：转储设备存储在 `dumpfile` 结构中。若要检查设备的设置，请使用下列命令：

```
echo dumpfile+0x10/s | adb -k /dev/ksyms /dev/mem
```

可以通过重新启动来更改设置，该设置将转储设备配置到第一交换分区。

对于 Solaris 2.7 和后续版本：

可以使用 `dumpadm` 命令查看和设置转储设备。获取有关详细资料，请键入：

```
man dumpadm
```

## 替换故障引导磁盘

当要替换引导磁盘时，必须先将系统从备用引导磁盘中引导出来。如果故障磁盘未从其设备分离，则应使用 `vxdiskadm` 的“为替换而删除磁盘”功能手动将其分离。有关 `vxdiskadm` 的更多信息，请参见 `vxdiskadm(1M)` 帮助手册。一旦磁盘分离，关闭计算机并替换硬件。

替换磁盘的存储容量至少应与被替换的磁盘上使用的存储容量相同。替换磁盘必须足够大，以便使用于存储子磁盘的磁盘区域可以以当前的磁盘偏移容纳初始磁盘的所有子磁盘。若要确定替换磁盘的最小大小，需要确定在故障磁盘上已使用的空间大小。

若要近似估计替换磁盘的大小，请使用下列命令：

```
vxprint -st -e 'sd_disk="diskname"'
```

根据所得的输出，将列出的最后一个磁盘在 `DISKOFFS` 和 `LENGTH` 栏下的值相加。总数是 512 字节的倍数。将总和除以 2，得到单位为千字节的总数。



**注意** 制造商所公布的磁盘大小通常并不表示可用容量。而且，某些制造商公布的容量单位是百万字节，而不是兆字节，它们并不相等。

---

一旦查找到了替换磁盘，关闭好系统并替换必需的硬件。在硬件替换完成后引导系统。使用 `vxdiskadm` 的“替换出错的或已删除的磁盘”功能，用添加的新设备替换故障磁盘。

## 重新挂接磁盘

如果磁盘发生完全故障并且不可能进行热重定位，或在某些磁盘驱动程序已卸载或不可装载（导致磁盘进入故障状态）的情况下启动 **Volume Manager** 时，可以重新挂接磁盘。如果问题已修复，则可以使用 `vxreattach` 命令重新挂接磁盘而无 `plex` 标志为失效状态。但是，重新挂接必须在启动磁盘上的所有卷之前进行。

`vxreattach` 命令作为 `vxdiskadm` 菜单中磁盘恢复的部分以及在引导进程期间被调用。如果可能，`vxreattach` 将故障磁盘介质记录重新挂接到设备名相同的磁盘上。重新挂接将在其以前所属的同一磁盘组中发生，并且将保留其初始的磁盘介质名称。

重新挂接发生后，便可能不再需要恢复。如果磁盘故障的初始（或另一个）原因仍然存在，则重新挂接将失败。

命令 `vxreattach -c` 将检查重新挂接是否可能，但却不执行任何操作。相反，它会显示可以重新挂接磁盘的磁盘组和磁盘介质名称。

有关 `vxreattach` 命令的更多信息，请参阅 `vxreattach(1M)` 帮助手册。

## 重新安装恢复

如果根（引导）磁盘的所有副本受到损坏，或某些关键文件因文件系统损坏而损失，则必需重新安装。当发生其中任何一种类型的故障时，必须重新安装整个系统，因为当前没有从备份中恢复根文件系统的方法。

如果发生这些类型的故障，尝试保留尽可能多的初始 **Volume Manager** 配置。未直接涉及故障的任何卷可能被保存。所以不必重新配置任何已保存的卷。

### 一般重新安装信息

本节描述用于重新安装 **Volume Manager** 的过程以及在发生故障后如何保存尽可能多的初始配置。

系统重新安装将破坏用于重新安装的任何磁盘上的内容。

在重新安装中将删除所有与 **Volume Manager** 相关的信息。删除的数据包括含有磁盘标识符和 **Volume Manager** 配置副本的删除磁盘专用区域中的数据。该信息的删除使磁盘无法作为 **Volume Manager** 磁盘使用。

在重新安装中总会涉及系统根磁盘。也会涉及其它磁盘。如果使根磁盘受 Volume Manager 控制，则无论在 Volume Manager 安装中还是在后来的封装中，其上的磁盘和任何卷或镜像都将在重新安装中损失。在重新安装中涉及的任何其它磁盘或删除和替换的磁盘都将损失 Volume Manager 配置数据（包括卷和镜像）。

如果磁盘（包括根磁盘）在发生故障前不受 Volume Manager 控制，则在重新安装时不会损失 Volume Manager 配置数据。可以按照“磁盘和磁盘组”中描述的过程替换任何其它磁盘。

虽然简化了重新安装后的恢复进程，但不使根磁盘受 Volume Manager 控制增加了需要重新安装的可能性。通过使根磁盘受 Volume Manager 控制并创建根磁盘内容的镜像，可以消除许多需要重新安装系统的问题。

当需要重新安装时，唯一被保存的卷是驻留在未直接涉及故障和重新安装的磁盘上的卷，或是在该磁盘上有副本的卷。根磁盘上的任何卷和其它涉及故障和 / 或重新安装的磁盘都将在重新安装中损失。如果这些卷的备份副本可用，则卷可以在重新安装后恢复。在某些系统上，root、stand 和 usr 文件系统例外，它们无法从备份中恢复。

## 重新安装和重新配置过程

若要重新安装系统和恢复 Volume Manager 配置，请按照此过程进行。这些步骤将在以下几节中进行详细描述：

1. 为安装准备系统。

这包括替换任何故障磁盘或其它硬件和分离任何未在重新安装中涉及的磁盘。

2. 安装操作系统。

通过重新安装基础系统和任何其它非相关的 Volume Manager 软件包来完成此工作。

3. 安装 Volume Manager。

添加 Volume Manager 软件包，但不执行 vxinstall 命令。

4. 恢复 Volume Manager 配置。

5. 清除 Volume Manager 配置。

这包括恢复受故障或重新安装影响的卷中的任何信息和重新创建系统卷（rootvol、swapvol、usr 和其它系统卷）。



## 为重新安装准备系统

为防止不涉及重新安装的磁盘上的数据丢失，在重新安装过程中应只涉及根磁盘。

---

**注意** 用于安装的若干选项在不用得到管理员确认的情况下 *自动* 存取根磁盘以外的磁盘。建议在重新安装操作系统前从系统断开其它所有包含卷的磁盘。

---

断开其它磁盘可以确保这些磁盘不受重新安装的影响。例如，如果操作系统进行初始安装时 `home` 文件系统在第二磁盘上，则 `home` 文件系统仍然可以恢复。删除第二磁盘可以确保起始文件系统的完整性。

## 重新安装操作系统

一旦替换了任何故障磁盘或正在发生故障的磁盘并分离了未涉及重新安装的磁盘，则应按操作系统文档所述重新安装操作系统。先安装操作系统，然后再安装 `Volume Manager`。

当进行操作系统安装时，必须确保不以任何方式存取除根磁盘外的任何其它磁盘。无论将任何内容写入了除根磁盘外的其它磁盘，都将破坏该磁盘上的 `Volume Manager` 配置。

---

**注意** 在重新安装中，可以更改主机 ID（或名称）。建议保留现有的主机 ID（名称），因为以下几节假定您未更改主机 ID（名称）。

---

## 重新安装 Volume Manager

重新安装 `Volume Manager` 分为两个步骤：

- ◆ 从 CD-ROM 装入 `Volume Manager`
- ◆ 初始化 `Volume Manager`

若要重新安装 `Volume Manager`，请按照 《*VERITAS Volume Manager 安装指南*》中用于装入 `Volume Manager`（从 CD-ROM）的指令。

---

**注意** 若要重建保留在非根磁盘上的 `Volume Manager` 配置，在重新安装后 *不要* 初始化 `Volume Manager`（使用 `vxinstall`）。

---

在某些系统上，可以使用 `vxserial` 安装 `Volume Manager` 许可证密钥（请参见 `vxserial (1M)` 帮助手册）。

## 恢复 Volume Manager 配置

一旦装入了 Volume Manager 软件包，请按照下列步骤恢复 Volume Manager 配置：

1. 关闭系统。
2. 重新挂接从系统中删除的磁盘。
3. 重新启动系统。
4. 系统启动时，用下列命令使系统变为单用户模式：  

```
shutdown -g0 -is -y
```
5. 提示时，请输入密码并按回车键继续。
6. 删除安装中涉及的在装入 Volume Manager 时创建的但不再需要的文件。请使用下列命令：  

```
rm -rf /etc/vx/reconfig.d/state.d/install-db
```
7. 删除文件后，启动一些 Volume Manager I/O 后台驻留程序。通过输入下列命令启动后台驻留程序：  

```
vxiod set 10
```
8. 使用下列命令以禁用模式启动 Volume Manager 配置后台驻留程序 vxconfigd：  

```
vxconfigd -m disable
```
9. 用下列命令初始化 vxconfigd 后台驻留程序：  

```
vxdctl init
```
10. 用下列命令初始化 DMP 子系统：  

```
vxdctl initdmp
```
11. 用下列命令启用 vxconfigd：  

```
vxdctl enable
```

现在，恢复了保留在未涉及重新安装的磁盘上的配置。但是，由于重新安装了根磁盘，因此，对 Volume Manager 来说，根磁盘不再是 Volume Manager 磁盘。保留磁盘的配置不包括作为 Volume Manager 配置组成部分的根磁盘。



如果系统根磁盘和涉及重新安装的任何其它磁盘在发生故障和进行重新安装时不受 Volume Manager 控制，那么将在此时完成重新配置。有多种方法可用于替换磁盘；选择所喜欢的方法。

如果根磁盘（或另一个磁盘）涉及重新安装，则该磁盘（或其它不再挂接到系统的磁盘）上的任何卷或镜像现在都不可存取。如果重新安装、删除或替换的磁盘上包含只有一个 plex 的卷，那么该卷中的数据将损失，并且必须从备份中恢复。

另外，系统的 root 文件系统、swap 区域（和某些系统上的 stand 区域）和 /usr 文件系统都不再位于卷上。若要更正这些问题，请按照“配置清除”中的指令进行。

可以在 vxctl enable 成功后启动热重定位设备，但只能在管理员确保其服务（已被启用并正常运转）不妨碍其它重新配置过程时才可启动。建议在完成“最终重新配置”步骤后启动热重定位。有关启动热重定位的更多信息，请参见第 195 页的“启动热重定位”。

## 配置清除

以下几节描述在重新安装 Volume Manager 之后如何清除系统配置。

将描述下列清除类型：

- ◆ 根目录可置性清除
- ◆ 卷清除
- ◆ 磁盘清除

这几节之后是重配置信息：

- ◆ 根目录可置性重配置
- ◆ 最终重配置

### 根目录可置性清除

若要开始 Volume Manager 配置的清除，请删除与根目录可置性关联的所有卷。如果根磁盘（和任何涉及系统引导进程的磁盘）受 Volume Manager 控制，则必须做这一步。要删除的卷是：

- ◆ rootvol，它包含 root 文件系统
- ◆ swapvol，它包含 swap 区域
- ◆ （在某些系统上）standvol，它包含 stand 文件系统
- ◆ usr，它包含 /usr 文件系统

若要删除根卷，请使用 vxedit 命令：

```
vxedit -fr rm rootvol
```

重复使用该命令，使用 swapvol 和 usr (standvol) 代替 rootvol，来删除 swap、stand 和 usr 卷。

## 卷清除

在完成根目录可置性清除之后，必须确定需要从备份中恢复的卷。要恢复的卷包括那些全部镜像（该卷的所有副本）都驻留在被重新安装或被删除的磁盘上的卷。这些卷是无效的并且必须将它们删除、重新创建和从备份中恢复。如果只有卷的一些镜像存在重新初始化的磁盘或删除的磁盘上，则必须删除这些镜像。可以在以后重新添加这些镜像。

若要恢复这些卷，请执行下列步骤：

1. 通过使用下列命令，确定哪些 VM 磁盘已被删除或重新安装：

```
vxdisk list
```

Volume Manager 显示系统磁盘设备列表和这些设备的状态。例如，对于有三个磁盘和一个重新安装的根磁盘的重新安装系统，vxdisk list 命令的输出与下面类似：

| 设备       | 类型     | 磁盘     | 组      | 状态              |
|----------|--------|--------|--------|-----------------|
| c0t0d0s2 | sliced | -      | -      | 错误              |
| c0t1d0s2 | sliced | disk02 | rootdg | 联机              |
| c0t2d0s2 | sliced | disk03 | rootdg | 联机              |
| -        | -      | disk01 | rootdg | 故障 位于: c0t0d0s2 |

**注意** 您的系统可以使用与本实例不同的设备名称。有关设备名称的更多信息，请参见第 1 页的“[了解 Volume Manager](#)”。

显示表明重新安装的根设备 c0t0d0s2 不与 VM 磁盘关联，并被标记为错误状态。disk02 和 disk03 未涉及到重新安装中，由 Volume Manager 辨识，并与它们的设备（c0t1d0s2 和 c0t2d0s2）关联。以前的 disk01，即与被替换的磁盘设备相关联的 VM 磁盘，不再与该设备（c0t0d0s2）相关联。

如果在重新安装过程中删除或替换了其它磁盘（其上有卷或镜像），则那些磁盘也会有一个列出错误状态的磁盘设备和一个列出与设备不相关联的 VM 磁盘。

2. 一旦知道删除或替换了哪些磁盘，就要在故障磁盘上定位所有的镜像。使用此命令：

```
vxprint -sF "%vname" -e'sd_disk = "disk"'
```

其中，*disk* 是一个处于故障状态的磁盘名称。确保在此命令中将磁盘名称括在引号里面。否则，此命令将返回出错讯息。vxprint 命令返回在故障磁盘上有镜像的卷的列表。对每个处于故障状态的磁盘重复使用此命令。

3. 检查每个卷的状态。使用下列命令打印卷信息：

```
vxprint -th volume_name
```

其中，*volume\_name* 是要检查的卷的名称。vxprint 命令显示该卷的状态、它的 plex 和构成那些 plex 的磁盘部分。例如，只有一个 plex 的称为 v01 的卷驻留在名为 disk01 的重新安装的磁盘上。vxprint -th v01 命令输出以下显示：





```

V NAME USETYPE KSTATE STATE LENGTH READPOL
 PREFPLEX
PL NAME VOLUME KSTATE STATE LENGTH LAYOUT
 NCOL/WID MODE
SD NAME PLEX DISK DISKOFFS LENGTH [COL/]OFF
 DEVICE MODE
v v01 fsgen DISABLED ACTIVE 24000 SELECT
-
pl v01-01 v01 DISABLED NODEVICE 24000 CONCAT
-
 RW
sd disk01-06 v0101 disk01 245759 24000 0
 c1t5d1 ENA

```

此卷唯一的 plex 显示在以 p1 开头的行中。名为 v01-01 的卷的 STATE 字段是 NODEVICE。此 plex 在被替换、删除或重新安装的磁盘上有空间。此 plex 不再有效，必须将其删除。

因为 v01-01 是该卷唯一的 plex，因此，除非从备份中恢复该卷，否则该卷的内容无法恢复。也必须删除该卷。如果存在该卷的一个备份副本，则可以稍后再恢复该卷。保留该卷的名称及其长度的记录，因为在备份过程中需要它。

#### 4. 若要删除卷 v01，请使用 vxedit 命令：

```
vxedit -r rm v01
```

有可能只有 plex 的一部分位于故障磁盘上。如果卷有与之关联的一个条带 plex，该卷就被划分到几个磁盘上。例如，名为 v02 的卷有一个条带横跨三个磁盘的条带 plex，其中一个磁盘是重新安装的磁盘 disk01。vxprint -th v02 命令显示：

```

V NAME USETYPE KSTATE STATE LENGTH
 READPOL PREFPLEX
PL NAME VOLUME KSTATE STATE LENGTH
 LAYOUT NCOL/WID MODE
SD NAME PLEX DISK DISKOFFS LENGTH
 [COL/]OFF DEVICE MODE
v v02 fsgen DISABLED ACTIVE 30720
 SELECT v02-01
pl v02-01 v02 DISABLED NODEVICE 30720
 STRIPE 3/128 RW
sd disk02-02 v02-01 disk02 424144 10240
 0/0 c1t2d0 ENA
sd disk01-05 v02-01 disk01 620544 10240
 1/0 c1t2d1 DIS
sd disk03-01 v02-01 disk03 620544 10240
 2/0 c1t2d2 ENA

```



显示的内容中显示了的三个磁盘，条带化的 `plex v02-01` 横跨这三个磁盘（以 `sd` 开始的行表示条带）。条带区域中的一个位于故障磁盘上。此磁盘不再有效，因此名为 `v02-01` 的 `plex` 处于 `NODEVICE` 状态。由于这是该卷唯一的 `plex`，因此该卷无效，必须将其删除。如果备份介质上有 `v02` 的一个副本，则可以在以后恢复它。若打算将卷从备份中恢复，则保留该卷名称和长度的记录。

#### 5. 使用 `vxedit` 命令删除卷，如前所述。

在故障磁盘上有镜像的卷也可以在有效磁盘上有其它镜像。在这种情况下，不需要从备份中恢复该卷，因为有效磁盘上的数据仍然有效。

对在故障磁盘 (`disk01`) 上有一个 `plex` 并在有效磁盘 (`disk02`) 上有另一个 `plex` 的卷的 `vxprint -th` 命令的输出与以下实例类似：

| V  | NAME      | USETYPE  | KSTATE   | STATE    | LENGTH |
|----|-----------|----------|----------|----------|--------|
|    | READPOL   | PREFPLEX |          |          |        |
| PL | NAME      | VOLUME   | KSTATE   | STATE    | LENGTH |
|    | LAYOUT    | NCOL/WID | MODE     |          |        |
| SD | NAME      | PLEX     | DISK     | DISKOFFS | LENGTH |
|    | [COL/]OFF | DEVICE   | MODE     |          |        |
| v  | v03       | fsgen    | DISABLED | ACTIVE   | 30720  |
|    | SELECT    | -        |          |          |        |
| pl | v03-01    | v03      | DISABLED | ACTIVE   | 30720  |
|    | CONCAT    | -        | RW       |          |        |
| sd | disk02-01 | v03-01   | disk01   | 620544   | 30720  |
|    | 0         | c1t3d0   | ENA      |          |        |
| pl | v03-02    | v03      | DISABLED | NODEVICE | 30720  |
|    | CONCAT    | -        | RW       |          |        |
| sd | disk01-04 | v0302    | disk03   | 262144   | 30720  |
|    | 0         | c1t2d2   | DIS      |          |        |

此卷有两个 `plex`，`v03-01` 和 `v03-02`。第一个 `plex` (`v03-01`) 不使用无效磁盘上的任何空间，所以它仍然可以使用。第二个 `plex` (`v03-02`) 使用无效磁盘 `disk01` 上的空间，其处于 `NODEVICE` 状态。必须将 `plex v03-02` 删除。但是，该卷仍然有包含有效数据的一个有效 `plex`。如果需要镜像该卷，可以在稍后添加另一个 `plex`。注意稍后创建另一个 `plex` 的卷名称。

#### 6. 若要删除一个无效的 `plex`，必须将此 `plex` 从卷分离，然后将其删除。这是使用 `vxplex` 命令完成的。若要删除 `plex v03-02`，请使用以下命令：

```
vxplex -o rm dis v03-02
```

#### 7. 一旦清除了所有卷，就按照第 194 页的“磁盘清除”中描述的方法清除磁盘配置。



## 磁盘清除

一旦删除了全部无效的卷和 plex，就可以清除磁盘配置了。必须将每个被删除、重新安装或替换的磁盘（根据 `vxdisk list` 命令的输出来确定）从配置中删除。

若要删除磁盘，请使用 `vxdg` 命令。若要删除故障磁盘 `disk01`，请使用以下命令：

```
vxdg rmdisk disk01
```

如果 `vxdg` 命令返回一个出错讯息，则存在一些无效的镜像。重复在第 191 页的“卷清除”一节中描述的进程，直到删除完所有无效的卷和镜像。

## 根目录可置性配置

一旦删除了全部无效磁盘，就可以将替换或重新安装的磁盘添加到 Volume Manager 控制中。如果根磁盘原先受 Volume Manager 的控制，或者现在想要将根磁盘置于 Volume Manager 的控制，则首先添加根磁盘。

若要将根磁盘添加到 Volume Manager 控制中，请使用 Volume Manager 支持操作 (`vxdiskadm`)。使用以下命令：

```
vxdiskadm
```

从 `vxdiskadm` 主菜单中，选择菜单项 2（封装一个或多个磁盘）。按照有关指令为系统封装根磁盘。

当完成封装时，将系统重新启动为多用户模式。

## 最终重配置

一旦封装了根磁盘，应使用 `vxdiskadm` 添加所有其它被替换的磁盘。如果在操作系统重新安装过程中重新安装了这些磁盘，应将它们封装；否则，可以添加它们。

一旦将全部磁盘添加到了系统，就可以重新创建曾作为配置清除的一部分被完全删除的任何卷，并可以从备份中恢复其内容。可以使用 `vxassist` 或图形用户界面完成卷的重新创建。

可以使用 `vxassist` 命令重新创建卷 `v01` 和 `v02`：

```
vxassist make v01 24000
vxassist make v02 30720 layout=stripe nstripe=3
```

一旦创建了卷，就可以使用正常备份 / 恢复过程从备份中将其恢复。

对于删除了 plex（作为卷清除一部分）的任何卷，按照本手册中所描述的，可以使用 `vxassist` 镜像卷的指令重新创建卷的这些镜像。若要使用 `vxassist` 替换从卷 `v03` 中删除的 plex，请使用以下命令：

```
vxassist mirror v03
```

一旦恢复了在重新安装过程中损失的卷和 plex，就完成了恢复，应将系统配置为故障发生之前的样子。

## 启动热重定位

这时，管理员应该重新启动系统，或手动启动热重定位（如果需要热重定位服务）。每种操作都将启动重定位后台驻留程序（以及它的 `vxnotify` 进程）。

若要启动热重定位，请使用下列命令。

启动监视后台驻留程序。它在发现任何问题时，将发送电子邮件给管理员。若要更改用于发送问题报告的地址，请更改 `vxrelocd` 的参数：

```
nohup /usr/lib/vxvm/bin/vxrelocd root &
```

下列命令也是有效的：

```
nohup /usr/lib/vxvm/bin/vxrelocd root > /dev/null 2>&1 &
```

下列命令可以检测热重定位是否已启动：

```
ps -ef | grep vxrelocd | grep -v grep
```

## Plex 和卷状态

下面几节描述 plex 和卷的状态。

### Plex 状态

Plex 状态反映 plex 是不是卷内容的完整且一致的副本（镜像）。Volume Manager 实用程序自动维护 plex 状态。但是，如果不想把对 plex 所关联卷的更改写入该 plex，则可以更改 plex 的状态。例如，如果带有一个特别 plex 的磁盘开始发生故障，就可以暂时禁用该 plex。

---

**注意** Plex 不必与卷关联。可以使用 `vxmake plex` 命令创建 plex。可以在以后将此命令创建的 plex 挂接到卷。

---

Volume Manager 实用程序使用 plex 状态来：

- ◆ 指出是否已将卷的内容初始化为已知的状态
- ◆ 确定 plex 是否包含卷内容的有效副本（镜像）
- ◆ 跟踪 plex 在系统故障时是否仍处在使用中
- ◆ 监视 plex 上的操作

本节将详细解释 plex 状态，供希望对 plex 状态有详细了解的管理员使用。

与卷关联的 plex 具有下列状态中的一种：

- ◆ EMPTY
- ◆ CLEAN



- ◆ ACTIVE
- ◆ STALE
- ◆ OFFLINE
- ◆ TEMP
- ◆ TEMPRM
- ◆ TEMPRMSD
- ◆ IOFAIL

脏区日志或 RAID-5 日志 plex 是一个特例，因为其状态总是设置为 LOG。

### EMPTY plex 状态

卷创建操作将所有与卷关联的 plex 设置为 EMPTY 状态，以表明还没有初始化这些 plex。

### CLEAN plex 状态

当已知 plex 包含了卷内容的一致副本（镜像），并且操作已禁用该卷时，则 plex 处于 CLEAN 状态。因此，如果卷的所有 plex 都处于 CLEAN 状态，则启动卷时无需任何措施来保证 plex 都是完全相同的。

### ACTIVE plex 状态

Plex 在以下两种情况下可以处于 ACTIVE 状态：

- ◆ 当启动了卷，且 plex 完全参与到正常 I/O 时（卷内容更改时，plex 内容也随之更改）
- ◆ 当由于系统崩溃而导致卷停止并且 plex 在崩溃发生时是 ACTIVE

在后一种情况下，系统故障可以使 plex 内容处于不一致的状态。当启动卷时，Volume Manager 进行恢复操作，保证使标记为 ACTIVE 的 plex 的内容完全相同。

---

**注意** 在运行正常的系统上，对于任何卷 plex，ACTIVE 应是最一般的状态。

---

### STALE plex 状态

在某种可能情况下，如果 plex 没有完整的当前的卷内容，则该 plex 被置于 STALE 状态。另外，如果在 plex 上发生 I/O 错误，内核就停止对该 plex 内容的使用和更新，并有一个操作将该 plex 的状态设置为 STALE。

vxplex att 操作从 ACTIVE plex 恢复 STALE plex 的内容。原子复制操作将卷内容复制到 STALE plex。系统管理员可以使用 vxplex det 操作将 plex 强制为 STALE 状态。

## OFFLINE plex 状态

通过将 plex 状态设置为 OFFLINE, `vxmend off task` 无限期地将 plex 从卷上分离。虽然分离的 plex 仍保留与卷的关联, 但是对卷的更改却不更新 OFFLINE plex。直到使用 `vxplex att` 任务将 plex 联机并重新挂接后, 才会更新 plex。当以上情况发生时, plex 被置于 STALE 状态, 使其内容可在下一次 `vxvol start` 操作时恢复。

## TEMP plex 状态

将 plex 设置为 TEMP 状态可以使一些以真正原子方式不能发生的 plex 操作变得容易。例如, 将 plex 挂接到启用的卷要求将卷内容复制到 plex 上, 否则, 不能认为 plex 已完全挂接到启用的卷。

一个实用程序在这样的操作开始时将 plex 状态设置为 TEMP, 并在该操作结束时将 plex 设置为适当的状态。如果系统由于任何原因而出现故障, TEMP plex 状态表明该操作没有完成。随后的 `vxvol start` 分离处于 TEMP 状态的 plex。

## TEMPRM plex 状态

TEMPRM plex 状态与 TEMP 状态类似, 不同之处是在操作完成后 TEMPRM plex 被删除。一些子磁盘操作需要临时的 plex。例如, 将子磁盘与 plex 关联要求用卷的内容更新此子磁盘, 而在此之前子磁盘并未真正与 plex 关联。这种更新要求将此子磁盘与标记为 TEMPRM 的临时 plex 关联直到操作完成, 并删除 TEMPRM plex。

如果系统由于任何原因而出现故障, TEMPRM 状态表明该操作没有成功地完成。随后的操作分离并删除 TEMPRM plex。

## TEMPRMSD plex 状态

当挂接新的 plex 时, `vxassist` 使用 TEMPRMSD plex 状态。如果操作没有完成, 则删除 plex 和它的子磁盘。

## IOFAIL plex 状态

IOFAIL plex 状态与永久状态日志相关联。检测到 ACTIVE plex 故障时, `vxconfigd` 将该 plex 置于 IOFAIL 状态, 以便在卷启动时将此 plex 排除在恢复选择进程之外。

## Plex 状态循环

更改 plex 状态是正常操作的一部分。Plex 状态中的更改表明出现了异常情况, Volume Manager 必须进行正常化操作。系统启动时, 自动启动卷并且 `vxvol start` 任务使全部 CLEAN plex 变成 ACTIVE。如果关机前所有程序都运行正常, 则卷停止的操作标记全部 ACTIVE plex 为 CLEAN 并且循环继续。启动时所有的 plex 处于 CLEAN (在 `vxvol start` 使它们处于 ACTIVE 前) 表明这是正常关机并优化启动。



## Plex 内核状态

*plex* 内核状态表明 *plex* 的可存取性。Plex 内核状态由卷驱动程序监视并允许 *plex* 处于脱机 (DISABLED)、维护 (DETACHED) 或联机 (ENABLED) 操作模式。

下列是 *plex* 内核状态：

- ◆ DISABLED —— 无法存取 *plex*。
- ◆ DETACHED —— 对卷的写没有反映到 *plex*。来自卷的读请求未反映来自 *plex* 的请求。接受 *plex* 操作和 *ioctl* 函数。
- ◆ ENABLED —— 对卷的写请求反映到了 *plex*。卷的读请求满足 *plex* 的请求。

---

**注意** 设置这些状态不需要用户介入，它们实行内部维护。在运转正常的系统上，*plex* 被全部启用。

---

## 卷状态

有几种卷状态，其中一些和 *plex* 状态相似：

- ◆ CLEAN —— 卷未启动（内核状态是 DISABLED）并且卷的 *plex* 是同步的。
- ◆ ACTIVE —— 卷已经启动（当前内核状态是 ENABLED）或当重新启动机器时卷处于使用状态（内核状态是 ENABLED）。如果卷当前是 ENABLED，则在任何时候卷的 *plex* 状态不确定（因为正使用卷）。如果卷当前是 DISABLED，这意味着不能保证 *plex* 一致，但当启动卷时可使 *plex* 一致。
- ◆ EMPTY —— 卷内容没有初始化。当卷是 EMPTY 时，内核状态总是 DISABLED。
- ◆ SYNC —— 卷处在读取 - 写回恢复模式（当前内核状态是 ENABLED）或者当重新启动机器时处在读取 - 写回模式（内核状态是 DISABLED）。在读取 - 写回恢复模式下，通过从一个 *plex* 的块读数据并将数据写到全部其它可写的 *plex* 来恢复 *plex* 的一致性。如果卷是 ENABLED，这意味着正通过读取 - 写回恢复使 *plex* 再同步。如果卷是 DISABLED，这意味着当重新启动机器时通过读取 - 写回使 *plex* 再同步，因此仍然需要使 *plex* 再同步。
- ◆ NEEDSYNC —— 当卷下次启动时，它需要一个再次同步操作。

卷的永久状态日志在卷启动期间修改这些标志的解释（例如，DIRTY/CLEAN 标志）。如果设置了 CLEAN 标志，则 ACTIVE 卷没有被任何进程写入或在重新启动时没有被打开，因此可认为它是 CLEAN。在标记卷为 CLEAN 的任何情况下，总是设置 CLEAN 标志。

## RAID-5 卷状态

RAID-5 卷有它们自己的卷状态：

- ◆ CLEAN —— 卷没启动（内核状态是 DISABLED）并且它的奇偶校验良好。RAID-5 *plex* 条带是一致的。



- ◆ **ACTIVE** —— 卷已经启动（当前内核状态是 **ENABLED**）或重新启动系统时卷正处于使用状态（内核状态是 **ENABLED**）。如果卷当前是 **ENABLED**，则任何时候卷的 RAID-5 plex 的状态不确定（因为正使用卷）。如果卷当前是 **DISABLED**，不能保证奇偶校验是同步的。
- ◆ **EMPTY** —— 卷内容没有初始化。当卷是 **EMPTY** 时，内核状态总是 **DISABLED**。
- ◆ **SYNC** —— 卷正经受奇偶校验再同步（当前内核状态是 **ENABLED**）或当重新启动机器时卷正使它的奇偶校验再同步（内核状态是 **DISABLED**）。
- ◆ **NEEDSYNC** —— 下次启动卷时，卷需要一个奇偶校验再同步操作。
- ◆ **REPLAY** —— 卷处于日志重放的过渡状态。当需要使用日志奇偶校验和数据时，发生日志重放。

## 卷内核状态

卷内核状态表明卷的可存取性。卷内核状态允许卷处于脱机 (**DISABLED**)、维护 (**DETACHED**) 或联机 (**ENABLED**) 操作模式。

下列是卷内核状态：

- ◆ **DISABLED** —— 不能存取卷。
- ◆ **DETACHED** —— 不能读或写卷，但是接受 plex 设备操作和 **ioctl** 功能。
- ◆ **ENABLED** —— 能读或写卷。

## RAID-5 卷恢复

本节描述 RAID-5 卷的操作和恢复。有关 RAID-5 卷的更多信息，请参见第 1 页的“[了解 Volume Manager](#)”。

## RAID-5 卷布局

RAID-5 卷包含一个或多个 plex，每个 plex 包含一个或多个子磁盘。和镜像卷不同，不是所有 RAID-5 卷中的 plex 都适于保留卷数据的镜像副本。

RAID-5 卷可以有两种类型的 plex：

- ◆ *RAID-5 plex* 用于为卷同时保留数据和奇偶校验
- ◆ *日志 plex* 保留写到卷上的数据日志，使恢复更快并更有效

## RAID-5 plex

RAID-5 卷将数据和奇偶校验信息同时保存在单一的 RAID-5 plex 上。RAID-5 plex 包含按照列来排列的子磁盘，这和条带化模型相似。以下的显示内容显示一个 RAID-5 plex 和与它关联的 **vxprint** 输出：



```

PL NAME VOLUME KSTATE STATE LENGTH LAYOUT NCOL/WID MODE
SD NAME PLEX DISK DISKOFFS LENGTH [COL/]OFF DEVICE MODE

pl rvol-01 rvol ENABLED ACTIVE 20480 RAID 3/16 RW
sd disk00-00 rvol-01 disk00 0 10240 0/0 c1t4d1 ENA
sd disk01-00 rvol-01 disk01 0 10240 1/0 c1t2d1 ENA
sd disk02-00 rvol-01 disk02 0 10240 2/0 c1t3d1 ENA

```

Plex 行显示 plex 布局是 RAID 并显示它有三列和 16 个扇区的条带单元大小。每个子磁盘行显示 plex 中的列和列（子磁盘所处的列）中的偏移。

**注意** 您的系统可使用一个不同于实例的设备名。有关设备名的更多信息，请参见第 1 页的“了解 Volume Manager”。

## RAID-5 日志

每个 RAID-5 卷有一个存储数据和奇偶校验的 RAID-5 plex。任何和卷关联的其它 plex 用于记录有关写到卷上的数据和奇偶校验的信息。这些 plex 指 RAID-5 日志 plex 或 RAID-5 日志。

RAID-5 日志可以是连续或条带化 plex，并且每个和 RAID-5 卷关联的 RAID-5 日志有该卷日志信息的完整副本。建议每个 RAID-5 卷最少有两个 RAID-5 日志 plex，这些日志 plex 应位于不同的磁盘上。每个 RAID-5 卷有两个 RAID-5 日志 plex 可以避免因单个磁盘故障而损失日志信息。

为支持对 RAID-5 阵列的并行存取，日志应该是 RAID-5 plex 条带大小的几倍。

通过检查 vxprint 输出，可以看出 RAID-5 卷的 RAID-5 日志 plex 和 RAID-5 plex 之间的差异。日志 plex 的 STATE 字段标记为 LOG。以下显示内容显示 RAID-5 卷的 vxprint 输出。

```

V NAME USETYPE KSTATE STATE LENGTH READPOL PREFPLEX
PL NAME VOLUME KSTATE STATE LENGTH LAYOUT NCOL/WID MODE
SD NAME PLEX DISK DISKOFFS LENGTH [COL/]OFF DEVICE MODE

v r5vol raid5 ENABLED ACTIVE 20480 RAID -
pl r5vol-01 r5vol ENABLED ACTIVE 20480 RAID 3/16 RW
sd disk00-00 r5vol-01 disk00 0 10240 0/0 c1t4d1 ENA
sd disk01-00 r5vol-01 disk01 0 10240 1/0 c1t2d1 ENA
sd disk02-00 r5vol-01 disk02 0 10240 2/0 c1t3d1 ENA
pl r5vol-11 r5vol ENABLED LOG 1024 CONCAT - RW
sd disk03-01 r5vol-11 disk00 0 1024 0 c1t3d0 ENA
pl r5vol-12 r5vol ENABLED LOG 1024 CONCAT - RW
sd disk04-01 r5vol-12 disk02 0 1024 0 c1t1d1 ENA

```



通过卷的读策略为 RAID，可以将 RAID-5 卷 (`r5vol`) 标识为 RAID-5 卷。RAID-5 卷有一个 RAID-5 plex (`r5vol-01`)，与前面所述类似。在 plex `r5vol-11` 和 `r5vol-12` 中，有两个 RAID-5 日志。这些 plex 由状态字段 LOG 所标识，它们与 RAID-5 卷关联并且有一个不是 RAID 的布局。

## 创建 RAID-5 卷

可以使用 `vxassist`（推荐使用）或者 `vxmake` 创建 RAID-5 卷。本节将描述这两种方法。

RAID-5 卷包含 RAID-5 plex，该 RAID-5 plex 由两个或两个以上的位于两个或两个以上物理磁盘上的子磁盘组成。每个卷上只能存在一个 RAID-5 plex。

### vxassist 和 RAID-5 卷

可以使用 `vxassist` 命令创建 RAID-5 卷：

```
vxassist make volume_name length layout=raid5
```

例如，若要创建名为 `volraid` 的 10M RAID-5 卷，请输入：

```
vxassist make volraid 10m layout=raid5
```

此命令在默认数量的磁盘上创建具有默认条带单元大小的 RAID-5 卷。

## 初始化 RAID-5 卷

如果 RAID-5 卷是由 `vxmake` 命令创建的并且尚未初始化，或者它被设置为未初始化的状态，则必须进行 RAID-5 卷初始化。

可以用 `vxvol` 和以下命令之一初始化 RAID-5 卷：

```
vxvol init zero volume_name
```

or

```
vxvol start volume_name
```

`vxvol init zero` 将 0 写入所有的 RAID-5 日志 plex 和卷的整个长度。然后，该命令使卷保留在 ACTIVE 状态。

`vxvol start` 通过 XOR 所有其它列中对应的数据条带单元来恢复奇偶校验。虽然 `vxvol start` 比 `vxvol init zero` 操作慢，但是此命令使 RAID-5 卷立即可用。



## 故障和 RAID-5 卷

可以看到两种故障：*系统故障*和*磁盘故障*。系统故障指由于操作系统应急或电源故障导致系统突然停止。磁盘故障指由于系统故障（如磁头失效、磁盘上的电子故障或磁盘控制器故障）导致一些磁盘上的数据不可用。

### 系统故障

如果没有磁盘故障，RAID-5 卷的设计使磁盘空间的开销最小而保持最大的可用性。但是在系统故障之后，许多形式的 RAID-5 可能会损失数据。发生数据损失是由于系统故障导致 RAID-5 卷中的数据 and 奇偶校验不同步。发生同步损失是由于无法确定出现故障时的未完成写入状态。

如果正在存取 RAID-5 卷时发生同步损失，则将卷描述为有*失效的奇偶校验*。然后，必须通过读取每个条带内的所有非奇偶校验列、重新计算奇偶校验并在条带中写出奇偶校验条带单元来重构奇偶校验。卷中的每个条带都必须进行该操作，因此可能会花很长时间才能完成。

---

**警告** 在再同步正在进行时，阵列内的任何磁盘故障都将导致卷中数据损失。此操作只应用于无日志 plex 的 RAID-5 卷。

---

除了容易发生故障，再同步进程还占用计算机资源使系统操作变慢。

由于 RAID-5 日志保留着发生故障时写入数据副本，因此 RAID-5 日志能够减小因系统故障导致的损失。再同步进程由从日志中读取数据与奇偶校验并将数据与奇偶校验写入 RAID-5 卷适当的区域构成。此进程大大减少了数据和奇偶校验再同步所需的时间。它还意味着卷永远不会真正地失效。由于在任何时间都能知道卷中所有条带的数据和奇偶校验，因此单一磁盘的故障不能引起卷中数据的损失。

### 磁盘故障

磁盘故障可以导致磁盘上的数据不可用。对 RAID-5 卷来说，这意味着子磁盘不可用。

在写入磁盘时出现不可更正的 I/O 错误时，可能发生这种情况。当启动系统时，I/O 错误可能导致子磁盘从阵列中分离或磁盘不可用（例如，电缆问题或由于驱动器断电）。

当发生这种情况时，子磁盘不能用于保留数据并被认为是 STALE 和 DETACHED。如果基础磁盘变得可用或被替换，则子磁盘仍被认为失效和未使用。

如果试图读取失效子磁盘中包含的数据，则从条带中所有其它条带单元上的数据重构数据。该操作称为*重构读取*。重构读取是比简单读取更繁重的操作，它可以导致读取性能的下降。在 RAID-5 卷有失效子磁盘时，则被认为处于*退化模式*。

从 vxprint 的输出中可以辩识处在退化模式中的 RAID-5 卷，如下所示：

```

V NAME USETYPE KSTATE STATE LENGTH READPOL PREFPLEX
PL NAME VOLUME KSTATE STATE LENGTH LAYOUT NCOL/WID MODE
SD NAME PLEX DISK DISKOFFS LENGTH [COL/]OFF DEVICE MODE
v r5vol RAID-5 ENABLED DEGRADED 20480 RAID -
pl r5vol-01 r5vol ENABLED ACTIVE 20480 RAID 3/16 RW
sd disk00-00 r5vol-01 disk00 0 10240 0/0 c1t4d1
sd disk01-00 r5vol-01 disk01 0 10240 1/0 c1t2d1 dS
sd disk02-00 r5vol-01 disk02 0 10240 2/0 c1t3d1 -
pl r5vol-11 r5vol ENABLED LOG 1024 CONCAT - RW
sd disk03-01 r5vol-11 disk00 10240 1024 0 c1t3d0 -
pl r5vol-12 r5vol ENABLED LOG 1024 CONCAT - RW
sd disk04-01 r5vol-12 disk02 10240 1024 0 c1t1d1 -

```

r5vol 卷处于退化模式，它由 STATE 所示，所列出的 STATE 为 DEGRADED。故障子磁盘是 disk01-00，如最后一列中的标志所示——d 表示子磁盘已分离，s 表示子磁盘内容失效。

包含 RAID-5 日志的磁盘可能发生故障。这对卷操作没有直接影响。但是，卷上所有 RAID-5 日志的损失使卷易于发生完全故障。在 `vxprint -ht` 的输出中，plex 状态为 BADLOG 指示 RAID-5 日志 plex 中的故障。这种情况显示在下列显示内容中，下列显示内容中的 RAID-5 日志 plex r5vol-11 已失败：

```

V NAME USETYPE KSTATE STATE LENGTH READPOL PREFPLEX
PL NAME VOLUME KSTATE STATE LENGTH LAYOUT NCOL/WID MODE
SD NAME PLEX DISK DISKOFFS LENGTH [COL/]OFF DEVICE MODE
v r5vol RAID-5 ENABLED ACTIVE 20480 RAID -
pl r5vol-01 r5vol ENABLED ACTIVE 20480 RAID 3/16 RW
sd disk00-00 r5vol-01 disk00 0 10240 0/0 c1t4d1 ENA
sd disk01-00 r5vol-01 disk01 0 10240 1/0 c1t2d1 dS
sd disk02-00 r5vol-01 disk02 0 10240 2/0 c1t3d1 ENA
pl r5vol-11 r5vol DISABLED BADLOG 1024 CONCAT - RW
sd disk03-01 r5vol-11 disk00 10240 1024 0 c1t3d0 ENA
pl r5vol-12 r5vol ENABLED LOG 1024 CONCAT - RW
sd disk04-01 r5vol-12 disk02 10240 1024 0 c1t1d1 ENA

```

## RAID-5 恢复

这里是 RAID-5 卷通常需要的恢复类型：

- ◆ 奇偶校验再同步
- ◆ 失效子磁盘恢复
- ◆ 日志 plex 恢复

在下面各段中描述恢复的类型。奇偶校验再同步和失效子磁盘恢复通常在以下时间执行：

- ◆ 启动 RAID-5 卷时



- ◆ 系统启动之后不久
- ◆ 通过调用 `vxrecover` 命令

有关启动 RAID-5 卷的更多信息，请参见第 207 页的“启动 RAID-5 卷”。

如果在磁盘故障时启用热重定位，除非没有重定位可用的磁盘空间，否则不要求系统管理员干预。热重定位由故障和通过电子邮件注意到故障的系统管理员触发。

热重定位自动尝试重新定位发生故障的 RAID-5 plex 子磁盘。在任何重定位发生之后，热重定位后台驻留程序 (`vxrelocd`) 还启动奇偶校验再同步。

在 RAID-5 日志 plex 发生故障的情况下，只有日志 plex 已镜像才发生重定位；然后 `vxrelocd` 启动镜像再同步以再创建 RAID-5 日志 plex。如果在发生故障时禁止热重定位，则系统管理员可能需要启动再同步或进行恢复。

### 奇偶校验恢复

大多数情况下，RAID-5 阵列中没有失效奇偶校验。失效奇偶校验只发生在 RAID-5 卷的 RAID-5 日志失败之后，这时只要存在系统故障就发生失效奇偶校验。即使 RAID-5 卷有失效奇偶校验，通常作为卷启动过程的部分来修复它。

如果启动有无效 RAID-5 日志的卷，并且卷在再同步之前终止，其结果是带失效奇偶校验的一个活动卷。可以通过检查下面显示在 `vxprint -ht` 命令输出结果中的卷状态来证实这一点。

```
V NAME USETYPE KSTATE STATE LENGTH READPOL PREFPLEX
PL NAME VOLUME KSTATE STATE LENGTH LAYOUT NCOL/WID MODE
SD NAME PLEX DISK DISKOFFS LENGTH [COL/]OFF DEVICE MODE
v r5vol1 RAID-5 ENABLED NEEDSYNCSYNC 20480 RAID -
pl r5vol1-01 r5vol1 ENABLED ACTIVE 20480 RAID 3/16 RW
sd disk00-00 r5vol1-01 disk00 0 10240 0/0 c1t4d1 ENA
sd disk01-00 r5vol1-01 disk01 0 10240 1/0 c1t2d1 ENA
sd disk02-00 r5vol1-01 disk02 0 10240 2/0 c1t3d1 ENA
```

此输出列出卷状态为 NEEDSYNCSYNC，指示需要再同步奇偶校验。卷状态还可能是 SYNC，表明在启动时试图进行过同步并且同步进程应正在完成同步。如果这种进程不存在，或如果卷处在 NEEDSYNCSYNC 状态，则可以通过使用 `vxvol` 命令的 `resync` 关键字手动启动同步。例如，若要再同步上一个显示中的 RAID-5 卷，则使用此命令：

```
vxvol resync r5vol1
```

通过向 RAID-5 卷发布 `VOL_R5_RESYNCSYNC` ioctl 再生奇偶校验。在 RAID-5 卷开始时启动再同步进程，并再同步与 `-o iosize` 选项指定的扇区值相等的区域。如果没有指定 `-o iosize`，则使用默认最大 I/O 大小。然后再同步操作移动到下一个区域，直到再同步 RAID-5 卷的全长。

对于更大的卷，奇偶校验再生会占用很长时间。在操作完成前，系统可能关闭或崩溃。在系统关闭的情况下，经过重新启动必须保持奇偶校验再生过程。否则，此过程将从头再来。

若要避免重新启动进程，则需要为奇偶校验设置检查点。这意味着在配置数据库中保存到再生奇偶校验的偏移量。-o checkpoint=size 选项控制多长时间保存一次检查点。如果没有指定该选项，则使用默认检查点大小。

由于保存检查点偏移要求事务，因此检查点太小会延长再生奇偶校验所需的时间。在重新启动系统后，检查点偏移小于卷长的 RAID-5 卷在检查点偏移处启动奇偶校验再同步。

### 子磁盘恢复

失效子磁盘恢复通常在卷启动时完成。但是，恢复进程可能会崩溃，或者启动卷时带有防止子磁盘恢复的选项。还可能不执行恢复操作就将子磁盘所在的磁盘替换。但不论何种情况，都可以通过使用 vxvol 命令的 recover 关键字完成子磁盘恢复。例如，若要恢复第 208 页的图 26 “无效 RAID-5 卷”中显示的 RAID-5 卷中的失效子磁盘，请使用此命令：

```
vxvol recover r5vol disk01-00
```

可同时抓住 RAID-5 卷中的多个失效子磁盘。只用卷名调用 vxvol recover 就可以抓住多个失效子磁盘：

```
vxvol recover r5vol
```

### 故障之后恢复日志

可以由于磁盘故障而使 RAID-5 日志 plex 分离。通过使用 vxplex 命令的 att 关键字可以重新挂接这些 RAID-5 日志。若要重新挂接第 212 页的图 27 “读取 - 修改 - 写入”中显示的失败的 RAID-5 日志 plex，请使用此命令：

```
vxplex att r5vol r5vol-11
```

## 杂项 RAID-5 操作

许多操作是处理 RAID-5 卷及其关联对象的专用操作。这些操作通常由其它命令如 vxassist 和 vxrecover 作为更大操作（如迁移磁盘）的一部分来执行。如果只是少量使用 Volume Manager，则这些命令行操作不是必需的。

### 处理 RAID-5 日志

将 RAID-5 日志表示为 RAID-5 卷的 plex 并使用 vxplex 命令处理 RAID-5 日志。可以使用 vxplex att 命令添加 RAID-5 日志：

```
vxplex att r5vol r5log
```

只有新日志的大小足够保留条带上的所有数据，挂接操作才能进行。如果 RAID-5 卷已经有日志，则新日志长度是每个单独日志长度的最小值。这是因为新日志是旧日志的镜像。



如果没有启用 RAID-5 卷，则将新日志标记为 BADLOG 并在启动卷时将其启用。但是日志的内容将被忽略。

如果已将 RAID-5 卷启用并且有其它已启用的 RAID-5 日志，则新日志将自己的内容通过 ATOMIC\_COPY ioctl 与其它日志同步。

如果 RAID-5 卷当前无已启用的日志，则在启用 RAID-5 卷前将新日志设为 0。

可以通过使用 `vxplex dis` 命令从卷中删除日志 `plex`：

```
vxplex dis r5log3
```

如果删除日志使卷的有效日志少于两个，则打印出警告并且不允许操作继续。必需通过使用 `-o force` 选项强制执行操作。

## 处理 RAID-5 子磁盘

和处理其它子磁盘一样，使用 `vxsd` 命令处理 RAID-5 卷中 RAID-5 plex 的子磁盘。通过对条带 `plex` 的相同方式使用 `assoc` 关键字来完成关联。例如，若要向上一个显示中 RAID-5 卷的每一列的末尾添加子磁盘，请使用此命令：

```
vxsd assoc r5vol-01 disk10-01:0 disk11-01:1 disk12-01:2
```

如果子磁盘填充 `plex` 中的“洞”（就是用子磁盘映射卷逻辑地址中的某些部分），则认为子磁盘失效。如果已启用 RAID-5 卷，则关联操作通过使用 `VOL_R5_RECOVER ioctls` 命令再生属于子磁盘的数据。否则，将 RAID-5 卷标记为失效并在启动卷时将其恢复。

可以通过使用 `vxsd dis:` 命令从 RAID-5 plex 中删除子磁盘

```
vxsd dis disk10-01
```

---

**警告** 如果子磁盘映射 RAID-5 卷地址空间的部分，则该命令将卷设置为 DEGRADED 方式。在这种情况下，`dis` 操作打印警告并且必须强制使用 `-o force` 选项。同样，如果删除子磁盘使 RAID-5 卷不可用，则由于同一个条带上的另一个子磁盘不可用或丢失并且卷不是 DISABLED 与空状态，从而不允许该操作。

---

可以通过使用 `vxsd mv` 命令移动子磁盘来更改 RAID-5 卷所占用的子磁盘。例如，如果迁移了 `disk03` 并且 `disk22` 通过使用其空间的两个部分而有了足够的空间，则可以使用此命令：

```
vxsd mv disk03-01 disk22-01 disk22-02
```

虽然此命令与用于条带 `plex` 的命令相似，但操作的内部机制却相差很大。

## RAID-5 子磁盘移动

若要完成 RAID-5 子磁盘移动，需要从 RAID-5 plex 中删除当前子磁盘并用新子磁盘替换它。将新子磁盘标记为失效，然后用 `VOL_R5_RECOVER` 操作恢复新子磁盘。即可以用 `vxsd` 命令也可以（如果卷是不活动的）在启动卷时完成恢复。这意味着 RAID-5 卷在操作期间是退化方式。

在条带中涉及移动的另一个故障使卷不可用。如果卷的奇偶校验失效，则 RAID-5 卷也可以变为无效。

若要避免这些情况，则在发生下列情况时 `vxsd` 实用程序不允许子磁盘移动：

- ◆ 失效子磁盘占用与正在移动的子磁盘相同的条带
- ◆ RAID-5 卷已停止但没有干净地关闭（认为奇偶校验失效）
- ◆ RAID-5 卷活动并且无有效日志区域

只有第三种情况可以通过使用 `-o force` 选项覆盖。

还可以通过使用 `vxsd split` 和 `vxsd join` 命令来分割和结合 RAID-5 卷的子磁盘。这些操作以对镜像卷使用的操作相同的方式工作。

---

**注意** RAID-5 子磁盘以与移动其它子磁盘相同的方式移动，移动时没有降级冗余损失。

---

## 启动 RAID-5 卷

在启动 RAID-5 卷时，它的状态可以是许多状态中的一种。在系统正常关闭后，卷应处于 CLEAN 状态且不需要恢复。但是，如果没有关闭卷，或者在崩溃之前没有卸载，在启动 RAID-5 卷并使它可用之前可以请求恢复。本节描述可以在某些条件下采取的措施。

在正常条件下，在发生重新启动和任何恢复之后或通过 `vxrecover` 命令都可以自动启动卷。

## 不可启动的 RAID-5 卷

如果 RAID-5 plex 的一些部分未映射卷长，则 RAID-5 卷不可用：

- ◆ RAID-5 plex 不能比 RAID-5 卷长稀疏
- ◆ RAID-5 plex 不映射一个条带中有两个子磁盘发生故障的区域，这里的故障或者是子磁盘失效，或者是因为子磁盘建立在故障磁盘上。

发生这种情况时，`vxvol start` 命令返回该出错讯息：

```
vxvm:vxvol: 错误: 卷 r5vol 不能启动; RAID-5 plex 未映射整个卷长。
```

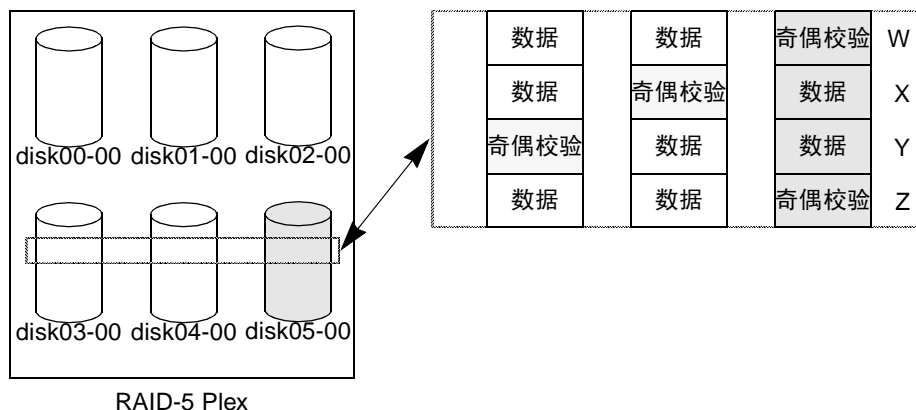
此时，RAID-5 卷的内容不可用。





奇偶校验无法启动并且子磁盘分离或失效是导致 RAID-5 卷可能无法启动的另一种方式。发生这种情况是由于在包含故障子磁盘的条带内，奇偶校验条带单元无效（因为奇偶校验失效）并且 错误子磁盘上的条带单元也无效。在图 26 “无效 RAID-5 卷” 中显示这种情况，该图显示了由于奇偶校验失效和子磁盘故障导致 RAID-5 卷无效。

图 26. 无效 RAID-5 卷



本实例在 RAID-5 阵列中显示四个条带。所有奇偶校验失效，子磁盘 disk05-00 发生故障。这使条带 X 和 Y 由于在这些条带中发生两个故障而不可用。

这种情况描述为一个条带内的两个故障并阻止卷的使用。在这种情况下，从 `vxvol start` 中的输出显示是：

```
vxvm:vxvol: 错误: 无法启动卷 r5vol ; 一些子磁盘不可用并且奇偶校验失效。
```

可以通过始终使用 RAID-5 卷中两个以上的 RAID-5 日志来避免这种情况。RAID-5 日志 plex 防止卷内的奇偶校验失效借以防止这种情况的发生（有关详细信息，请参见第 202 页的“系统故障”）。

## 强制启动 RAID-5 卷

即使卷被标记为失效也可以启动卷。例如，如果停止的卷有失效奇偶校验并且无 RAID-5 日志，则磁盘分离然后重新挂接。

即使数据没有过期（因为在子磁盘不可用时使用卷）也认为子磁盘无效，并认为 RAID-5 卷无效。若要防止这种情况，则要始终有多个与阵列相关联的有效 RAID-5 日志。但是，不可能始终保持这种状态。



若要用失效子磁盘启动 RAID-5 卷，可以使用 `-f` 选项和 `vxvol start` 命令。这样导致将所有的失效子磁盘都标记为非失效。标记发生在 `start` 操作求取 RAID-5 卷的有效性并确定启动所需的条件之前。同样，也可以通过使用 `vxmend fix unstale subdisk` 命令将单个的子磁盘标记为非失效。

## 在启动 RAID-5 卷时恢复

要全面恢复 RAID-5 卷的内容并使它可用，有几个操作是必需的。无论何时启动卷，在启动卷之前都将所有的 RAID-5 日志 `plex` 置零。这样做是为了防止将随机数据解释为日志项和破坏卷内容。同样，可能需要恢复一些子磁盘，或可能需要再同步奇偶校验（如果 RAID-5 日志已失败）。

在启动 RAID-5 卷时采用下列步骤：

1. 如果 RAID-5 卷未能干净地关闭，它将检查有效的 RAID-5 日志 `plex`。
  - ◆ 如果存在有效的日志 `plex`，则将重放这些日志 `plex`。通过将卷置于 DETACHED 内核状态并将卷状态设置为 REPLAY 和激活 RAID-5 日志 `plex` 来重放这些日志 `plex`。如果能够成功地读取和重放日志，则进入步骤 2。
  - ◆ 如果不存在有效日志，则必须再同步奇偶校验。通过将卷置于 DETACHED 内核状态并将卷状态设置为 SYNC 来完成再同步。所有的日志 `plex` 都继续处在 DISABLED 状态

由于在再同步奇偶校验时任何子磁盘失效都将导致卷不可用，因此在这一时期使卷不可用。通过使用 `-o unsafe` 启动选项和 `vxvol` 命令来忽略这种情况。如果存在任何失效子磁盘，则 RAID-5 卷不可用。

---

**警告** `-o unsafe start` 选项是危险的，它可能导致卷的内容不可用。因此不推荐使用该选项。

---

2. 将全部现有日志 `plex` 置零并激活这些日志。如果在该进程期间全部日志失败，则终止启动进程。
3. 如果不存在失效子磁盘或可以修复那些失效子磁盘，则将卷置于 ENABLED 内核状态并将卷状态设置为 ACTIVE。现在卷启动。
4. 如果存在一些需要恢复的失效子磁盘，有效日志也存在，则通过将卷置于 ENABLED 内核状态激活卷，卷在子磁盘恢复期间可用。否则，将卷内核状态设置为 DETACHED，卷在子磁盘恢复期间不可用。

这样做是由于在卷处于活动状态时系统崩溃或卷非从容停止，从而使奇偶校验失效，卷不可用。如果不喜欢这种状态，则可以用 `-o unsafe start` 选项启动卷。



---

**警告** `-o unsafe start` 启动选项是危险的，它可能导致卷的内容不可用。因此不推荐使用该选项。

---

5. 将卷状态设置为 RECOVER 并还原失效子磁盘。由于每个子磁盘上的数据都变为有效，因此将子磁盘标记为不再失效。
6. 如果子磁盘恢复失败并且无有效日志，则由于子磁盘保持失效和系统崩溃使 RAID-5 卷不可用，将导致卷启动终止。通过使用 `-o unsafe start` 选项也可以忽略这种情况。

---

**警告** `-o unsafe start` 选项是危险的，它可能导致卷的内容不可用。因此不推荐使用该选项。

---

如果卷中包含有效日志，则将通知子磁盘恢复失败但不停止启动过程。

7. 当已恢复了所有子磁盘，将卷置于 ENABLED 内核状态并将其标记为 ACTIVE。现在卷启动。

## 更改 RAID-5 卷的属性

可以更改 RAID-5 卷的某些属性。对于 RAID-5 卷，可以通过使用 `vxvol set` 命令来更改卷长度和 RAID-5 日志长度。使用下面的命令来更改 RAID-5 卷的长度：

```
vxvol set len=10240 r5vol
```

卷的长度不能超出 RAID-5 plex 的映射区域（称做*相邻长度*或 *contiglen*）。长度不能长到使卷不可用。如果 RAID-5 卷是活动的并且长度正被缩短，则必须通过使用 `-o force usage` 选项来强制完成操作。这样做是为了防止使用卷的应用程序删除空间。

也可以通过使用 `vxvol set` 和下面的命令来更改 RAID-5 日志的长度：

```
vxvol set loglen=2M r5vol
```

记住，RAID-5 日志 plex 只有映射 RAID-5 卷日志全长时才有效。如果增加日志长度使 RAID-5 日志无效，则不允许执行操作。同样，如果卷不是活动的并且是 **dirty**（关闭不彻底），则不能更改日志长度。这样避免了丢失任何日志内容（如果减小了日志长度）或将随机数据引入日志（如果增大了日志长度）。

## 向 RAID-5 阵列写入

本段描述 RAID-5 阵列的写入过程。

## 读取 - 修改 - 写入

在向 RAID-5 阵列写入时，对于 I/O 中涉及到的每个条带可以遵照下列步骤：

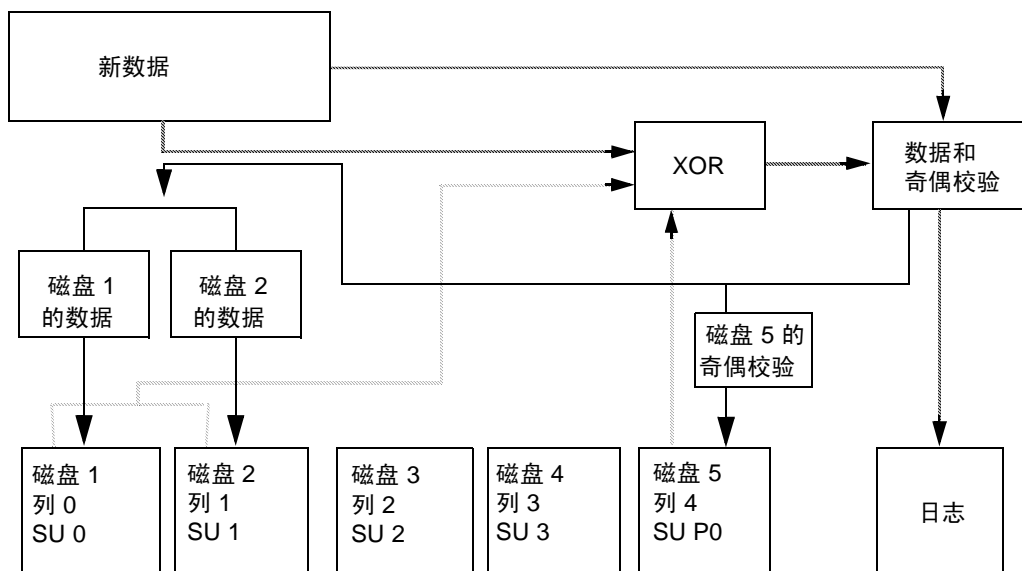
1. 存取用新写入数据更新的数据条带单元，并将其读入内部缓冲区。将奇偶校验单元读入内部缓冲区。
2. 更新奇偶校验以反映新数据区域的内容。首先，旧数据的内容经过与奇偶校验的异或 (XOR) 操作（逻辑删除旧数据）。然后将新数据 XOR 为奇偶校验（逻辑添加新数据）。将新数据和新奇偶校验写入日志。
3. 将新奇偶校验写入奇偶校验条带单元。将新数据写入数据条带单元。一次性地写入所有条带单元。

本过程称为 *读取 - 修改 - 写入* 循环，它是 RAID-5 的默认写入类型。如果磁盘失败，则磁盘上的数据和奇偶校验条带将不可用。这时称磁盘阵列在 *退化* 方式中操作。

读取 - 修改 - 写入序列如图 27 “*读取 - 修改 - 写入*” 中所示



图 27. 读取 - 修改 - 写入



SU = 条带单位

- = 第 1 步: 读取数据 (从奇偶校验条带单元 P0 和数据条带单元 0 和 1 中)。
- = 第 2 步: 在数据和奇偶校验之间完成 XOR 以计算新奇偶校验。记录新数据和新奇偶校验。
- = 第 3 步: 将新奇偶校验 (由于 XOR 的结果) 写入奇偶校验条带单元 P0, 将新数据写入数据条带单元 0 和 1。

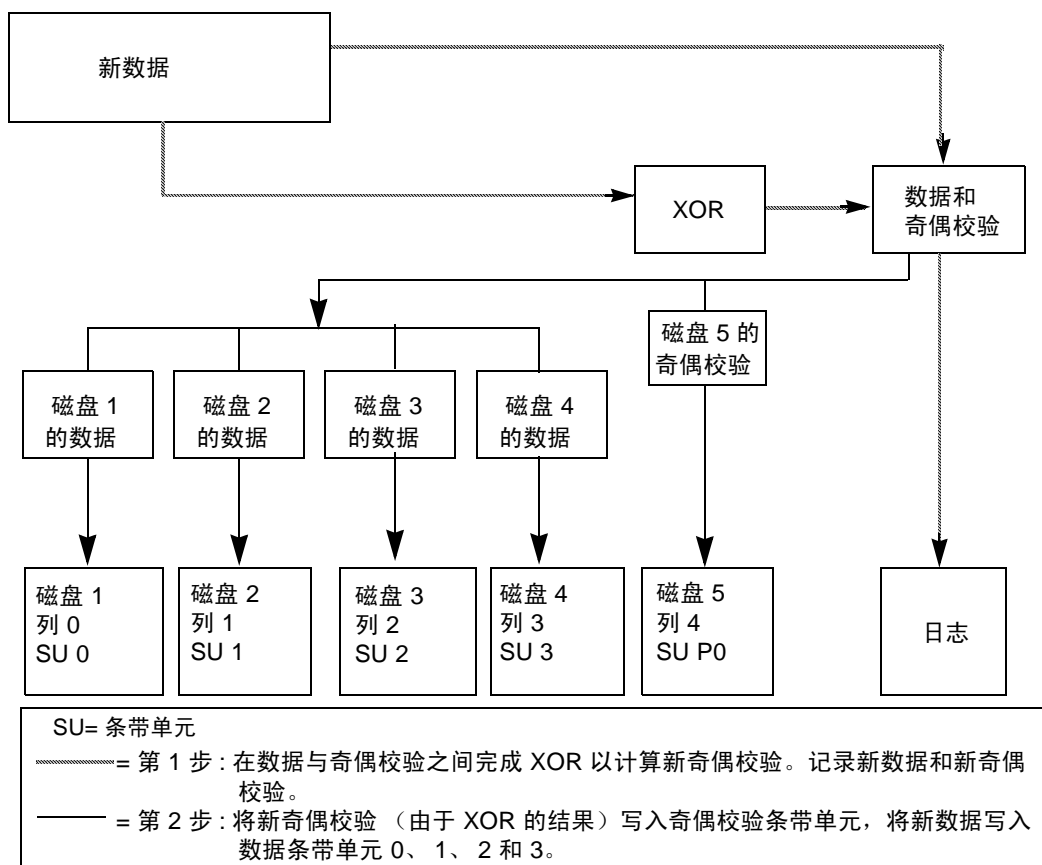
## 全条带写入

当发生大量写入 (写入包括整个数据条带) 时, 可以利用全条带写入而不必使用读取 - 修改 - 写入过程。由于全条带写入不需要发生读取过程, 因此它比读取 - 修改 - 写入的速度快。消除读取循环, 减少向磁盘写入所需的 I/O 时间。全条带写入过程由下列步骤组成:

1. 将所有的新数据条带单元 XOR 在一起, 产生一个新的奇偶校验值。将新数据和新奇偶校验写入日志。
2. 将新奇偶校验写入奇偶校验条带单元。将新数据写入数据条带单元。一次性地写入全部条带。

图 28 “全条带写入” 显示全条带写入。

图 28. 全条带写入



## 重构 - 写入

当一次 I/O 写入 50% 或 50% 以上的数据磁盘时，可以使用 **重构 - 写入**。重构 - 写入通过 XOR 节省 I/O 时间。XOR 不要求读取奇偶校验区域，只要求读取不受影响的数据。不受影响的数据的数量不到条带中条带单元的 50%。

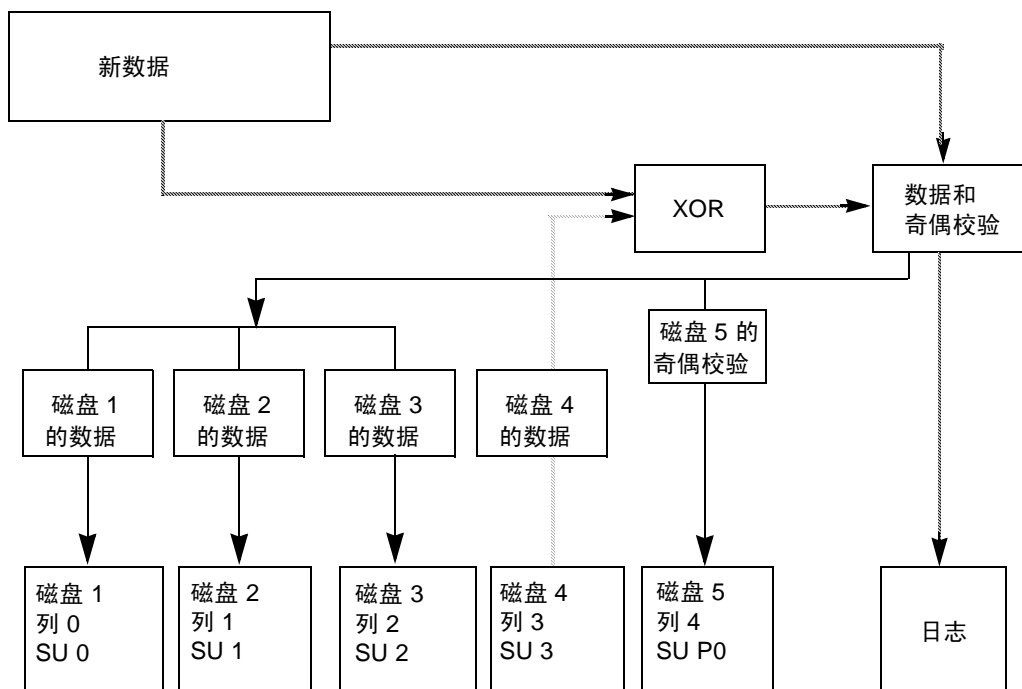
重构 - 写入过程由下列步骤组成：

1. 从未更改的数据条带单元中读取不受影响的数据。
2. 将新数据与旧的未受影响的数据 XOR，产生新奇偶校验条带单元。新数据和产生的奇偶校验记入日志。

3. 将新奇偶校验写入奇偶校验条带单元。将新数据写入数据条带单元。一次性地写入所有条带单元。

图 29 “重构 - 写入” 显示重构 - 写入。由于重构 - 写入只读取必需的数据磁盘而不是读取磁盘和奇偶校验磁盘，因此在这种情况下重构 - 写入比读取 - 修改 - 写入更优越。

图 29. 重构 - 写入



SU= 条带单元

- = 第 1 步: 从受影响的数据条带单元 3 中读取数据。
- = 第 2 步: 在旧的、未受影响的数据和新数据之间完成 XOR。记录新数据和新奇偶校验。
- = 第 3 步: 将新奇偶校验 (由于 XOR 的结果) 写入奇偶校验条带单元 P0, 将新数据写入数据条带单元 0、1 和 2。

## 介绍

逻辑卷管理是一种工具，它可以提高系统的总体性能。本章的性能管理和配置原则可以帮助您从 Volume Manager 提供的优点中获益。本章提供有关建立性能优先级的信息，并描述获得和使用适当数据的方法。

本章讲述以下主题：

- ◆ 性能准则
- ◆ 性能监视
- ◆ 调整 Volume Manager

## 性能准则

借助 Volume Manager，可以灵活地对存储器进行配置，从而提高系统性能。可以使用两种基本策略优化性能：

- ◆ 将数据分配给多个物理磁盘以均衡可用磁盘间的 I/O 负载
- ◆ 识别存取最频繁的数据并通过使用条带化和镜像增加那些数据的存取带宽

Volume Manager 也通过镜像和 RAID-5 提供数据冗余，允许在磁盘发生故障时继续存取数据。

## 数据分配

当确定在哪里定位文件系统时，系统管理员通常试图平衡可用磁盘间的 I/O 负载。预测未来使用模式很困难，并且无法跨驱动器分割文件系统，这些都限制了该方法的有效性。例如，如果单个文件系统接收大部分磁盘存取，当将该文件系统放在另一个磁盘上时，瓶颈也随之转移到这个磁盘上。

Volume Manager 可以跨多个磁盘分割卷，因此可以达到更精细的数据放置最小单位级别。衡量了实际的存取模式后，系统管理员可以调整文件系统放置决策。建立或更改性能模式之后可以联机重新配置卷，而不会对卷的可用性造成负面影响。



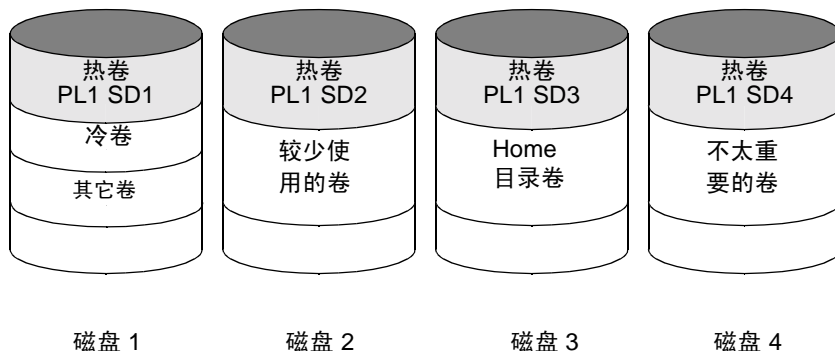
## 条带化

条带化是一种“分片”数据并在多个设备上存储数据以提高存取性能的方法。条带化为 plex 提供增加的存取带宽。条带 plex 提高了读写操作的存取性能。

如果可以识别存取最频繁的卷（包含文件系统或数据库），则可以了解性能收益。通过跨多个磁盘的各部分条带化这些“通信繁重的”数据，可以增加这些数据的存取带宽。

图 30 “利用条带化优化数据存取”是已被识别为数据存取瓶颈的单个卷（热卷）的实例。该卷跨四个磁盘条带化，四个磁盘的剩余部分则被释放，供不经常用到的卷使用。

图 30. 利用条带化优化数据存取



## 镜像

镜像是一种在系统上存储多个数据拷贝的技术。如果使用正确，镜像可用于提供连续的数据可用性，防止由于物理介质故障引起的数据丢失。使用镜像可提高系统崩溃或磁盘发生故障时恢复数据的可能性。

在某些情况下，镜像也可以用于提高系统性能。镜像频繁存取的数据不仅可以防止由于磁盘故障引起的数据丢失，而且可以提高 I/O 性能。但是与条带化不同，通过使用镜像获得的性能取决于磁盘存取的读 / 写比率。如果系统工作量主要是写密集的（例如，写的比例大于百分之 30），则镜像会导致性能稍微降低。

为了提供适合不同类型镜像卷的优化性能，Volume Manager 支持这些读策略：

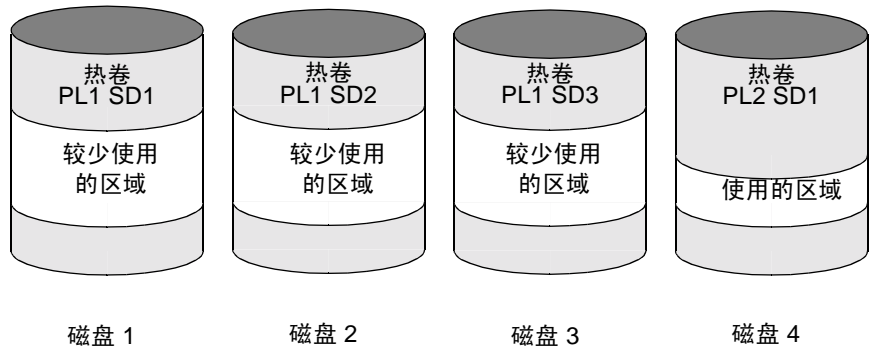
- ◆ **循环读策略 (round)**，以循环方式满足卷中所有 plex 对卷的读请求。（这意味着 plex 依次请求。）
- ◆ **首选 plex 读策略 (prefer)**，满足一个特定的 plex（大概是具有最高性能的 plex）的读请求，除非该 plex 失败，这时才存取另一个 plex。



- ◆ 默认读策略 (`select`)，选择适合配置的读策略。例如，当卷只有一个关联的条带 `plex` 时选择首选 `plex` 读策略，其它大多数情况下选择循环读策略。

在图 31 “利用镜像和条带化提高性能”中显示的配置实例中，对于带 `PL1` 标签的条带 `plex`，带热卷标签的卷的读策略应设置为 `prefer`。这样，指向 `PL1` 的读取将在许多其它相对于单个磁盘使用较少的磁盘间分布负载。

图 31. 利用镜像和条带化提高性能



为了提高读密集性工作量的性能，同一卷上可以挂接多达 32 个 `plex`。但是，该方案将导致减少有效磁盘空间的利用率。通过跨一半可用磁盘条带化形成一个 `plex`，跨另一半可用磁盘形成另一个 `plex`，也可以提高性能。如果可行，该方案通常是在一组磁盘上配置 `Volume Manager` 以获得最佳性能和合理可靠性的最好方法。

## 镜像和条带化

当一起使用镜像和条带化时，它们提供跨多个磁盘分布数据和数据冗余的优点。

当有多个 I/O 流时，一起使用镜像和条带化可以极大地提高性能。当 I/O 正好适合一个条带中的所有条带单元时，条带化可以改进串行存取。由于并行 I/O 流可以在不同的设备上并行地运行，从而达到更好的吞吐量。

由于镜像最常用于防止因磁盘故障引起的数据丢失，写密集的工作负荷有时可能需要用到镜像。在这些情况中，可以将镜像和条带化结合起来以提供高可用性和性能。

## 条带化和镜像

当一起使用条带化和镜像时，它们提供跨多个磁盘分布数据和数据冗余的优点。

当有多个 I/O 流时，一起使用条带化和镜像可以极大地提高性能。当 I/O 完全适合一个条带中的所有条带单元时，条带化可以改进串行存取。由于并行 I/O 流可以在不同的设备上并行地运行，从而达到更好的吞吐量。

由于镜像最常用于防止因磁盘故障引起的数据丢失，写密集的工作量有时可能需要使用镜像。在这些情况中，可以将镜像和条带化结合起来以提供高可用性和性能。请参见本指南中的“[分层卷](#)”。

## RAID-5

RAID-5 提供一起使用镜像和条带化的许多优点，而 RAID-5 需要的磁盘空间更少。RAID-5 读性能类似于条带化的读性能，RAID-5 奇偶校验提供的冗余类似于镜像。RAID-5 的缺点包括写速度相对较慢。

---

**注意** 通常不将 RAID-5 视为性能提高机制，但应用程序的存取模式中显示的高读写比率的情况除外。

---

## 性能监视

系统管理员有两组优先级。一组是 *物理的*，与硬件有关。另一组是 *逻辑的*，与管理软件及其操作有关。

### 性能优先级

物理性能特点解决每个驱动器上的 I/O 平衡和集中驱动器的 I/O 以最小化查找时间。基于监视结果，可移动子磁盘的位置以平衡磁盘。

逻辑优先级涉及软件操作及其管理方式。基于监视，可以镜像或条带化某些卷以提高其性能。为提高重要卷的性能，可以牺牲总体吞吐量。只有系统管理员可以决定系统中什么是重要的以及做怎样的权衡。

通过跨合理数量的磁盘条带化和镜像所有卷并在可能时在控制器之间进行镜像，通常可以获得最佳性能。这往往会均衡所有磁盘间的负载。但是，这通常使 **Volume Manager** 变得更难管理。如果有大量的磁盘（数以百计或数以千计），可以按 10 个一组放置磁盘（使用磁盘组），其中每个组用于条带化和镜像一个卷集。这仍提供了好的性能并减轻了管理任务。

## 获取性能数据

Volume Manager 提供两种类型的性能信息：I/O 统计和 I/O 跟踪。每种信息都有助于性能监视。vxstat 实用程序用于检索 I/O 统计信息，vxtrace 实用程序用于检索 I/O 跟踪。本章包括对每个实用程序的简要论述。

### 获得 I/O 统计 (vxstat)

vxstat 实用程序存取受 Volume Manager 控制的卷、plex、子磁盘和磁盘的活动信息。vxstat 报告的统计数据则反映了自引导以来 Volume Manager 对象的活动程度。可以同时显示一个特定 Volume Manager 对象和所有对象的统计数据。也可以指定一个磁盘组，在此情况下只显示该磁盘组中对象的统计数据。如果未指定磁盘组，则使用 rootdg。

显示的信息量取决于给 vxstat 指定的选项。有关可用选项的详细信息，请参阅 vxstat (1M) 帮助手册。

Volume Manager 记录下列 I/O 统计：

- ◆ 操作计数
- ◆ 传送的块数（一个操作可以涉及多个块）
- ◆ 平均操作时间（反映经过 Volume Manager 接口的总时间，不适合与其它统计程序比较）

Volume Manager 记录逻辑 I/O 以前的 I/O 统计。统计信息包括对每个卷的读、写、原子拷贝、验证的读出、验证的写入、plex 读和 plex 写。结果，对含 2 个 plex 的卷的一个写操作至少产生 5 个操作：每个 plex 一个，每个子磁盘一个，该卷一个。同样，跨 2 个子磁盘的一个读操作至少显示 4 个读操作——每个子磁盘一个，plex 一个，该卷一个。

Volume Manager 也维护其它统计数据。维护每个 plex 的读故障和写故障。对于卷，纠正的读故障和写故障与读故障和写故障相伴而生。

vxstat 也可以将统计信息重置为 0。使用命令 vxstat -r 可清除所有统计。这可以针对所有对象或只针对指定的对象。若在操作即将开始前重置，则可以测量那个操作的影响。

下面是 vxstat 输出的一个实例：

| 类型  | 名称        | 操作    |        | 块      |         | 平均时间（毫秒） |       |
|-----|-----------|-------|--------|--------|---------|----------|-------|
|     |           | 读     | 写      | 读      | 写       | 读        | 写     |
| vol | blop      | 0     | 0      | 0      | 0       | 0.0      | 0.0   |
| vol | foobarvol | 0     | 0      | 0      | 0       | 0.0      | 0.0   |
| vol | rootvol   | 73017 | 181735 | 718528 | 1114227 | 26.8     | 27.9  |
| vol | swapvol   | 13197 | 20252  | 105569 | 162009  | 25.8     | 397.0 |
| vol | testvol   | 0     | 0      | 0      | 0       | 0.0      | 0.0   |

对于 RAID-5 配置还有其它的卷统计数据可用。有关更多信息，请参见 vxstat (1M) 帮助手册。



## 跟踪 I/O (vxtrace)

vxtrace 命令跟踪卷上的操作。vxtrace 或者将内核 I/O 错误或 I/O 跟踪记录打印到标准输出，或者将记录以二进制格式写入文件。跟踪可以应用于特定的内核 I/O 对象类型，或者应用于指定的对象或设备。有关其它信息，请参阅 vxtrace(1M) 帮助手册。

## 使用性能数据

一旦收集到性能数据，便可以使用这些数据决定最佳系统配置，以有效地利用系统资源。以下各节概述如何使用性能数据。

### 使用 I/O 统计

检查 I/O 统计可以确定是否需要重新配置。有两种主要统计：卷 I/O 活动和磁盘 I/O 活动。

获得统计数据之前，清除（重置）所有现有的统计数据。使用命令 `vxstat -r` 清除所有统计数据。清除统计将消除由于创建卷引起的卷或磁盘之间的任何差异，也可以删除引导时的统计（这些统计通常不重要）。

清除统计之后，允许系统在典型系统活动期间运行。若要测量具体某个应用程序或工作负荷的效果，应在该应用程序或工作负荷上运行系统。当监视多用途系统时，使用任何一个应用程序的次数尽量不要比正常情况下多。当监视具有许多用户的分时系统时，尽量在一天内几个小时的正常使用期间累积统计数据。

若要显示卷统计，请不带参数使用命令 `vxstat`。下面是卷统计的典型显示：

| 类型  | 名称      | 操作     |        | 块       |         | 平均时间（毫秒） |       |
|-----|---------|--------|--------|---------|---------|----------|-------|
|     |         | 读      | 写      | 读       | 写       | 读        | 写     |
| vol | archive | 865    | 807    | 5722    | 3809    | 32.5     | 24.0  |
| vol | home    | 2980   | 5287   | 6504    | 10550   | 37.7     | 221.1 |
| vol | local   | 49477  | 49230  | 507892  | 204975  | 28.5     | 33.5  |
| vol | rootvol | 102906 | 342664 | 1085520 | 1962946 | 28.1     | 25.6  |
| vol | src     | 79174  | 23603  | 425472  | 139302  | 22.4     | 30.9  |
| vol | swapvol | 22751  | 32364  | 182001  | 258905  | 25.3     | 323.2 |

该输出有助于识别具有异乎寻常的大量操作或过多读 / 写次数的卷。

若要显示磁盘统计，请使用命令 `vxstat -d`。下面是磁盘统计的典型显示：

| 类型 | 名称     | 操作    |        | 块      |        | 平均时间（毫秒） |       |
|----|--------|-------|--------|--------|--------|----------|-------|
|    |        | 读     | 写      | 读      | 写      | 读        | 写     |
| dm | disk01 | 40473 | 174045 | 455898 | 951379 | 29.5     | 35.4  |
| dm | disk02 | 32668 | 16873  | 470337 | 351351 | 35.2     | 102.9 |
| dm | disk03 | 55249 | 60043  | 780779 | 731979 | 35.3     | 61.2  |
| dm | disk04 | 11909 | 13745  | 114508 | 128605 | 25.0     | 30.7  |

您可能希望将卷从一个磁盘移到另一个磁盘。若要将卷 archive 移到另一个磁盘上，首先使用以下命令识别卷所在的磁盘：

```
vxprint -tvh archive
```

下面是一个实例显示：

```
V NAME USETYPE KSTATE STATE LENGTH READPOL PREFPLEX
PL NAME VOLUME KSTATE STATE LENGTH LAYOUT NCOL/WDTH MODE
SD NAME PLEX PLOFFS DISKOFFS LENGTH [COL/]OFF FLAGS
v archive fsgen ENABLED ACTIVE 204800 SELECT -
pl archive-01 archive ENABLED ACTIVE 204800 CONCAT - RW
sd disk03-03 archive-01 0 409600 204800 0 device
```

**注意** 您的系统使用的设备名称可能不同于这些实例。有关设备名称的更多信息，请参见本指南中的“[Volume Manager 操作](#)”。

相关子磁盘列表指出 archive 卷在磁盘 disk03 上。若要将该卷从 disk03 移出，请使用此命令：

```
vxassist move archive !disk03 dest_disk
```

这里 *dest\_disk* 是希望将该卷移到的磁盘。没有必要指定 *dest\_disk*。如果未指定，该卷将移到有足够空间容纳该卷的可用磁盘。

例如，使用以下命令将该卷从 disk03 移到 disk04：

```
vxassist move archive !disk03 disk04
```

此命令表明将重新组织该卷，使得 disk03 上没有剩余部分。

**注意** 图形用户界面提供了更容易的方法在磁盘之间移动部分卷，因此可能优于使用命令行。

如果有两个忙卷（根卷以外的卷），移动它们使得每个卷在不同的磁盘上。

如果有一个卷特别忙（尤其当该卷有异乎寻常的大量平均读写时间时），则条带化该卷（或将该卷分割成多个部分，每个部分在不同的磁盘上）。如果将卷联机转换为使用条带化，则需要足够的空闲空间存储卷的额外副本。如果没有足够的空闲空间可用，则可以改为创建一个备份副本。若要转换为条带化，先创建卷的条带 plex，然后删除旧的 plex。例如，若要跨磁盘 disk02、disk03 和 disk04 条带化卷 archive，请使用这些命令：

```
vxassist mirror archive layout=stripe disk02 disk03 disk04
vxplex -o rm dis archive-01
```

重新组织任何特别忙的卷之后，检查磁盘统计数据。如果有些卷已经重新组织，则首先清除统计，然后累积一段合理时间内的统计数据。

如果一些磁盘看起来非常忙（或有特别长的读写时间），则可能需要重新配置某些卷。如果一个磁盘上有两个相对较忙的卷，则将它们移得更近些以缩短磁盘上的查找时间。如果一个磁盘上有太多相对较忙的卷，则将它们移到不太忙的磁盘上。

使用 I/O 跟踪（或子磁盘统计）确定在卷的特定区域内是否有过多的活动。如果可以识别活动区域，则分割卷中的子磁盘并将这些区域移到不太忙的磁盘上。



**警告** 条带化卷或跨多个磁盘分割卷将增加因磁盘故障导致卷发生故障的可能性。例如，如果跨相同的五个磁盘条带化五个卷，则只要其中任何一个磁盘发生故障，都需要从备份恢复所有五个卷。如果每个卷在不同的磁盘上，则只需要恢复一个卷。使用镜像或 RAID-5 可减少单个磁盘的故障导致许多卷故障的可能性。

---

注意，文件系统和数据库通常随着时间改变已分配空间的使用，因此卷上的这些位置特有的信息经常是无用的。对于数据库，识别由特别忙的索引或表使用的空间或许是可能的。如果可以识别这些空间，则应将它们移到不忙的磁盘上。

检查读写比率有助于识别可通过镜像提高性能的卷。如果读写比率高，则镜像可以增强性能和可靠性。通过镜像可提高性能的读写比率主要取决于磁盘、磁盘控制器、是否使用多个磁盘控制器以及系统总线速度等因素。如果特别忙的卷的读写比率高，则镜像可以显著提高该卷的性能。

### 使用 I/O 跟踪

I/O 统计提供基本性能分析数据；I/O 跟踪用于更详细的分析。使用 I/O 跟踪，焦点缩小到获得特定工作负荷的事件跟踪。这有助于直接识别热点的位置和大小以及导致热点的应用程序。

使用 I/O 跟踪中的数据，可以模拟磁盘上的真实工作负荷并跟踪结果。通过使用这些统计信息，系统管理员可以预测系统限制并规划其它资源。

## 调整 Volume Manager

本节描述控制由 Volume Manager 使用的资源的机制。可能需要调整一些可调值以获得最佳性能（取决于可用系统资源的类型）。

### 一般调整原则

对于从小系统到大型服务器的大多数配置都可以调整 Volume Manager。如果在大型系统上可以通过调整增强性能，但要以有价值的资源（如内存）为代价，则通常将 Volume Manager 调整到在支持的最小配置上运行。在执行这些调整更改时应小心，因为它们可能会对系统的总体性能造成负面影响，甚至可能会使 Volume Manager 变得不可用。

存在各种调整 Volume Manager 的机制。在一些系统上，可以使用全局可调文件 `/etc/system` 调整数个参数。其它值只能使用 Volume Manager 的命令行接口调整。

## 可调值

在有些系统上，`idttune` 命令应用于修改可调值。有关详细信息，请参阅 `idttune(1M)` 帮助手册。

在其它系统上，可以通过向 `/etc/system` 文件中添加行然后重新引导系统来修改可调值。更改过的可调值随后生效。

例如，若要将称为 `vol_tuneme` 的可调值的默认值更改为 5000，应将下行插入到 `/etc/system` 文件的适当部分：

```
set vxio:vol_tuneme=5000
```

在许多情况下，可调值包含在 `volinfo` 结构中，如 `vxio(7)` 帮助手册所描述的那样。

以下各节描述特定的可调值。

### `vol_maxvol`

该值控制系统上可以创建的最大卷数。可以将该值设置在 1 与系统可以表示的最大次要数之间。

该可调值的默认值是系统中最大次要数值的一半。

### `voliomem_maxpool_sz`

该可调值的目的是防止一个 I/O 使用系统中的所有内存。

VxVM 分配两个 `voliomem_maxpool_sz` 池，一个用于 RAID-5，一个用于镜像卷。

当对 RAID-5 卷的写进入并且大于 `volio_maxpoll_sz/10` 时，将被分解并按 `volio_maxpoll_sz/10` 块执行。

当对镜像卷的写进入并且大于 `volio_maxpoll_sz/2` 时，将被分解并按 `volio_maxpoll_sz/2` 块执行。

### `vol_subdisk_num`

该可调值用于控制可挂载在单个 `plex` 上的最大子磁盘数。在理论上对这个数字没有限制，但出于实用目的，已将该数字限制为默认值 4096。如果需要可以更改该默认值。

### `vol_maxioctl`

该值控制可通过 `ioctl` 调用传递到 Volume Manager 中的数据的最大大小。增大该限制将允许执行较大的操作。建议通常情况下不要减小该限制，因为一些实用程序依赖于执行特定大小的操作，如果它们发出过大的 `ioctl` 请求，可能会意外地失败。

该可调值的默认值为 32768 字节 (32K)。





### `vol_maxspecialio`

该可调值控制由 `ioctl` 调用发出的 I/O 的最大大小。`ioctl` 请求本身可能很小，但可能会请求执行很大的 I/O。该可调值限制这些 I/O 的大小。如有必要，超过该值的请求可能失败，或者 I/O 可能分解并同步执行。

该可调值的默认值为 512 个扇区 (256K)。

### `vol_maxio`

该值控制不用中断请求即可执行的逻辑 I/O 操作的最大大小。大于该值的物理 I/O 请求将分解并同步执行。多个物理 I/O 的分解基于磁盘设备的容量，不受该最大逻辑请求限制的影响。

该可调值的默认值为 512 个扇区 (256K)。

如果 I/O 的大小导致进程所使用的内存或内核映射空间比现有的多，从而发生死锁，则增加该限制会导致困难。`vol_maxio` 的最大限制是物理内存或内核虚拟内存中较小者的 20%。超过该限制是不可取的，因为有可能发生死锁。

如果有大于 `vol_maxio` 的条带，则整个条带的 I/O 将分解，从而阻止全条带读 / 写。这控制了有序 I/O 或大型 I/O 的卷 I/O 吞吐量。

应将该可调值至少设置为最大条带的大小。该可调值原则适用于 Raid0 条带化和 Raid5 条带化。

### `vol_maxkiocount`

该可调值控制 Volume Manager 可以并行执行的最大 I/O 数。试图使用卷设备的附加 I/O 将排队等候，直到当前活动计数降到该值以下。

该可调值的默认值为 2048。

由于大多数进程线程一次只能发出一个 I/O，当达到内核中的活动 I/O 限制时，需要并行执行 2K I/O 请求。除了在最大系统上以外，增加该限制似乎不太可能提供更多的优点。

### `vol_default_iodelay`

如果已经指示实用程序减慢发出 I/O 的速度，但是未给出具体的延迟时间，该值是实用程序在发出 I/O 之间暂停的时钟周期计数。执行诸如再同步镜像或重构 RAID-5 列等操作的实用程序将使用该值。

该值的默认值为 50 个时钟周期。

增加该值将导致减慢恢复操作，从而减少执行恢复操作时的系统影响。



### `voldrl_min_regionsz`

使用 Dirty 区日志，Volume Manager 将卷在逻辑上划分成一系列连续的区域。`voldrl_min_regionsz` 可调值指定 DRL 卷区的最小扇区数。

Volume Manager 内核当前将该可调值的默认值设置为 1024 个扇区。

区域大小越大，越有可能提高区域的高速缓存命中率。这将提高写性能，但也将延长恢复时间。

### `voldrl_max_dirty`

一些卷（例如用于 Oracle 重放日志的卷）按有序方式写入，不从这种 DRL 位的惰性清理中获益。对于这些卷，可使用有序 DRL 进一步限制 dirty 位的数目并加快恢复速度。有序 DRL 所允许的 dirty 位数目由可调值 `voldrl_max_dirty` 限制。在有序写入的卷上使用有序 DRL 可能会严重影响 I/O 吞吐量。

### `voldrl_max_drtregs`

该可调值指定任何时候都可以存在于系统上的最大 Dirty 区数。这是应用于整个系统的全局值，与系统上的活动卷数无关。

该可调值的默认值为 2048。

可调值 `voldrl_max_drtregs` 可用于调整系统在失败后的更坏情况时的恢复时间。较大值可能导致以恢复时间为代价提高系统性能。

### `vol_maxparallelio`

该可调值控制每个 `VOL_VOLDIO_WRITE` ioctl 调用允许 `vxconfigd(1M)` 后台驻留程序从单个 `VOL_VOLDIO_READ` 的内核中请求的 I/O 操作数。

该可调值的默认值为 256，并且最好不要更改该值。

### `vol_mvr_maxround`

该值控制用于从镜像中读取的循环策略的最小单位。如果某个读的偏移在上个读的该可调值所描述的扇区数以内，则从相同的镜像中读取。

该值的默认值为 512 个扇区 (256K)。

增加该值将导致较少切换到替换镜像读取。如果执行的 I/O 主要在 I/O 之间少数几个小查找之后顺序发生，则最好增加该值。从替换镜像中读通常适用于大量随机分布式卷读。



### `voliot_iobuf_limit`

该值设置可用于在内核中存储跟踪缓冲区的内存大小限制。Volume Manager 内核使用跟踪缓冲区存储跟踪事件记录。由于要求将跟踪缓冲区存储在内核中，它们使用的内存从该池中提取。

增加该大小可允许以系统内存使用为代价执行附加跟踪。将该值设置为大于系统易于适应的大小是不可取的。

该可调值的默认值为 131072 字节 (128K)。

### `voliot_iobuf_max`

该值控制可用于单个跟踪缓冲区的最大缓冲区大小。大于该大小的缓冲区请求自动截尾到该大小。从跟踪接口请求最大缓冲区大小将获得（以使用限制为准）该大小的缓冲区。

该缓冲区的默认大小为 65536 字节 (64K)。

增大该缓冲区可以进行较大的跟踪，同时不会丢失使用非常频繁的卷。增加该值时应小心，不能超出可调值 `voliot_iobuf_limit` 的值。

### `voliot_iobuf_default`

如果对作为跟踪 `ioctl` 的一部分所需的内核缓冲区大小没有任何其它指定，则该值是创建跟踪缓冲区的默认大小。

该可调值的默认值为 8192 字节 (8K)。

如果由于该缓冲区太小经常丢失跟踪数据，则可以将该值调得更大些。

### `voliot_errbuf_default`

该可调值包含为错误跟踪事件维护的缓冲区的默认大小。该缓冲区在驱动程序装入时分配，并且当 Volume Manager 正在运行时，其大小是不可调的。

该缓冲区的默认值为 16384 字节 (16K)。

增加该缓冲区可以为更多的错误事件提供存储，但这是以系统内存为代价。减小该缓冲区的大小可能导致无法通过跟踪设备检测错误的情况。依赖错误跟踪执行某些响应操作的应用程序取决于该缓冲区。

### `voliot_max_open`

该值控制可以同时打开的最大跟踪通道数。跟踪通道是进入跟踪设备驱动程序的克隆入口点。每次在系统上运行 `vxtrace` 命令都消耗一个跟踪通道。

通道的默认数为 32。分配每个通道大约要占用 20 字节，即使不使用该通道。

### `vol_checkpoint_default`

该可调值控制时间间隔，在此间隔内，执行恢复或再同步操作的实用程序将当前偏移装入内核，这样，当系统失败时，不需要完全恢复，而是可以从上次已到达的检查点继续。

该检查点的默认值为 20480 个扇区 (10M)。

增加该大小将减少恢复操作的检查点开销，但代价是如果系统在恢复期间失败，则需要更长的时间恢复失败的操作。

### `volraid_rsrtransmax`

该 RAID-5 可调值控制可以并行执行的最大瞬态重构操作数。瞬态重构操作发生在非降级的 RAID-5 卷上，因此无法预测。通过限制这些操作可以同时发生的数目，消除了系统被同时发生的许多重构操作淹没的可能性，从而减小了导致内存不足状态的风险。

可以并行执行的这些瞬态重构操作的默认数目为 1。

故障首次发生并且在分离故障对象之前，增加该大小可以提高系统的初始性能，但会导致可能的内存不足状态。

### `voliomem_chunk_size`

使用该最小单位为 Volume Manager 分配系统内存和从 Volume Manager 中释放内存。较大的最小单位允许 Volume Manager 保留大量内存，从而可稍微减少内存分配开销。

该可调值的默认大小为 64K。

## 调整大型系统

在小型系统（少于 100 个驱动器）上不需要调整，Volume Manager 有能力为所有配置参数采用合理的默认值。但是，在大型系统上，由于能力和性能的原因，有些配置可能要求对调整这些参数有更多的控制。

在大型系统上设置 Volume Manager 时，通常只进行少数几个重要的决策。一个是决定磁盘组的大小和为每个磁盘组维护的配置副本数。另一个是决定磁盘组中所有磁盘的专用区域大小。

大型磁盘组的优点是可以为 `vxassist (1M)` 命令提供选择用的较大空闲空间池，并且允许创建较大的阵列。不过，小型磁盘组不需要那样大的配置数据库，因此可与较小的专用区域共存。超大型磁盘组最终会耗尽磁盘组中的专用区域大小，结果无法向该磁盘组添加更多的配置对象。此时，必须将配置分割到多个磁盘组中，或者必须扩大专用区域。这涉及重新初始化磁盘组中的每个磁盘（并且会涉及重新配置一切并从备份恢复）。

对磁盘阵列子系统用户的一般建议是，为每个阵列创建一个单独的磁盘组，以便磁盘组可以作为一个单元在系统之间物理移动。



## 磁盘组的配置副本数

磁盘组的配置副本数取决于冗余与性能之间的权衡。一般的规则是，磁盘组中存在的配置副本越少，初始存取该组的速度越快，`vxconfigd(1M)` 的初始启动越快，在磁盘组上执行事务的速度也越快。

---

**警告** 数据库副本的低冗余风险是丢失配置数据库。丢失数据库将导致丢失数据库中的所有对象和磁盘组包含的所有数据。

---

磁盘组配置副本的默认策略是为磁盘组中标识的每个控制器分配一个配置副本，或者为在同一目标上包含多个可寻址磁盘的每个目标分配一个配置副本。从冗余角度看这已足够了，但在某些情况下会导致大量的配置副本。

如果情况是这样，建议将配置副本数限制为最小值 4。象前面那样，根据最大范围的控制器或目标分布选择副本的位置。

设置磁盘组副本数的机制是使用命令 `vxdg init` 设置新组（有关详细信息，请参见 `vxdg(1M)` 帮助手册）。而且，可以使用 `vxedit set` 命令更改现有组的副本（有关详细信息，请参见 `vxedit(1M)` 帮助手册）。例如，若要将称为 `foodg` 的磁盘组设置为包含 5 个副本，请使用此命令：

```
vxedit set nconfig=5 foodg
```

# 词汇表

---

## 活动 / 活动磁盘阵列

这种多路径磁盘阵列允许在不降低性能的情况下，通过指向磁盘阵列中某个磁盘的所有路径同时存取该磁盘。

## 活动 / 被动磁盘阵列

这种多路径磁盘阵列允许指定磁盘的一条路径作为主路径并在任何时候通过该路径存取磁盘。在一些磁盘阵列中，使用指定的活动路径以外的路径将导致性能严重下降。请参见“*路径*”、“*主路径*”、“*辅助路径*”。

## 关联

建立 Volume Manager 对象之间关系的过程。例如，创建一个子磁盘并定义为在某个 plex 中有起始点，则称该子磁盘与那个 plex 关联。

## 关联的 plex

与卷关联的 plex。

## 关联的子磁盘

与 plex 关联的子磁盘。

## 原子操作

一种操作方式，操作或者完全成功，或者失败而令一切保持操作开始前的状态。如果操作成功，该操作的所有方面立即生效并且看不见更改的中间状态。如果操作的任何一方面失败，则操作终止并且不保留局部更改。

## 已挂接

一种状态，在此状态下 VxVM 对象既与另一对象关联又可使用。

## 块

传送到磁盘或阵列的最小数据单元。



---

## 引导磁盘

用于引导的磁盘。此磁盘可能受 VxVM 的控制。

## 干净节点关闭

当对共享卷的所有存取停止时节点平稳离开集群的能力。

## 集群

共享磁盘集的一组主机。

## 集群管理器

由外部提供并运行于集群内所有节点上的后台驻留程序。每个节点上的集群管理器相互通信并将集群成员的更改通知 VxVM。

## 集群共享磁盘组

磁盘由多个主机共享的磁盘组（也称为 *共享磁盘组*）。

## 列

条带 plex 内的一个或多个子磁盘的集合。通过将数据交替并均匀地分配在 plex 内的列中来完成条带化。

## 连续

以子磁盘的有序和连续排列为特点的布局式样。

## 配置数据库

包含关于现有 Volume Manager 对象（如磁盘和卷属性）详细信息的记录集。配置数据库的一个副本称为一个配置副本。

## 数据条带

指条带的可用数据部分，等于条带减去奇偶校验区域。

## 分离的

一种状态，在该状态下 VxVM 对象与另一对象关联，但不能使用。

## 设备名称

用于存取物理磁盘的设备名称或地址，例如 c0t0d0。c#t#d#s# 语法分别标识控制器、目标地址、磁盘和区域（某些系统上为分区）。



---

## 脏区日志

Volume Manager 监视和记录对 plex 的修改的过程。更改区域的位图保存在称为 *日志子磁盘* 的关联子磁盘中。

## 禁用的路径

不可用于 I/O 的磁盘路径。由于实际的硬件故障或用户在那个控制器上使用了 `vxddmpadm disable` 命令，会导致路径被 *禁用*。

## 磁盘

已建立了索引并可以相当快地被存取的读 / 写数据块的集合。每个磁盘具有一个通用唯一标识符。

## 磁盘存取名

用于存取物理磁盘的名称，例如 `c0t0d0`。`c#t#d#s#` 语法分别标识控制器、目标地址、磁盘和分区。术语 *设备名称* 也可用于指磁盘存取名。

## 磁盘存取记录

用于指定特定磁盘的存取路径的配置记录。每个磁盘的存取记录均包含名称和类型，可能的话还包含某些类型特有的信息。Volume Manager 使用这些信息决定如何存取和操作由磁盘存取记录定义的磁盘。

## 磁盘阵列

逻辑上排列为一个对象的磁盘集合。阵列有住有助于提供诸如冗余或提高的性能等优点。

## 磁盘阵列序列号

这是磁盘阵列的序列号。通常印在磁盘阵列外匣上，也可以通过对磁盘阵列上的磁盘发出供应商特定的 SCSI 命令获得。DMP 子系统用此数字唯一地标识磁盘阵列。

## 磁盘控制器

与主机 或 与由操作系统表示为磁盘父节点的磁盘阵列相连的控制器 (HBA)，Volume Manager 的多路径子系统称它们为磁盘控制器。例如，如果磁盘由以下设备名表示：

```
/devices/sbus@1f,0/QLGC,isp@2,10000/sd@8,0:c
```

则磁盘 `sd@8,0:c` 的磁盘控制器是：

```
QLGC,isp@2,10000
```

此控制器 (HBA) 与主机相连。



---

## 磁盘组

共享公用配置的磁盘集合。磁盘组配置是一个记录集，包含现有 Volume Manager 对象（如磁盘和卷属性）及其相互关系的详细信息。每个磁盘组有一个管理员分配的名称和一个内部定义的唯一 ID。根磁盘组 (rootdg) 是一个始终存在的特殊专用磁盘组。

## 磁盘组 ID

用于标识磁盘组的唯一标识符。

## 磁盘 ID

给每个磁盘提供的通用唯一标识符，可用于标识磁盘，即使磁盘已经移动。

## 磁盘介质名

为磁盘选择的逻辑或管理名称，例如 disk03。术语 *磁盘名* 也用于指磁盘介质名。

## 磁盘介质记录

用磁盘 ID 标识具体某个磁盘并给该磁盘提供逻辑（或管理）名称的配置记录。

## 分离

删除存在于两个 Volume Manager 对象间的任何链接的过程。例如，从 plex 分离子磁盘意味着从 plex 中删除子磁盘并将其添加到空闲空间池中。

## 分离的 plex

从卷分离的 plex。

## 分离的子磁盘

从 plex 分离的子磁盘。

## 分布式锁管理器

运行在不同系统上并保证一致地存取分布式资源的锁管理器。

## 启用的路径

可用于 I/O 的磁盘路径。

## 封装

将指定磁盘上的现有分区转换为卷的过程。如果任何分区包含文件系统，则将修改 /etc/vfstab 项，以便将文件系统安装到卷上。封装不适用于某些系统。

## 文件系统

组织成某种结构的文件集合。UNIX 文件系统是由目录和文件组成的分层结构。





---

## 空闲空间

受 VxVM 控制的磁盘区，未分配给任何子磁盘或留给任何其它 Volume Manager 对象使用。

## 空闲子磁盘

与任何 plex 没有关联的子磁盘，具有空的 `putil[0]` 字段。

## 主机 ID

向 Volume Manager 标识主机的字符串。主机的 *主机 ID* 存储于它的 `volboot` 文件中，并用于定义磁盘和磁盘组的拥有权。

## 热重定位

当磁盘发生故障时，自动恢复冗余及对镜像卷和 RAID-5 卷的存取的技术。通过将受影响的子磁盘重新定位到同一磁盘组中指定为备用和（或）空闲空间的磁盘来实现。

## 起始节点

一类节点，系统管理员在其上运行请求对 Volume Manager 对象进行更改的实用程序。该节点启动卷重配置。

## 日志 plex

用于存储 RAID-5 日志的 plex。术语 *日志 plex* 还可用于指脏区日志 plex。

## 日志子磁盘

用于存储脏区日志的子磁盘。请参见“*脏区日志*”。

## 主节点

由软件指定为“主”节点的节点。任何节点都可以作为主节点。主节点协调特定的 Volume Manager 操作。

## 主控节点

磁盘挂接在的节点。也称为 *磁盘属主*。

## 镜像

卷及卷数据（以有序子磁盘集合的形式）的重复副本。每个镜像是与镜像关联的卷的一个副本。术语 *镜像* 等同于 *plex*。

## 镜像

将卷内容镜像到多个 plex 上的布局技术。每个 plex 复制存储在卷上的数据，但 plex 本身可能有不同的布局。



---

## 多路径

如果与系统相连的磁盘有多条物理存取路径，则称该磁盘是多路径的。我们说驻留在主机上并让用户无法察觉此事实的任何软件（如 DMP 驱动程序）提供多路径功能性。

## 节点

集群中的主机之一。

## 节点异常终止

节点在紧急情况下不试图停止正在进行的操作便离开集群的情况。

## 节点加入

节点加入集群并获得对共享磁盘的存取的过程。

## 对象

为 Volume Manager 定义并由 Volume Manager 在内部识别的实体。VxVM 对象包括：卷、plex、子磁盘、磁盘和磁盘组。实际上有两种类型的磁盘对象：物理磁盘对象和逻辑磁盘对象。

## 奇偶校验

一个计算出的值，可用于在故障后重建数据。当数据写入 RAID-5 卷时，还通过在数据上执行异或 (XOR) 过程来计算奇偶校验。所得到的奇偶校验值随后写入卷。如果 RAID-5 卷的一部分失败，可以从其余的数据和奇偶校验重新创建失败卷上的那部分数据。

## 奇偶校验条带单元

包含奇偶校验信息的 RAID-5 卷存储区域。包含在奇偶校验条带单元中的数据可用于帮助重建因 I/O 故障或磁盘故障而丢失的 RAID-5 卷的区域。

## 分区

物理磁盘设备的标准划分，直接由操作系统和磁盘驱动器支持。

## 路径

磁盘连接到主机后，磁盘的路径由主机上的 HBA（主机总线适配器）、SCSI 或光纤电缆连接器以及磁盘或磁盘阵列上的控制器组成。这些组件构成磁盘的路径。这些组件中的任何一个发生故障都将导致 DMP 试图将该磁盘的所有 I/O 操作转移到其余（替代）路径上。

## 永久状态日志

确保在恢复时只使用活动镜像并防止选择失败镜像的日志类型。也称为 *内核日志*。



---

## 物理磁盘

基础存储设备，不一定受 Volume Manager 的控制。

## plex

卷及卷数据（以有序子磁盘集合的形式）的重复副本。每个 plex 是所关联的卷的一个副本。术语 *镜像* 等同于 *plex*。

## 主路径

在主动 / 被动类型的磁盘阵列中，可以将磁盘绑定到磁盘阵列中某个具体的控制器上或由控制器拥有。这样，就可以使用通过此控制器的路径存取该磁盘。请参见“*路径*”、“*辅助路径*”。

## 专用磁盘组

磁盘仅由一个特定主机存取的磁盘组。

## 专用区域

物理磁盘中用于存储专用的结构化 Volume Manager 信息的区域。*专用区域* 包含磁盘磁头、内容表目录和配置数据库。内容表概括磁盘的内容。磁盘磁头包含磁盘 ID。复制专用区域中的所有数据以获得额外的可靠性。

## 公共区域

物理磁盘中由 Volume Manager 管理的区域，包含可用空间并用于分配子磁盘。

## 独立磁盘冗余阵列

独立磁盘冗余阵列 (RAID) 是一种磁盘阵列，设置为用组合存储容量的一部分来存储关于该阵列中所存储数据的重复信息。如果发生了磁盘故障，这使得有可能重新生成数据。

## 读取 - 写回模式

一种恢复模式，此模式的每次读操作恢复所读区域的 plex 的一致性。恢复 plex 一致性的方法是从一个 plex 的块中读数据，然后将数据写入所有其它可写的 plex。

## 根配置

根磁盘组的配置数据库。根配置的特别之处在于它总是包含其它磁盘组的记录，而这些记录仅用于备份目的。它还包含定义系统上所有磁盘设备的磁盘记录。

## 根磁盘

包含根文件系统的磁盘。该磁盘可以受 VxVM 的控制。



---

## 根磁盘组

始终存在于系统上的特殊专用磁盘组。根磁盘组的名称是 `rootdg`。

## 根文件系统

作为 UNIX 内核启动序列一部分安装的初始文件系统。

## 根分区

根文件系统驻留的磁盘区域。

## 根卷

包含根文件系统的 VxVM 卷（如果系统配置指定了这样的卷）。

## 根目录可置性

将根文件系统和交换设备置于 Volume Manager 控制之下的能力。然后，可以镜像所得的卷以提供冗余并允许在磁盘发生故障时恢复。

## 辅助路径

在活动 / 被动类型的磁盘阵列中，主路径之外的磁盘路径都称为辅助路径。在磁盘发生故障前，应只通过主路径存取磁盘；磁盘发生故障后，磁盘的拥有权转让给其中一个辅助路径。请参见“路径”、“主路径”。

## 扇区

一种大小单位，可能因系统而异。一个扇区通常为 512 字节。

## 共享磁盘组

磁盘由多个主机共享的磁盘组（也称为*集群共享磁盘组*）。

## 共享卷

属于共享磁盘组并且在不只一个节点上同时打开的卷。

## 共享 VM 磁盘

属于共享磁盘组的 VM 磁盘。

## 从属节点

未被指定为主节点的节点。

## 片

逻辑磁盘设备的标准划分。术语*分区*有时等同于*片*。



---

## 跨接

一种布局技术，允许单个磁盘无法容纳的卷（及其文件系统或数据库）跨在多个物理磁盘上。

## 稀疏 plex

与卷长不相等或存在洞（plex 中没有后备子磁盘的区域）的 plex。

## 条带

在一连串列中占据同一位置的条带单元集。

## 条带大小

横跨所有条带列组成单个条带的条带单元大小的总和。

## 条带单元

交替地在每个条带 plex 的子磁盘上（在列内）分配的大小相等的区域。在阵列中，条带单元是一系列在逻辑上相邻的块，在从阵列的下一个磁盘进行分配前存在于每个磁盘上。条带单元也称为条带元素。

## 条带单元大小

每个条带单元的大小。默认条带单元的大小是 32 个扇区 (16K)。条带单元大小过去也称为条带宽度。

## 条带化

使用条带将数据分布在多个物理磁盘上的布局技术。数据交替地分配到每个 plex 的子磁盘内的条带上。

## 子磁盘

形成逻辑磁盘段的一系列连续的相邻磁盘块。子磁盘可与 plex 关联以形成卷。

## 交换区

用于容纳由系统分页进程换出的内存页副本的磁盘区域。

## 交换卷

配置作为交换区的 VxVM 卷。

## 事务

作为一个整体而非个体成功或失败的一系列配置更改。事务用于在内部维护一致的配置。



---

## volboot 文件

用于定位根配置副本的小文件。该文件可以列出在标准位置包含配置副本的磁盘，并且还可以包含直接指向配置副本位置的指针。volboot 的存储位置因系统而异。

## VM 磁盘

受 Volume Manager 控制并分配给磁盘组的磁盘。VM 磁盘有时称为 *Volume Manager 磁盘* 或简单地称为 *磁盘*。在图形用户界面中，形象地用标有 D 的柱面表示 VM 磁盘。

## 卷

即虚拟磁盘，表示由诸如文件系统或数据库等应用程序使用的可寻址的磁盘块范围。卷是 1 个到 32 个 plex 的集合。

## 卷配置设备

卷配置设备 (/dev/vx/config) 是对卷设备驱动程序执行所有配置更改的接口。

## 卷设备驱动程序

在应用程序级和物理设备驱动程序级之间形成虚拟磁盘驱动器的驱动程序。卷设备驱动程序通过特定的虚拟磁盘设备节点存取，该节点的字符设备节点出现在 /dev/vx/rdisk 中，块设备节点出现在 /dev/vx/dsk 中。

## 卷事件日志

卷事件日志设备 (/dev/vx/event) 是向实用程序报告卷驱动程序事件的接口。

## vxconfigd

Volume Manager 配置后台驻留程序，负责更改 VxVM 配置。必须运行该后台驻留程序才能执行 VxVM 操作。



# 索引

---

## B

- 保留磁盘用于特殊目的 67
- 备份 35, 147
  - 镜像 137
  - vxassist 147
- 备份卷 133
- 标准磁盘设备 56
- 布局
  - 左对称 19
- 不可用卷 207, 208

## C

- 菜单界面 91
- 处理子磁盘 206
- 初始化 59
- 创建
  - 磁盘组 116
  - 跨接卷 124
  - 连续卷 124
  - RAID-5 卷 126
  - 条带卷 125
  - VM 磁盘上的卷 125
- 创建磁盘组
  - vx dg 82
- 创建镜像
  - vxmake 135
- 创建卷
  - vxassist 123
- 创建子磁盘 142
  - vxmake 142
- 磁盘 96
  - 初始化 59
  - 分离 78
  - 封装 35, 49, 89, 93, 103, 175
  - 根磁盘 35, 49, 68, 174, 175, 183
  - 更换 80
  - 故障 71, 202
    - 和热重定位 42, 71
    - 与恢复 173

- 集群中 160
- 禁止 112
- 镜像卷 150
- 启用 113
- 热重定位备用磁盘 74, 108
- 删除 112
- 替换 183
- 添加 59
- 添加供将来使用 105
- VM
  - 创建卷在 125
- VM 磁盘 4
- 物理
  - 保留 67
  - 更换 93, 114
  - 禁止 95
  - 启用 94
  - 删除 69, 93
  - 使脱机 68
  - 受 VxVM 的控制 61
  - 添加 61, 93
  - 添加到磁盘组 63
  - 显示信息 70
  - 移动卷 94
- 显示信息 111
- 移动 71
- 移动卷 151
- 易失性 90
- 引导磁盘 68, 183
- 重新初始化 105
- 重新挂接 186
- 重新添加 183
- 磁盘存取名称 95
- 磁盘故障 202
  - 与恢复 173
- 磁盘介质名称 4, 56, 96
- 磁盘名称 56, 57
- 磁盘信息, 显示 111



- 磁盘组 4, 57
  - 初始化 116
  - 创建 116
  - 导入 85, 86, 119
  - 禁止 118
  - 默认 116
  - 启用 94
  - 删除 84
  - 添加磁盘 116
  - 移动 85, 86
  - 在系统间移动 85
  - 重命名 86
  - 逐出 85, 86, 118
- 磁盘组实用程序 58
- 次要编号
  - 保留 88
- 从 DMP 设备引导
  - DMP 52
- 从节点 155
- 存储布局
  - 转换 38

## D

- devname 95
- Dirty 区日志 45, 130
  - 日志子磁盘 144
  - 原则 33
  - 在集群环境中 161
- DMP
  - 从 DMP 设备引导 52
  - 动态多路径 51
  - 负载平衡 52
  - 路径故障切换机制 52
- DMP 显示 90
- DRL 130
- 大小
  - 卷
    - 减少 128
    - 增加 126
- 导入
  - 磁盘组 85
- 导入磁盘组 94, 119
- 调整
  - Volume Manager 222
- 动态多路径
  - DMP 51
- 读
  - 策略 216

## F

- 分割子磁盘
  - vxsd 146
- 分离镜像
  - vxplex 130, 137
- 分区 3
  - UNIX 95
- 封装 35, 49, 89, 93, 103, 175
- 封装磁盘 93, 103
- 负载平衡
  - DMP 52
- 复制镜像
  - vxplex 142

## G

- 格式化实用程序 59
- 根磁盘 35, 49, 174, 175, 183
  - 镜像 68
- 根卷
  - 引导 50
- 根卷限制 50
- 根目录可置性 49
  - 清除 190
- 跟踪 I/O 220
  - vxtrace 220
- 更改卷属性 210
- 更换磁盘 80, 93, 114
  - vxdiskadm 80
- 公共区域 57
- 共享对象 154
- 故障 178, 205
  - 磁盘 202
  - 和恢复过程 178
  - 系统 202
- 故障磁盘 71
  - 检测 78
- 故障切换
  - DMP 52
- 关联 plex
  - vxmake 136
- 关联镜像
  - vxmake 136
- 关联日志子磁盘 144
  - vxsd 144
- 关联子磁盘
  - vxmake 143
  - vxsd 143

## H

- 后台驻留程序 28





- 热重定位 42
- Volume Manager 28
- vxrelocd 73
- 恢复 173, 209
  - 过程 178
  - 卷 134
  - RAID-5 卷 203, 209
  - 日志 205
- 获取性能数据 219

## J

- I/O
  - 跟踪 220, 222
  - 统计 220
    - 获得 219
- I/O 后台驻留程序 29
- 集群
  - 磁盘 160
  - 共享对象 154
  - 集群功能性 54, 153
  - 集群共享磁盘组 154
  - 集群环境 54, 153
  - 集群协议版本 169
  - 集群协议范围 169
  - 集群重配置 157
  - 加入子磁盘
    - vxsd 146
  - 检查点 205
  - 减少卷的大小 128
  - 降级方式 202
  - 交换卷限制 50
  - 解除重定位实用程序 44
  - 节点 153
  - 禁止
    - 磁盘 95, 112
    - 磁盘组 94, 118
  - 镜像 5, 6, 15, 94, 216, 217
    - 备份 137
    - 创建 135
    - 磁盘上的卷 150
    - 分离 130, 136
    - 根磁盘 68
    - 恢复 79
    - 删除 130, 136
    - 所有卷 129
    - 脱机 139
    - 显示 138
    - 引导磁盘 68
    - 原则 32

- 卷 6
  - 备份 133
  - 布局 199
  - 创建
    - 连续 124
  - 从磁盘移动 151
  - 定义 2, 6
  - 读策略 132
  - 和 plex 6
  - 恢复 134
  - 减少大小 128
  - 镜像 129
  - 镜像所有现有的卷 129
  - 跨接
    - 创建 124
  - 连续, 创建 124
  - 启动 132
  - 启用 131
  - 清除 191
  - RAID-5
    - 创建 126
  - 删除 128
  - 条带
    - 创建 125
    - 停止 131, 132
    - 显示配置 134
    - 在磁盘上镜像 150
    - 增加大小 126
- 卷内核状态 199
  - DETACHED 199
  - DISABLED 199
  - ENABLED 199
- 卷限制
  - 引导时 50
- 卷再同步 44
- 卷重配置 158
- 卷状态 198
  - ACTIVE 198, 199
  - CLEAN 198
  - EMPTY 198, 199
  - SYNC 198, 199

## K

- 可调值 222
- 控制器 95
- 跨接卷
  - 创建 124
- 跨越 10
- 快速镜像再同步 (FMR) 46



快照 147, 148  
RAID-5 147

## L

联机备份 147  
联机重新布局 38  
    故障恢复 41  
    联机重新布局的工作机制 38  
    转换和卷长 42  
    转换类型 39  
    转换特性 41  
连续 10  
连续卷  
    创建 124  
列, 在条带化中 11  
列出镜像  
    vxprint 138

## M

名称  
    磁盘存取 3, 95  
    磁盘介质 4, 96  
    设备 95  
命令行实用程序 58

## N

nopriv 89  
    设备 90

## O

OFFLINE 139

## P

plex 5, 199  
    创建 135  
    定义 5  
    分离 139  
    复制 142  
    更改信息 138  
    挂接 139, 140  
    和卷 6  
    列出 138  
    条带 11  
    显示 138  
    移动 141  
    作为镜像 6  
plex 内核状态 198  
    DETACHED 198  
    DISABLED 198  
    ENABLED 198  
plex 状态 195

ACTIVE 196  
CLEAN 196  
EMPTY 195  
IOFAIL 197  
OFFLINE 197  
STALE 196  
TEMP 197  
TEMPRM 197

plex 状态循环 197  
putil 138  
配置原则 215  
片 95

## Q

启动  
    vxdiskadm 92  
启动卷 207  
奇偶校验 14, 17, 205  
奇偶校验恢复 204, 205  
启用  
    磁盘 94, 113  
    磁盘组 94  
    对磁盘组的存取 119  
强制启动卷 208

## R

RAID-0 11  
RAID-1 15  
RAID-5 199, 200, 202, 204, 205, 206,  
207, 208, 210, 218  
    恢复 203, 209  
    快照 147  
    原则 34  
    子磁盘移动 207  
RAID-5 卷 209  
    创建 126  
RAID-5 plex 199  
RAID-5 日志 130  
resilvering 53  
rootdg 4, 57  
    重命名 86  
热重定位 42, 71, 95, 108  
    删除备用磁盘 108  
    修改 vxrelocd 73  
    引导磁盘 183  
    指定备件 43  
    指定备用磁盘 108  
日志 20, 200, 205  
    RAID-5 130  
    添加 130



日志 plex 199  
日志子磁盘 33, 45, 144, 161  
    关联 144

## S

Storage Administrator 23

删除

    磁盘 93, 112  
    DRL 130  
    卷 128  
    物理磁盘 69

删除磁盘

    vxdg 65

删除磁盘组 84

    vxdg 84

删除子磁盘

    vxedit 142

设备的路径名 96

设备名称 56, 95

使用 I/O 统计 220

使用磁盘组

    vxassist 58

使用特殊设备 89

使用性能数据 220

数据

    保留 35

    冗余 15

数据分配 215

## T

tutil 138

特殊封装

    vxdisk 89

特殊设备

    使用 89

替换磁盘 183

添加

    磁盘 61, 93

    磁盘到磁盘组 63, 116

    磁盘供将来使用 105

    DRL 日志 130

    RAID-5 日志 130

添加磁盘 59

    格式化 59

条带 plex 11

条带单元 12

条带化 11, 216, 217

条带卷

    创建 125

条带列 11

图形用户界面 23, 58

退出

    vxdiskadm 120

脱机

    磁盘 68, 95, 112

## W

VM 磁盘 4

    定义 4

vm 磁盘操作 69

UNIX 分区 95

Volume Manager 23, 37

    布局 22

    根目录可置性 49

    和操作系统 22

    后台驻留程序 28

Volume Manager 图形用户界面 58

Volume Manager 支持操作 56

vxassist 35, 58, 81, 129, 147, 148

    备份 147

    创建卷 123

    growby 127

    growto 127

    shrinkby 127

    shrinkto 127

vxassist addlog 130

vxassist growby 127

vxassist growto 127

vxassist make 67, 125

vxassist snapshot 133

vxassist snapstart 133

vxclust 164

vxconfigd 28, 50, 165

vxctl 169

vxdg 58, 65, 82, 85, 86, 166

    删除磁盘组 84

    移动磁盘组 85

vxdg rmdisk 69

vxdisk 58, 65, 89, 168

    rm 65

    特殊封装 89

vxdisk list 70

vxdisk rm 69

vxdiskadd 56, 58, 59, 61, 70, 81

vxdiskadm 50, 56, 58, 59, 65, 66, 80,

86

    更换磁盘 80

    启动 92

vxedit 137, 138, 142, 145, 190



- 删除子磁盘 142
- vxedit rename 66
- vxedit set 67
- vxinfo 80
- vxiod 28, 29
- vxmake 129, 135, 136, 142, 143
  - 创建镜像 135
  - 创建子磁盘 142
  - 关联镜像 136
  - 关联子磁盘 143
- vxmend 134, 139, 141
- vxmirror 175
- vxplex 129, 130, 136, 137, 139, 140, 141
  - 分离镜像 130, 137
  - 复制镜像 142
  - 移动镜像 141
- vxprint 78, 134, 138, 143
  - 列出镜像 138
  - 显示子磁盘 143
- vxreattach 186
- vxrecover 79, 169
  - 移动磁盘组 85
- vxrelocd 42, 73
  - 修改 73
- vxsd 143, 145, 146
  - 分割子磁盘 146
  - 关联日志子磁盘 144
  - 关联子磁盘 143
  - 加入子磁盘 146
  - 移动子磁盘 146
- VxSmartSync 53
- vxstat 78, 170, 219, 220
- vxtrace 219, 220
- VxVM 23, 37
- vxvol 80, 140

## X

- 系统故障 202
- 显示
  - 磁盘信息 70, 111
  - 卷配置 134
- 显示子磁盘
  - vxprint 143
- 写入
  - 读取 - 修改 211
  - 全条带 212
  - 重构 213
- 信息 145
- 性能 218

- 管理 215
- 监视 218
- 优化 215
- 优先级 218
- 原则 215
- 性能数据 219
  - 获取 219
  - 使用 220

## Y

- 移动
  - 磁盘上的卷 151
  - 磁盘中的卷 94
- 移动磁盘 71
- 移动磁盘组
  - vxdbg 85
  - vxrecover 85
- 移动镜像 141
  - vxplex 141
- 移动子磁盘
  - vxsd 146
- 引导
  - 发生故障后 174
- 引导磁盘
  - 故障 183
  - 故障和热重定位 183
  - 镜像 68
  - 替换 183, 185
  - 重新添加 183, 184
- 引导进程 173
- 原则
  - Dirty 区日志 33
  - 镜像 32
  - 镜像和条带化 33
  - RAID-5 34

## Z

- 再同步
  - 和 Oracle 数据库 53
  - 卷 44
- 增加卷的大小 126
- 重构 - 读取 202
- 重命名磁盘组 86
- 重新安装 187, 188
- 重新初始化磁盘 105
- 重新挂接磁盘 186
- 重新配置过程 187
- 重新添加磁盘 183
- 逐出
  - 磁盘组 85, 94, 118



---

主机  
    多个 153  
主节点 155  
转换  
    存储布局 38  
专用磁盘组 154  
专用区域 57  
状态  
    卷 198  
    plex 195  
子磁盘  
    分割 146  
    分离 145  
    更改信息 145  
    关联 143  
    加入 146  
    日志 45, 144, 161  
    删除 142  
    显示 143  
    移动子磁盘 146  
子磁盘移动  
    RAID-5 207  
自动引导标志 173

