

# **Solaris Volume Manager 管理指南**

版权所有 © 2006, 2013, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

#### U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品和服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

# 目录

---

前言 .....	17
<b>1 Solaris Volume Manager 入门 .....</b>	<b>21</b>
Solaris Volume Manager 任务列表—新增功能 .....	21
Solaris Volume Manager 任务列表—存储容量 .....	22
Solaris Volume Manager 任务列表—可用性 .....	23
Solaris Volume Manager 任务列表—I/O 性能 .....	23
Solaris Volume Manager 任务列表—管理 .....	24
Solaris Volume Manager 任务列表—故障排除 .....	24
<b>2 存储管理概念 .....</b>	<b>25</b>
存储管理介绍 .....	25
存储硬件 .....	25
RAID 级别 .....	26
配置规划原则 .....	26
选择存储 .....	27
一般性能原则 .....	28
随机 I/O 和顺序 I/O 优化 .....	29
随机 I/O .....	29
顺序访问 I/O .....	29
<b>3 Solaris Volume Manager 概述 .....</b>	<b>31</b>
Solaris Volume Manager 的新增功能 .....	31
Solaris Volume Manager 介绍 .....	31
Solaris Volume Manager 如何管理存储 .....	32
如何管理 Solaris Volume Manager .....	32
▼ 如何访问 Solaris Volume Manager 图形用户界面 (Graphical User Interface, GUI) .....	33

Solaris Volume Manager 要求 .....	34
Solaris Volume Manager 组件概述 .....	34
卷概述 .....	35
状态数据库和状态数据库副本 .....	38
热备用池 .....	39
磁盘集 .....	39
Solaris Volume Manager 配置原则 .....	40
一般原则 .....	40
文件系统原则 .....	40
创建 Solaris Volume Manager 组件概述 .....	40
创建 Solaris Volume Manager 组件的先决条件 .....	41
Solaris Volume Manager 中的多 TB 支持概述 .....	41
大型卷支持的限制 .....	41
使用大型卷 .....	42
升级到 Solaris Volume Manager .....	42
<b>4 Solaris Volume Manager for Sun Cluster (概述) .....</b>	<b>45</b>
Solaris Volume Manager for Sun Cluster 介绍 .....	45
先决条件：多属主磁盘卷功能所需的软件组件 .....	46
多属主磁盘集概念 .....	47
与多属主磁盘集关联的任务 .....	48
Solaris Volume Manager for Sun Cluster 配置 .....	49
多属主磁盘集中的 RAID-1 (镜像) 卷 .....	50
多属主磁盘集的镜像所有权 .....	50
数据管理和恢复过程 .....	50
<b>5 配置和使用 Solaris Volume Manager (方案) .....</b>	<b>53</b>
方案背景信息 .....	53
硬件配置 .....	53
初始物理存储配置 .....	54
最终的 Solaris Volume Manager 配置 .....	54
<b>6 状态数据库 (概述) .....</b>	<b>57</b>
关于 Solaris Volume Manager 状态数据库和副本 .....	57

了解多数一致算法 .....	58
管理状态数据库副本 .....	59
处理状态数据库副本错误 .....	60
方案一状态数据库副本 .....	61
<b>7 状态数据库（任务） .....</b>	<b>63</b>
状态数据库副本（任务列表） .....	63
创建状态数据库副本 .....	64
▼如何创建状态数据库副本 .....	64
维护状态数据库副本 .....	66
▼如何检查状态数据库副本的状态 .....	66
▼如何删除状态数据库副本 .....	67
<b>8 RAID-0（条带和串联）卷（概述） .....</b>	<b>69</b>
RAID-0 卷概述 .....	69
RAID-0（条带）卷 .....	70
RAID-0（串联）卷 .....	72
RAID-0（串联条带）卷 .....	73
创建 RAID-0 卷的背景信息 .....	75
RAID-0 卷要求 .....	75
RAID-0 卷的原则 .....	75
方案一 RAID-0 卷 .....	76
<b>9 RAID-0（条带和串联）卷（任务） .....</b>	<b>77</b>
RAID-0 卷（任务列表） .....	77
创建 RAID-0（条带）卷 .....	77
▼如何创建 RAID-0（条带）卷 .....	78
创建 RAID-0（串联）卷 .....	79
▼如何创建 RAID-0（串联）卷 .....	79
扩展存储容量 .....	80
▼如何扩展现有数据的存储容量 .....	80
▼如何扩展现有的 RAID-0 卷 .....	82
删除 RAID-0 卷 .....	83
▼如何删除 RAID-0 卷 .....	83

<b>10 RAID-1 (镜像) 卷 (概述)</b>	85
RAID-1 (镜像) 卷概述	85
子镜像概述	85
方案—RAID-1 (镜像) 卷	86
提供 RAID-1+0 和 RAID-0+1	87
RAID-1 卷 (镜像) 重新同步	88
完全重新同步	88
优化重新同步	88
部分重新同步	88
创建和维护 RAID-1 卷	89
RAID-1 卷的配置原则	89
RAID-1 卷的性能原则	90
关于 RAID-1 卷选项	90
了解子镜像状态以确定维护操作	91
引导至单用户模式对 RAID-1 卷产生的影响	93
方案—RAID-1 卷 (镜像)	93
<b>11 RAID-1 (镜像) 卷 (任务)</b>	95
RAID-1 卷 (任务列表)	95
创建 RAID-1 卷	96
▼ 如何从未使用的分片创建 RAID-1 卷	96
▼ 如何从文件系统创建 RAID-1 卷	98
▼ SPARC: 如何从根 (/) 文件系统创建 RAID-1 卷	103
x86: 从根 (/) 文件系统创建 RAID-1 卷	106
了解镜像根 (/) 文件系统时出现的引导时警告	115
使用子镜像	115
▼ 如何附加子镜像	115
▼ 如何分离子镜像	117
▼ 如何使子镜像脱机或联机	117
▼ 如何启用子镜像中的分片	118
维护 RAID-1 卷	119
▼ 如何查看镜像和子镜像的状态	119
▼ 如何更改 RAID-1 卷选项	121
▼ 如何扩展 RAID-1 卷	122
响应 RAID-1 卷的组件故障	123

▼ 如何替换子镜像中的分片 .....	123
▼ 如何替换子镜像 .....	124
删除 RAID-1 卷（取消镜像） .....	126
▼ 如何取消镜像文件系统 .....	126
▼ 如何取消对无法取消挂载的文件系统的镜像 .....	128
备份 RAID-1 卷上的数据 .....	131
▼ 如何对 RAID-1 卷执行联机备份 .....	131
 12 软分区（概述） .....	135
软分区概述 .....	135
软分区的配置原则 .....	136
方案一软分区 .....	136
 13 软分区（任务） .....	137
软分区（任务列表） .....	137
创建软分区 .....	137
▼ 如何创建软分区 .....	137
维护软分区 .....	139
▼ 如何检查软分区的状态 .....	139
▼ 如何扩展软分区 .....	140
▼ 如何删除软分区 .....	140
 14 RAID-5 卷（概述） .....	143
RAID-5 卷概述 .....	143
示例一 RAID-5 卷 .....	144
示例一 串联（扩展） RAID-5 卷 .....	145
创建 RAID-5 卷的背景信息 .....	146
RAID-5 卷的要求 .....	146
RAID-5 卷的原则 .....	146
检查 RAID-5 卷状态概述 .....	147
在 RAID-5 卷中替换和启用分片概述 .....	149
方案一 RAID-5 卷 .....	149

<b>15 RAID-5 卷（任务）</b>	151
RAID-5 卷（任务列表）	151
创建 RAID-5 卷	152
▼ 如何创建 RAID-5 卷	152
维护 RAID-5 卷	153
▼ 如何检查 RAID-5 卷的状态	153
▼ 如何扩展 RAID-5 卷	154
▼ 如何在 RAID-5 卷中启用组件	155
▼ 如何在 RAID-5 卷中替换组件	156
<b>16 热备用池（概述）</b>	159
热备件和热备用池概述	159
热备件	160
热备用池	160
热备件的工作原理	160
热备用池状态	161
示例－热备用池	161
方案－热备件	162
<b>17 热备用池（任务）</b>	163
热备用池（任务列表）	163
创建热备用池	164
▼ 如何创建热备用池	164
▼ 如何向热备用池中添加更多分片	165
将热备用池与卷关联	166
▼ 如何将热备用池与卷关联	166
▼ 如何更改关联的热备用池	167
维护热备用池	168
▼ 如何检查热备件和热备用池的状态	168
▼ 如何替换热备用池中的热备件	169
▼ 如何从热备用池中删除热备件	170
▼ 如何启用热备件	171

<b>18 磁盘集（概述）</b> .....	173
磁盘集方面的新增功能 .....	173
磁盘集介绍 .....	173
磁盘集类型 .....	174
本地磁盘集 .....	174
已命名磁盘集 .....	174
Solaris Volume Manager 磁盘集管理 .....	176
保留磁盘集 .....	176
释放磁盘集 .....	177
导入磁盘集 .....	177
自动磁盘分区 .....	178
磁盘集名称要求 .....	180
示例—两个共享磁盘集 .....	180
磁盘集使用原则 .....	181
磁盘集中的异步共享存储 .....	182
方案—磁盘集 .....	182
<b>19 磁盘集（任务）</b> .....	183
磁盘集（任务列表） .....	183
创建磁盘集 .....	184
▼ 如何创建磁盘集 .....	184
扩展磁盘集 .....	185
▼ 如何向磁盘集添加磁盘 .....	185
▼ 如何向磁盘集添加其他主机 .....	187
▼ 如何在磁盘集中创建 Solaris Volume Manager 组件 .....	188
维护磁盘集 .....	189
▼ 如何检查磁盘集的状态 .....	189
▼ 如何从磁盘集中删除磁盘 .....	190
▼ 如何获取磁盘集 .....	190
▼ 如何释放磁盘集 .....	192
▼ 如何删除主机或磁盘集 .....	193
导入磁盘集 .....	195
▼ 如何打印有关可导入的磁盘集的报告 .....	195
▼ 如何将磁盘集从一个系统导入到另一个系统 .....	196

<b>20 维护 Solaris Volume Manager (任务)</b>	197
Solaris Volume Manager 维护 (任务列表)	197
查看 Solaris Volume Manager 配置	198
▼ 如何查看 Solaris Volume Manager 卷配置	198
下一步执行的操作	201
重命名卷	201
重命名卷的背景信息	201
交换卷名称	202
▼ 如何重命名卷	203
使用配置文件	204
▼ 如何创建配置文件	204
▼ 如何根据配置文件初始化 Solaris Volume Manager	204
更改 Solaris Volume Manager 缺省值	206
使用 growfs 命令扩展文件系统	206
有关扩展分片和卷的背景信息	207
▼ 如何扩展文件系统	207
替换和启用 RAID-1 和 RAID-5 卷中的组件概述	208
启用组件	208
将一个组件替换为另一个可用组件	209
"Maintenance" (维护) 和 "Last Erred" (最近出错) 状态	210
替换和启用 RAID-1 和 RAID-5 卷中组件的背景信息	211
<b>21 Solaris Volume Manager 的最佳做法</b>	213
部署小型服务器	213
将 Solaris Volume Manager 与网络存储设备结合使用	215
<b>22 自上而下创建卷 (概述)</b>	217
自上而下创建卷概述	217
使用磁盘集实现自上而下创建卷	218
自上而下的卷创建过程	218
确定可用于自上而下创建卷的磁盘	220
<b>23 自上而下创建卷 (任务)</b>	221
自上而下创建卷 (任务列表)	221

自上而下创建卷的先决条件 .....	222
自动创建卷 .....	222
通过指定输出的详细级别来分析卷的创建过程 .....	223
▼ 如何使用 metassist 命令创建 RAID-1（镜像）卷 .....	223
使用 metassist 命令处理基于文件的数据 .....	226
使用 metassist 命令创建命令文件（shell 脚本） .....	226
▼ 如何使用 metassist 命令创建命令文件（shell 脚本） .....	226
使用保存的 shell 脚本（由 metassist 命令创建）创建卷 .....	229
使用 metassist 命令创建卷配置文件 .....	230
▼ 如何使用 metassist 命令创建卷配置文件 .....	230
更改 metassist 命令的缺省行为 .....	232
更改卷缺省文件 .....	232
<b>24 监视和错误报告（任务） .....</b>	<b>235</b>
Solaris Volume Manager 监视与报告（任务列表） .....	235
配置 mdmonitord 命令以定期检查错误 .....	236
▼ 如何配置 mdmonitord 命令以定期检查错误 .....	236
Solaris Volume Manager SNMP 代理概述 .....	237
配置 Solaris Volume Manager SNMP 代理 .....	237
▼ 如何配置 Solaris Volume Manager SNMP 代理 .....	237
Solaris Volume Manager SNMP 代理的限制 .....	239
使用 cron 作业监视 Solaris Volume Manager .....	240
▼ 如何自动检查卷中的错误 .....	240
<b>25 Solaris Volume Manager 故障排除（任务） .....</b>	<b>249</b>
Solaris Volume Manager 故障排除（任务列表） .....	249
系统故障排除概览 .....	250
系统故障排除先决条件 .....	250
Solaris Volume Manager 故障排除的一般原则 .....	250
一般故障排除方法 .....	251
替换磁盘 .....	251
▼ 如何替换故障磁盘 .....	252
从磁盘移动问题中恢复 .....	253
磁盘移动和设备 ID 概述 .....	254
解决未命名设备错误消息 .....	254

升级到 Solaris 10 发行版之后设备 ID 的改变 .....	255
从引导问题中恢复 .....	256
引导问题的背景信息 .....	257
如何从不正确的 <code>/etc/vfstab</code> 项恢复 .....	257
▼ 恢复根 ( <code>/</code> ) RAID-1 (镜像) 卷 .....	257
▼ 如何从引导设备故障中恢复 .....	259
从状态数据库副本故障中恢复 .....	263
▼ 如何从不足的状态数据库副本中恢复 .....	263
从软分区问题中恢复 .....	265
▼ 如何恢复软分区的配置数据 .....	265
从另一个系统恢复存储 .....	267
▼ 如何从本地磁盘集恢复存储 .....	267
从已知磁盘集恢复存储 .....	271
从磁盘集问题中恢复 .....	273
无法获取磁盘集所有权时应执行的操作 .....	273
使用 <code>ufsdump</code> 命令对已挂载的文件系统执行备份 .....	275
▼ 如何对 RAID-1 卷上的已挂载文件系统执行备份 .....	275
执行系统恢复 .....	276
▼ 如何使用 Solaris Volume Manager 配置来恢复系统 .....	276
 A <b>重要的 Solaris Volume Manager 文件</b> .....	279
系统文件和启动文件 .....	279
手动配置的文件 .....	280
<code>md.tab</code> 文件概述 .....	280
 B <b>Solaris Volume Manager 快速参考</b> .....	283
命令行参考 .....	283
 C <b>Solaris Volume Manager CIM/WBEM API</b> .....	285
管理 Solaris Volume Manager .....	285
 索引 .....	287



图 3-1	Solaris Management Console 中 "Enhanced Storage" (增强的存储) 工具 (Solaris Volume Manager) 视图 .....	33
图 3-2	卷、物理磁盘和分片之间的关系 .....	37
图 4-1	群集配置样例 .....	46
图 5-1	基本硬件存储方案图 .....	54
图 8-1	RAID-0 (条带) 卷示例 .....	71
图 8-2	RAID-0 (串联) 卷示例 .....	73
图 8-3	RAID-0 (串联条带) 卷示例 .....	74
图 10-1	RAID-1 (镜像) 示例 .....	86
图 10-2	RAID-1+0 示例 .....	87
图 14-1	RAID-5 卷示例 .....	144
图 14-2	扩展的 RAID-5 卷示例 .....	145
图 16-1	热备用池示例 .....	162
图 18-1	磁盘集示例 .....	181
图 21-1	小型系统配置 .....	214
图 22-1	自上而下创建卷的处理选项 .....	219



# 表

---

表 2-1	存储类型比较 .....	27
表 2-2	优化冗余存储 .....	27
表 3-1	Solaris Volume Manager 功能摘要 .....	34
表 3-2	卷的类别 .....	35
表 10-1	RAID-1 卷读取策略 .....	91
表 10-2	RAID-1 卷写入策略 .....	91
表 10-3	子镜像状态 .....	92
表 10-4	子镜像分片状态 .....	92
表 14-1	RAID-5 卷状态 .....	147
表 14-2	RAID-5 分片状态 .....	148
表 16-1	热备用池状态（命令行） .....	161
表 18-1	磁盘集卷名示例 .....	180
表 25-1	Solaris Volume Manager 的常见引导问题 .....	257
表 B-1	Solaris Volume Manager 命令 .....	283



# 前言

---

《Solaris Volume Manager 管理指南》介绍如何使用 Solaris Volume Manager 来管理系统的存储需求。使用 Solaris Volume Manager，可以创建、修改和使用 RAID-0（串联和条带）卷、RAID-1（镜像）卷、RAID-5 卷以及软分区。

---

注 – 此 Oracle Solaris 发行版支持使用 SPARC 和 x86 系列处理器体系结构的系统。支持的系统可以在《Oracle Solaris OS: Hardware Compatibility Lists》（《Oracle Solaris OS：硬件兼容性列表》）中找到。本文档列举了在不同类型的平台上进行实现时的所有差别。

在本文档中，这些与 x86 相关的术语表示以下含义：

- x86 泛指 64 位和 32 位的 x86 兼容产品系列。
- x64 特指 64 位的 x86 兼容 CPU。
- “32 位 x86”指出了有关基于 x86 的系统的特定 32 位信息。

有关支持的系统，请参见《[Oracle Solaris OS: Hardware Compatibility Lists](#)》（《Oracle Solaris OS：硬件兼容性列表》）。

---

## 目标读者

系统和存储管理员可以使用本书了解以下内容：

- Solaris Volume Manager 所支持的任务
- 使用 Solaris Volume Manager 提供更可靠且更易访问数据的方法

## 本书的结构

《Solaris Volume Manager 管理指南》包含以下信息：

[第 1 章，Solaris Volume Manager 入门](#)提供了有关本书中介绍的概念和任务的详细“任务列表”。可使用本章作为本书内容的导航帮助。

[第 2 章，存储管理概念](#)向初次接触该技术的读者介绍一般存储管理概念。

[第 3 章，Solaris Volume Manager 概述](#)介绍 Solaris Volume Manager。本章介绍与产品相关的基本概念并说明如何访问 Solaris Volume Manager 工具。

第 4 章，[Solaris Volume Manager for Sun Cluster（概述）](#)介绍多属主磁盘集。多属主磁盘集增强了 Solaris Volume Manager 在 Sun Cluster 环境中的用途。

第 5 章，[配置和使用 Solaris Volume Manager（方案）](#)提供了本书中所使用的存储配置方案。此方案可用于帮助您了解 Solaris Volume Manager 产品。

第 6 章，[状态数据库（概述）](#)介绍与状态数据库和状态数据库副本相关的概念。

第 7 章，[状态数据库（任务）](#)说明如何执行与状态数据库和状态数据库副本相关的任务。

第 8 章，[RAID-0（条带和串联）卷（概述）](#)介绍与 RAID-0（条带和串联）卷相关的概念。

第 9 章，[RAID-0（条带和串联）卷（任务）](#)说明如何执行与 RAID-0（条带和串联）卷相关的任务。

第 10 章，[RAID-1（镜像）卷（概述）](#)介绍与 RAID-1（镜像）卷相关的概念。

第 11 章，[RAID-1（镜像）卷（任务）](#)说明如何执行与 RAID-1（镜像）卷相关的任务。

第 12 章，[软分区（概述）](#)介绍与 Solaris Volume Manager 的软分区功能相关的概念。

第 13 章，[软分区（任务）](#)说明如何执行与软分区相关的任务。

第 14 章，[RAID-5 卷（概述）](#)介绍与 RAID-5 卷相关的概念。

第 15 章，[RAID-5 卷（任务）](#)说明如何执行与 RAID-5 卷相关的任务。

第 16 章，[热备用池（概述）](#)介绍与热备件和热备用池相关的概念。

第 17 章，[热备用池（任务）](#)说明如何执行与热备件和热备用池相关的任务。

第 18 章，[磁盘集（概述）](#)介绍与磁盘集相关的概念。

第 19 章，[磁盘集（任务）](#)说明如何执行与磁盘集相关的任务。

第 20 章，[维护 Solaris Volume Manager（任务）](#)说明一些与特定 Solaris Volume Manager 组件无关的一般维护任务。

第 21 章，[Solaris Volume Manager 的最佳做法](#)提供了有关配置和使用 Solaris Volume Manager 的一些“最佳做法”信息。

第 23 章，[自上而下创建卷（任务）](#)介绍与 Solaris Volume Manager 自上而下卷创建功能相关的概念和任务。

第 24 章，[监视和错误报告（任务）](#)提供了有关使用 Solaris Volume Manager SNMP 代理和其他错误检查方法的概念及说明。

- 第 25 章，[Solaris Volume Manager 故障排除（任务）](#) 提供了有关在 Solaris Volume Manager 环境中排除故障及解决常见问题的信息。
- 附录 A，[重要的 Solaris Volume Manager 文件](#) 列出了重要的 Solaris Volume Manager 文件。
- 附录 B，[Solaris Volume Manager 快速参考](#) 提供了汇总命令和其他有用信息的表。
- 附录 C，[Solaris Volume Manager CIM/WBEM API](#) 简要介绍了 CIM/WBEM API，利用该 API，可通过符合 WBEM 标准的管理工具对 Solaris Volume Manager 进行开放式管理。

## 相关书籍

Solaris Volume Manager 是可用于 Solaris 操作系统的若干系统管理工具之一。以下书籍提供了有关总体系统管理特征和功能以及相关工具的信息：

- [《Oracle Solaris 管理：基本管理》](#)
- [《系统管理指南：高级管理》](#)
- [《System Administration Guide: Devices and File Systems》](#)

## 获取 Oracle 支持

Oracle 客户可以通过 My Oracle Support 获取电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>，或访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>（如果您听力受损）。

## 印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

字体或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>machine_name% you have mail.</code>
<b>AaBbCc123</b>	用户键入的内容，与计算机屏幕输出的显示不同	<code>machine_name% su</code> Password:
<i>aabbcc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <code>rm filename</code> 。

表 P-1 印刷约定 (续)

字体或符号	含义	示例
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词	这些称为 <i>Class</i> 选项。 <b>注意：</b> 有些强调的项目在联机时以粗体显示。
<b>新词术语强调</b>	新词或术语以及要强调的词	<b>高速缓存</b> 是存储在本地的副本。 请勿保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

## 命令中的 shell 提示符示例

下表显示了 Oracle Solaris OS 中包含的缺省 UNIX shell 系统提示符和超级用户提示符。请注意，在命令示例中显示的缺省系统提示符可能会有所不同，具体取决于 Oracle Solaris 发行版。

表 P-2 shell 提示符

shell	提示符
Bash shell、Korn shell 和 Bourne shell	\$
Bash shell、Korn shell 和 Bourne shell 超级用户	#
C shell	machine_name%
C shell 超级用户	machine_name#

# Solaris Volume Manager 入门

《Solaris Volume Manager 管理指南》介绍如何设置和维护使用 Solaris Volume Manager 管理存储的系统，以实现高可用性、灵活性和可靠性。

本章可用作查找某些 Solaris Volume Manager 任务（如设置存储容量）信息的概括性指南，其中并没有说明使用 Solaris Volume Manager 需要执行的所有任务，而是概述了新增的功能，提供了查找与 Solaris Volume Manager 概念相关的常见任务的说明过程的简单方法。

本章包含以下任务列表：

- 第 21 页中的“Solaris Volume Manager 任务列表—新增功能”
- 第 22 页中的“Solaris Volume Manager 任务列表—存储容量”
- 第 23 页中的“Solaris Volume Manager 任务列表—可用性”
- 第 23 页中的“Solaris Volume Manager 任务列表—I/O 性能”
- 第 24 页中的“Solaris Volume Manager 任务列表—管理”
- 第 24 页中的“Solaris Volume Manager 任务列表—故障排除”



**注意** – 如果没有正确地使用 Solaris Volume Manager，则可能会销毁数据。Solaris Volume Manager 提供了一种可靠管理磁盘和磁盘上的数据的有效方法。但是，您应当始终维护数据的备份，特别是在修改活动的 Solaris Volume Manager 配置之前。

## Solaris Volume Manager 任务列表—新增功能

任务	说明	参考
管理存储，其中一个或多个组件大于 1 TB	使用大小大于 1 TB 的物理逻辑单元号 (Logical Unit Number, LUN) 或创建大于 1 TB 的逻辑卷。	第 41 页中的“Solaris Volume Manager 中的多 TB 支持概述”

任务	说明	参考
将磁盘集从一个系统导入另一个系统	使用 <code>metainport</code> 命令导入磁盘集，包括在不同系统上创建的磁盘集。此命令使用扩展的设备 ID 支持，自动跟踪磁盘在已命名的磁盘集中的移动。	第 177 页中的“导入磁盘集” 第 182 页中的“磁盘集中的异步共享存储”
创建和管理多属主磁盘集	使用 <code>metaset -M</code> 可管理 Sun Cluster 环境中的多属主磁盘集。	第 48 页中的“与多属主磁盘集关联的任务”

## Solaris Volume Manager 任务列表－存储容量

任务	说明	参考
设置存储	通过创建 RAID-0 或 RAID-5 卷来创建跨多个分片的存储。然后，可将 RAID-0 或 RAID-5 卷用于文件系统或任何访问原始设备的应用程序，如数据库。	第 78 页中的“如何创建 RAID-0（条带）卷” 第 79 页中的“如何创建 RAID-0（串联）卷” 第 96 页中的“如何从未使用的分片创建 RAID-1 卷” 第 98 页中的“如何从文件系统创建 RAID-1 卷” 第 152 页中的“如何创建 RAID-5 卷”
扩展现有文件系统	通过创建 RAID-0（串联）卷，然后向该卷中添加其他分片，增加现有文件系统的容量。	第 80 页中的“如何扩展现有数据的存储容量”
扩展现有的 RAID-0（串联或条带）卷	通过串联其他分片扩展现有的 RAID-0 卷。	第 82 页中的“如何扩展现有的 RAID-0 卷”
扩展 RAID-5 卷	通过串联其他分片扩展 RAID-5 卷的容量。	第 154 页中的“如何扩展 RAID-5 卷”
增加扩展的卷上 UFS 文件系统的大小	使用 <code>growfs</code> 命令在不中断数据访问的情况下扩展已挂载的 UFS 大小，从而扩展文件系统。	第 207 页中的“如何扩展文件系统”
将分片或逻辑卷进一步细分为更小的分区，突破 8 个分片的硬分区限制	通过使用软分区细分逻辑卷或分片。	第 137 页中的“如何创建软分区”

任务	说明	参考
创建文件系统	在 RAID-0（条带或串联）、RAID-1（镜像）、RAID-5 或软分区上创建文件系统。	《System Administration Guide: Devices and File Systems》中的第 15 章“Creating and Mounting File Systems (Tasks)”

## Solaris Volume Manager 任务列表 – 可用性

任务	说明	参考
最大程度地提高数据可用性	使用 Solaris Volume Manager 的镜像功能维护数据的多个副本。可以从为数据准备的、尚未使用的分片创建 RAID-1 卷，或者可以镜像现有的文件系统，包括根 (/) 和 /usr。	第 96 页中的“如何从未使用的分片创建 RAID-1 卷” 第 98 页中的“如何从文件系统创建 RAID-1 卷”
以最低硬件成本增加数据可用性	通过使用 Solaris Volume Manager 的 RAID-5 卷，以最少的硬件来提高数据可用性。	第 152 页中的“如何创建 RAID-5 卷”
提高现有的 RAID-1 或 RAID-5 卷的数据可用性	通过创建热备用池，然后将其与 RAID-1 卷或 RAID-5 卷的子镜像相关联，来提高 RAID-1 或 RAID-5 卷的数据可用性。	第 164 页中的“创建热备用池” 第 166 页中的“将热备用池与卷关联”

## Solaris Volume Manager 任务列表 – I/O 性能

任务	说明	参考
调优 RAID-1 卷读取和写入策略	指定 RAID-1 卷的读取和写入策略以提高给定配置的 I/O 性能。	第 91 页中的“RAID-1 卷读写策略” 第 121 页中的“如何更改 RAID-1 卷选项”
优化设备性能	创建 RAID-0（条带）卷以优化组成该条带的设备的 I/O 性能。可以优化交错值以进行随机或顺序访问。	第 77 页中的“创建 RAID-0（条带）卷”
维护 RAID-0（条带）内的设备性能	通过串联一个新组件来扩展空间已经不足的条带或串联。条带串联的 I/O 性能优于分片串联。	第 80 页中的“扩展存储容量”

# Solaris Volume Manager 任务列表－管理

任务	说明	参考
以图形方式管理卷管理配置	使用 Solaris Management Console 图形用户界面 (Graphical User Interface, GUI) 管理卷管理配置。	Solaris Management Console 应用程序的 Solaris Volume Manager（增强的存储）节点内的联机帮助
以图形方式管理分片和文件系统	使用 Solaris Management Console GUI 管理磁盘和文件系统，执行诸如对磁盘进行分区、构建 UFS 文件系统等此类任务。	Solaris Management Console 应用程序内的联机帮助
优化 Solaris Volume Manager	Solaris Volume Manager 性能取决于良好设计的配置。一旦创建，该配置需要监视和调优。	第 40 页中的“Solaris Volume Manager 配置原则” 第 204 页中的“使用配置文件”
计划未来扩展	由于文件系统空间即将用完，可以通过将文件系统置于串联卷中来应对未来增长。	第 79 页中的“创建 RAID-0（串联）卷” 第 80 页中的“扩展存储容量”

# Solaris Volume Manager 任务列表－故障排除

任务	说明	参考
替换有故障的分片	如果磁盘出现故障，必须替换 Solaris Volume Manager 配置中使用的分片。如果是 RAID-0 卷，必须使用新的分片，并删除和重新创建卷，然后从备份中恢复数据。可以在不丢失数据的情况下替换并重新同步 RAID-1 和 RAID-5 卷中的分片。	第 123 页中的“响应 RAID-1 卷的组件故障” 第 156 页中的“如何在 RAID-5 卷中替换组件”
从引导问题中恢复	由于硬件问题或操作错误，引导系统时可能会产生特殊问题。	第 257 页中的“如何从不正确的 /etc/vfstab 项恢复” 第 263 页中的“如何从不足的状态数据库副本中恢复” 第 259 页中的“如何从引导设备故障中恢复”

## 存储管理概念

---

本章简要介绍了一些常用存储管理概念。

本章包含以下信息：

- 第 25 页中的“存储管理介绍”
- 第 26 页中的“配置规划原则”
- 第 28 页中的“一般性能原则”
- 第 29 页中的“随机 I/O 和顺序 I/O 优化”

### 存储管理介绍

选择管理存储的方式决定了控制系统上用于存储活动数据的设备的方式。要使活动的数据有用，则这些数据必须可用，且始终保持不变，即使是在出现意外事件（如硬件或软件故障）后也是如此。

### 存储硬件

在许多种不同的设备上都可以存储数据。选择最能满足存储需求的设备主要取决于三个因素：

- 性能
- 可用性
- 成本

使用 Solaris Volume Manager 有助于管理性能、可用性和成本三者的权衡因素。通常，使用 Solaris Volume Manager 可以减少许多权衡因素。

Solaris Volume Manager 可以与运行 Solaris 操作系统的任何系统上的任何受支持存储设备很好地协作。

## RAID 级别

RAID 是廉价/独立磁盘冗余阵列 (Redundant Array of Inexpensive/Independent Disks) 的首字母缩略词。RAID 指一组称为阵列或卷的磁盘，作为单个的大型磁盘驱动器呈现给用户。根据配置，该阵列可以提供改进的可靠性、响应时间或存储容量。

从技术上讲，有六个 RAID 级别（0-5）。每个级别指一种分布数据并同时确保数据冗余的方法。（RAID 级别 0 不提供数据冗余，但通常仍然作为 RAID 等级包括在内。RAID 级别 0 为所使用的大多数 RAID 配置提供基础。）支持 RAID 级别 2、3 和 4 的存储环境非常少，所以此处不再介绍。

Solaris Volume Manager 支持以下 RAID 级别：

- **RAID 级别 0**—尽管串联和条带不提供冗余，但这些卷通常称为 RAID-0。实质上，数据分布在相对较小且大小相等的段中，这些段交替且均匀地分配在多个物理磁盘上。任何单个驱动器故障都会导致数据丢失。RAID-0 提供了较高的数据传输速率和 I/O 吞吐量，但与单个磁盘相比，可靠性和可用性都较低。
- **RAID 级别 1**—镜像使用同等数量的磁盘容量来存储数据和数据副本（镜像）。数据在两个或更多物理磁盘上进行复制或镜像。可以同时从两个驱动器读取数据，这意味着任一个驱动器都可以为任何请求提供服务，从而提高了性能。如果一个物理磁盘出现故障，可以继续使用镜像，从而不会造成性能降低或数据丢失。

Solaris Volume Manager 支持 RAID-0+1 镜像，并透明地支持 RAID-1+0 镜像（取决于底层卷）。有关详细信息，请参见第 87 页中的“提供 RAID-1+0 和 RAID-0+1”。

- **RAID 级别 5**—RAID-5 使用条带化将数据分布于阵列中的磁盘上。RAID-5 还记录奇偶校验信息，以提供一定程度的数据冗余性。RAID-5 卷可以承受基础设备的故障而继续运行。如果 RAID-5 卷与热备件结合使用，可以承受多个故障而仍然继续运行。但是，若在运行时有设备发生故障，RAID-5 卷的性能会显著下降。

在 RAID-5 模型中，每个设备都有一个包含奇偶校验条带的区域和其他包含数据的区域。奇偶校验信息分布于阵列中的所有磁盘上，这样可以缩短写入时间。因为写入操作不必等到专用的奇偶校验磁盘可以接受数据后再进行，所以缩短了写入时间。

## 配置规划原则

规划存储管理配置时，请牢记，任何给定的配置都需要在性能、可用性和硬件成本这几个方面进行权衡。您可能需要试验不同的变量来确定最适合您配置的变量。

本部分提供有关使用以下卷类型的原则：

- RAID-0（串联和条带）卷
- RAID-1（镜像）卷
- RAID-5 卷
- 软分区

- 在 Solaris Volume Manager 卷上构造的文件系统

## 选择存储

实施存储管理方法之前，需要确定要使用何种类型的存储设备。此组原则可比较不同的存储类型，帮助您做出选择。其他组原则适用于在 Solaris Volume Manager 中实现的特定存储类型。有关详细信息，请参见关于每个卷类型的特定章节。

注- 此处列出的存储类型并不互相排斥。可以结合使用这些卷以满足多个目标。例如，可以首先创建一个 RAID-1 卷来实现冗余性。接下来，可以在该 RAID-1 卷上创建软分区，以增加独立文件系统的可能数量。

下表比较了每种存储类型可用的功能。

表 2-1 存储类型比较

要求	RAID-0（串联）	RAID-0（条带）	RAID-1（镜像）	RAID-5	软分区
冗余数据	否	否	是	是	否
提高了读取性能	否	是	取决于基础设备	是	否
提高了写入性能	否	是	否	否	否
每个设备超过 8 个分片	否	否	否	否	是
更大的可用存储空间	是	是	否	是	否

下表概述了 RAID-1 和 RAID-5 卷之间在写入操作、随机读取和硬件成本三个方面的权衡。

表 2-2 优化冗余存储

	RAID-1（镜像）	RAID-5
写入操作	较快	较慢
随机读取	较快	较慢
硬件成本	较高	较低

以下列表总结了表中概述的信息：

- RAID-0 卷（条带和串联）和软分区不提供任何数据冗余。
- 串联适用于小型随机 I/O 操作。
- 条带化适用于大型顺序 I/O 操作和随机 I/O 操作。
- 镜像可能会提高读取性能，但始终会降低镜像中的写入性能。
- 由于 RAID-5 卷具有读取-修改-写入特性，写入量超过 20% 的卷不应该是 RAID-5。如果需要冗余，请考虑采用镜像。
- RAID-5 写入不如镜像写入的速度快，而镜像写入又不如未保护写入的速度快。
- 软分区对于管理非常大的存储设备很有用。

---

注-除这些通用存储选项外，有关使用 Solaris Volume Manager 支持冗余设备的更多信息，请参见第 39 页中的“热备用池”。

---

## 一般性能原则

设计存储配置时，请考虑以下性能原则：

- 条带化通常性能最佳，但不提供数据冗余性。对于写入密集型应用程序，RAID-1 卷的性能通常比 RAID-5 卷好。
- RAID-1 卷和 RAID-5 卷都可以提高数据可用性，但一般情况下，这两种卷的写入操作性能都比较低。镜像可以提高随机读取性能。
- RAID-5 卷的硬件成本比 RAID-1 卷低，而 RAID-0 卷没有其他硬件成本。
- 条带和 RAID-5 卷均可以跨多个磁盘驱动器分布数据并帮助平衡 I/O 负载。
- 确定最常访问的数据，并使用镜像或条带化来增加这些数据的访问带宽。
- 使用可用的性能监视功能和通用工具（如 `iostat` 命令）确定最常访问的数据。确定之后，即可使用条带化、RAID-1 卷或 RAID-5 卷增加这些数据的访问带宽。
- 多次更改软分区大小后，软分区的性能可能会降低。
- RAID-5 卷的写入操作性能低于条带。该性能损失是由于需要执行多次 I/O 操作来计算和存储 RAID-5 卷的奇偶校验信息。
- 对于原始随机 I/O 读取，条带和 RAID-5 卷相当。条带和 RAID-5 卷都可以将数据分布在多个磁盘上。RAID-5 卷在读取时不会进行奇偶校验计算，除非是在分片出现故障之后。
- 对于原始随机 I/O 写入，条带优于 RAID-5 卷。

有关特定于 Solaris Volume Manager 的配置原则，请参见第 40 页中的“Solaris Volume Manager 配置原则”。

# 随机 I/O 和顺序 I/O 优化

本节说明用于优化配置的策略。

如果不清楚在将要创建的 Solaris Volume Manager 卷上是以顺序 I/O 还是随机 I/O 为主，请不要实施这些性能调优提示。如果不当地实施这些提示，可能会降低性能。

以下优化建议假定您要优化 RAID-0 卷。通常，您需要优化 RAID-0 卷，然后镜像该卷以提供最佳的性能和数据冗余性。

## 随机 I/O

在随机 I/O 环境（如用于数据库和通用文件服务器的环境）中，所有磁盘都应花费相同的时间为 I/O 请求提供服务。

例如，假定有 40 GB 的存储用于数据库应用程序。如果跨四个 10 GB 磁盘轴进行条带化，且 I/O 是随机、均匀地分散于卷上，则每个磁盘都会同等繁忙，这样通常可提高性能。

磁盘上最大随机 I/O 性能的目标是 35% 或更低的使用率，如 `iostat` 命令所报告。通常，磁盘使用率超过 65% 就会出现严重问题。磁盘使用率超过 90% 就会出现严重问题。解决磁盘使用率值太高的办法就是创建包含更多磁盘（轴）的新 RAID-0 卷。

---

注 - 只是向现有卷附加更多磁盘并不能提高性能。必须创建具有理想参数的新卷才能优化性能。

---

条带的交错大小无关紧要，因为您只是想将数据分布于所有磁盘上。任何大于一般 I/O 请求的交错值都可满足要求。

## 顺序访问 I/O

可以在顺序 I/O 环境（比如以全表扫描为主的 DBMS 服务器，以及数据非常密集的环境中的 NFS 服务器）中优化配置性能。要充分利用顺序 I/O 环境，请相对于一般 I/O 请求的大小将交错值设置为较小的值。

例如，假定一般 I/O 请求的大小为 256 KB，并跨 4 个轴进行条带化。此示例中条带单元大小的较好选择将是：

$256 \text{ KB} / 4 = 64 \text{ KB}$ ，或更小

此策略可确保一般 I/O 请求跨多个磁盘轴分布，从而增加了连续带宽。

---

注 - 在顺序 I/O 环境中，查找时间和旋转时间实际上都为零。优化顺序 I/O 时，磁盘的内部传输速率最为重要。

---

在顺序的应用程序中，一般 I/O 大小通常很大，超过 128 KB 或甚至超过 1 MB。假定应用程序的一般 I/O 请求大小为 256 KB，并假定跨 4 个磁盘轴进行条带化，因而：

$$256 \text{ KB} / 4 = 64 \text{ KB}$$

因此，交错大小的较好选择将是 32-64 KB。

## Solaris Volume Manager 概述

---

本章介绍了 Solaris Volume Manager 的总体结构。本章包含以下信息：

- 第 31 页中的“Solaris Volume Manager 的新增功能”
- 第 31 页中的“Solaris Volume Manager 介绍”
- 第 34 页中的“Solaris Volume Manager 要求”
- 第 34 页中的“Solaris Volume Manager 组件概述”
- 第 40 页中的“Solaris Volume Manager 配置原则”
- 第 40 页中的“创建 Solaris Volume Manager 组件概述”
- 第 41 页中的“Solaris Volume Manager 中的多 TB 支持概述”
- 第 42 页中的“升级到 Solaris Volume Manager”

### Solaris Volume Manager 的新增功能

本节介绍了此 Solaris 发行版中在使用 Solaris Volume Manager 方面的新增功能。

有关 Solaris 新增功能的完整列表和 Solaris 发行版的说明，请参见《[Oracle Solaris 10 1/13 新增功能](#)》。

### Solaris Volume Manager 介绍

Solaris Volume Manager 是用于管理大量磁盘及磁盘上数据的软件产品。尽管 Solaris Volume Manager 有很多种用途，但主要任务包括：

- 增加存储容量
- 提高数据可用性
- 简化大型存储设备的管理

在某些情况下，Solaris Volume Manager 还可以提高 I/O 性能。

有关 Solaris 操作系统所支持的磁盘类型的信息，请参见《[System Administration Guide: Devices and File Systems](#)》中的第 7 章“Managing Disks (Overview)”。

## Solaris Volume Manager 如何管理存储

Solaris Volume Manager 使用虚拟磁盘管理物理磁盘及其关联数据。在 Solaris Volume Manager 中，虚拟磁盘称作**卷**。由于历史原因，一些命令行实用程序也将卷称为**元设备**。

从应用程序或文件系统的角度来看，卷在功能上等同于物理磁盘。Solaris Volume Manager 可将定向到卷的 I/O 请求转换成定向到底层成员磁盘的 I/O 请求。

Solaris Volume Manager 卷是由磁盘分片或其他 Solaris Volume Manager 卷构建的。构建卷的一种简单方法是使用 Solaris Management Console 中内置的图形用户界面 (Graphical User Interface, GUI)。Solaris Management Console 中增强的存储工具可提供所有现有卷的视图。按照向导中的步骤，可轻松构建各种 Solaris Volume Manager 卷或组件。您还可以使用 Solaris Volume Manager 命令行实用程序构建和修改卷。

例如，如果需要更多存储容量作为单个卷，则可以使用 Solaris Volume Manager 使系统将分片集合视为一个较大的卷。从这些分片创建卷后，可以立即开始使用此卷，就像使用任何真正的分片或设备一样。

有关卷的更详细论述，请参见第 35 页中的“[卷概述](#)”。

Solaris Volume Manager 通过使用 RAID-1（镜像）卷和 RAID-5 卷可以提高数据的稳定性和可用性。Solaris Volume Manager 热备件可以进一步提高镜像卷和 RAID-5 卷的数据可用性。

设置完配置后，可以使用 Solaris Management Console 中增强的存储工具来报告其操作。

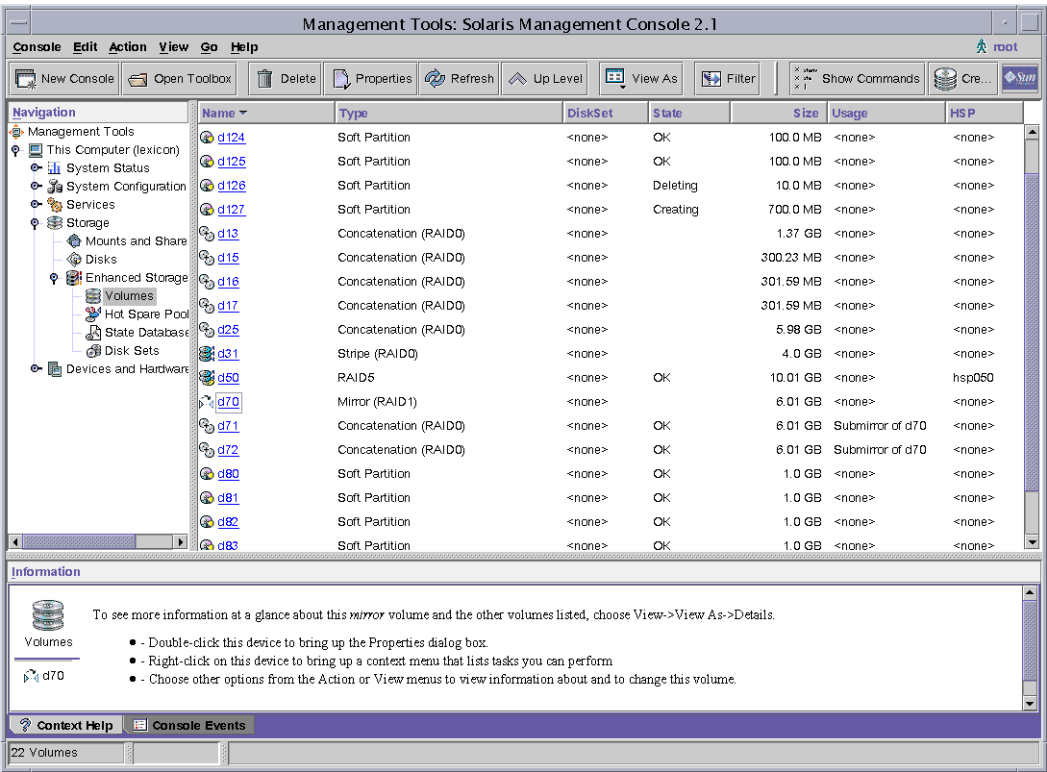
## 如何管理 Solaris Volume Manager

使用以下方法之一来管理 Solaris Volume Manager：

- Solaris Management Console—此工具提供用于管理卷管理功能的 GUI。使用 Solaris Management Console 中增强的存储工具。有关增强的存储工具的示例，请参见 [图 3-1](#)。此界面提供 Solaris Volume Manager 组件（包括卷、热备用池和状态数据副本）的图形视图，并提供对 Solaris Volume Manager 组件的基于向导的处理功能，使您可以快速配置磁盘或更改现有配置。
- 命令行—可以使用多种命令执行卷管理功能。Solaris Volume Manager 核心命令以 meta 开头，例如 metainit 和 metastat 命令。有关 Solaris Volume Manager 命令的列表，请参见[附录 B，Solaris Volume Manager 快速参考](#)。

注 – 请不要尝试同时使用命令行和 GUI 来管理 Solaris Volume Manager。这样可能会导致对配置所做的更改出现冲突，从而使得行为不可预测。您可以使用这两种工具来管理 Solaris Volume Manager，但不能同时使用。

图 3-1 Solaris Management Console 中 "Enhanced Storage"（增强的存储）工具 (Solaris Volume Manager) 视图



## ▼ 如何访问 Solaris Volume Manager 图形用户界面 (Graphical User Interface, GUI)

Solaris Volume Manager GUI (Enhanced Storage) 是 Solaris Management Console 的一部分。要访问此 GUI，请按照以下说明进行操作：

- 1 使用以下命令在主机系统上启动 Solaris Management Console：

```
% /usr/sbin/smc
```

- 2 在 "Navigation" ( 导航 ) 窗格中双击 "This Computer" ( 本计算机 ) 。
- 3 在 "Navigation" ( 导航 ) 窗格中双击 "Storage" ( 存储 ) 。
- 4 在 "Navigation" ( 导航 ) 窗格中双击 "Enhanced Storage" ( 增强的存储 ) 以装入 Solaris Volume Manager 工具。
- 5 如果系统提示登录，请以 root 用户身份或具有等效访问权限的用户身份登录。
- 6 双击相应图标来管理卷、热备用池、状态数据库副本和磁盘集。

提示 – Solaris Management Console 中的所有工具都会在控制台窗口的底部或向导面板的左侧显示信息。您可随时选择 "Help" ( 帮助 )，以查找有关在此界面中执行任务的其他信息。

## Solaris Volume Manager 要求

Solaris Volume Manager 要求包括以下内容：

- 必须具有 root 用户权限才能管理 Solaris Volume Manager。通过 Solaris Management Console 中的 "User Profile" ( 用户配置文件 ) 功能授予的等效特权允许通过 Solaris Management Console 进行管理。但是，只有 root 用户才可以使用 Solaris Volume Manager 命令行界面。
- 在使用 Solaris Volume Manager 创建卷之前，Solaris Volume Manager 系统上必须存在状态数据库副本。状态数据库副本包含所有卷、热备件和磁盘集的配置和状态信息。应该至少存在三个副本，而且为了达到最高的可靠性，这些副本应置于不同的控制器和不同的磁盘上。有关状态数据库副本的更多信息，请参见第 57 页中的“关于 Solaris Volume Manager 状态数据库和副本”。有关如何创建状态数据库副本的说明，请参见第 64 页中的“创建状态数据库副本”。

## Solaris Volume Manager 组件概述

使用 Solaris Volume Manager 创建的五种基本类型的组件为卷、软分区、磁盘集、状态数据库副本和热备用池。下表概述了这些 Solaris Volume Manager 功能。

表 3-1 Solaris Volume Manager 功能摘要

Solaris Volume Manager 功能	定义	目的	更多信息
■ RAID-0 卷（条带、串 联、串联条带）	一组物理分片，在系统中显示为单个逻辑设备	提高存储容量、性能或数据可用性。	第 35 页中的“卷概述”
■ RAID-1（镜像）卷			
■ RAID-5 卷			

表 3-1 Solaris Volume Manager 功能摘要 (续)

Solaris Volume Manager 功能	定义	目的	更多信息
软分区	物理分片或逻辑卷的细分，提供更小、更易于管理的存储单元	提高大型存储卷的可管理性。	<a href="#">第 12 章，软分区（概述）</a>
状态数据库（状态数据库副本）	包含所有卷、热备件和磁盘集的配置和状态信息的数据库。在创建状态数据库副本之前，Solaris Volume Manager 无法运行。	存储有关 Solaris Volume Manager 配置状态的信息	<a href="#">第 38 页中的“状态数据库和状态数据库副本”</a>
热备用池	保留的分片（热备件）的集合。当子镜像或 RAID-5 卷组件出现故障时，会自动使用这些分片进行替换。	提高 RAID-1 和 RAID-5 卷的数据可用性。	<a href="#">第 39 页中的“热备用池”</a>
磁盘集	独立名称空间中的一组共享磁盘驱动器，其中包含卷和热备件，可由多个主机非同时共享	提供数据冗余性和数据可用性，并提供独立名称空间以简化管理	<a href="#">第 39 页中的“磁盘集”</a>

## 卷概述

卷是一组物理分片，在系统中显示为单个逻辑设备。在标准的 UNIX 术语中，卷实际上称作伪设备或虚拟设备。

注 – 在以前，Solstice DiskSuite 产品将这些逻辑设备称为元设备。但是，为了简化和标准化，本书将这些设备统称为卷。

## 卷的类别

创建卷时，可以创建 RAID-0（串联或条带）卷、RAID-1（镜像）卷、RAID-5 卷或软分区。

可以使用 Solaris Management Console 中增强的存储工具或命令行实用程序来创建和管理卷。

下表汇总了卷的类别。

表 3-2 卷的类别

卷	说明
RAID-0（条带或串联）	可直接使用，也可用作镜像的基本生成块。RAID-0 卷不直接提供数据冗余性。

表 3-2 卷的类别 (续)	
卷	说明
RAID-1 (镜像)	通过维护多个副本来复制数据。RAID-1 卷由一个或多个称为子镜像的 RAID-0 卷组成。
RAID-5	通过使用奇偶校验信息来复制数据。在磁盘出现故障时，可以使用可用数据和奇偶校验信息重新生成丢失的数据。RAID-5 卷通常由分片组成。将为奇偶校验信息分配相当于一个分片大小的空间，但是奇偶校验信息分布在 RAID-5 卷的所有分片中。
软分区	将分片或逻辑卷分成一个或多个更小的可扩展卷。

## 如何使用卷

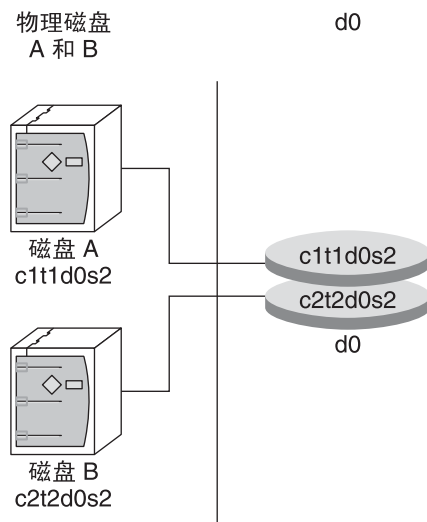
使用卷可提高存储容量和性能并增加数据的可用性。在某些情况下，卷还可以提高 I/O 性能。在功能方面，卷与分片特性相同。卷与分片很相像，因此，对于最终用户、应用程序和文件系统是透明的。与物理设备一样，可通过块或原始设备名称访问卷。根据使用的是块或原始设备，卷名称也会有所不同。有关卷名称的详细信息，请参见第 38 页中的“卷名称”。

可以对卷使用大多数文件系统命令，包括 `mkfs`、`mount`、`umount`、`ufsdump`、`ufsrestore` 以及其他命令。但是，您不能使用 `format` 命令。只要卷包含挂载的文件系统，就可以在该卷中读取、写入或复制文件。

## 示例一由两个分片组成的卷

图 3-2 显示了由两个分片组成的卷，一个分片来自磁盘 A，一个分片来自磁盘 B。应用程序或 UFS 会将该卷视为一个物理磁盘。向该卷添加更多分片可增加其存储容量。

图 3-2 卷、物理磁盘和分片之间的关系



## 使用 growfs 命令扩展卷和磁盘空间

使用 Solaris Volume Manager，可以通过添加其他分片来扩展卷。您可以使用 Solaris Management Console 中增强的存储工具或命令行界面向现有卷添加分片。

您可以扩展卷中包含的已挂载或已卸载的 UFS 文件系统，而无需停止或备份系统。不过，备份数据始终是好的做法。扩展卷后，可使用 **growfs** 命令扩展文件系统。

---

**注** – 文件系统扩展后，其大小将无法缩小。无法缩小文件系统大小是 UFS 的局限。同样，Solaris Volume Manager 分区增大后，也无法缩小。

---

使用原始卷的应用程序和数据库必须自身具备“扩展”增加空间的方法，以便应用程序可识别这些空间。Solaris Volume Manager 不提供此功能。

可以采用以下方式扩展卷中的磁盘空间：

- 向 RAID-0 卷添加一个或多个分片
- 向 RAID-1 卷的所有子镜像添加一个或多个分片
- 向 RAID-5 卷添加一个或多个分片
- 通过添加来自底层组件的空间而扩展软分区

使用 **growfs** 命令可以扩展 UFS 文件系统，而不会丢失服务或数据。但是，在 **growfs** 命令运行时，会暂停对卷的写入访问。您可以将文件系统扩展到包含该文件系统的分片或卷的大小。

通过对 **growfs** 命令使用 **-s size** 选项，可以在文件系统扩展时仅使用所增加磁盘空间的一部分。

---

注-扩展镜像时，先向镜像的底层子镜像添加空间。然后，再对 RAID-1 卷运行 `growfs` 命令。一般原则是向基础设备添加空间，然后对顶层设备运行 `growfs` 命令。

---

## 卷名称

与物理分片一样，卷也会在文件系统中显示其逻辑名称。逻辑卷名称在块设备的 `/dev/md/dsk` 目录和原始设备的 `/dev/md/rdsk` 目录中均存在相应条目。对于任何 `meta*` 命令，您通常不用指定完整的卷名（例如 `/dev/md/dsk/volume-name`），而只使用缩写形式的卷名称（例如 `d1`）即可。一般情况下，只要卷当前未被使用，且新名称未被其他卷使用，就可以对卷重命名。有关更多信息，请参见第 202 页中的“交换卷名称”。

最初，卷名称必须以 "d" 字母开头，后跟一个数字（例如 `d0`）。此格式现在仍可接受。下面是使用 "d\*" 命名结构的卷名称示例：

<code>/dev/md/dsk/d0</code>	块卷 <code>d0</code>
<code>/dev/md/dsk/d1</code>	块卷 <code>d1</code>
<code>/dev/md/rdsk/d126</code>	原始卷 <code>d126</code>
<code>/dev/md/rdsk/d127</code>	原始卷 <code>d127</code>

## 卷命名原则

使用标准的卷名可简化管理并使您能够快速识别卷类型。以下是几点建议：

- 为每种类型的卷指定范围。例如，为 RAID-1 卷分配数字 0 至 20，为 RAID-0 卷分配数字 21 到 40，等等。
- 对镜像使用命名关系。例如，使用以零 (0) 结尾的数字命名镜像，使用以 (1)、二 (2) 等结尾的数字命名子镜像。例如，可以按如下方式命名镜像：镜像 `d10`、子镜像 `d11` 和 `d12`；镜像 `d20`、子镜像 `d21`、`d22`、`d23` 和 `d24`。
- 使用将分片编号和磁盘编号映射到卷编号的命名方法。

## 状态数据库和状态数据库副本

**状态数据库**是存储有关 Solaris Volume Manager 配置状态的信息的数据库。状态数据库会记录并跟踪对配置所做的更改。当配置或状态发生更改时，Solaris Volume Manager 将自动更新状态数据库。例如，创建新卷是一种配置更改，而子镜像失败则是一种状态更改。

状态数据库实际上是多个复制的数据库副本的集合。每一份数据库（称作**状态数据库副本**）都可以确保状态数据库中的数据始终有效。拥有多份状态数据库副本可以防止发生单点故障时丢失数据。状态数据库可以跟踪所有已知状态数据库副本的位置和状态。

在创建状态数据库及其状态数据库副本之前，Solaris Volume Manager 无法进行操作。Solaris Volume Manager 配置中必须具有操作状态数据库。

在设置配置时，您可以将状态数据库副本放在以下位置之一：

- 在专用分片上
- 在以后要成为卷组成部分的分片上

Solaris Volume Manager 可识别分片是否包含状态数据库副本，如果分片已在某个卷中使用，将自动跳过其中的副本。为状态数据库副本保留的分片部分不应用作任何其他用途。

一个分片上可以存储多个状态数据库副本。但是，这样做可能会使系统更容易出现单点故障。

如果删除所有状态数据库副本，Solaris 操作系统仍可继续正常运行。但是，如果在磁盘上不存在现有状态数据库副本的情况下重新引导系统，则系统会丢失所有 Solaris Volume Manager 配置数据。

## 热备用池

**热备用池**是 Solaris Volume Manager 保留的用于自动替换故障组件的分片（**热备件**）的集合。这些热备件在子镜像或 RAID-5 卷中均可使用。热备件可提高 RAID-1 和 RAID-5 卷的数据可用性。您可以使用 Solaris Management Console 中增强的存储工具或命令行界面创建热备用池。

当组件出现错误时，Solaris Volume Manager 将查找第一个等于或大于故障组件大小的可用热备件。如果找到这样的热备件，Solaris Volume Manager 将自动替换组件并重新同步数据。如果在热备件列表中未发现适当大小的分片，则系统将认为子镜像或 RAID-5 卷出现故障。有关更多信息，请参见第 16 章，**热备用池（概述）**。

## 磁盘集

**磁盘集**是指一组包含逻辑卷和热备件的物理存储卷。必须从该磁盘集内部的驱动器上构建卷和热备用池。一旦在该磁盘集内部创建了卷，您就可以像使用物理分片一样来使用该卷了。

磁盘集在群集环境中提供数据可用性。如果一个主机出现故障，则另一个主机可接管故障主机的磁盘集。（此类配置称为**故障转移配置**。）此外，磁盘集可用于帮助管理 Solaris Volume Manager 名称空间，实现对网络连接存储设备的随时访问。

有关更多信息，请参见第 18 章，**磁盘集（概述）**。

# Solaris Volume Manager 配置原则

设计很差的 Solaris Volume Manager 配置会降低性能。本节提供实现 Solaris Volume Manager 良好性能的提示。有关存储配置性能原则的信息，请参见第 28 页中的“一般性能原则”。

## 一般原则

- **磁盘和控制器**—将卷中的驱动器置于单独的驱动器路径中，而对于 SCSI 驱动器，则置于单独的主机适配器上。将 I/O 负载分布在多个控制器中可以提高卷的性能和可用性。
- **系统文件**—绝不要编辑或删除 `/etc/lvm/mddb.cf` 或 `/etc/lvm/md.cf` 文件。  
确保持定期备份这些文件。
- **卷完整性**—如果将一个分片定义为一个卷，请勿将底层分片用于任何其他目的，包括将此分片用作转储设备。
- **关于磁盘和分区的信息**—保留 `prtvtoc` 和 `metastat -p` 命令输出的副本，以备重新格式化损坏的磁盘或重新创建 Solaris Volume Manager 配置时使用。

## 文件系统原则

请勿在卷的底层分片上挂载文件系统。如果要将分片用于任意类型的卷，则不得将该分片作为文件系统进行挂载。如有可能，请在激活卷之前卸载要用作卷的任何物理设备。

# 创建 Solaris Volume Manager 组件概述

创建 Solaris Volume Manager 组件时，可以为逻辑 Solaris Volume Manager 名称（例如 `d0`）指定物理分片。可创建的 Solaris Volume Manager 组件包括以下各项：

- 状态数据库副本
- 各种卷，比如 RAID-0（条带、串联）、RAID-1（镜像）、RAID-5 和软分区
- 热备用池
- 磁盘集

---

注—有关如何命名卷的建议，请参见第 38 页中的“卷名称”。

---

## 创建 Solaris Volume Manager 组件的先决条件

创建 Solaris Volume Manager 组件的先决条件如下所示：

- 创建初始状态数据库副本。如果尚未执行此操作，请参见第 64 页中的“创建状态数据库副本”。
- 确定可供 Solaris Volume Manager 使用的分片。如有必要，请使用 `format` 命令、`fmthard` 命令或 Solaris Management Console 对现有磁盘重新分区。
- 确保您具有 root 用户权限。
- 具有所有数据的最新备份。
- 如果是使用 GUI，请启动 Solaris Management Console 并导航到 Solaris Volume Manager 功能。有关信息，请参见第 33 页中的“如何访问 Solaris Volume Manager 图形用户界面 (Graphical User Interface, GUI)”。

## Solaris Volume Manager 中的多 TB 支持概述

从 Solaris 9 4/03 发行版开始，Solaris Volume Manager 支持在运行 64 位内核的系统上使用大于 1 TB 的存储设备和逻辑卷。

---

注 - 请使用 `isainfo -v` 确定系统运行的是否为 64 位内核。如果出现字符串 "64-bit"，则说明运行的是 64 位内核。

---

使用 Solaris Volume Manager 可以执行以下操作：

- 创建、修改和删除基于（或通过）大小超过 1 TB 的逻辑存储单元 (Logical Storage Unit, LSU) 而构建的逻辑卷。
- 创建、修改和删除大小超过 1 TB 的逻辑卷。

系统自动支持大型卷。如果创建大于 1 TB 的设备，则 Solaris Volume Manager 会对其进行相应配置，无需用户干预。

## 大型卷支持的限制

Solaris Volume Manager 仅支持在运行 64 位内核的 Solaris 9 4/03 或更高版本的系统上使用大型卷（大于 1 TB）。如果在以前的 Solaris 9 发行版上的 32 位内核下运行带有大型卷的系统，会影响 Solaris Volume Manager 功能。具体而言，请注意以下几点：

- 如果在 32 位 Solaris 9 4/03 或更高版本的内核下重新引导带有大型卷的系统，则可通过 `metastat` 输出查看大型卷，但无法访问、修改或删除这些卷。此外，无法创建新的大型卷。大型卷上的任何卷或文件系统也都不可用。

- 如果在 Solaris 9 4/03 之前的 Solaris 发行版下重新引导带有大型卷的系统，则 Solaris Volume Manager 将不会启动。在其他版本的 Solaris 平台上运行 Solaris Volume Manager 之前，必须删除所有大型卷。



**注意** – 如果希望运行具有 32 位内核的 Solaris 软件，或希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，请不要创建大型卷。

## 使用大型卷

所有 Solaris Volume Manager 命令均适用于大型卷。无需改变语法或执行任何特殊任务即可利用大型卷支持功能。因此，熟悉 Solaris Volume Manager 的系统管理员可以立即使用 Solaris Volume Manager 大型卷。

**提示** – 如果创建了大型卷，随后确定需要在以前发行版的 Solaris 下使用 Solaris Volume Manager，或者需要在 32 位 Solaris 9 4/03 或更高版本的内核下运行，则需要删除这些大型卷。请在 64 位内核下使用 `metaclear` 命令从 Solaris Volume Manager 配置中删除大型卷，然后在以前的 Solaris 发行版或在 32 位内核下重新引导。

## 升级到 Solaris Volume Manager

Solaris Volume Manager 完全支持从 Solstice DiskSuite 版本 4.1、4.2 和 4.2.1 进行无缝升级。请确保所有卷都处于 "Okay"（正常）状态（而不是 "Needs Maintenance"（需要维护）或 "Last Erred"（最近出错）状态），且没有任何热备件正在使用之中。您无需对 Solaris Volume Manager 执行任何特殊操作即可进行升级，也就是说，无需更改配置或细分根镜像。升级系统时，将显示 Solstice DiskSuite 配置，并可在升级后通过 Solaris Volume Manager 工具访问该配置。

Solaris 10 OS 引入了服务管理工具 (Service Management Facility, SMF)，该工具提供了用于扩展传统 UNIX 启动脚本、`init` 运行级别和配置文件的基础结构。从以前版本的 Solaris OS 升级时，请验证与 Solaris Volume Manager 关联的 SMF 服务是否为联机状态。如果 SMF 服务未联机，则在管理 Solaris Volume Manager 时可能会遇到问题。

要检查与 Solaris Volume Manager 关联的 SMF 服务，请使用以下格式的 `svcs` 命令：

```
# svcs -a | egrep "md|meta"
disabled      12:05:45 svc:/network/rpc/mdcomm:default
disabled      12:05:45 svc:/network/rpc/metamed:default
disabled      12:05:45 svc:/network/rpc/metamh:default
online        12:05:39 svc:/system/metainit:default
online        12:05:46 svc:/network/rpc/meta:default
online        12:05:48 svc:/system/fmd:default
online        12:05:51 svc:/system/mdmonitor:default
```

如果 Solaris Volume Manager 配置仅包含本地集，则这些服务应处于联机状态：

```
svc:/system/metainit  
svc:/network/rpc/meta  
svc:/system/mdmonitor
```

如果 Solaris Volume Manager 配置包含磁盘集，则其他的这些服务应处于联机状态：

```
svc:/network/rpc/metamed  
svc:/network/rpc/metamh
```

如果 Solaris Volume Manager 包含多节点磁盘集，则除已提到的其他服务外，此服务也应处于联机状态：

```
svc:/network/rpc/mdcomm
```

有关 SMF 的更多信息，请参见《[Oracle Solaris 管理：基本管理](#)》中的第 18 章“管理服务（概述）”。



## Solaris Volume Manager for Sun Cluster（概述）

---

本章概述了 Solaris Volume Manager for Sun Cluster。

本章包括以下信息：

- 第 45 页中的“Solaris Volume Manager for Sun Cluster 介绍”
- 第 47 页中的“多属主磁盘集概念”
- 第 49 页中的“Solaris Volume Manager for Sun Cluster 配置”
- 第 50 页中的“多属主磁盘集中的 RAID-1（镜像）卷”

### Solaris Volume Manager for Sun Cluster 介绍

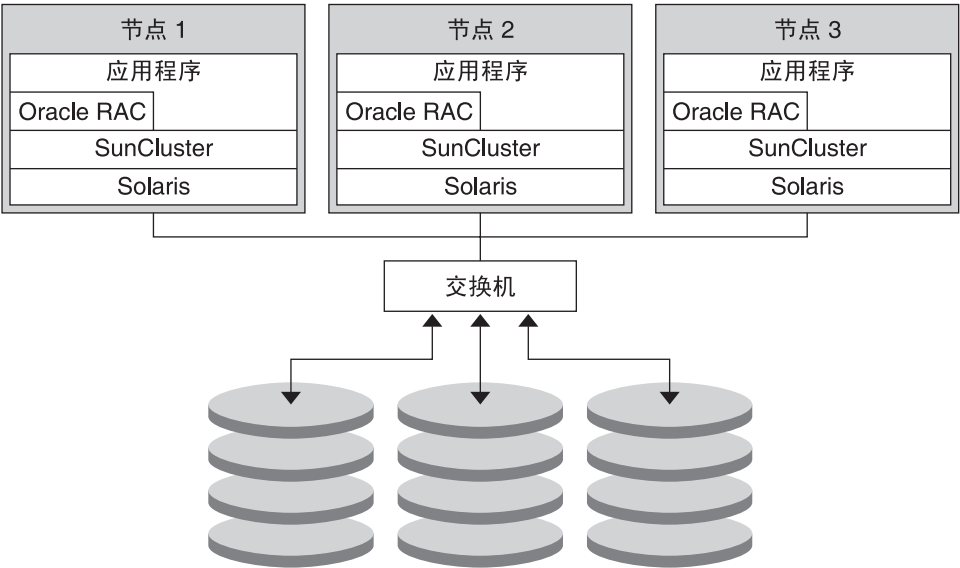
从 Solaris 9 9/04 发行版开始，Solaris Volume Manager 可在 Sun Cluster 环境中使用多属主磁盘集管理存储。**多属主磁盘集**允许多个节点共享磁盘的所有权以及同时向共享磁盘进行写入。以前，磁盘集中的所有参与主机都可以看到共享磁盘集，但是一次只能有一个主机进行访问。多属主磁盘集可与 Sun Cluster 和 Oracle Real Application Clusters 之类的应用程序结合使用。

多属主磁盘集和 Solaris Volume Manager 共享磁盘可共存于同一节点上。但是，不支持在这两种配置之间移动磁盘集。

Solaris Volume Manager for Sun Cluster 创建的组件与使用 Solaris Volume Manager 创建的组件相同，包括条带、串联、镜像、软分区和热备件。Solaris Volume Manager for Sun Cluster 不支持 RAID-5 卷和事务卷。

下图显示了典型群集配置中软件和共享存储之间的关联。

图 4-1 群集配置样例



每个节点都有本地存储以及至少一个指向共享存储的路径。群集中的多属主磁盘集由 Solaris Volume Manager for Sun Cluster 进行管理，后者是 Solaris 操作系统 (Solaris Operating System, Solaris OS) 的一部分。

## 先决条件：多属主磁盘卷功能所需的软件组件

要使用 Solaris Volume Manager for Sun Cluster，除 Solaris OS 之外，还必须安装以下软件：

- Sun Cluster 初始群集框架
- Sun Cluster Support for Oracle Real Application Clusters 软件
- Oracle Real Application Clusters 软件

注 - 有关设置 Sun Cluster 和 Oracle Real Application Clusters 软件的信息，请参见《[Oracle Solaris Cluster 软件安装指南](#)》和《[适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南](#)》。

# 多属主磁盘集概念

由 Solaris Volume Manager for Sun Cluster 管理的存储可以划分为多属主磁盘集。多属主磁盘集允许多个节点共享磁盘集的所有权以及同时向共享磁盘进行写入。应用程序（例如 Oracle Real Application Clusters）的实例运行在群集的每个节点上，因此，多属主磁盘集提供了可伸缩性。因为应用程序的每个实例都可以直接访问共享存储，所以，多属主磁盘集还提高了应用程序的性能。

注 - 多属主磁盘集功能只能在 Sun Cluster 环境中启用。**节点**是作为 Sun Cluster 系统一部分的物理计算机。

每个多属主磁盘集均与一个节点列表关联。这些节点共享磁盘集的所有权。以下 `metaset -s disk-set` 命令显示了多属主磁盘集的输出。

```
# metaset -s blue

Multi-owner Set name = blue, Set number = 1, Master = nodeone

Host          Owner          Member
  nodeone      multi-owner    Yes
  nodetwo      multi-owner    Yes

Drive    Dbase

d9        Yes

d13       Yes
```

此输出显示了节点列表中共享磁盘集所有权的 `nodeone` 和 `nodetwo`。此外，`nodeone` 被指定为**主节点**。

每个多属主磁盘集都有一个主节点。创建磁盘集后，向其中添加第一个磁盘的节点将成为该磁盘集的主节点。主节点可以创建、删除和更新磁盘集中的状态数据库副本。

注 - 有关状态数据库副本的更多信息，请参见[第 6 章，状态数据库（概述）](#)。

Solaris Volume Manager for Sun Cluster 可支持带有不同的甚至重叠的节点列表的磁盘集。因为每个磁盘集都具有一个主节点，所以多个主节点可同时存在于同一个群集中。

以下 `metaset` 命令的输出显示，在向磁盘集添加第一个磁盘后，`nodeone` 成为了主节点。

```
nodeone# metaset -s red
Multi-owner Set name = red, Set number = 1, Master =
```

```
Host          Owner      Member
nodeone
nodetwo      Yes
nodeone# metaset -s red -a /dev/did/dsk/d9
nodeone# metaset -s red

Multi-owner Set name = red, Set number = 1, Master = nodeone

Host          Owner      Member
nodeone      multi-owner  Yes
nodetwo      multi-owner  Yes

Drive      Dbase
d9          Yes
```

Solaris Volume Manager for Sun Cluster 可支持带有不同的甚至重叠的节点列表的磁盘集。因为每个磁盘集都具有一个主节点，所以多个主节点可同时存在于同一个群集中。

## 与多属主磁盘集关联的任务



注意 - 配置多属主磁盘集之前，除 Solaris OS 之外，还必须安装以下软件：

- Sun Cluster 初始群集框架
- Sun Cluster Support for Oracle Real Application Clusters 软件
- Oracle Real Application Clusters 软件

有关设置 Sun Cluster 和 Oracle Real Application Clusters 软件的信息，请参见 [《Oracle Solaris Cluster 软件安装指南》](#) 和 [《适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南》](#)。

Solaris Volume Manager for Sun Cluster 通常使用同一组 Solaris Volume Manager 命令来执行与磁盘集关联的任务。多属主磁盘集所特有的一些命令选项已添加到 `metaset` 命令中。例如，用于创建多属主磁盘集的任务需要在 `metaset` 命令中使用 `-M` 选项。以下输出显示了如何使用 `metaset -s diskset-name -a -M -h hostname` 命令创建多属主磁盘集。

```
# metaset -s red -a -M -h nodeone
# metaset
Multi-owner Set name = red, Set number = 1, Master =

Host          Owner      Member
nodeone      Yes
```

此外，一些 `metaset` 命令选项（如用于获取或释放磁盘集的命令）不能用于多属主磁盘集。有关更多信息，请参见 `metaset(1M)` 手册页。

下面的另一个示例将说明在 Sun Cluster 环境中使用磁盘的任务有何不同之处。Sun Cluster 为每个磁盘分配一个唯一设备 ID (Device ID, DID) 号。如果不想使用 *cntndn* 格式来标识磁盘，则可以使用 Sun Cluster DID 路径名 */dev/did/dsk/dN*。变量 *N* 是 Sun Cluster 分配的设备号。

以下输出显示了如何使用 `metaset -s diskset-name -a disk-name` 命令向多属主磁盘集添加磁盘，以及如何使用 Sun Cluster DID 路径名标识该磁盘。

```
nodeone# metaset -s red
Multi-owner Set name = red
Multi-owner Set name = red, Set number = 1, Master =

Host          Owner          Member
  nodeone                Yes
  nodetwo                Yes
nodeone# metaset -s red -a /dev/did/dsk/d13
nodeone# metaset -s red
Multi-owner Set name = red, Set number = 1, Master = nodeone

Host          Owner          Member
  nodeone      multi-owner    Yes

Drive Dbase

d13   Yes
```

有关为 Oracle Real Application Clusters 创建多属主磁盘集的信息，请参见《适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南》。

有关与磁盘集关联的任务，请参见第 19 章，磁盘集（任务）。

# Solaris Volume Manager for Sun Cluster 配置

Solaris Volume Manager for Sun Cluster 支持以下配置：

- Solaris Volume Manager for Sun Cluster 最多可支持 32 个磁盘集。这些磁盘集可以是多属主磁盘集、共享磁盘集和本地磁盘集的任意组合。

---

注 – 有关不同类型磁盘集的更多信息，请参见第 174 页中的“磁盘集类型”。

---

- 每个多属主磁盘集可支持每个磁盘集最多有 8192 个卷。
- 状态数据库副本的缺省大小为 16 MB。最小大小为 16 MB。最大大小为 256 MB。

Sun Cluster Support for Oracle Real Application Clusters 的许多扩展属性为重新配置过程中的各个步骤指定了超时。有关设置超时的详细信息，请参阅《适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南》。

## 多属主磁盘集中的 RAID-1（镜像）卷

在多属主磁盘集功能中创建的 RAID-1 卷或镜像与 Solaris Volume Manager 共享磁盘集中的 RAID-1 卷相同。但是，多属主磁盘集中的 RAID-1 卷具有另外一些功能。

### 多属主磁盘集的镜像所有权

镜像所有权概念是多属主磁盘集所特有的。与 Solaris Volume Manager 共享磁盘集中的 RAID-1 卷不同，多属主磁盘集中的 RAID-1 卷通常具有与其关联的属主。镜像卷的所有权由卷管理器进行选择。卷的属主是节点列表中为磁盘集指定的一个节点。只有 RAID-1 卷的属主可以对卷执行写入操作。如果非属主节点要对卷执行写入操作，则所有权会转给执行写入操作的节点。`metastat -s diskset-name` 命令的以下输出显示，`nodeone` 是 RAID-1 卷 `d24` 的属主。

```
# metastat -s red
red/d24: Mirror
  Submirror 0: red/d20
    State: Okay
  Submirror 1: red/d21
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Resync option: optimizedresync
  Owner: nodeone
  Size: 825930 blocks (403 MB)
```

### 数据管理和恢复过程

与 Solaris Volume Manager 中的 RAID-1 卷一样，Solaris Volume Manager for Sun Cluster 中的 RAID-1 卷也会执行相关操作以确保数据一致性。Solaris Volume Manager for Sun Cluster 为 RAID-1 卷提供了两个数据管理和恢复选项。

### Solaris Volume Manager for Sun Cluster 的优化的重新同步

Solaris Volume Manager for Sun Cluster 功能中的优化的重新同步与 Solaris Volume Manager 中的优化的重新同步相同。但是，在多属主磁盘集中，将重新同步选项设置为优化的重新同步的 RAID-1 卷将始终具有镜像属主。`metastat -s diskset-name` 命令的以下输出显示，重新同步选项已设置为 `optimizedresync`（用于优化的重新同步）。

```
# metastat -s red
red/d24: Mirror
  Submirror 0: red/d20
    State: Okay
  Submirror 1: red/d21
    State: Okay
  Pass: 1
```

```

Read option: roundrobin (default)
Write option: parallel (default)
Resync option: optimizedresync
Owner: nodeone
Size: 825930 blocks (403 MB)

```

有关优化的重新同步的更多信息，请参见第 88 页中的“优化重新同步”。

## 基于应用程序的恢复和有向镜像读取

为了优化 Solaris Volume Manager for Sun Cluster 中的数据恢复，Oracle Real Application Clusters 之类的应用程序需要具有管理和控制数据恢复的功能。让应用程序控制恢复可以提高恢复的性能。ioctl DKIOGETVOLCAP、DKIOSETVOLCAP 和 DKIODMR 为应用程序在群集环境中的数据管理恢复提供了支持。这些 ioctl 为应用程序提供以下功能：

- 基于应用程序的恢复 (Application Based Recovery, ABR) — 允许应用程序控制镜像卷上的数据恢复
- 有向镜像读取 — 允许应用程序直接读取特定子镜像并确定数据的状态

有关 ioctl 与基于应用程序的数据管理恢复功能结合使用的更多信息，请参见 [dkio\(71\)](#) 手册页。

如果 RAID-1 卷将重新同步选项设置为基于应用程序的恢复，则仅在基于应用程序的恢复过程中才具有镜像属主。metastat -s *diskset-name* 命令的以下输出显示 RAID-1 卷处于常规状态。重新同步选项已设置为基于应用程序的恢复。没有镜像属主。

```

# metastat -s red
red/d24: Mirror
  Submirror 0: red/d20
    State: Okay
  Submirror 1: red/d21
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Resync option: application based
  Owner: None
  Size: 825930 blocks (403 MB)

```



## 配置和使用 Solaris Volume Manager（方案）

---

在《Solaris Volume Manager 管理指南》中，只要可能，实例通常与单个存储配置有关。本章介绍了这些示例中使用的方案，还提供了有关后续章节中使用的初始存储配置的详细信息。

本章包含以下信息：

- 第 53 页中的“方案背景信息”
- 第 54 页中的“最终的 Solaris Volume Manager 配置”

### 方案背景信息

在本书中，方案和许多示例都与单个配置有关。尽管此配置很小（以便于简化文档），但在概念上可延伸到更大的存储环境。

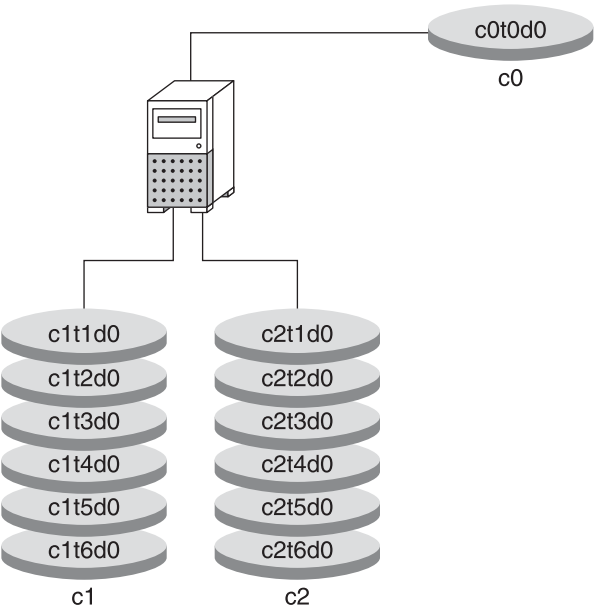
### 硬件配置

硬件系统配置如下：

- 有三个物理上独立的控制器（c0 - IDE、c1 - SCSI 和 c2 - SCSI）。
- 每个 SCSI 控制器都连接到包含六个内置 9 GB 磁盘（c1t1 到 c1t6 和 c2t1 到 c2t6）的 MultiPack。这将创建一个镜像配置。
- 每个控制器/终结器对（c ntn）都有 8.49 GB 可用存储空间。
- 根 (/) 驱动器 c0t0d0 上的存储空间已划分为六个分区。

下图所示的配置有助于理解以上的配置。

图 5-1 基本硬件存储方案图



## 初始物理存储配置

以下是配置 Solaris Volume Manager 前的存储配置：

- SCSI 控制器/终结器对 (`c ntn`) 有大约 20 GB 的存储空间。
- 每个磁盘（如 `c1t1d0`）上的存储空间均划分为七个分区（`cntnd0s0` 到 `cntnd0s6`）。  
对磁盘进行分区，请按照《[System Administration Guide: Devices and File Systems](#)》中的“[Formatting a Disk](#)”中说明的过程执行操作。

## 最终的 Solaris Volume Manager 配置

本书针对特定任务提供了特定的方案。因此，您可以更好地理解本书中的示例，但最终配置大体上与 `metastat -p` 命令显示的以下内容相同：

```
[root@lexicon:/]$ metastat -p
d50 -r c1t4d0s5 c1t5d0s5 c2t4d0s5 c2t5d0s5 c1t1d0s5 c2t1d0s5 -k -i 32b
d1 1 1 c1t2d0s3
d2 1 1 c2t2d0s3
d12 1 1 c1t1d0s0
d13 1 1 c2t1d0s0
d16 1 1 c1t1d0s1
d17 1 1 c2t1d0s1
d25 2 2 c1t1d0s3 c2t1d0s3 -i 32b \
      1 c0t0d0s3
```

```

d31 1 2 c1t4d0s4 c2t4d0s4 -i 8192b
d80 -p d70 -o 1 -b 2097152
d81 -p d70 -o 2097154 -b 2097152
d82 -p d70 -o 4194307 -b 2097152
d83 -p d70 -o 6291460 -b 2097152
d84 -p d70 -o 8388613 -b 2097152
d85 -p d70 -o 10485766 -b 2097152
d70 -m d71 d72 1
d71 3 1 c1t3d0s3 \
      1 c1t3d0s4 \
      1 c1t3d0s5
d72 3 1 c2t3d0s3 \
      1 c2t3d0s4 \
      1 c2t3d0s5
d123 -p c1t3d0s6 -o 1 -b 204800
d124 -p c1t3d0s6 -o 204802 -b 204800
d125 -p c1t3d0s6 -o 409603 -b 204800
d126 -p c1t3d0s7 -o 3592 -b 20480
d127 -p c2t3d0s7 -o 3592 -b 1433600
hsp010
hsp014 c1t2d0s1 c2t2d0s1
hsp050 c1t2d0s5 c2t2d0s5
hsp070 c1t2d0s4 c2t2d0s4

```

有关 `-p` 选项的更多信息，请参见 [metastat\(1M\)](#) 命令。



## 状态数据库（概述）

---

本章提供了有关状态数据库副本的概念性信息。有关执行相关任务的信息，请参见第 7 章，[状态数据库（任务）](#)。

本章包含以下信息：

- 第 57 页中的“关于 Solaris Volume Manager 状态数据库和副本”
- 第 58 页中的“了解多数一致算法”
- 第 60 页中的“处理状态数据库副本错误”
- 第 61 页中的“方案一状态数据库副本”

## 关于 Solaris Volume Manager 状态数据库和副本

Solaris Volume Manager 状态数据库包含有关所有卷、热备件和磁盘集的配置和状态信息。Solaris Volume Manager 可维护状态数据库的多个副本以提供冗余，并防止数据库在系统崩溃时受到损坏（最多只会会有一个数据库副本损坏）。

状态数据库副本可确保状态数据库中的数据始终有效。当状态数据库被更新时，所有状态数据库副本也被更新。每次只更新一个副本的内容，以防止如果系统崩溃，所有更新副本都遭到破坏。

如果系统丢失了一个状态数据库副本，Solaris Volume Manager 必须确定哪些状态数据库副本中仍包含有效数据。Solaris Volume Manager 将使用**多数一致算法**确定此信息。在此算法中，只有当多数（半数 + 1 个）数据库副本可用并一致时，才会认为其中的任何数据库副本是有效的。根据此多数一致算法，设置磁盘配置时，您必须至少创建三个状态数据库副本。如果三个状态数据库副本中至少有两个可用，则可能取得一致。

在引导期间，Solaris Volume Manager 会忽略损坏的状态数据库副本。在某些情况下，Solaris Volume Manager 会尝试重写损坏的状态数据库副本。否则，这些副本将被忽略，直到将其修复。如果某个状态数据库副本因其底层分片出错而被损坏，则需要修复或替换该分片，然后再启用该副本。



**注意** – 请勿将状态数据库副本放置在与光纤网络连接的存储、SAN 或未直接连接到系统的其他存储上。否则可能无法引导 Solaris Volume Manager。副本所在的存储设备必须与传统的 SCSI 或 IDE 驱动器在引导过程中同时可用。

如果所有状态数据库副本丢失，则从理论上讲，也会丢失存储在 Solaris Volume Manager 卷上的所有数据。为此，最好在不同的驱动器上以及控制器之间创建足够的状态数据库副本，以防止发生灾难性故障。此外，最好将初始 Solaris Volume Manager 配置信息以及磁盘分区信息保存下来。

有关向系统中添加更多状态数据库副本的信息，请参见第 7 章，[状态数据库（任务）](#)。有关在丢失状态数据库副本时进行恢复的信息，请参见第 263 页中的“[从状态数据库副本故障中恢复](#)”。

也可对 RAID-1 卷重新同步区域使用状态数据库副本。如果相对于镜像数量来说，状态数据库副本过少，则可能会使副本 I/O 影响 RAID-1 卷的性能。也就是说，如果您有大量镜像，请确保每个 RAID-1 卷至少有两个状态数据库副本，而每个磁盘集最多有 50 个副本。

缺省情况下，卷（本地集）和磁盘集的每个状态数据库副本占用 4 MB 的磁盘存储空间（即，8192 个磁盘扇区）。多属主磁盘集中的状态数据库副本的缺省大小为 16 MB。

副本可以存储在以下设备中：

- 专用本地磁盘分区
- 要成为卷的一部分的本地分区
- 要成为 UFS 日志设备的一部分的本地分区

副本不能存储在根 (/)、swap 或 /usr 分片上。也不能存储在包含现有文件系统或数据的分片上。副本存储后，可以将卷或文件系统放置在同一个分片上。

## 了解多数一致算法

复制的数据库存在一个固有问题，那就是可能很难确定哪个数据库包含有效且正确的数据。为解决此问题，Solaris Volume Manager 使用了多数一致算法。此算法要求，只要多数数据库副本达成一致，就会认为其中任何数据库副本有效。此算法还要求，至少创建三个初始副本。这样，只要这三个副本中至少两个副本可用，就可以达成一致。如果只有一个副本，而系统崩溃，则所有卷配置数据都可能丢失。

为保护数据，只有当所有状态数据库副本中有一半可用时，Solaris Volume Manager 才会正常运行。因此，此算法可确保数据不被损坏。

此多数一致算法将提供以下逻辑：

- 如果状态数据库副本至少有一半可用，则系统就会继续运行。

- 如果可用状态数据库副本数不到一半，则系统将出现紧急情况。
- 只有当状态数据库副本总数中大多数（半数 + 1）可用时，系统才可重新引导至多用户模式。

如果可用状态数据库副本数不够，则您必须引导至单用户模式，并删除足够多的已损坏或丢失的副本，以达到法定数目。请参见第 263 页中的“如何从不足的状态数据库副本中恢复”。

---

注 - 如果状态数据库副本总数为奇数，则 Solaris Volume Manager 会将此值除以二，然后向下舍入为最接近的整数，并在该整数的基础上加 1（一），从而得出多数值。例如，如果系统包含七个副本，则多数值为 4（即，7 除以 2 得到 3.5，然后将 3.5 向下舍入为 3，最后加 1，从而得到 4）。

---

## 管理状态数据库副本

- 缺省情况下，状态数据库副本的大小为 4 MB 或 8192 个块。应在专用分片上创建状态数据库副本，每个副本至少需要 4 MB。由于磁盘分片可能不会这么小，因此您可能需要调整分片的大小以存储状态数据库副本。有关调整分片大小的信息，请参见《System Administration Guide: Devices and File Systems》中的第 9 章“Administering Disks (Tasks)”。
- 为避免出现单点故障，应将状态数据库副本分布在多个分片、驱动器和控制器中。一般希望在单个组件发生故障时大部分副本完好。如果丢失一个副本（例如，设备出现故障），则在运行 Solaris Volume Manager 或重新引导系统时可能会出现問題。要运行 Solaris Volume Manager，需要至少有一半副本可用，但要重新引导至多用户模式，则需要多数副本（半数 + 1）可用。

建议每个 Solaris Volume Manager 磁盘集的状态数据库副本数量最少为 3，最多为 50。建议遵循以下原则：

- 对于只有一个驱动器的系统：将所有三个副本放置在一个分片中。
- 对于具有两个到四个驱动器的系统：一个驱动器上放置两个副本。
- 对于具有五个或更多驱动器的系统：一个驱动器上放置一个副本。
- 如果存在多个控制器，则副本应在所有控制器上尽可能地均匀分布。如果某个控制器出现故障，此策略可以提供冗余性，并有助于平衡负载。如果控制器上存在多个磁盘，则每个控制器上至少有两个磁盘应存储副本。
- 如果需要，您可以在将要用作 RAID-0、RAID-1 或 RAID-5 卷或软分区一部分的分片上创建状态数据库副本。必须先创建副本，然后再向卷中添加分片。Solaris Volume Manager 会预留出分片的起始位置，用于存放状态数据库副本。

将状态数据库副本放在作为卷的一部分的分片上时，由于副本占用了空间，卷的容量减小。一个副本所使用的空间将向上舍入为下一个柱面边界。此卷会跳过此空间。

- RAID-1 卷用于小型随机 I/O（例如，用于数据库的 I/O）。要获得最佳性能，请确保未连接到 RAID-1 卷的分片上（最好位于不同的磁盘和控制器上）的每个 RAID-1 卷都至少具有两个附加副本。
- 不能在现有文件系统、根 (/) 文件系统、/usr 文件系统和 swap 文件系统中创建状态数据库副本。如果需要，可以通过使用 swap 分配空间来创建新分片（假设已提供分片名称）。然后，将状态数据库副本放置在该新分片上。
- 可以在未使用的分片上创建状态数据库副本。
- 您可以随时向系统中添加更多状态数据库副本。添加更多状态数据库副本有助于确保 Solaris Volume Manager 的可用性。



---

**注意** – 如果已从 Solstice DiskSuite 产品升级到 Solaris Volume Manager，并且状态数据库副本与文件系统或逻辑卷共享分片（而不是使用不同的分片），请不要删除现有副本并在同一位置替换为新副本。

Solaris Volume Manager 中的缺省状态数据库副本大小为 8192 个块，而 Solstice DiskSuite 产品中的缺省大小为 1034 个块。如果要删除在 Solstice DiskSuite 产品中创建的缺省大小的状态数据库副本，然后在 Solaris Volume Manager 中添加缺省大小的新副本，则应谨慎操作。您会覆盖占用共享分片其余部分的任何文件系统的前 7158 个块，从而会销毁数据。

---

## 处理状态数据库副本错误

如果某个状态数据库副本出现故障，则只要剩余副本中至少有一半可用，系统就会继续运行。如果可用副本数不到一半，则系统将出现紧急情况。

如果可用副本数超过一半，则系统可重新引导至多用户模式。如果可用副本数未达到多数，则必须重新引导至单用户模式，然后使用 `metadb` 命令删除不可用的副本。

例如，假设您有四个副本。只要有两个副本（即，总数的一半）可用，系统就可以继续运行。但是，要重新引导系统，必须有三个副本（即，总数的一半加 1）可用。

在双磁盘配置中，应始终在每个磁盘上至少创建两个副本。例如，假设您的配置包含两个磁盘，并且您仅创建了三个副本（即，在第一个磁盘上创建两个副本，在第二个磁盘上创建一个副本）。如果包含两个副本的磁盘出现故障，则系统将出现紧急情况，因为剩余磁盘只包含一个副本。而该副本数未达到副本总数的一半。

---

**注** – 在一个双磁盘配置中，如果在每个磁盘上创建了两个副本，则当其中一个磁盘出现故障后，Solaris Volume Manager 仍可正常运行。但是，要使系统重新引导，可用副本数必须超过总副本数的一半，因此，此时无法重新引导。

---

如果包含状态数据库副本的分片出现故障，则该配置的其余部分仍应正常运行。Solaris Volume Manager 会在引导期间查找有效的状态数据库（前提是，可用的有效状态数据库副本数超过一半）。

手动修复或启用状态数据库副本时，Solaris Volume Manager 将使用有效数据更新这些副本。

## 方案一 状态数据库副本

状态数据库副本可提供有关总体 Solaris Volume Manager 配置的冗余数据。以下示例是基于第 5 章，[配置和使用 Solaris Volume Manager（方案）](#)所提供的方案中的样例系统的。此示例说明了如何分布状态数据库副本可提供足够冗余。

此样例系统包含一个内部 IDE 控制器和驱动器，以及两个 SCSI 控制器。每个 SCSI 控制器连接有六个磁盘。可以为系统配置这三个控制器，以避免出现任何单点故障。相对于 Solaris Volume Manager 而言，只包含两个控制器的系统将无法避免单点故障。通过将副本均匀分布在这三个控制器上，并且在每个控制器上至少分布在一个磁盘上（如果可能，可以分布在两个磁盘上），系统便可承受任何单点硬件故障。

在最低配置中，您可以将一个状态数据库副本放置在根磁盘的分片 7 上，然后将其他副本放置在其他两个控制器中每个控制器上的一个磁盘的分片 7 上。为了防止可能发生的远程介质故障，可以向根磁盘中再添加一个副本，然后在每个控制器上的两个不同磁盘中分别放置两个副本，总共六个副本，从而提供足够的安全性。

要进一步增强安全性，可以再添加 12 个副本，并均匀分布在两个镜像每一端的 6 个磁盘中。通过这种配置，便可获得总共 18 个副本，其中，在根磁盘上有 2 个副本，在每个 SCSI 控制器上有 8 个副本（分布在每个控制器上的磁盘中）。



## 状态数据库（任务）

---

本章提供有关执行与 Solaris Volume Manager 状态数据库副本相关联的任务的信息。有关这些任务中所涉及的概念的信息，请参见第 6 章，[状态数据库（概述）](#)。

### 状态数据库副本（任务列表）

以下任务列表列出了管理 Solaris Volume Manager 状态数据库副本所需的过程。

任务	说明	参考
创建状态数据库副本	使用 Solaris Volume Manager GUI 或 <code>metadb -a</code> 命令创建状态数据库副本。	<a href="#">第 64 页中的“如何创建状态数据库副本”</a>
检查状态数据库副本的状态	使用 Solaris Volume Manager GUI 或 <code>metadb</code> 命令检查现有副本的状态。	<a href="#">第 66 页中的“如何检查状态数据库副本的状态”</a>
删除状态数据库副本	使用 Solaris Volume Manager GUI 或 <code>metadb -d</code> 命令删除状态数据库副本。	<a href="#">第 67 页中的“如何删除状态数据库副本”</a>

# 创建状态数据库副本



注意 - 如果已从 Solstice DiskSuite 产品升级到 Solaris Volume Manager，并且状态数据库副本与文件系统或逻辑卷共享分片（而不是使用不同的分片），请不要删除现有副本并在同一位置替换为新的缺省副本。

Solaris Volume Manager 中的缺省状态数据库副本大小为 8192 个块，而 Solstice DiskSuite 产品中的缺省大小为 1034 个块。如果要删除在 Solstice DiskSuite 产品中创建的缺省大小的状态数据库副本，然后在 Solaris Volume Manager 中添加缺省大小的新副本，则应谨慎操作。您会覆盖占用共享分片其余部分的任何文件系统的前 7158 个块，从而会销毁数据。



注意 - 请勿将状态数据库副本放置在与光纤网络连接的存储、SAN 或未直接连接到系统的其他存储上。否则可能无法引导 Solaris Volume Manager。副本所在的存储设备必须与传统的 SCSI 或 IDE 驱动器在引导过程中同时可用。

## ▼ 如何创建状态数据库副本

开始之前 检查第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要创建状态数据库副本，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "State Database Replicas"（状态数据库复制）节点。选择 "Action"（操作）⇒ "Create Replicas"（创建复制）并遵循屏幕说明执行操作。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metadb` 命令：请参见 [metadb\(1M\)](#)。

```
# metadb -a -c number -l length-of-replica -f ctds-of-slice
```

-a 指定添加或创建状态数据库副本。

-f 指定强制执行操作，即使不存在副本。使用 -f 强制创建初始副本。

-c number 指定要添加到指定分片的副本数量。

-l length-of-replica 指定新副本的大小（以块为单位）。缺省大小为 8192。该大小应适用于几乎所有配置，包括那些包含数以千计的逻辑卷的配置。

ctds-of-slice 指定将保存副本的组件的名称。

---

注 – 在命令行中输入不带选项的 `metadb` 命令会报告所有状态数据库副本的状态。

---

### 示例 7-1 创建第一个状态数据库副本

```
# metadb -a -f c0t0d0s7
# metadb
      flags      first blk      block count
...
  a      u        16          8192          /dev/dsk/c0t0d0s7
```

必须使用 `-f` 选项与 `-a` 选项创建第一个状态数据库副本。`-a` 选项会将状态数据库副本添加到系统，而 `-f` 选项会强制创建第一个副本（因此将补充副本添加到系统时可以省略）。

### 示例 7-2 将两个状态数据库副本添加到同一分片

```
# metadb -a -c 2 c1t3d0s1
# metadb
      flags      first blk      block count
...
  a      u        16          8192          /dev/dsk/c1t3d0s1
  a      u       8208          8192          /dev/dsk/c1t3d0s1
```

`-a` 选项会将状态数据库副本添加到系统。`-c 2` 选项会将两个副本置于指定的分片上。`metadb` 命令会检查这些副本是否处于活动状态，如 `metadb` 命令输出中的 `a` 标志所指示。

### 示例 7-3 添加特定大小的状态数据库副本

如果要替换现有的状态数据库副本，则可能需要指定副本的大小。尤其是当现有状态数据库副本（可能是在从 Solstice DiskSuite 产品升级而来的系统上）与文件系统共享分片时，必须将现有副本替换为相同大小的其他副本或者在其他位置添加新的副本。

```
# metadb -a -c 3 -l 1034 c0t0d0s7
# metadb
      flags      first blk      block count
...
  a      u        16          1034          /dev/dsk/c0t0d0s7
  a      u       1050          1034          /dev/dsk/c0t0d0s7
  a      u       2084          1034          /dev/dsk/c0t0d0s7
```

`-a` 选项会将状态数据库副本添加到系统。`-l` 选项指定了要添加的副本中块的长度。

# 维护状态数据库副本

## ▼ 如何检查状态数据库副本的状态

- 1 成为超级用户。
- 2 要检查状态数据库副本的状态，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "State Database Replicas"（状态数据库复制）节点以查看所有现有的状态数据库副本。有关更多信息，请参见联机帮助。
  - 使用 `metadb` 命令查看状态数据库副本的状态。添加 `-i` 选项以显示状态标志的说明，如以下示例所示。请参见 [metadb\(1M\)](#)。

示例 7-4 检查所有状态数据库副本的状态

```
# metadb -i
      flags      first blk      block count
a m p luo      16              8192      /dev/dsk/c0t0d0s7
a   p luo      8208              8192      /dev/dsk/c0t0d0s7
a   p luo      16400             8192      /dev/dsk/c0t0d0s7
a   p luo       16              8192      /dev/dsk/c1t3d0s1
W   p l         16              8192      /dev/dsk/c2t3d0s1
a   p luo       16              8192      /dev/dsk/c1t1d0s3
a   p luo      8208              8192      /dev/dsk/c1t1d0s3
a   p luo      16400             8192      /dev/dsk/c1t1d0s3
r - replica does not have device relocation information
o - replica active prior to last mddb configuration change
u - replica is up to date
l - locator for this replica was read successfully
c - replica's location was in /etc/lvm/mddb.cf
p - replica's location was patched in kernel
m - replica is master, this is replica selected as input
W - replica has device write errors
a - replica is active, commits are occurring to this replica
M - replica had problem with master blocks
D - replica had problem with data blocks
F - replica had format problems
S - replica is too small to hold current data base
R - replica had device read errors
```

状态后面有对所有标志的说明。设备名称前面的字符表示状态。大写字母指示问题状态。小写字母指示“正常”状态。

## ▼ 如何删除状态数据库副本

您可能需要删除状态数据库副本以维护 Solaris Volume Manager 配置。例如，如果要替换磁盘驱动器，需要先删除状态数据库副本，然后再移除驱动器。否则 Solaris Volume Manager 会报告错误。

1 成为超级用户。

2 要删除状态数据库副本，请使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "State Database Replicas"（状态数据库复制）节点以查看所有现有的状态数据库副本。选择要删除的副本，然后选择 "Edit"（编辑）⇒ "Delete"（删除）将其删除。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metadb` 命令：

```
# metadb -d -f ctds-of-slice
-d           指定此项以删除状态数据库副本。
-f           指定强制执行操作，即使不存在副本。
ctds-of-slice 指定包含副本的组件的名称。
```

请注意，您需要指定要从其中删除状态数据库副本的每个分片。有关更多信息，请参见 [metadb\(1M\)](#) 手册页。

### 示例 7-5 删除状态数据库副本

```
# metadb -d -f c0t0d0s7
```

本示例显示如何从分片中删除最后一个副本。

您必须添加 `-f` 选项以强制删除系统上的最后一个副本。



## RAID-0（条带和串联）卷（概述）

---

本章介绍 Solaris Volume Manager 中可用的 RAID-0（条带和串联）卷。有关相关任务的信息，请参见第 9 章，[RAID-0（条带和串联）卷（任务）](#)。

本章提供以下信息：

- 第 69 页中的“RAID-0 卷概述”
- 第 75 页中的“创建 RAID-0 卷的背景信息”
- 第 76 页中的“方案—RAID-0 卷”

### RAID-0 卷概述

RAID-0 卷由分片或软分区组成。通过这些卷，可以扩展磁盘存储容量。可以直接使用这些卷，也可以将它们用作 RAID-1（镜像）卷和软分区的生成块。RAID-0 卷有三种：

- 条带卷
- 串联卷
- 串联条带卷

---

注 - 组件是指在其他逻辑卷中使用的任何设备（从分片到软分区等）。

---

条带卷将数据平均分布在卷的所有组件中，而串联卷则先将数据写入第一个可用组件，直到该组件写满后再移至下一个可用组件。串联条带卷也是一种条带卷，此类条带卷只是通过添加额外组件而对原始配置进行了扩展。

通过 RAID-0 卷，可以快速方便地扩展磁盘存储容量。缺点是，这些卷不会像 RAID-1 或 RAID-5 卷一样提供任何数据冗余。只要 RAID-0 卷中的一个组件发生故障，数据就会丢失。

对于在条带卷中执行的顺序 I/O 操作，Solaris Volume Manager 会先读取第一个组件上的一个块区段（称为**交错**）中的所有块，然后读取第二个组件上的一个块区段中的所有块，依此类推。

对于在串联卷上执行的顺序 I/O 操作，Solaris Volume Manager 会先读取第一个组件上的所有块，然后读取第二个组件的所有块，依此类推。

在串联卷和条带卷上，所有 I/O 操作都是并行执行的。

可以对任何文件系统使用只包含一个分片的 RAID-0 卷。

可以对除以下文件系统之外的任何文件系统使用包含多个组件的 RAID-0 卷：

- 根 (/)
- /usr
- swap
- /var
- /opt
- 升级或安装操作系统期间访问的任何文件系统

---

注 - 在镜像根 (/)、/usr、swap、/var 或 /opt 时，可以将该文件系统放置在一个充当子镜像的单向串联或条带（单分片串联）中。此单向串联由另一个子镜像进行镜像，该子镜像也必须是一个串联。

---

## RAID-0（条带）卷

RAID-0（条带）卷是指将数据分布在一个或多个组件中的卷。条带化可以在两个或更多组件之间交替分布大小相等的数据区段，从而形成一个逻辑存储单元。这些区段循环交错，从而利用各个组件来交替构成组合空间，其效果就像洗牌一样。

---

注 - 要增加条带卷的容量，需要生成串联条带卷。请参见第 73 页中的“RAID-0（串联条带）卷”。

---

通过条带化，多个控制器可以同时访问数据，这也称为**并行访问**。并行访问可以增加 I/O 吞吐量，因为在大多数时间，卷中的所有磁盘都会忙于为 I/O 请求提供服务。

不能将现有文件系统直接转换为条带。要将现有文件系统放置在条带卷上，必须先备份该文件系统并创建卷，然后将该文件系统恢复到该条带卷上。

## RAID-0（条带）卷的交错值

交错是指条带卷上逻辑数据区段的大小（以 KB、MB 或块为单位）。可以根据应用程序使用不同的交错值来提高配置的性能。性能的提高源于使用多个磁盘装备来管理 I/O 请求。当 I/O 请求超过交错大小时，性能可能会有所提高。

注 – RAID-5 卷也可使用交错值。有关更多信息，请参见第 143 页中的“RAID-5 卷概述”。

创建条带卷时，可以设置交错值，也可以使用 Solaris Volume Manager 的缺省交错值 16 KB。一旦创建条带卷，该交错值就无法更改了。但是，可以备份条带卷上的数据，删除条带卷，使用新交错值创建新条带卷，然后恢复数据。

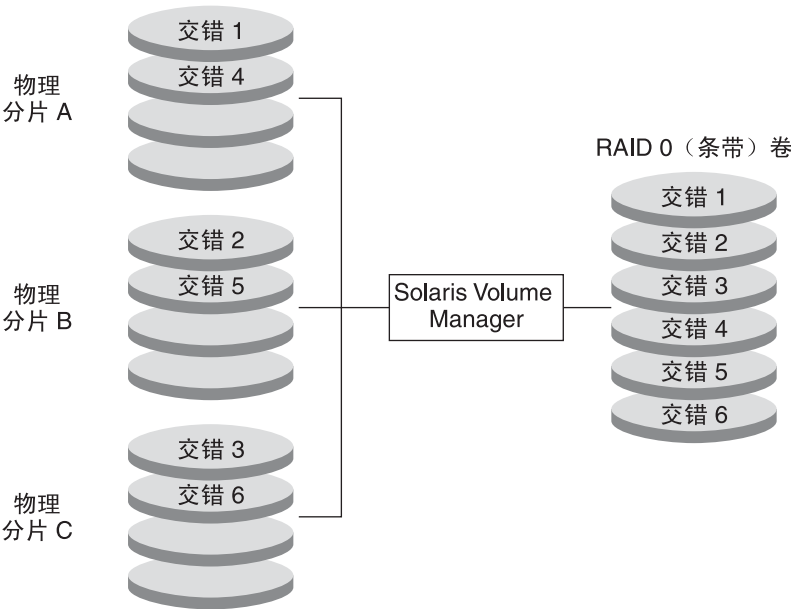
### 方案 – RAID-0（条带）卷

图 8-1 显示了一个由三个组件（分片）构建的条带卷。该图还显示了数据是如何根据交错大小并使用循环技术写入卷组件的。

Solaris Volume Manager 在将数据写入条带卷的组件时，它会将宽度为交错值的数据块写入磁盘 A（交错 1）、磁盘 B（交错 2）和磁盘 C（交错 3）。然后，Solaris Volume Manager 会重复这一模式，从而写入磁盘 A（交错 4）、磁盘 B（交错 5）和磁盘 C（交错 6），依此类推。

交错值用于设置每次向分片中写入的数据大小。条带卷的总容量等于组件数量乘以最小组件的大小。（如果以下示例中的每个分片为 2 GB，则卷为 6 GB。）

图 8-1 RAID-0（条带）卷示例



## RAID-0（串联）卷

在 RAID-0（串联）卷中，数据按顺序逐个排列在各个组件中，从而形成一个逻辑存储单元。

使用串联卷可以将多个组件的容量组合在一起，从而获得更多的存储容量。随着存储需求的增长，可以向串联卷中添加更多组件。

通过串联卷，可以联机动态扩展存储容量和文件系统大小。此外，使用串联卷还可以在其他组件处于活动状态时添加组件。

串联卷还可以扩展任何已挂载的活动 UFS 文件系统，而无需关闭该系统。一般情况下，串联卷的总容量等于卷中所有组件的容量总和。如果串联卷包含具有状态数据库副本的分片，则卷的总容量等于各个组件容量之和减去为该副本保留的空间。

您也可以仅使用一个组件来创建串联卷。日后需要更多存储时，可以向该卷中添加更多组件。

---

注 – 在镜像根 (/)、swap、/usr、/opt 或 /var 文件系统时，必须使用串联卷来封装它们。

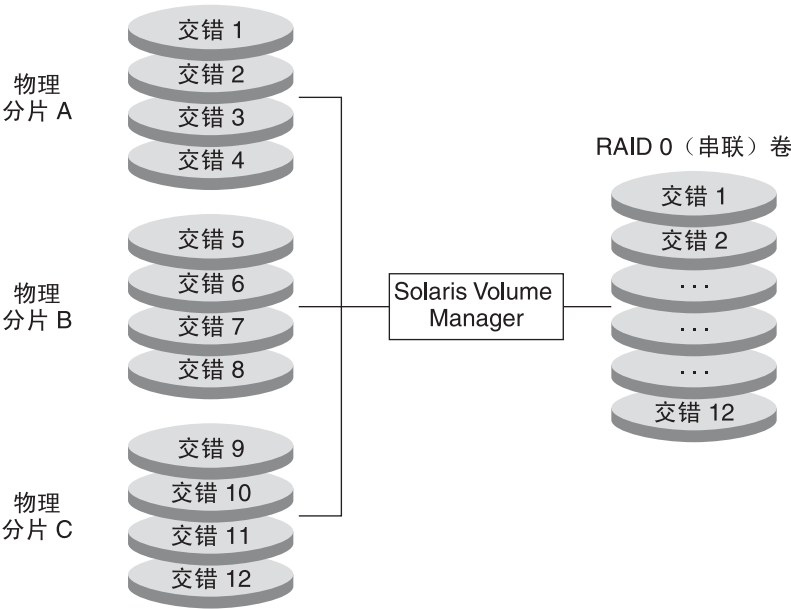
---

### 方案 – RAID-0（串联）卷

图 8-2 说明了一个由三个组件（分片）构建的串联卷。该图还显示了数据是如何根据交错大小写入卷组件并按顺序写入每个分片的。

数据块会按顺序写入各个组件（从分片 A 开始）。您可以假设分片 A 包含逻辑数据块 1 到 4。磁盘 B 包含逻辑数据块 5 到 8。驱动器 C 包含逻辑数据块 9 到 12。卷的总容量即为这三个分片的组合容量。如果每个分片为 2 GB，则卷的总容量为 6 GB。

图 8-2 RAID-0（串联）卷示例



# RAID-0（串联条带）卷

RAID-0（串联条带）卷是指通过添加附加组件（条带）进行扩展的条带。

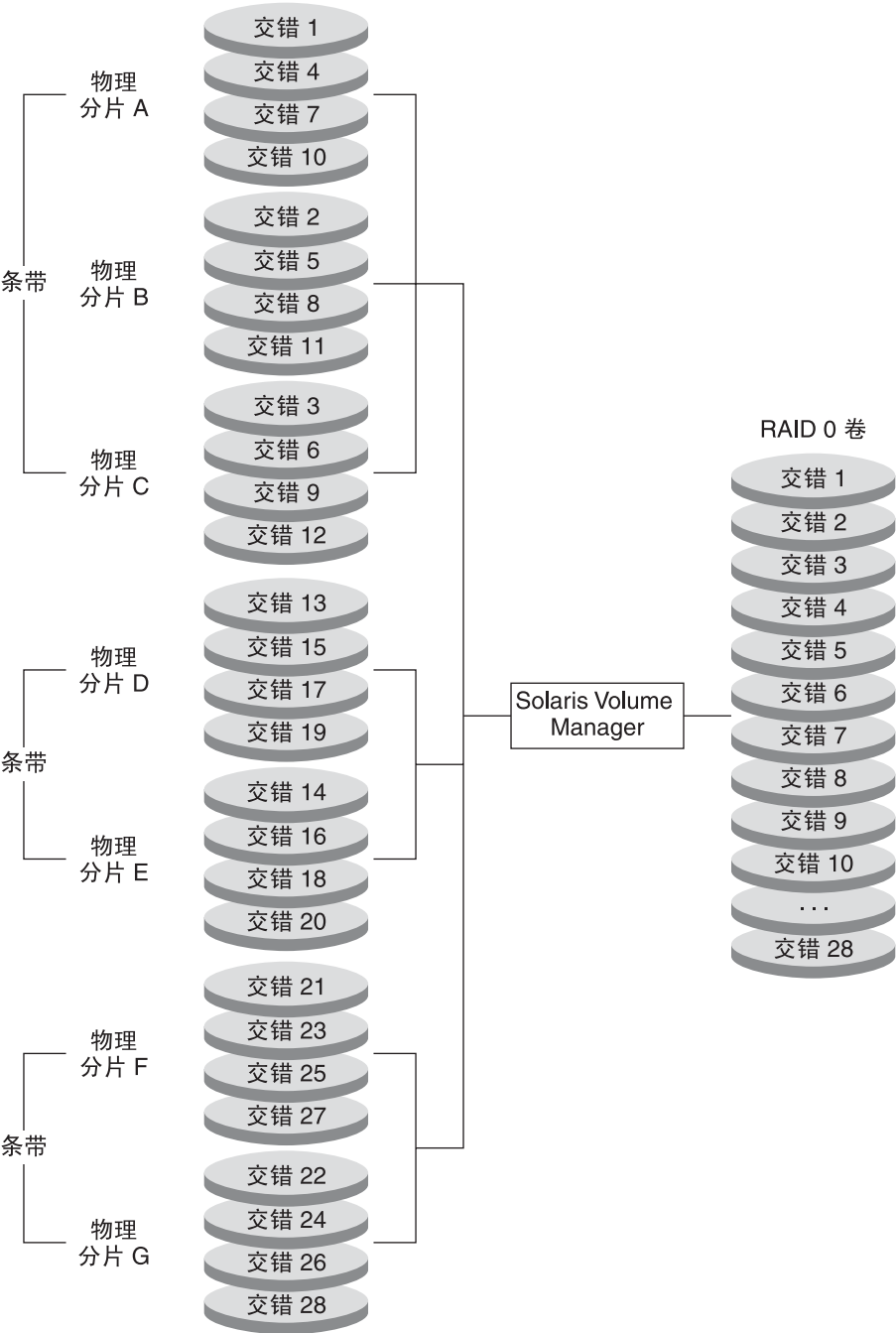
要在条带级别上为串联条带卷设置交错值，请使用 Solaris Management Console 中增强的存储工具或 `metattach -i` 命令。串联条带卷中的每个条带都可以具有自己的交错值。新建串联条带卷时，如果不对特定条带指定交错值，则该条带将沿用添加到该卷中的前一个条带的交错值。

## 示例—RAID-0（串联条带）卷

图 8-3 显示了一个由三个条带串联而成的串联条带卷。

第一个条带包含三个分片，即，分片 A 到 C，交错值为 16 KB。第二个条带包含两个分片，即，分片 D 和 E，并使用交错值 32 KB。最后一个条带包含两个分片，即，分片 F 和 G。由于没有为第三个条带指定任何交错值，因此，该条带将沿用在它之前添加的条带的值，此处为 32 KB。系统会将顺序数据块添加到第一个条带中，直到该条带空间已满为止。然后，会将数据块添加到第二个条带中。当此条带空间已满后，会将数据块添加到第三个条带中。在每个条带中，数据块会根据指定的交错值交错分布。

图 8-3 RAID-0（串联条带）卷示例



# 创建 RAID-0 卷的背景信息

## RAID-0 卷要求

使用 RAID-0 卷时，请考虑以下要求：

- 使用分别位于不同控制器上的组件，以增加可同时执行的读取和写入操作的数量。
- 请勿基于现有文件系统或数据创建条带，这样做会销毁数据。请改用串联。（可以使用现有数据创建条带，但必须将该数据转储并恢复到卷中。）
- 使用相同大小的磁盘组件来创建条带。如果对不同大小的组件进行条带化，则会浪费磁盘空间。
- 设置条带的交错值，使其与系统或应用程序发出的 I/O 请求更好地匹配。
- 由于条带或串联不包含复制数据，因此，如果此类卷发生组件故障，则必须替换该组件，重新创建条带或串联，然后从备份中恢复数据。
- 重新创建条带或串联时，请使用至少与发生故障的组件大小相等的替换组件。

## RAID-0 卷的原则

- 与条带化相比，串联使用的 CPU 周期更少，因此，非常适用于小型随机 I/O 以及均匀分布的 I/O。
- 如有可能，请将条带或串联的组件分布在不同的控制器和总线中。使用位于不同控制器上的条带，可以增加可同时执行的读写操作数。
- 如果在发生故障的控制器上定义了一个条带，而系统上还有另一个可用的控制器，则可以通过将磁盘移动到这一新控制器并重新定义条带来将该条带“移动”到该控制器上。
- 条带数量：考虑条带化的另一因素是，首先确定性能要求。例如，您可能需要为选定应用程序提供 10.4 MB/秒的性能，而每个磁盘的速度可能大约为 4 MB/秒。根据以下公式可以确定需要在多少个磁盘轴上进行条带化：

$$10.4 \text{ Mbyte/sec} / 4 \text{ Mbyte/sec} = 2.6$$

由此可知，需要三个能够并行执行 I/O 操作的磁盘。

## 方案—RAID-0 卷

RAID-0 卷为创建较复杂的存储配置或生成镜像提供了基础生成块。以下示例（即，在[第 5 章，配置和使用 Solaris Volume Manager（方案）](#)中所述的方案）说明了 RAID-0 卷是如何提供更大的存储空间并构建现有文件系统（包括根 (/) 的镜像的）。

该方案中的样例系统包含一组较小 (9 GB) 的磁盘，但某些应用程序可能需要更大的存储空间。要创建更大的空间（并提高性能），可以创建一个跨多个磁盘的条带。例如，以下磁盘（c1t1d0、c1t2d0、c1t3d0、c2t1d0、c2t2d0 和 c2t3d0）中的每一个磁盘都可以格式化为一个跨整个磁盘的分片 0。然后，由一个控制器中所有三个磁盘构成的条带可提供大约 27 GB 的存储空间，并可加快访问速度。第二个控制器中的第二个条带可用作冗余，如[第 11 章，RAID-1（镜像）卷（任务）](#)中所述，具体信息请参见[第 93 页](#)中的“方案—RAID-1 卷（镜像）”。

## RAID-0（条带和串联）卷（任务）

---

本章包含有关与 RAID-0 卷相关的任务的信息。有关相关概念的信息，请参见第 8 章，[RAID-0（条带和串联）卷（概述）](#)。

### RAID-0 卷（任务列表）

以下任务列表列出了管理 Solaris Volume Manager RAID-0 卷所需的过程。

任务	说明	参考
创建 RAID-0（条带）卷	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建新卷。	<a href="#">第 78 页中的“如何创建 RAID-0（条带）卷”</a>
创建 RAID-0（串联）卷	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建新卷。	<a href="#">第 79 页中的“如何创建 RAID-0（串联）卷”</a>
扩展存储空间	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令扩展现有的文件系统。	<a href="#">第 80 页中的“如何扩展现有数据的存储容量”</a>
扩展现有的 RAID-0 卷	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令扩展现有的卷。	<a href="#">第 82 页中的“如何扩展现有的 RAID-0 卷”</a>
删除 RAID-0 卷	使用 Solaris Volume Manager GUI 或 <code>metaclear</code> 命令删除卷。	<a href="#">第 83 页中的“如何删除 RAID-0 卷”</a>

### 创建 RAID-0（条带）卷



**注意** - 请勿基于现有文件系统或数据创建条带，这样做会销毁数据。要从现有数据创建条带，必须先备份数据，创建条带卷，然后将数据恢复到该卷上。

---



**注意** – 如果您希望运行具有 32 位内核的 Solaris 软件，请勿创建大于 1 TB 的卷。此外，如果希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，也请勿创建大于 1 TB 的卷。有关 Solaris Volume Manager 中的大型卷支持的更多信息，请参见第 41 页中的“Solaris Volume Manager 中的多 TB 支持概述”。

## ▼ 如何创建 RAID-0 (条带) 卷

**开始之前** 请查看第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 75 页中的“创建 RAID-0 卷的背景信息”。

### ● 要创建条带卷，请使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes" (卷) 节点。选择 "Action" (操作) ⇒ "Create Volume" (创建卷)，然后遵循向导中的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes
    components-per-stripe
    component-names
    [ -i interlace]
```

<i>volume-name</i>	指定要创建的卷的名称。有关命名卷的信息，请参见第 38 页中的“卷名称”。
<i>number-of-stripes</i>	指定要创建的条带数。
<i>components-per-stripe</i>	指定每个条带应包含的组件数。
<i>component-names</i>	指定所使用的组件的名称。如果使用多个组件，请使用空格分隔各个组件。
<i>-i interlace</i>	指定要用于条带的交错宽度。交错宽度是一个数值，后跟 "k" (表示千字节)、"m" (表示兆字节) 或 "b" (表示块)。指定的交错值不能小于 16 个块，也不能大于 100 MB。缺省的交错宽度为 16 KB。

有关更多信息，请参见以下示例和 `metainit(1M)` 手册页。

### 示例 9-1 创建包含三个分片的 RAID-0 (条带) 卷

```
# metainit d20 1 3 c0t1d0s2 c0t2d0s2 c0t3d0s2
d20: Concat/Stripe is setup
```

本示例显示了由单个条带 (由数字 1 指示) 组成的条带 d20。该条带由三个分片 (由数字 3 指示) 组成。由于未指定交错值，条带将使用缺省的 16 KB。系统确认了卷已设置。

## 示例 9-2 创建由两个分片组成且交错值为 32 KB 的 RAID-0 ( 条带 ) 卷

```
# metainit d10 1 2 c0t1d0s2 c0t2d0s2 -i 32k
d10: Concat/Stripe is setup
```

本示例显示了由单个条带（由数字 1 指示）组成的条带 d10。该条带由两个分片（由数字 2 指示）组成。-i 选项将交错值设置为 32 KB。（交错值不能小于 8 KB 或大于 100 MB。）系统会验证卷是否已设置。

**另请参见** 要针对某个文件系统准备新创建的条带，请参见《[System Administration Guide: Devices and File Systems](#)》中的第 15 章“[Creating and Mounting File Systems \(Tasks\)](#)”。某些应用程序（例如数据库）不使用文件系统，而是使用原始设备。这些应用程序必须有自己访问原始设备的方法。

## 创建 RAID-0 ( 串联 ) 卷

### ▼ 如何创建 RAID-0 ( 串联 ) 卷



**注意** - 如果希望运行具有 32 位内核的 Solaris 软件，请勿创建大于 1 TB 的卷。此外，如果希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，也请勿创建大于 1 TB 的卷。有关 Solaris Volume Manager 中的多 TB 卷的更多信息，请参见第 41 页中的“[Solaris Volume Manager 中的多 TB 支持概述](#)”。

**开始之前** 请查看第 41 页中的“[创建 Solaris Volume Manager 组件的先决条件](#)”和第 75 页中的“[创建 RAID-0 卷的背景信息](#)”。

- **要创建串联卷，请使用以下方法之一：**

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes" ( 卷 ) 节点。选择 "Action" ( 操作 ) ⇒ "Create Volume" ( 创建卷 )，然后遵循向导中的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 metainit 命令：

```
# metainit volume-name number-of-stripes
components-per-stripe
component-names
```

*volume-name*                      指定要创建的卷的名称。

*number-of-stripes*                指定要创建的条带数。

*components-per-concatenation*    指定每个串联应包含的组件数。

*component-names*

指定所使用的组件的名称。如果使用多个组件，请使用空格分隔各个组件。

有关详细信息，请参见以下示例和 [metainit\(1M\)](#) 手册页。

### 示例 9-3 创建包含一个分片的串联

```
# metainit d25 1 1 c0t1d0s2
d25: Concat/Stripe is setup
```

本示例说明如何创建串联 d25。此串联由一个条带组成（由第一个数字 1 指示），该条带由单个分片组成（由分片前的第二个数字 1 指示）。系统会验证卷是否已设置。

本示例显示了一个可以安全地封装现有数据的串联。

### 示例 9-4 创建包含四个分片的串联

```
# metainit d40 4 1 c0t1d0s2 1 c0t2d0s2 1 c0t2d0s3 1 c0t2d1s3
d40: Concat/Stripe is setup
```

本示例说明如何创建串联 d40。该串联由四个条带组成（由数字 4 指示），每个条带由单个分片组成（由每个分片前的数字 1 指示）。系统会验证卷是否已设置。

另请参见 要针对某个文件系统准备新创建的串联，请参见 [《System Administration Guide: Devices and File Systems》](#) 中的第 15 章 “Creating and Mounting File Systems (Tasks)”。

## 扩展存储容量

要向文件系统添加存储容量，请创建串联卷。要向现有条带添加存储容量，请创建串联的条带卷。

### ▼ 如何扩展现有数据的存储容量



**注意** – 如果您希望运行具有 32 位内核的 Solaris 软件，请勿创建大于 1 TB 的卷。此外，如果希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，也请勿创建大于 1 TB 的卷。有关 Solaris Volume Manager 中的多 TB 卷支持的更多信息，请参见第 41 页中的 [“Solaris Volume Manager 中的多 TB 支持概述”](#)。

**开始之前** 请查看第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 75 页中的“创建 RAID-0 卷的背景信息”。

## 1 取消挂载文件系统。

```
# umount /filesystem
```

## 2 要创建串联，请使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择 "Action"（操作）⇒ "Create Volume"（创建卷），然后遵循向导中的说明进行操作。有关更多信息，请参见联机帮助。

- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes
      components-per-stripe
      component-names
```

`volume-name`                      指定要创建的卷的名称。

`number-of-stripes`                指定要创建的条带数。

`components-per-stripe`           指定每个条带应包含的组件数。

`component-names`                指定所使用的组件的名称。如果使用多个组件，请使用空格分隔各个组件。

有关更多信息，请参见 [metainit\(1M\)](#) 手册页。

## 3 编辑 `/etc/vfstab` 文件，以便文件系统可以引用串联的名称。

## 4 重新挂载文件系统。

```
# mount /filesystem
```

### 示例 9-5 通过创建串联来扩展文件系统

```
# umount /docs
# metainit d25 2 1 c0t1d0s2 1 c0t2d0s2
d25: Concat/Stripe is setup
      (Edit the /etc/vfstab file so that the file system references the volume d25 instead of slice c0t1d0s2)
# mount /docs
```

本示例说明如何从两个分片 `/dev/dsk/c0t1d0s2`（该分片包含挂载在 `/docs` 上的文件系统）和 `/dev/dsk/c0t2d0s2` 创建串联 `d25`。必须首先取消挂载文件系统。请注意，`metainit` 命令中的第一个分片必须是包含该文件系统的分片。否则，将会损坏数据。

接下来，应更改 `/etc/vfstab` 文件中与该文件系统相对应的条目（第一次请输入该条目），以便引用该串联。例如，最初 `/etc/vfstab` 文件中显示以下行：

```
/dev/dsk/c0t1d0s2 /dev/rdisk/c0t1d0s2 /docs ufs 2 yes -
```

应将该行更改为以下内容：

```
/dev/md/dsk/d25 /dev/md/rdisk/d25 /docs ufs 2 yes -
```

最后，将重新挂载该文件系统。

**另请参见** 对于 UFS 文件系统，请针对该串联运行 `growfs` 命令。请参见第 207 页中的“如何扩展文件系统”。

某些应用程序（例如数据库）不使用文件系统，应用程序（例如数据库）可使用原始串联，但必须有自己识别串联或扩展添加的空间的方法。

## ▼ 如何扩展现有的 RAID-0 卷

通过串联的条带，可以扩展现有条带。例如，如果某个条带的存储容量已经用完，可将其转换为串联的条带。这样做可以扩展存储容量而无需备份和恢复数据。

此过程假设要向现有条带再添加一个条带。



**注意** – 如果您希望运行具有 32 位内核的 Solaris 软件，请勿创建大于 1 TB 的卷。此外，如果希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，也请勿创建大于 1 TB 的卷。有关 Solaris Volume Manager 中的多 TB 支持的更多信息，请参见第 41 页中的“Solaris Volume Manager 中的多 TB 支持概述”。

**开始之前** 请查看第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 75 页中的“创建 RAID-0 卷的背景信息”。

### ● 要创建串联条带，请使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择 "Action"（操作）⇒ "Create Volume"（创建卷），然后遵循向导中的说明进行操作。有关更多信息，请参见联机帮助。
- 要从命令行串联现有条带，请使用以下格式的 `metattach` 命令：

```
# metattach volume-name component-names
```

*volume-name*            指定要扩展的卷的名称。

*component-names*        指定所使用的组件的名称。如果使用多个组件，请使用空格分隔各个组件。

有关更多信息，请参见 `metattach(1M)` 手册页。

**示例 9-6 通过附加单个分片创建串联的条带**

```
# metattach d2 c1t2d0s2
d2: components are attached
```

本示例说明如何向现有条带 d2 附加分片。系统确认了该分片已附加。

**示例 9-7 通过添加多个分片创建串联的条带**

```
# metattach d25 c1t2d0s2 c1t2d1s2 c1t2d3s2
d25: components are attached
```

本示例以现有的三向条带 d25 为例，将另一个三向条带与其串联。由于没有为附加的分片提供交错值，条带将继承为 d25 配置的交错值。系统会验证卷是否已设置。

**另请参见** 对于 UFS 文件系统，请针对该卷运行 growfs 命令。请参见第 207 页中的“如何扩展文件系统”。

某些应用程序（例如数据库）不使用文件系统，应用程序（如数据库）可使用原始卷，但必须有自己识别串联或扩展添加的空间的方法。

要针对某个文件系统准备新创建的串联条带，请参见《System Administration Guide: Devices and File Systems》中的第 15 章“Creating and Mounting File Systems (Tasks)”。

## 删除 RAID-0 卷

### ▼ 如何删除 RAID-0 卷

- 1 确保您拥有所有数据的当前备份并且具有 root 用户权限。
- 2 确保您不再需要该卷。  
如果删除某个条带或串联并重复使用删除的卷中的分片，则将从系统中删除该卷上的所有数据。
- 3 取消挂载文件系统（如果需要）。  

```
# umount /filesystem
```
- 4 要删除卷，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择 "Edit"（编辑）⇒ "Delete"（删除），然后按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。

- 使用以下格式的 `metaclear` 命令删除卷：

```
metaclear volume-name
```

有关更多信息，请参见以下示例和 [metaclear\(1M\)](#) 手册页。

### 示例 9-8 删除串联

```
# umount d8
# metaclear d8
d8: Concat/Stripe is cleared
    (Edit the /etc/vfstab file)
```

本示例说明如何删除串联 `d8`，该串联还包含一个挂载的文件系统。必须先卸载文件系统，才可以删除卷。系统显示了一条确认消息，说明串联已删除。如果 `/etc/vfstab` 文件中存在与该卷相对应的条目，请删除该条目。如果要求在不存在的卷上挂载文件系统，系统会出现混乱，您一定不希望出现这种情况。

## RAID-1（镜像）卷（概述）

---

本章说明与镜像和子镜像相关的基本 Solaris Volume Manager 概念。有关执行相关任务的信息，请参见第 11 章，[RAID-1（镜像）卷（任务）](#)。

本章包含以下信息：

- 第 85 页中的“[RAID-1（镜像）卷概述](#)”
- 第 88 页中的“[RAID-1 卷（镜像）重新同步](#)”
- 第 89 页中的“[创建和维护 RAID-1 卷](#)”
- 第 93 页中的“[引导至单用户模式对 RAID-1 卷产生的影响](#)”
- 第 93 页中的“[方案—RAID-1 卷（镜像）](#)”

## RAID-1（镜像）卷概述

RAID-1 卷（即**镜像**）是指在 RAID-0（条带或串联）卷中保留相同数据副本的卷。已镜像的 RAID-0 卷称为**子镜像**。镜像要求购买磁盘。所需的磁盘空间至少要为必须镜像的数据量的两倍。由于 Solaris Volume Manager 必须对所有子镜像写入数据，因此镜像还会增加将要写入的请求写入磁盘所需的时间量。

配置镜像后，可以像使用物理分片一样使用此镜像。

您可以镜像任何文件系统，包括现有文件系统。这些文件系统包括根 (/)、swap 和 /usr。也可以将镜像用于任何应用程序，如数据库。

---

**提示** – 将 Solaris Volume Manager 的热备用功能与镜像结合使用，可以使数据始终安全且可用。有关热备件的信息，请参见第 16 章，[热备用池（概述）](#)和第 17 章，[热备用池（任务）](#)。

---

## 子镜像概述

一个镜像由一个或多个 RAID-0 卷（条带或串联）组成，这些卷称为子镜像。

一个镜像最多可以包含四个子镜像。但是，双向镜像通常就可以为大多数应用程序提供足够的数据冗余，并且在磁盘驱动器方面的开销较小。第三个子镜像使您可以在其中一个子镜像处于脱机状态下进行联机备份，同时又能保持数据的冗余性。

如果使某个子镜像“脱机”，则镜像将停止读取和写入该子镜像。此时，可以访问子镜像本身（例如，在执行备份时）。但是，该子镜像处于只读状态。在某个子镜像处于脱机状态时，Solaris Volume Manager 会跟踪镜像的所有写入。在子镜像恢复联机后，只有在该子镜像处于脱机状态时写入的那部分镜像（**重新同步区域**）才会进行重新同步。此外，还可以使子镜像脱机来对出错的物理设备进行故障排除或修复。

可以随时在镜像中附加或分离子镜像，但必须始终至少附加一个子镜像。

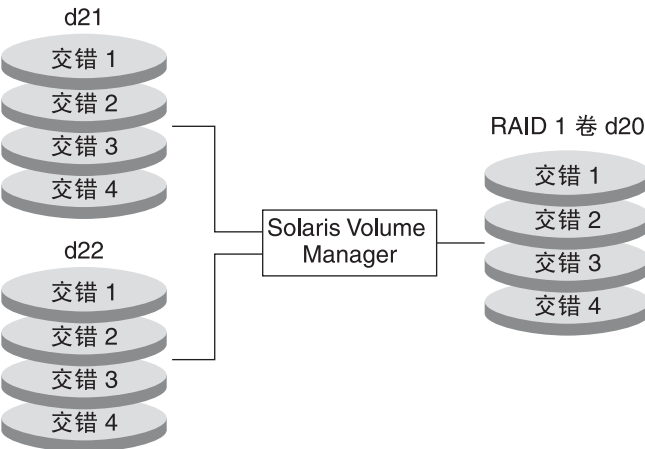
通常，在创建镜像时只具有一个子镜像。之后，您可以在创建镜像后附加第二个子镜像。

## 方案一 RAID-1（镜像）卷

图 10-1 显示了一个镜像 d20。此镜像由两个卷（子镜像）d21 和 d22 组成。

Solaris Volume Manager 将在多个物理磁盘上创建数据的副本，并为应用程序提供一个虚拟磁盘，在此示例中为 d20。所有磁盘写入都会被复制。而磁盘读取则通过一个底层子镜像进行。镜像 d20 的总容量是最小子镜像的大小（假设这些子镜像大小不等）。

图 10-1 RAID-1（镜像）示例



## 提供 RAID-1+0 和 RAID-0+1

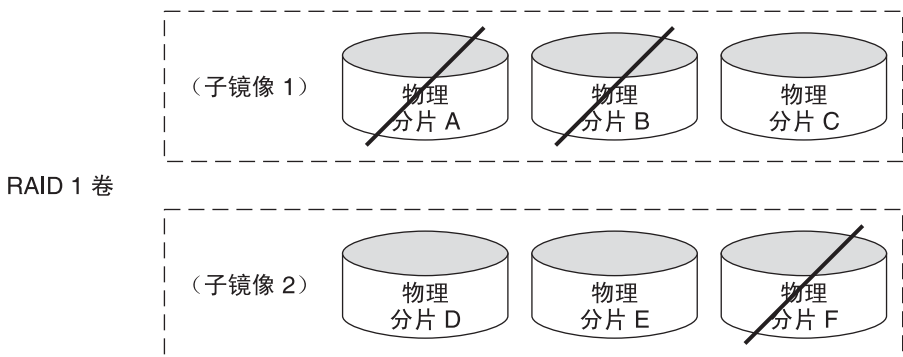
Solaris Volume Manager 既支持 RAID-1+0 冗余，也支持 RAID-0+1 冗余。RAID-1+0 冗余表示要条带化的镜像配置。而 RAID-0+1 冗余则表示要镜像的条带配置。在 Solaris Volume Manager 界面中，所有 RAID-1 设备看起来都是完全遵从 RAID-0+1 配置的。但是，Solaris Volume Manager 会识别底层组件并分别对其进行镜像（如果可能）。

注 - Solaris Volume Manager 不能始终提供 RAID-1+0 功能。但是，如果两个子镜像彼此完全相同，并且由磁盘分片（而不是软分区）组成，则可以提供 RAID-1+0。

假设要实施一个 RAID-0+1，其中包括一个双向镜像，该镜像包括三个条带化分片。如果不使用 Solaris Volume Manager，则只要一个分片出现故障，就可能会使镜像的一端出现故障。假设没有使用任何热备件，则如果第二个分片出现故障，该镜像就会出现故障。如果使用 Solaris Volume Manager，则最多可能会使三个分片出现故障，而不会使镜像出现故障。由于这三个条带化分片中的每一个分片都已分别镜像到该镜像的另一端上的对应分片，因此，该镜像不会出现故障。

图 10-2 显示了 RAID-1 卷是如何发生分片丢失并通过实施 RAID-1+0 来防止数据丢失的。

图 10-2 RAID-1+0 示例



RAID-1 卷由两个子镜像组成。而每个子镜像又由三个具有相同交错值的相同物理磁盘组成。允许三个磁盘（A、B 和 F）出现故障。而该镜像的整个逻辑块范围仍包含在至少一个正常运行的磁盘上。卷中的所有数据均可用。

但是，如果磁盘 A 和磁盘 D 出现故障，则该镜像的一部分数据将在任何磁盘上不再可用。访问这些逻辑块将失败。但是，仍然能够成功访问该镜像中具有可用数据的部分。在这种情况下，该镜像就像是一个出现坏块的磁盘。受损的部分不再可用，但其余部分仍然可用。

## RAID-1 卷（镜像）重新同步

RAID-1 卷（镜像）重新同步是指在发生以下情况之一后将数据从一个子镜像复制到另一个子镜像的过程：

- 子镜像出现故障
- 系统发生故障
- 子镜像脱机之后又重新联机
- 已添加新的子镜像

执行重新同步时，用户仍然可以读取和写入镜像。

镜像重新同步可通过保证所有子镜像具有相同数据来确保正确执行镜像操作（正在执行的写入操作除外）。

---

注 - 不应跳过镜像重新同步。您无需手动启动镜像重新同步。此过程会自动执行。

---

## 完全重新同步

在向镜像中附加（添加）新的子镜像后，该镜像中另一个子镜像的所有数据将自动写入新附加的子镜像。完成镜像重新同步后，即可读取新的子镜像。子镜像会始终附加到镜像，直到将其分离为止。

如果正在执行重新同步时系统崩溃，则在系统完成重新引导后会重新启动重新同步。

## 优化重新同步

在系统出现故障后进行重新引导期间或在已脱机的子镜像恢复联机后，Solaris Volume Manager 将执行**优化镜像重新同步**。元盘驱动程序将跟踪子镜像区域。通过此功能，元盘驱动程序可以知道在出现故障后哪些子镜像区域可能不同步。优化镜像重新同步只会对不同步的区域执行。您可以指定在重新引导期间对镜像进行重新同步的顺序。可以通过将子镜像传送号设置为零来忽略镜像重新同步。有关与更改传送号相关的任务，请参见[示例 11-16](#)。



---

注意 - 只能对以只读形式挂载的镜像使用值为零的传送号。

---

## 部分重新同步

在替换子镜像中的分片后，Solaris Volume Manager 会对数据执行**部分镜像重新同步**。Solaris Volume Manager 会将数据从另一个子镜像中正常运行的其余分片复制到替换的分片。

# 创建和维护 RAID-1 卷

本节提供的原则可以帮助您创建镜像。同时，还为所创建的镜像提供了一些性能原则。

## RAID-1 卷的配置原则

- 在创建镜像之前，请先创建构成该镜像的 RAID-0（条带或串联）卷。
- 创建镜像时，请先创建单向镜像，然后附加第二个子镜像。此策略会启动重新同步操作。它还可以确保数据不受损。您还可以创建单向镜像并在日后用作双向镜像或多向镜像。
- 只需使用一个命令即可从单向镜像创建双向镜像、三向镜像或四向镜像。您可以通过使用一个命令创建所有子镜像来加快创建过程。只有当您不需要镜像现有数据，以及不在意销毁所有子镜像上的数据时，才使用此过程。
- 可以从在分片上构建的现有文件系统创建 RAID-1 卷。主 RAID-0 卷（子镜像）只能包含一个分片。如果您要镜像根文件系统或其他对系统很重要的文件系统，则所有子镜像都必须只包含一个分片。
- 使用 `swap -l` 命令可检查所有 `swap` 设备。指定为 `swap` 的每个分片都必须单独进行镜像。
- Solaris Management Console 中增强的存储工具不支持取消镜像根（/）、/opt、/usr 或 `swap`。实际上，此工具不支持取消镜像在系统运行时无法卸载的任何文件系统。对于这些文件系统，请改用命令行过程。
- 使用大小相同的子镜像。不同大小的子镜像会导致部分磁盘空间不能使用。
- 在镜像中仅使用配置相似的子镜像。尤其是，在创建镜像时，如果某一子镜像不带标签，则无法附加包含磁盘标签的任何子镜像。
- 假设在一个镜像文件系统中，附加的第一个子镜像不是从柱面 0 开始的，则您附加的所有其他子镜像也不能从柱面 0 开始。如果尝试附加从柱面 0 开始的子镜像，则会显示以下错误消息：

```
can't attach labeled submirror to an unlabeled mirror
```

要在特定镜像中使用的所有子镜像要么都必须从柱面 0 开始，要么都不能从柱面 0 开始。

所有子镜像的起始柱面不必相同。但是，所有子镜像要么都包括柱面 0，要么都不包括柱面 0。

- 可以通过在创建镜像之前添加更多状态数据库副本来提高镜像的性能。一般情况下，请为添加到系统中的每个镜像再添加两个副本。Solaris Volume Manager 使用这两个额外添加的副本来存储脏区日志 (Dirty Region Log, DRL)，以便提供优化重新同步。通过提供足够数量的副本，可以最大限度地降低 I/O 对 RAID-1 卷性能的影响。在相同的磁盘或控制器上使用至少两个副本作为副本记录的镜像，还可以帮助提高整体性能。

- 仅直接挂载镜像设备。不要尝试直接挂载子镜像，除非子镜像处于脱机状态且挂载为只读。不要挂载子镜像中的分片。此过程会销毁数据并使系统崩溃。

## RAID-1 卷的性能原则

- 使不同子镜像的分片位于不同的磁盘和控制器中。如果同一镜像的两个和多个子镜像的分片位于同一个磁盘上，则对数据的保护作用将大大降低。同样，将子镜像分布在不同的控制器中，因为控制器及相关电缆比磁盘更容易发生故障。此做法还可以提高镜像性能。
- 对一个镜像使用同一类型的磁盘和控制器。特别是对于旧的 SCSI 存储设备，不同模型或品牌的磁盘或控制器的性能差别很大。如果在一个镜像中使用具有不同性能级别的磁盘和控制器，则性能会显著降低。
- 镜像可能会提高读取性能，但始终会降低写入性能。镜像只有在线程式或异步 I/O 操作中才会提高读取性能。从卷进行读取的单个线程不会提高性能。
- 可以对镜像读取策略进行实验以提高性能。例如，缺省读取模式是在磁盘之间以循环方式交替读取。此策略为缺省策略，因为循环方式最适合 UFS 多用户和多处理器活动。

在某些情况下，`geometric` 读取选项可通过最大限度地减少磁头运动并缩短访问时间来提高性能。该选项在以下情况下效果最佳：

- 每个磁盘只有一个分片
- 一次只有一个进程使用分片或文件系统
- I/O 模式的顺序性很强，或者需要读取所有访问
- 您可以在不中断服务的情况下向镜像中附加子镜像。您可以向镜像中附加子镜像以创建双向、三向和四向镜像。
- 使某个子镜像脱机，可以防止镜像从该子镜像读取和向该子镜像写入。但是，子镜像与镜像的逻辑关联将会保留下来。在子镜像处于脱机状态时，Solaris Volume Manager 会跟踪镜像的所有写入。当子镜像恢复联机后，写入内容将写入该子镜像。通过执行优化重新同步，Solaris Volume Manager 只需重新同步已更改的数据即可，而无需重新同步整个子镜像。分离子镜像后，该子镜像与镜像之间的逻辑关联将被断开。通常，可以使子镜像脱机来执行维护操作。要删除子镜像，可以将该子镜像分离。

## 关于 RAID-1 卷选项

以下选项可以优化镜像性能：

- 镜像读取策略
- 镜像写入策略
- 重新同步镜像的顺序（传送号）

可以在首次创建镜像时定义镜像选项。也可以在设置完镜像后，在该镜像运行时更改镜像选项。有关与更改这些选项相关的任务，请参见第 121 页中的“如何更改 RAID-1 卷选项”。

RAID-1 卷读写策略

Solaris Volume Manager 可以为 RAID-1 卷配置不同的读写策略。正确设置读写策略可以提高给定配置的性能。

表 10-1 RAID-1 卷读取策略

读取策略	说明
循环（缺省）	尝试在子镜像间平衡负载。所有读取均按循环顺序（依次）从镜像中的所有子镜像进行。
几何	可以根据逻辑磁盘块地址在子镜像之间划分读取范围。例如，如果使用双向子镜像，则镜像上的磁盘空间将划分为两个大小相同的逻辑地址范围。从一个子镜像读取的内容仅限于其中一个逻辑范围。而从另一个子镜像读取的内容则限于另一个逻辑范围。几何读取策略可以有效地缩短读取所需的查找时间。该读取策略是否能够提高性能取决于系统 I/O 负载以及应用程序访问模式。
第一	将所有读取定向到第一个子镜像。只有当构成第一个子镜像的设备比第二个子镜像的设备速度明显快时，才应使用此策略。

表 10-2 RAID-1 卷写入策略

写入策略	说明
并行（缺省）	对镜像执行的写入同时复制并分发到所有子镜像。
串行	按顺序对子镜像执行写入，即，第一个子镜像写入完成后，再开始第二个子镜像写入。根据此策略，必须先完成对一个子镜像的写入后，再启动对下一个子镜像的写入。在某一子镜像因电源故障而无法读取时可使用此策略。

传送号

传送号（介于 0-9 之间的数字）用于确定某一特定镜像在系统重新引导期间进行重新同步的顺序。缺省传送号为 1。传送号较低的镜像会先进行重新同步。如果使用零，则会跳过镜像重新同步。只能对以只读形式挂载的镜像使用值为零的传送号。具有相同传送号的镜像将同时进行重新同步。

了解子镜像状态以确定维护操作

Solaris Volume Manager 的 `metastat` 命令可报告有关 RAID 1 卷和子镜像的状态信息。此状态信息有助于确定是否需要 RAID-1 卷执行维护操作。下表说明了对 RAID-1 卷运行 `metastat` 命令时所显示的子镜像状态。

表 10-3 子镜像状态

状态	含义
Okay	表明子镜像未出现错误且运行正常。
Resyncing	表明子镜像正在进行重新同步。出现错误但已更正、子镜像刚刚恢复联机或添加了新的子镜像。
Needs Maintenance	表明子镜像中的分片遇到 I/O 错误或打开错误。在子镜像中对此分片进行的所有读取和写入均已中断。

此外，对于子镜像中的每个分片，`metastat` 命令还会显示以下信息：

Device	指示条带中分片的设备名称
Start Block	指示分片的起始块
Dbase	指示分片是否包含状态数据库副本
State	指示分片的状态
Hot Spare	指示某一分片正用作出现故障的分片的热备件。

该子镜像状态只会提供有关子镜像状态的一般信息。在对镜像错误进行故障排除时，需要查看的最重要信息可能是分片状态。如果子镜像报告 "Needs Maintenance"（需要维护）状态，则必须参考分片状态以获取更多信息。

根据分片是处于 "Maintenance"（维护）状态还是处于 "Last Erred"（最近出错）状态，您可以执行不同的恢复操作。如果分片均处于 "Maintenance"（维护）状态，则可以按任意顺序修复这些分片。如果一些分片处于 "Maintenance"（维护）状态，另一些分片处于 "Last Erred"（最近出错）状态，则必须先修复处于 "Maintenance"（维护）状态的分片。修复处于 "Maintenance"（维护）状态的分片后，再修复处于 "Last Erred"（最近出错）状态的分片。有关更多信息，请参见第 208 页中的[“替换和启用 RAID-1 和 RAID-5 卷中的组件概述”](#)。

下表说明了子镜像的分片状态以及可能采取的操作。

表 10-4 子镜像分片状态

状态	含义	操作
Okay	表明分片未出现错误且运行正常。	无。
Resyncing	表明分片正在进行重新同步。出现错误但已更正、子镜像刚刚恢复联机或添加了新的子镜像。	如果需要，请监视子镜像状态，直到完成重新同步为止。

表 10-4 子镜像分片状态 （续）

状态	含义	操作
Maintenance	表明分片遇到 I/O 错误或打开错误。对此组件进行的所有读取和写入均已中断。	启用或替换出现故障的分片。请参见第 118 页中的“如何启用子镜像中的分片”或第 123 页中的“如何替换子镜像中的分片”。 <code>metastat</code> 命令将显示一条 <code>invoke</code> 恢复消息，其中包含需要使用 <code>metareplace</code> 命令执行的相应操作。您也可以使用 <code>metareplace -e</code> 命令。
Last Erred	表明分片遇到 I/O 错误或打开错误。但是，数据不会复制到其他位置，因为另一个分片出现故障。I/O 仍在此分片上执行。如果出现 I/O 错误，镜像 I/O 将失败。	首先，启用或替换处于 "Maintenance"（维护）状态的分片。请参见第 118 页中的“如何启用子镜像中的分片”或第 123 页中的“如何替换子镜像中的分片”。通常，此错误会导致某些数据丢失，因此，请在修复后对镜像进行验证。对于文件系统，请使用 <code>fsck</code> 命令，然后检查数据。应用程序或数据库必须具有自己的设备验证方法。

## 引导至单用户模式对 RAID-1 卷产生的影响

有时，您可能需要使用 `boot -s` 命令将包含根 (/)、`/usr` 和 `swap`（即所谓的“引导”文件系统）的镜像的系统引导至单用户模式。在这种情况下，如果使用 `metastat` 命令查看这些镜像以及该系统中所有可能存在的镜像，则这些镜像将显示为 "Needing Maintenance"（需要维护）状态。此外，如果对这些分片进行写入，`metastat` 命令将显示镜像上的脏区有所增长。

此情形似乎存在潜在危险。但是，在将系统引导至单用户模式时，`metasync -r` 命令（通常会在引导期间运行以重新同步镜像）将被中断。系统重新引导后，`metasync -r` 命令将运行并重新同步所有镜像。

如果此情况会造成较大影响，您可以手动运行 `metasync -r` 命令。

## 方案—RAID-1 卷（镜像）

RAID-1 卷提供了一种构建冗余卷的方式。这样，当其中一个底层 RAID-0 卷出现部分或完全故障时，文件系统不会发生数据丢失或访问中断的情况。以下示例（即，在第 5 章，配置和使用 Solaris Volume Manager（方案）中说明并在第 76 页中的“方案—RAID-0 卷”中继续说明的方案）说明了 RAID-1 卷是如何提供冗余存储的。

如第 76 页中的“方案—RAID-0 卷”中所述，样例系统具有两个 RAID-0 卷。每个卷的大小约为 27 GB，并且跨三个磁盘。通过创建 RAID-1 卷来镜像这两个 RAID-0 卷，可以利用完全冗余存储空间来提供具有复原能力的数据存储。

在此 RAID-1 卷中，任何一个磁盘控制器出现故障都不会中断对卷的访问。同样，在不中断访问的情况下，最多可允许三个独立磁盘出现故障。

要提供额外的保护，以避免出现可能中断访问的问题，请使用热备件，如第 16 章，[热备用池（概述）](#)中所述。具体来说，请参见第 160 页中的“热备件的工作原理”。

## RAID-1（镜像）卷（任务）

---

本章介绍如何执行与 RAID-1 卷相关的 Solaris Volume Manager 任务。有关相关概念的信息，请参见第 10 章，[RAID-1（镜像）卷（概述）](#)。

### RAID-1 卷（任务列表）

以下任务列表列出了管理 Solaris Volume Manager RAID-1 卷所需的过程。

任务	说明	参考
从未使用的分片创建镜像	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令从未使用的分片创建镜像。	第 96 页中的“如何从未使用的分片创建 RAID-1 卷”
从现有文件系统创建镜像	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令从现有文件系统创建镜像。	第 98 页中的“如何从文件系统创建 RAID-1 卷”
从根 (/) 文件系统创建镜像	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令从根 (/) 文件系统创建镜像。	第 103 页中的“SPARC: 如何从根 (/) 文件系统创建 RAID-1 卷” 第 110 页中的“x86: 如何使用 DCA 从根 (/) 文件系统创建 RAID-1 卷”
附加子镜像	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令附加子镜像。	第 115 页中的“如何附加子镜像”
分离子镜像	使用 Solaris Volume Manager GUI 或 <code>metadetach</code> 命令分离子镜像。	第 117 页中的“如何分离子镜像”

任务	说明	参考
使子镜像联机或脱机	使用 Solaris Volume Manager GUI 或 <code>metaonline</code> 命令使子镜像联机。使用 Solaris Volume Manager GUI 或 <code>metaoffline</code> 命令使子镜像脱机。	第 117 页中的“如何使子镜像脱机或联机”
启用子镜像中的分片	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令启用子镜像中的分片。	第 118 页中的“如何启用子镜像中的分片”
检查镜像状态	使用 Solaris Volume Manager GUI 或 <code>metastat</code> 命令检查 RAID-1 卷的状态。	第 119 页中的“如何查看镜像和子镜像的状态”
更改镜像选项	使用 Solaris Volume Manager GUI 或 <code>metaparam</code> 命令更改特定 RAID-1 卷的选项。	第 121 页中的“如何更改 RAID-1 卷选项”
扩展镜像	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令扩展镜像的容量。	第 122 页中的“如何扩展 RAID-1 卷”
替换子镜像中的分片	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令替换子镜像中的分片。	第 123 页中的“如何替换子镜像中的分片”
替换子镜像	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令替换子镜像。	第 124 页中的“如何替换子镜像”
删除镜像（取消镜像）	使用 Solaris Volume Manager GUI、 <code>metadetach</code> 命令或 <code>metaclear</code> 命令取消对文件系统的镜像。	第 126 页中的“如何取消镜像文件系统”
删除无法卸载的文件系统的镜像（取消镜像）	使用 Solaris Volume Manager GUI、 <code>metadetach</code> 命令或 <code>metaclear</code> 命令取消对无法取消挂载的文件系统的镜像。	第 128 页中的“如何取消对无法取消挂载的文件系统的镜像”
使用镜像执行备份	使用 Solaris Volume Manager GUI、 <code>metaonline</code> 命令或 <code>metaoffline</code> 命令通过镜像执行备份。	第 131 页中的“如何对 RAID-1 卷执行联机备份”

## 创建 RAID-1 卷

### ▼ 如何从未使用的分片创建 RAID-1 卷

以下过程说明如何创建双向镜像。如果要创建三向镜像或四向镜像，请使用同一过程。

**开始之前** 请查看第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 89 页中的“创建和维护 RAID-1 卷”。

## 1 创建两个条带或串联。这些组件将成为子镜像。

请参见第 78 页中的“如何创建 RAID-0（条带）卷”或第 79 页中的“如何创建 RAID-0（串联）卷”。

## 2 要创建镜像，请使用下列方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）⇒ "Create Volume"（创建卷），并按照屏幕说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令创建单向镜像：

```
# metainit volume-name -m submirror-name
volume-name      指定要创建的卷的名称
-m              指定创建镜像
submirror-name   指定将成为镜像中的第一个子镜像的组件的名称
```

有关更多信息，请参见以下示例和 [metainit\(1M\)](#) 手册页。

## 3 要添加第二个子镜像，请使用下列方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择要修改的镜像。选择 "Action"（操作）⇒ "Properties"（属性），然后选择 "Submirrors"（子镜像）。请按照屏幕上的说明附加子镜像。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metattach` 命令：

```
# metattach volume-name submirror-name
volume-name      指定要向其添加子镜像的 RAID-1 卷的名称
submirror-name   指定将附加到该镜像的第二个子镜像的组件的名称
```

有关更多信息，请参见以下示例和 [metattach\(1M\)](#) 手册页。

### 示例 11-1 创建双向镜像

```
# metainit d51 1 1 c0t0d0s2
d51: Concat/Stripe is setup
# metainit d52 1 1 c1t0d0s2
d52: Concat/Stripe is setup
# metainit d50 -m d51
d50: Mirror is setup
# metattach d50 d52
d50: Submirror d52 is attached
```

以下示例显示如何创建双向镜像 `d50`。`metainit` 命令创建了两个子镜像（`d51` 和 `d52`），均为 RAID-0 卷。`metainit -m` 命令从 `d51` RAID-0 卷创建单向镜像。`metattach`

命令附加 d52，从而创建一个双向镜像并导致重新同步。在重新同步过程中，所附加的子镜像中的所有数据都会被其他子镜像覆盖。

### 示例 11-2 创建双向镜像而不进行重新同步

```
# metainit d51 1 1 c0t0d0s2
d51: Concat/Stripe is setup
# metainit d52 1 1 c1t0d0s2
d52: Concat/Stripe is setup
# metainit d50 -m d51 d52
metainit: d50: WARNING: This form of metainit is not recommended.
The submirrors may not have the same data.
Please see ERRORS in metainit(1M) for additional information.
d50: Mirror is setup
```

本示例说明如何创建双向镜像 d50。metainit 命令创建了两个子镜像（d51 和 d52），均为 RAID-0 卷。然后，运行 metainit -m 命令使用这两个子镜像创建镜像。使用 metainit 命令（而不是 metattach 命令）创建镜像时，将不会发生重新同步操作。因此，当 Solaris Volume Manager 假定镜像的两面相同且可以互换使用时，数据可能会损坏。

**另请参见** 要针对某个文件系统准备新创建的镜像，请参见《[System Administration Guide: Devices and File Systems](#)》中的第 15 章“[Creating and Mounting File Systems \(Tasks\)](#)”。某些应用程序（例如数据库）不使用文件系统，而是使用原始设备。这些应用程序必须有自己访问原始设备的方法。

## ▼ 如何从文件系统创建 RAID-1 卷

使用此过程镜像现有文件系统。如果可以卸载文件系统，则无需重新引导就可以完成整个过程。对于无法卸载的文件系统（如 /usr 和 /swap），必须重新引导系统才能完成该过程。

如果从构建在分片上的现有文件系统创建 RAID-1 卷，则主 RAID-0 卷（子镜像）中只能包含这一个分片。如果要镜像对系统非常关键的文件系统，则所有的子镜像都必须仅包含一个分片。

有关与镜像根 (/) 文件系统关联的过程，请参见第 103 页中的“[SPARC: 如何从根 \(/\) 文件系统创建 RAID-1 卷](#)”和第 110 页中的“[x86: 如何使用 DCA 从根 \(/\) 文件系统创建 RAID-1 卷](#)”。

在本过程所使用的示例中，现有分片为 c1t0d0s0。第二个分片（c1t1d0s0）可用于镜像的另一半。子镜像分别为 d1 和 d2，镜像为 d0。



注意 – 请确保使用 `metainit` 命令创建单向镜像，然后使用 `metattach` 命令附加其他子镜像。不使用 `metattach` 命令时，不会执行重新同步操作。因此，当 Solaris Volume Manager 假定镜像的两面相同且可以互换使用时，数据可能会损坏。

开始之前 请查看第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 89 页中的“创建和维护 RAID-1 卷”。

- 1 确定要进行镜像的现有文件系统所在的分片。本示例使用分片 `c1t0d0s0`。
- 2 通过以下方法之一，在上一步的分片上创建新的 RAID-0 卷：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）⇒ "Create Volume"（创建卷）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metainit` 命令：

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
-f                                强制继续执行该命令。如果分片包含已挂载的文件系统，则必须使用此选项。

volume-name                      指定要创建的卷的名称。有关命名卷的信息，请参见第 38 页中的“卷名称”。

number-of-stripes                指定要创建的条带数。

components-per-stripe            指定每个条带应包含的组件数。

component-names                  指定所使用的组件的名称。本示例使用根分片 c0t0d0s0。
```

- 3 在未使用的分片（本示例中为 `c1t1d0s0`）上创建另一个 RAID-0 卷（串联），将其用作第二个子镜像。第二个子镜像的大小必须大于或等于原来子镜像的大小。使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）→ "Create Volume"（创建卷），并按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes components-per-stripe component-name
```

---

注 – 有关各选项的相应说明，请参见步骤 2。

---

- 4 使用以下方法之一创建单向镜像：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）⇒ "Create Volume"（创建卷）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -m submirror-name
volume-name      指定要创建的卷的名称。
-m              指定创建镜像。
submirror-name   指定将成为镜像中的第一个子镜像的组件的名称。在本示例
                  中，为根分片所在的 RAID-0 卷。
```

有关更多信息，请参见 [metainit\(1M\)](#) 手册页。



注意 - 从现有文件系统创建镜像时，必须严格按照以下两个步骤进行操作以避免数据损坏。

- 5 编辑 `/etc/vfstab` 文件，以便文件系统挂载指令引用镜像，而不引用块设备。有关 `/etc/vfstab` 文件的更多信息，请参见《[System Administration Guide: Devices and File Systems](#)》中的“[Overview of Mounting and Unmounting File Systems](#)”。

例如，如果 `/etc/vfstab` file 包含以下用于文件系统的项：

```
/dev/dsk/slice /dev/rdisk/slice /var ufs 2 yes -
```

将此项更改为如下内容：

```
/dev/md/dsk/mirror-name /dev/md/rdsk/mirror-name /var ufs 2 yes -
```

- 6 按照以下过程之一重新挂载新镜像的文件系统：

- 如果要对可以卸载的文件系统进行镜像，请卸载该文件系统并重新挂载。

```
# umount /filesystem
# mount /filesystem
```

- 如果要对无法卸载的文件系统进行镜像，请重新引导系统。

```
# reboot
```

- 7 使用以下格式的 `metattach` 命令附加第二个子镜像。

```
# metattach volume-name submirror-name
volume-name      指定要向其添加子镜像的 RAID-1 卷的名称
submirror-name   指定将附加到该镜像的第二个子镜像的组件的名称
有关更多信息，请参见 metattach\(1M\) 手册页。
```

### 示例 11-3 从可以卸载的文件系统创建双向镜像

```
# metainit -f d1 1 1 c1t0d0s0
d1: Concat/Stripe is setup
# metainit d2 1 1 c1t1d0s0
d2: Concat/Stripe is setup
# metainit d0 -m d1
d0: Mirror is setup
# umount /master
    (Edit the /etc/vfstab file so that the file system references the mirror)
# mount /master
# metattach d0 d2
d0: Submirror d2 is attached
```

在本示例中，`-f` 选项强制创建第一个串联 `d1`，其中包含已挂载在 `/dev/dsk/c1t0d0s0` 上的文件系统 `/master`。第二个串联 `d2` 从 `/dev/dsk/c1t1d0s0` 创建。此分片的大小必须大于或等于 `d1` 的大小。带有 `-m` 选项的 `metainit` 命令用于从 `d1` 创建单向镜像 `d0`。

接下来，应在 `/etc/vfstab` 文件中更改该文件系统所对应的条目以便引用镜像。`/etc/vfstab` 文件中的以下行最初显示如下：

```
/dev/dsk/c1t0d0s0 /dev/rdisk/c1t0d0s0 /var ufs 2 yes -
```

应将条目更改为如下内容：

```
/dev/md/dsk/d0 /dev/md/rdisk/d0 /var ufs 2 yes -
```

最后，重新挂载文件系统，并将子镜像 `d2` 附加到镜像，从而导致镜像重新同步。系统确认了 RAID-0 和 RAID-1 卷均已设置，且子镜像 `d2` 已附加。

### 示例 11-4 从无法取消挂载的文件系统创建双向镜像

```
# metainit -f d12 1 1 c0t3d0s6
d12: Concat/Stripe is setup
# metainit d22 1 1 c1t0d0s6
d22: Concat/Stripe is setup
# metainit d2 -m d12
d2: Mirror is setup
    (Edit the /etc/vfstab file so that /usr references the mirror)
# reboot
...
# metattach d2 d22
d2: Submirror d22 is attached
```

本示例使用 `/usr` 文件系统所在的分片创建双向镜像。`-f` 选项强制创建第一个串联 `d12`，其中包含已挂载在 `/dev/dsk/c0t3d0s6` 上的文件系统 `/usr`。第二个串联 `d22` 从 `/dev/dsk/c1t0d0s6` 创建。此分片的大小必须大于或等于 `d12` 的大小。带有 `-m` 选项的 `metainit` 命令使用 `/usr` 文件系统所在的串联创建单向镜像 `d2`。接下来，必须编辑 `/etc/vfstab` 文件，以便将 `/usr` 所对应的条目更改为引用该镜像。

`/etc/vfstab` file 包含对应于 `/usr` 文件系统的以下条目：

```
/dev/dsk/c0t3d0s6 /dev/rdsk/c0t3d0s6 /usr ufs 1 yes -
```

将此项更改为如下内容：

```
/dev/md/dsk/d2 /dev/md/rdsk/d2 /usr ufs 1 yes -
```

重新引导后，将第二个子镜像 `d22` 附加到镜像，从而导致镜像重新同步。

### 示例 11-5 从 `/swap` 空间创建镜像

```
# metainit -f d11 1 1 c0t0d0s1
d11: Concat/Stripe is setup
# metainit d21 1 1 c1t0d0s1
d21: Concat/Stripe is setup
# metainit d1 -m d11
d1: Mirror is setup
    (Edit the /etc/vfstab file so that swap references the mirror)
# reboot
...
# metattach d1 d21
d1: Submirror d21 is attached
```

在本示例中，`-f` 选项强制创建第一个串联 `d11`，其中包含已挂载在 `/dev/dsk/c0t0d0s1` 上的文件系统 `swap`。第二个串联 `d21` 从 `/dev/dsk/c1t0d0s1` 创建。此分片的大小必须大于或等于 `d11` 的大小。带有 `-m` 选项的 `metainit` 命令使用 `swap` 所在的串联创建单向镜像 `d1`。接下来，如果 `/etc/vfstab` 文件中存在对应于 `swap` 的条目，则必须对该条目进行编辑以引用该镜像。

`/etc/vfstab` file 包含对应于 `swap` 空间的以下条目：

```
/dev/dsk/c0t0d0s1 - - swap - no -
```

将此项更改为如下内容：

```
/dev/md/dsk/d1 - - swap - no -
```

重新引导后，将第二个子镜像 `d21` 附加到镜像，从而导致镜像重新同步。

要在对 `swap` 空间进行镜像之后保存故障转储，请使用 `dumppadm` 命令将转储设备配置为卷。例如，如果交换设备名为 `/dev/md/dsk/d2`，请使用 `dumppadm` 命令将此设备设置为转储设备。

## ▼ SPARC: 如何从根 (/) 文件系统创建 RAID-1 卷

在 SPARC 平台上镜像根 (/) 文件系统的过程与镜像任何其他无法卸载的文件系统的过程类似。此过程的不同之处在于通过运行 `metaroot` 命令替代手动编辑 `/etc/vfstab` 文件。镜像根 (/) 文件系统还需要将路径记录到备用引导设备。如果子镜像出现故障，此设备会重新引导系统。

在本过程所使用的示例中，现有分片为 `c1t0d0s0`。第二个分片 (`c1t1d0s0`) 可用于镜像的另一半。子镜像分别为 `d1` 和 `d2`，镜像为 `d0`。



**注意** – 请确保使用 `metainit` 命令创建单向镜像，然后使用 `metattach` 命令附加其他子镜像。不使用 `metattach` 命令时，不会执行重新同步操作。因此，当 Solaris Volume Manager 假定镜像的两面相同且可以互换使用时，数据可能会损坏。

**开始之前** 请查看第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 89 页中的“创建和维护 RAID-1 卷”。

- 1 确定要进行镜像的现有根 (/) 文件系统所在的分片。本示例使用分片 `c1t0d0s0`。
- 2 通过以下方法之一，在上一步的分片上创建新的 RAID-0 卷。RAID-0 卷中只能包含这一个分片。
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）⇒ "Create Volume"（创建卷）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metainit` 命令：

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

<code>-f</code>	强制继续执行该命令。如果分片包含已挂载的文件系统，则必须使用此选项。
<code>volume-name</code>	指定要创建的卷的名称。有关命名卷的信息，请参见第 38 页中的“卷名称”。
<code>number-of-stripes</code>	指定要创建的条带数。
<code>components-per-stripe</code>	指定每个条带应包含的组件数。
<code>component-names</code>	指定所使用的组件的名称。本示例使用根分片 <code>c0t0d0s0</code> 。

- 3 在未使用的分片（本示例中为 `c1t1d0s0`）上创建另一个 RAID-0 卷，将其用作第二个子镜像。第二个子镜像的大小必须等于或大于原来子镜像的大小。使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）->"Create Volume"（创建卷），并按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes components-per-stripe component-name
```

---

注 - 有关各选项的相应说明，请参见步骤 2。

---

#### 4 使用以下方法之一创建单向镜像：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）⇒ "Create Volume"（创建卷）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -m submirror-name
```

`volume-name`            指定要创建的卷的名称。

`-m`                      指定创建镜像。

`submirror-name`        指定将成为镜像中的第一个子镜像的组件的名称。在本示例中，为根分片所在的 RAID-0 卷。

#### 5 重新挂载新镜像的文件系统。运行 `metaroot volume-name` 命令，使用已创建的镜像的名称替换 `volume-name`。然后，重新引导系统。

```
# metaroot volume-name
# reboot
```

有关更多信息，请参见 [metaroot\(1M\)](#) 手册页。

#### 6 使用以下格式的 `metattach` 命令附加第二个子镜像。

```
# metattach volume-name submirror-name
```

`volume-name`            指定要向其添加子镜像的 RAID-1 卷的名称

`submirror-name`        指定将附加到该镜像的第二个子镜像的组件的名称

有关更多信息，请参见 [metattach\(1M\)](#) 手册页。

#### 7 记录备用的引导路径。

- a. 确定备用根设备的路径。针对作为第二个子镜像附加到根 (/) 文件系统镜像的分片使用 `ls -l` 命令。

```
# ls -l /dev/dsk/clt1d0s0
lrwxrwxrwx 1 root root 55 Mar 5 12:54 /dev/rdisk/clt1d0s0 -> \
../..../devices/sbus@1,f8000000/esp@1,2000000/sd@3,0:a
```

- b. 记录 `/devices` 目录后面的字符串：`/sbus@1,f8000000/esp@1,200000/sd@3,0:a`。

---

注-由于系统可能会不可用，因此还应将这些信息记录在系统以外的某个位置。有关从备用引导设备引导的详细信息，请参见第 256 页中的“从引导问题中恢复”。

---

- c. 编辑该字符串，将主名称（在本例中为 `sd`）更改为 `disk`，产生的结果为 `/sbus@1,f8000000/esp@1,200000/disk@3,0:a`。如果系统使用 IDE 总线，则最初的全路径可能如下所示：

```
$ ls -l /dev/dsk/clt1d0s0
lrwxrwxrwx 1 root root 38 Mar 13 15:03 /dev/dsk/c0t0d0s0 -> \
../../../../devices/pci@1f,0/ide@d/dad@0,0:a
```

将主名称 `dad` 更改为 `disk` 之后，将变为 `/pci@1f,0/ide@d/disk@0,0:a`

- d. 使用 `OpenBoot PROM nvalias` 命令为辅助根 (`/`) 文件系统镜像定义“备份根”设备别名。例如：

```
ok nvalias backup_root /sbus@1,f8000000/esp@1,200000/disk@3,0:a
```

- e. 重新定义 `boot-device` 别名，使其按照主子镜像和辅助子镜像的使用顺序来进行引用，并存储该配置。

```
ok printenv boot-device
boot-device =          disk net
ok setenv boot-device disk backup_root net
boot-device =          disk backup_root net
ok nvstore
```

---

注-如果主子镜像出现故障，系统将自动引导到第二个子镜像。或者，如果手动引导而不是使用自动引导，则可输入以下内容：

```
ok boot backup_root
```

---

## 示例 11-6 SPARC：从根 (`/`) 文件系统创建镜像

```
# metainit -f d1 1 1 c0t0d0s0
d1: Concat/Stripe is setup
# metainit d2 1 1 c0t1d0s0
d2: Concat/Stripe is setup
# metainit d0 -m d1
d0: Mirror is setup
# metaroot d0
# lockfs -fa
# reboot
...
# metattach d0 d2
d0: Submirror d2 is attached
# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx 1 root root 88 Feb 8 15:51 /dev/rdisk/clt3d0s0 ->
../../../../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a
```

```
# init 0
.
.
.
ok nvalias backup_root /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
ok setenv boot-device disk backup_root net
ok nvstore
```

在本示例中，`-f` 选项强制创建第一个 RAID-0 卷 `d1`，其中包含已挂载在 `/dev/dsk/c0t0d0s0` 上的根 (`/`) 文件系统。第二个串联 `d2` 从 `/dev/dsk/c0t1d0s0` 创建。此分片的大小必须大于或等于 `d1` 的大小。带有 `-m` 选项的 `metainit` 命令使用根 (`/`) 所在的串联创建单向镜像 `d0`。

接下来，使用 `metaroot` 命令编辑 `/etc/vfstab` 和 `/etc/system` 文件，以便可以通过卷上的根 (`/`) 文件系统来引导系统。重新引导之前，最好先运行 `lockfs -fa` 命令。有关更多信息，请参见 [lockfs\(1M\)](#) 手册页。

在重新引导系统之前，请勿附加第二个子镜像。在运行 `metaroot` 命令之后、附加第二个子镜像之前，必须重新引导系统。

重新引导后，将子镜像 `d2` 附加到镜像，从而导致镜像重新同步。系统确认了串联和镜像均已设置，且子镜像 `d2` 已附加。

针对根原始设备运行了 `ls -l` 命令以确定指向备用根设备的路径，以备以后通过该路径引导系统。

## x86: 从根 (`/`) 文件系统创建 RAID-1 卷

从 Solaris 10 1/06 发行版开始，GRand Unified Bootloader (GRUB) 取代了基于 x86 的系统中用于引导过程和配置的 Device Configuration Assistant (DCA)。有关此功能的简要说明，请参阅《[Oracle Solaris 管理：基本管理](#)》中的第 15 章“[基于 GRUB 的引导（参考）](#)”。

本节中的过程描述了从根 (`/`) 文件系统创建 RAID-1 卷的步骤。如果系统正在运行 Solaris 10 1/06 OS 或后续发行版，请按照第一个过程（使用 GRUB）进行操作。否则，请执行第二个过程（使用 DCA）中的步骤。

在基于 x86 的系统上镜像根 (`/`) 文件系统的过程与在 SPARC 系统上镜像根的过程类似。但是，在基于 x86 的系统上，BIOS 和 `fdisk` 分区会将事情变得更为复杂。

在本过程所使用的示例中，现有分片为 `c1t0d0s0`。第二个分片 (`c1t1d0s0`) 可用于镜像的另一半。子镜像分别为 `d1` 和 `d2`，镜像为 `d0`。

---

注 – 在实现任何过程之前，请查看第 41 页中的“[创建 Solaris Volume Manager 组件的先决条件](#)”和第 89 页中的“[创建和维护 RAID-1 卷](#)”。

---

## ▼ x86: 如何使用 GRUB 从根 (/) 文件系统创建 RAID-1 卷

- 1 验证是否可将 BIOS 引导设备的顺序配置为允许系统从镜像中的第二个磁盘进行引导。  
启动内核之前，系统由只读存储器 (Read-Only-Memory, ROM) 基本输入/输出系统 (Basic Input/Output System, BIOS) 控制，该系统是基于 x86 的系统上的固件接口。BIOS 类似于基于 SPARC 的系统上的引导 PROM。以下是一些 BIOS 任务：
  - 执行启动功能。
  - 检测可从其引导系统的正确设备。
  - 从该设备装入主引导记录，以允许系统自行引导。

通常可以配置 BIOS 以选择要探测其引导记录的设备的顺序。此外，现在实现的大多数 BIOS 都允许将设备配置为自动向辅助子镜像进行故障转移。如果您系统的 BIOS 不具备此功能，则主子镜像出现故障时，需要在系统引导过程中访问 BIOS，以便将系统重新配置为从辅助根分片引导。有关如何配置 BIOS 中的设置的说明，请参阅 BIOS 的用户指南。

设置根镜像之前，请检查系统中的 BIOS，验证是否可以从多个磁盘引导。某些设备的驱动程序被配置为只能看到系统上的一个磁盘。

- 2 验证 fdisk 分区是否已配置为支持根镜像。

如果镜像根 (/) 文件系统时存在一个单独的 x86 引导分区，则会出现问题。由于该引导分区位于 Solaris fdisk 分区之外，因此无法通过 Solaris Volume Manager 对 x86 引导分区进行镜像。此外，由于 x86 引导分区仅存在一个副本，因此会出现单点故障。

Solaris 10 1/06 软件和后续发行版的基于 GRUB 的安装程序将不再自动创建 x86 引导分区。但是，如果系统中已经存在 x86 引导分区，则缺省情况下安装程序将保留该分区。

要确定系统是否有单独的 x86 引导分区，请检查 /etc/vfstab 文件。如果该文件包含类似于以下的条目，则存在 x86 引导分区：

```
/dev/dsk/c2t1d0p0:boot - /boot pcfs - no -
```

要使用 Solaris Volume Manager 镜像根 (/) 文件系统，文件系统必须使用单个 Solaris fdisk 分区。因此，如果系统中已存在 x86 引导分区，请使用 fdisk 命令删除此分区，然后重新安装 Solaris 软件。重新安装时，将不再重新创建引导分区。

---

注 - Solaris Volume Manager 只能镜像 Solaris fdisk 分区内的分片。如果您有多个 fdisk 分区，则需要使用其他方法保护 Solaris fdisk 分区以外的数据。

---

- 3 使辅助子镜像可以使用主引导程序进行引导。

- a. 指定主引导程序。

```
# fdisk -b /usr/lib/fs/ufs/mboot /dev/rdsd/c1t1d0p0
```

会出现以下屏幕：

```
Total disk size is 31035 cylinders
Cylinder size is 1146 (512 byte) blocks

Partition  Status  Type           Start  End  Length  %
=====  =====  =====
      1      Active  Solaris        1  31034  31034  100

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Enter Selection:
```

b. 从菜单中选择数字 5，然后按回车键。

4 使辅助磁盘可引导。

```
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t1d0s0
```

有关 `installgrub` 的更多信息，请参见 [installgrub\(1M\)](#) 手册页。

5 确定要进行镜像的现有根 (/) 文件系统所在的分片。

本示例使用分片 `c1t0d0s0`。

6 在上一步中的分片上创建新的 RAID-0 卷。

RAID-0 卷中只能包含这一个分片。使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）⇒ "Create Volume"（创建卷）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
-f                                强制继续执行该命令。如果分片包含已挂载的文件系统，则必须使用此选项。

volume-name                      指定要创建的卷的名称。有关命名卷的信息，请参见第 38 页中的“卷名称”。

number-of-stripes                指定要创建的条带数。

components-per-stripe            指定每个条带应包含的组件数。

component-names                  指定所使用的组件的名称。本示例使用根分片 c0t0d0s0。
```

- 7 在未使用的分片上创建第二个 RAID-0 卷（本示例中为 **c1t1d0s0**），将其用作第二个子镜像。

---

注- 第二个子镜像的大小必须等于或大于原来子镜像的大小。此外，用作第二个子镜像的分片必须带有分片标记 "root"，且根分片必须为分片 0。

有关配置分片标记字段的信息，请参见 [format\(1M\)](#) 手册页。

---

使用以下任一方法：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）-> "Create Volume"（创建卷），并按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes components-per-stripes component-names
```

---

注- 有关各选项的相应说明，请参见步骤 6。

---

- 8 使用以下方法之一创建单向镜像：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）⇒ "Create Volume"（创建卷）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -m submirror-name
volume-name      指定要创建的卷的名称。
-m              指定创建镜像。
submirror-name   指定将成为镜像中的第一个子镜像的组件的名称。在本示例
                  中，为根分片所在的 RAID-0 卷。
```

- 9 重新挂载新镜像的文件系统，然后重新引导系统。

```
# metaroot volume-name
# reboot
```

有关更多信息，请参见 [metaroot\(1M\)](#) 手册页。

- 10 附加第二个子镜像。

```
# metattach volume-name submirror-name
volume-name      指定要向其添加子镜像的 RAID-1 卷的名称。
submirror-name   定将附加到该镜像的第二个子镜像的组件的名称。
```

有关更多信息，请参见 [metattach\(1M\)](#) 手册页。

## 11 在 menu.lst 文件中定义备用引导路径。

要使系统可从用来存放辅助子镜像的磁盘进行引导，请将系统配置为将该磁盘视为备用引导设备。在当前示例中，备用路径 `c1t1d0s0` 位于第二个磁盘的第一个 fdisk 分区的第一个分片中。因此，应编辑包含以下条目的 `menu.lst`：

```
title alternate boot
  root (hd1,0,a)
  kernel /boot/multiboot
  module /boot/x86.miniroot-safe
```

---

注 – 要正确编辑 `menu.lst` 中的条目，您必须熟悉 GRUB 中的磁盘命名约定。有关详细信息，请参见《[Oracle Solaris 管理：基本管理](#)》中的“使用 GRUB 引导基于 x86 的系统（任务列表）”。

---

编辑完 `menu.lst` 文件之后，系统将设置为故障转移到第二个磁盘。如果主磁盘出现故障，则磁盘编号会更改，以便系统从辅助磁盘引导。



**注意** – 在某些情况下，BIOS 的磁盘自动重新编号功能可能会影响从不可用的主磁盘进行恢复。当磁盘重新编号功能强制系统从辅助磁盘引导时，主磁盘的引导归档文件将过时。如果该主磁盘稍后变为可用，则引导系统时磁盘编号功能将再次切换到使用缺省主磁盘引导系统。但是，此时主磁盘的引导归档文件仍处于过时状态。因此，系统可能无法引导。所以，在这种情况下，请确保从 GRUB 菜单中选择正确的项，以便从有效的引导归档文件引导系统。系统完成引导过程后，请执行常规的 `metadevice` 维护，这会对主磁盘和辅助磁盘均进行同步，并将有效的引导归档文件恢复到主磁盘。

---

## ▼ x86: 如何使用 DCA 从根 (/) 文件系统创建 RAID-1 卷

### 1 验证是否可将 BIOS 引导设备的顺序配置为允许系统从镜像中的第二个磁盘进行引导。

启动内核之前，系统由只读存储器 (Read-Only-Memory, ROM) 基本输入/输出系统 (Basic Input/Output System, BIOS) 控制，该系统是基于 x86 的系统上的固件接口。BIOS 类似于基于 SPARC 的系统上的引导 PROM。除了其他启动功能以外，BIOS 还负责查找要从其引导系统的正确设备，以及从允许系统自行引导的设备中装入主引导记录。通常可以配置 BIOS 以选择要探测其引导记录的设备的顺序。此外，现在实现的大多数 BIOS 都允许将设备配置为自动向辅助子镜像进行故障转移。如果系统不具备此功能，则主子镜像出现故障时，需要在系统引导过程中访问 BIOS，并将其重新配置为从辅助根分片进行引导。有关如何配置 BIOS 中的设置的说明，请参阅 BIOS 的用户指南。

可以在系统中使用 DCA 验证是否可以从多个磁盘引导。某些设备的驱动程序被配置为只能看到系统上的一个磁盘。

2 验证 fdisk 分区是否已配置为支持根镜像。

基于 x86 的系统的另一个功能是可使用 fdisk 分区。使用 Solaris OS 安装程序的缺省引导磁盘分区布局将创建一个 Solaris fdisk 分区和另一个大小大约为 10 MB、称为 x86 引导分区的小型 fdisk 分区。

对根 (/) 文件系统进行镜像时，x86 引导分区会产生问题。x86 引导分区位于 Solaris fdisk 分区以外的位置。出于这个原因，x86 引导分区无法通过 Solaris Volume Manager 进行镜像。此外，由于 x86 引导分区仅存在一个副本，因此会出现单点故障。

可以确定 Solaris OS 是否具有单独的 x86 引导分区。x86 引导分区会通过类似如下的条目挂载到 /etc/vfstab 文件中：

```
/dev/dsk/c2t1d0p0:boot - /boot pcfs - no -
```

如果不存在单独的 x86 引导分区，则 /etc/vfstab 文件中将不会显示此条目。

要镜像根 (/) 文件系统，需要定制 fdisk 分区来删除 x86 引导分区，然后使用单个 Solaris fdisk 分区。如果要使用 Solaris Volume Manager 根镜像，请勿在系统安装过程中创建单独的 x86 引导分区。如果已安装系统并创建了单独的 x86 引导分区，请使用 fdisk 命令删除该 fdisk 分区并重新安装系统。安装时，请避免通过在安装过程中定制磁盘分区来创建单独的 x86 引导分区。

注 - Solaris Volume Manager 只能镜像 Solaris fdisk 分区内的分片。如果您有多个 fdisk 分区，则需要使用其他方法保护 Solaris fdisk 分区以外的数据。

3 使辅助子镜像可以使用主引导程序进行引导。

a. 使用 fdisk 命令指定主引导程序。

```
# fdisk -b /usr/lib/fs/ufs/mboot /dev/rdsk/c1t1d0p0
```

会出现以下屏幕：

```
Total disk size is 31035 cylinders
Cylinder size is 1146 (512 byte) blocks

Partition  Status  Type           Start  End  Length  %
=====  =====  =====
      1      Active  Solaris        1 31034  31034  100

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Enter Selection:
```

b. 从菜单中选择数字 5，然后按回车键。

4 在辅助子镜像中安装引导块，以便可从该镜像引导系统。

辅助子镜像所驻留的磁盘中的分片 8 是从此 `fdisk` 分区引导 Solaris OS 所必需的。此分片包含分区引导记录 (pboot)、磁盘的 Solaris VTOC 和引导块。这些信息是特定于磁盘的信息，因此不能使用 Solaris Volume Manager 进行镜像。但是，必须确保这两个磁盘都是可引导的，以便可以在主磁盘出现故障时从辅助磁盘引导。使用 `installboot` 命令可将第二个磁盘设置为 Solaris 可引导磁盘。有关更多信息，请参见 [installboot\(1M\)](#) 手册页。

必须将该磁盘的分片 2 指定为设备，且分片 2 必须构成整个磁盘。

```
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot \
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdisk/c1t1d0s2
```

5 确定要进行镜像的现有根 (/) 文件系统所在的分片。本示例使用分片 `c1t0d0s0`。

6 通过以下方法之一，在上一步的分片上创建新的 RAID-0 卷。RAID-0 卷中只能包含这一个分片。

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes" (卷) 节点，然后选择 "Action" (操作) ⇒ "Create Volume" (创建卷)。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。

- 使用以下格式的 `metainit` 命令：

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

`-f` 强制继续执行该命令。如果分片包含已挂载的文件系统，则必须使用此选项。

`volume-name` 指定要创建的卷的名称。有关命名卷的信息，请参见第 38 页中的“卷名称”。

`number-of-stripes` 指定要创建的条带数。

`components-per-stripe` 指定每个条带应包含的组件数。

`component-names` 指定所使用的组件的名称。本示例使用根分片 `c0t0d0s0`。

7 在未使用的分片 (本示例中为 `c1t1d0s0`) 上创建另一个 RAID-0 卷，将其用作第二个子镜像。第二个子镜像的大小必须等于或大于原来子镜像的大小。使用以下方法之一：

---

注-用作第二个子镜像的分片必须带有分片标记 "root"，且根分片必须为分片 0。有关配置分片标记字段的信息，请参见 [format\(1M\)](#) 手册页。

---

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes" (卷) 节点，然后选择 "Action" (操作) -> "Create Volume" (创建卷)，并按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。

- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes components-per-stripes component-names
```

---

注 - 有关各选项的相应说明，请参见步骤 6。

---

## 8 使用以下方法之一创建单向镜像：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后选择 "Action"（操作）⇒ "Create Volume"（创建卷）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。

- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -m submirror-name
```

`volume-name`      指定要创建的卷的名称。

`-m`                指定创建镜像。

`submirror-name`    指定将成为镜像中的第一个子镜像的组件的名称。在本示例中，为根分片所在的 RAID-0 卷。

## 9 重新挂载新镜像的文件系统。运行 `metaroot volume-name` 命令，使用已创建的镜像的名称替换 `volume-name`。然后，重新引导系统。

```
# metaroot volume-name
```

```
# reboot
```

有关更多信息，请参见 [metaroot\(1M\)](#) 手册页。

## 10 使用以下格式的 `metattach` 命令附加第二个子镜像。

```
# metattach volume-name submirror-name
```

`volume-name`      指定要向其添加子镜像的 RAID-1 卷的名称。

`submirror-name`    定将附加到该镜像的第二个子镜像的组件的名称。

有关更多信息，请参见 [metattach\(1M\)](#) 手册页。

## 11 记录备用的引导路径。

您需要配置系统，以便在主子镜像出现故障的情况下从辅助子镜像引导系统。要使系统可从用来存放辅助子镜像的磁盘进行引导，请将系统配置为将该磁盘视为备用引导设备。

- 确定备用引导设备的路径。针对作为第二个子镜像附加到根 (/) 文件系统镜像的分片使用 `ls -l` 命令。

```
# ls -l /dev/dsk/clt1d0s0
lrwxrwxrwx 1 root root 55 Mar 5 12:54 /dev/rdisk/clt1d0s0 -> ../
./devices/eisa/eha@1000,0/cmdk@1,0:a
```

- b. 记录 `/devices` 目录后面的字符串：`/eisa/eha@1000,0/cmdk@1,0:a`。这是设备树路径。

注-由于系统可能会不可用，因此应将这些信息记录在系统以外的某个位置。如果必须使用 DCA 引导系统，这样可使您更轻松输入设备树路径信息。

- c. 使用 `eeeprom` 命令定义备用引导路径。例如：

```
# eeeprom altbootpath=/eisa/eha@1000,0/cmdk@1,0:a
```

如果主子镜像出现故障，则系统将尝试从辅助子镜像引导。如果可将 BIOS 配置为自动故障转移到第二个磁盘，则引导过程将是自动的。否则，需要进入 BIOS 并将其配置为从辅助磁盘引导。系统开始引导后，将尝试从 `bootpath` 设备引导。由于主引导磁盘是根镜像中的停用磁盘，因此系统随后会尝试从 `altbootpath` 设备引导。有关如何配置 BIOS 中的设置的说明，请参阅 BIOS 的用户指南。

如果系统没有自动进行引导，则可以尝试使用 DCA 来选择辅助子镜像。在某些系统中，可以选择在引导过程中进入 DCA。如果此选项不可用，则需要从 x86 引导软盘引导，然后使用 DCA 来选择辅助子镜像。操作系统引导之后，请使用设置为备用引导路径的值（`altbootpath` 值）来更新 `eeeprom bootpath` 值。然后，系统将自动引导。

有关使用 `eeeprom` 命令的更多信息，请参见 [eeeprom\(1M\)](#) 手册页。

示例 11-7 x86：使用 DCA 从根 (/) 文件系统创建镜像

```
# metainit -f d1 1 1 c0t0d0s0
d1: Concat/Stripe is setup
# metainit d2 1 1 c0t1d0s0
d2: Concat/Stripe is setup
# metainit d0 -m d1
d0: Mirror is setup
# metaroot d0
# lockfs -fa
# reboot
...
# metattach d0 d2
d0: Submirror d2 is attached
# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx 1 root root 88 Feb 8 15:51 /dev/dsk/c1t3d0s0 ->
../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a,raw
# eeeprom altbootpath=/pci@1f,0/pci@1,1/ide@3/dad@0,0:a,raw
# fdisk -b /usr/lib/fs/ufs/mboot /dev/dsk/c0t1d0p0
Total disk size is 31035 cylinders
Cylinder size is 1146 (512 byte) blocks
```

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====
1	Active	Solaris	1	31034	31034	100

```

SELECT ONE OF THE FOLLOWING:
  1. Create a partition
  2. Specify the active partition
  3. Delete a partition
  4. Change between Solaris and Solaris2 Partition IDs
  5. Exit (update disk configuration and exit)
  6. Cancel (exit without updating disk configuration)
Enter Selection: 5
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot \
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdisk/c0t1d0s2

```

## 了解镜像根 (/) 文件系统时出现的引导时警告

镜像根 (/) 文件系统后，控制台中将会显示错误消息，这些错误消息将记录在 `/etc/syslog.conf` 文件中所定义的系统日志中。这些错误消息并不表示存在问题。针对当前未使用的每种设备类型，都会显示这些消息，因为不能强制装入未使用的模块。错误消息可能与以下内容类似：

```

Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forcload of
misc/md_trans failed
Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forcload of
misc/md_raid failed
Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forcload of
misc/md_hotspares failed

```

您可以放心地忽略这些错误消息。

## 使用子镜像

### ▼ 如何附加子镜像

---

注 – 指示 "can't attach labeled submirror to an unlabeled mirror" 的错误消息表示您尝试将 RAID-0 卷附加到镜像而未能成功。带标签的卷（子镜像）的第一个组件从柱面 0 开始，而无标签的卷的第一个组件从柱面 1 开始。为了防止带标签的子镜像的标签遭到破坏，Solaris Volume Manager 不允许将带标签的子镜像附加到无标签的镜像。

---

**开始之前** 请阅读第 89 页中的“创建和维护 RAID-1 卷”。

#### 1 确定要用作子镜像的组件（串联或条带）。

该组件必须大于或等于镜像中的现有子镜像。如果尚未创建要作为子镜像的卷，请参见第 77 页中的“创建 RAID-0（条带）卷”或第 79 页中的“创建 RAID-0（串联）卷”。

2 确保您具有 root 用户权限并且拥有所有数据的当前备份。

3 使用 **metastat** 命令验证要处理的镜像的状态是否为 "Okay"。

```
# metastat mirror
```

4 使用以下方法之一附加子镜像。

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）⇒ "Properties"（属性）并单击 "Submirror"（子镜像）选项卡。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用 **metattachmirrorsubmirror** 命令。

```
# metattach mirror submirror
```

有关更多信息，请参见 [metattach\(1M\)](#) 手册页。

5 使用 **metastat** 命令查看镜像的状态。

```
# metastat mirror
```

## 示例 11-8 附加子镜像

```
# metastat d30
d30: mirror
    Submirror 0: d60
    State: Okay
...
# metattach d30 d70
d30: submirror d70 is attached
# metastat d30
d30: mirror
    Submirror 0: d60
    State: Okay
    Submirror 1: d70
    State: Resyncing
    Resync in progress: 41 % done
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 2006130 blocks
...
```

本示例说明如何将子镜像 d70 附加到单向镜像 d30。将子镜像附加到镜像后，即创建了一个双向镜像。镜像 d30 最初由子镜像 d60 构成。子镜像 d70 为 RAID-0 卷。使用 **metastat** 命令验证镜像的状态为 "Okay"，然后附加子镜像。运行 **metattach** 命令时，会将新的子镜像与现有镜像重新进行同步。将其他子镜像附加到镜像时，系统将显示一条消息。要验证镜像是否在重新同步，请使用 **metastat** 命令。

## ▼ 如何分离子镜像

开始之前 请阅读第 89 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有 root 用户权限。请确保已有所有数据的当前备份。
- 2 使用 `metastat` 命令验证要处理的镜像的状态是否为 "Okay"。
- 3 使用以下方法之一分离子镜像。
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）⇒ "Properties"（属性）并单击 "Submirror"（子镜像）选项卡。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用 `metadetach` 命令从镜像分离子镜像。

```
# metadetach mirror submirror
```

有关更多信息，请参见 [metadetach\(1M\)](#) 手册页。

### 示例 11-9 分离子镜像

```
# metastat
d5: mirror
    Submirror 0: d50
...
# metadetach d5 d50
d5: submirror d50 is detached
```

在本示例中，镜像 d5 具有子镜像 d50。您使用 `metadetach` 命令分离子镜像。d50 中的底层分片可以在其他位置重复使用。从镜像分离子镜像后，系统将显示一条确认消息。

## ▼ 如何使子镜像脱机或联机

仅当通过 `metaoffline` 命令使子镜像脱机后，才可以使用 `metaonline` 命令。运行 `metaonline` 命令后，Solaris Volume Manager 会自动开始将子镜像与镜像重新进行同步。

---

注 - `metaoffline` 命令的功能与 `metadetach` 命令所提供的功能类似。但是，`metaoffline` 命令不会切断子镜像与镜像之间的逻辑关联。

---

开始之前 请阅读第 89 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有 root 用户权限并且拥有所有数据的当前备份。

## 2 使用以下方法之一使子镜像联机或脱机。

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）⇒ "Properties"（属性）并单击 "Submirror"（子镜像）选项卡。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。

- 使用 `metaoffline` 命令使子镜像脱机。

```
# metaoffline mirror submirror
```

有关更多信息，请参见 [metaoffline\(1M\)](#) 手册页。

- 使用 `metaonline` 命令使子镜像联机。

```
# metaonline mirror submirror
```

有关更多信息，请参见 [metaonline\(1M\)](#) 手册页。

### 示例 11-10 使子镜像脱机

```
# metaoffline d10 d11
d10: submirror d11 is offlined
```

在本示例中，子镜像 `d11` 将从镜像 `d10` 脱机。系统将继续从另一个子镜像中进行读取。第一次写入之后，镜像将不再同步。使脱机子镜像恢复联机后，即可更正这种不一致性。

### 示例 11-11 使子镜像联机

```
# metaonline d10 d11d10: submirror d11 is onlined
```

在本示例中，将使镜像 `d10` 中的子镜像 `d11` 恢复为联机状态。

## ▼ 如何启用子镜像中的分片

**开始之前** 请阅读第 208 页中的“替换和启用 RAID-1 和 RAID-5 卷中的组件概述”和第 89 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有 `root` 用户权限并且拥有所有数据的当前备份。
- 2 使用以下方法之一来启用子镜像中的分片。
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）⇒ "Properties"（属性）并单击 "Submirror"（子镜像）选项卡。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。

- 使用 `metareplace` 命令启用子镜像中出现故障的分片。

```
# metareplace -e mirror failed-slice
```

`metareplace` 命令会自动开始重新同步，以便将已修复或替换的分片与镜像的其余部分同步。

有关更多信息，请参见 [metareplace\(1M\)](#) 手册页。

### 示例 11-12 启用子镜像中的分片

```
# metareplace -e d11 c1t4d0s7
d11: device c1t4d0s7 is enabled
```

在本示例中，镜像 `d11` 中的子镜像包含分片 `c1t4d0s7`，该分片有一个软错误。使用带有 `-e` 选项的 `metareplace` 命令可启用出现故障的分片。

如果物理磁盘有缺陷，则可以使用系统上的其他可用磁盘（及其分片）来替换该物理磁盘，如第 123 页中的“[如何替换子镜像中的分片](#)”中所述。或者，也可以修复或替换该磁盘，将其格式化，然后运行带有 `-e` 选项的 `metareplace` 命令，如本示例中所示。

## 维护 RAID-1 卷

### ▼ 如何查看镜像和子镜像的状态

**开始之前** 有关与 RAID-1 卷和子镜像关联的状态信息的概述，请参见第 91 页中的“[了解子镜像状态以确定维护操作](#)”。

- 使用以下方法之一检查镜像或子镜像的状态。

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）⇒ "Properties"（属性）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 针对镜像运行 `metastat` 命令以查看每个子镜像的状态。

```
# metastat mirror
```

请参见第 121 页中的“[如何更改 RAID-1 卷选项](#)”以更改镜像的传送号、读取选项和写入选项。

有关检查设备状态的详细信息，请参见 [metastat\(1M\)](#)。

示例 11-13 检查 RAID-1 卷的状态

以下是 `metastat` 命令的输出样例。使用不带镜像名称的 `metastat` 命令可显示所有镜像的状态。

```
# metastatd70: Mirror
  Submirror 0: d71
    State: Okay
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 12593637 blocks

d71: Submirror of d70
  State: Okay
  Size: 12593637 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Reloc Hot Spare
    clt3d0s3              0      No  Okay      Yes
  Stripe 1:
    Device          Start Block  Dbase State      Reloc Hot Spare
    clt3d0s4              0      No  Okay      Yes
  Stripe 2:
    Device          Start Block  Dbase State      Reloc Hot Spare
    clt3d0s5              0      No  Okay      Yes

d0: Mirror
  Submirror 0: d1
    State: Okay
  Submirror 1: d2
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 5600 blocks

d1: Submirror of d0
  State: Okay
  Size: 5600 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Hot Spare
    c0t2d0s7              0      No  Okay

...
```

使用带有镜像名称参数的 `metastat` 命令可显示针对特定镜像的输出。

```
metastat d70
d70: Mirror
  Submirror 0: d71
    State: Okay
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 12593637 blocks

d71: Submirror of d70
```

```
State: Okay
Size: 12593637 blocks
Stripe 0:
  Device          Start Block  Dbase State      Reloc Hot Spare
  clt3d0s3              0      No  Okay        Yes
Stripe 1:
  Device          Start Block  Dbase State      Reloc Hot Spare
  clt3d0s4              0      No  Okay        Yes
Stripe 2:
  Device          Start Block  Dbase State      Reloc Hot Spare
  clt3d0s5              0      No  Okay        Yes
```

对于镜像中的每一个子镜像，`metastat` 命令将显示其状态、"Invoke" 行（如果存在错误）、已分配的热备用池（如果有）、大小（块数）以及有关子镜像中每个分片的信息。

## ▼ 如何更改 RAID-1 卷选项

开始之前 查看第 90 页中的“关于 RAID-1 卷选项”。

- 1 确保您具有 `root` 用户权限并且拥有所有数据的当前备份。
- 2 使用以下方法之一更改 RAID-1 选项。
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）⇒ "Properties"（属性）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用 `metaparam` 命令显示并更改镜像的选项。

```
# metaparam [mirror options] mirror
```

有关镜像选项的说明，请参见第 90 页中的“关于 RAID-1 卷选项”。另请参见 [metaparam\(1M\)](#) 手册页。

### 示例 11-14 更改 RAID-1 卷的读取策略

```
# metaparam -r geometric d30
# metaparam d30
d30: mirror current parameters are:
  Pass: 1
  Read option: geometric (-g)
  Write option: parallel (default)
```

在本示例中，`-r` 选项将镜像的读取策略更改为 `geometric`。

### 示例 11-15 更改 RAID-1 卷的写入策略

```
# metaparam -w serial d40
# metaparam d40
d40: mirror current parameters are:
    Pass: 1
    Read option: roundrobin (default)
    Write option: serial (-S)
```

在本示例中，-w 选项将镜像的写入策略更改为 serial。

### 示例 11-16 更改 RAID-1 卷的传送号

```
# metaparam -p 5 d50
# metaparam d50
d50: mirror current parameters are:
    Pass: 5
    Read option: roundrobin (default)
    Write option: parallel (default)
```

在本示例中，-p 选项将镜像的传送号更改为 5。

## ▼ 如何扩展 RAID-1 卷

开始之前 请阅读第 89 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有 root 用户权限并且拥有所有数据的当前备份。
- 2 使用以下方法之一扩展镜像。
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）->"Properties"（属性）并单击 "Submirror"（子镜像）选项卡。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用 `metattach` 命令将其他分片附加到每个子镜像。

```
# metattach submirror slice
```

必须对镜像中的每个子镜像都进行扩展。有关更多信息，请参见 `metattach(1M)` 手册页。

- 3 使用 `metattach` 命令，以便基于子镜像的大小重新计算镜像的大小。

```
# metattach mirror
```

### 示例 11-17 扩展包含已挂载文件系统的双向镜像

```
# metastat
d8: Mirror
    Submirror 0: d9
        State: Okay
    Submirror 1: d10
        State: Okay
...
# metattach d9 c0t2d0s5
d9: component is attached
# metattach d10 c0t3d0s5
d10: component is attached
# metattach d8
```

本示例说明如何通过将两个磁盘驱动器串联到镜像的两个子镜像来扩展已挂载的镜像文件系统。镜像的名称为 **d8**，其中包含两个名为 **d9** 和 **d10** 的子镜像。

**另请参见** 对于 UFS，请针对镜像卷运行 **growfs(1M)** 命令。请参见第 207 页中的“如何扩展文件系统”。

使用原始卷的应用程序（如数据库）必须有自己扩展添加的存储的方法。

## 响应 RAID-1 卷的组件故障

### ▼ 如何替换子镜像中的分片

**开始之前** 请阅读第 208 页中的“替换和启用 RAID-1 和 RAID-5 卷中的组件概述”和第 89 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有 root 用户权限并且拥有所有数据的当前备份。
- 2 使用 **metastat** 命令查看 RAID-1 卷和关联的子镜像的状态。

```
# metastat mirror-name
```

- 3 使用以下方法之一来替换子镜像中的分片。

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）⇒ "Properties"（属性）并单击 "Submirror"（子镜像）选项卡。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 **metareplace** 命令替换子镜像中的分片：

```
# metareplace mirror-name component-name
```

- *mirror-name* 是要创建的卷的名称。
- *component-name* 用于指定要替换的组件的名称。

*mirror-name*            指定要创建的卷的名称

*component-name*      指定要替换的组件的名称

有关更多信息，请参见以下示例和 [metainit\(1M\)](#) 手册页。

### 示例 11-18 替换镜像中出现故障的分片

以下示例说明在未将系统配置为使用热备用池来自动替换出现故障的磁盘时，如何替换出现故障的分片。有关使用热备用池的详细信息，请参见第 16 章，[热备用池（概述）](#)。

```
# metastat d6
d6: Mirror
   Submirror 0: d16
     State: Okay
   Submirror 1: d26
     State: Needs maintenance
...
d26: Submirror of d6
   State: Needs maintenance
   Invoke: metareplace d6 c0t2d0s2 <new device>
...
# metareplace d6 c0t2d0s2 c0t2d2s2
d6: device c0t2d0s2 is replaced with c0t2d2s2
```

metastat 命令确认了镜像 d6 具有子镜像 d26，且其中一个分片处于 "Needs Maintenance"（需要维护）状态。metareplace 命令使用系统中的其他可用分片来替换 metastat 输出的 "Invoke" 行中所指定的分片。系统确认了该分片已被替换，并开始重新同步子镜像。

## ▼ 如何替换子镜像

开始之前 请阅读第 208 页中的“替换和启用 RAID-1 和 RAID-5 卷中的组件概述”和第 89 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有 root 用户权限并且拥有所有数据的当前备份。
- 2 使用 metastat 命令查看 RAID-1 卷和关联的子镜像的状态。  
# metastat *mirror-name*
- 3 使用以下方法之一替换子镜像。

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择该镜像。然后，选择 "Action"（操作）⇒ "Properties"（属性）并单击 "Submirror"（子镜像）选项卡。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。

- 使用 `metadetach`、`metaclear`、`metainit` 和 `metattach` 命令替换整个子镜像。

- a. 使用 `metadetach` 命令从镜像中分离出现故障的子镜像。

```
# metadetach -f mirror-name submirror
-f                强制执行分离
mirror-name       指定镜像的名称
submirror         指定要分离的子镜像
```

- b. 使用 `metaclear` 命令删除子镜像。

```
# metaclear -f submirror
-f                强制删除子镜像
submirror         指定要删除的子镜像
```

- c. 使用 `metainit` 命令创建新的子镜像。

```
# metainit volume-name number-of-stripes components-per-stripe component-name
volume-name       指定要创建的卷的名称。有关命名卷的信息，请参见
                  第 38 页中的“卷名称”。
number-of-stripes 指定要创建的条带数。
components-per-stripe 指定每个条带应包含的组件数。
component-names   指定所使用的组件的名称。本示例使用根分片
                  c0t0d0s0。
```

- d. 使用 `metattach` 命令附加新的子镜像。

```
# metattach mirror submirror
```

### 示例 11-19 替换镜像中的子镜像

以下示例说明如何替换活动镜像中的子镜像。

```
# metastat d20
d20: Mirror
    Submirror 0: d21
        State: Okay
    Submirror 1: d22
        State: Needs maintenance
...
# metadetach -f d20 d22
d20: submirror d22 is detached
```

```
# metaclear -f d22
d22: Concat/Stripe is cleared
# metainit d22 2 1 c1t0d0s2 1 c1t0d1s2
d22: Concat/Stripe is setup
# metattach d20 d22
d20: components are attached
```

在本示例中，`metastat` 命令确认了双向镜像 `d20` 的子镜像 `d22` 处于 "Needs Maintenance"（需要维护）状态。在这种情况下，需要清除整个子镜像，然后重新创建。`metadetach` 命令通过使用 `-f` 选项（强制执行分离）将出现故障的子镜像从镜像中分离。`metaclear` 命令清除子镜像。`metainit` 命令使用新的分片重新创建子镜像 `d22`。最后，通过 `metattach` 命令附加重新生成的子镜像。系统将开始自动重新同步镜像。

新卷 `d22` 的具体配置取决于要替换的组件。此处所示的串联足以替换串联。但是，使用串联来替换条带并不理想，因为可能会影响性能。

如果是单向镜像，则会暂时失去数据冗余功能。

## 删除 RAID-1 卷（取消镜像）

### ▼ 如何取消镜像文件系统

对于可以在系统运行时取消挂载的文件系统，可以使用此过程取消对它的镜像。要取消对根 (`/`)、`/var`、`/usr`、`swap` 或其他任何在系统运行时无法卸载的文件系统的镜像，请参见第 128 页中的“如何取消对无法取消挂载的文件系统的镜像”。

开始之前 请阅读第 89 页中的“创建和维护 RAID-1 卷”。

1 确保您具有 `root` 用户权限并且拥有所有数据的当前备份。

2 验证是否至少有一个子镜像处于 "Okay"（正常）状态。

```
# metastat mirror
```

3 取消挂载文件系统。

```
# umount /file-system
```

4 分离将继续用于文件系统的子镜像。

```
# metadetach mirror submirror
```

有关更多信息，请参见 [metadetach\(1M\)](#) 手册页。

## 5 清除镜像和剩余的子组件。

```
# metaclear -r mirror
```

有关更多信息，请参见 [metaclear\(1M\)](#) 手册页。

6 如有必要，请编辑 `/etc/vfstab` 文件，使其使用 [步骤 4](#) 中分离出的组件。

## 7 重新挂载文件系统。

```
# mount /file-system
```

示例 11-20 取消对 `/opt` 文件系统的镜像

```
# metastat d4
d4: Mirror
  Submirror 0: d2
    State: Okay
  Submirror 1: d3
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 2100735 blocks (1.0 GB)

d2: Submirror of d4
  State: Okay
  Size: 2100735 blocks (1.0 GB)
  Stripe 0:
    Device      Start Block  Dbase      State Reloc Hot Spare
    c0t0d0s0      0         No         Okay    Yes

d3: Submirror of d4
  State: Okay
  Size: 2100735 blocks (1.0 GB)
  Stripe 0:
    Device      Start Block  Dbase      State Reloc Hot Spare
    c1t0d0s0      0         No         Okay    Yes

...
# umount /opt
# metadetach d4 d2
d4: submirror d2 is detached
# metaclear -r d4
d4: Mirror is cleared
d3: Concat/Stripe is cleared
    (Edit the /etc/vfstab file so that the entry for /opt is changed from d4 to the underlying slice or volume)
# mount /opt
```

在本示例中，`/opt` 文件系统由双向镜像 `d4` 组成。该镜像的子镜像为 `d2` 和 `d3`。子镜像由分片 `/dev/dsk/c0t0d0s0` 和 `/dev/dsk/c1t0d0s0` 组成。`metastat` 命令验证了至少有一

个子镜像处于 "Okay"（正常）状态。（必须首先修复其子镜像都不处于 "Okay"（正常）状态的镜像。）已取消挂载此文件系统。然后，分离子镜像 d2。metaclear -r 命令用于删除镜像和另一个子镜像 d3。

接下来，将更改 /etc/vfstab 文件中的 /opt 条目以引用底层分片。

在本示例中，/etc/vfstab file 包含对应于 /opt 文件系统的以下条目：

```
/dev/md/dsk/d4 /dev/md/rdisk/d4 /opt ufs 2 yes -
```

将此项更改为如下内容：

```
/dev/md/dsk/d2 /dev/md/rdisk/d2 /opt ufs 2 yes -
```

通过使用子镜像名称，可以继续在该卷上挂载文件系统。最后，将重新挂载 /opt 文件系统。

通过在 /etc/vfstab 文件中使用 d2（而不是 d4），取消了镜像。由于 d2 由单个分片构成，如果不希望设备支持卷，则可以在该分片（其名称为 /dev/dsk/c0t0d0s0）上挂载文件系统。

## ▼ 如何取消对无法取消挂载的文件系统的镜像

使用此任务可以取消对正常系统运行过程中无法取消挂载的文件系统的镜像，其中包括根 (/)、/usr、/opt 和 swap。

- 1 确保您具有 root 用户权限并且拥有所有数据的当前备份。

- 2 验证是否至少有一个子镜像处于 "Okay"（正常）状态。

```
# metastat mirror
```

- 3 分离将继续用于文件系统的子镜像。

```
# metadetach mirror submirror
```

有关更多信息，请参见 [metadetach\(1M\)](#) 手册页。

- 4 根据要取消镜像的文件系统，使用以下命令之一：

- 对于 /usr、/opt 或 swap 文件系统，将 /etc/vfstab 文件中的文件系统条目更改为使用非 Solaris Volume Manager 设备（分片）。
- 仅适用于根 (/) 文件系统：运行 metaroot 命令。

```
# metaroot rootslice
```

有关更多信息，请参见 [metaroot\(1M\)](#) 手册页。

**5 重新引导系统。**

```
# reboot
```

**6 清除剩余的镜像和子镜像。**

```
# metaclear -r mirror
```

有关更多信息，请参见 [metaclear\(1M\)](#) 手册页。

**示例 11-21 取消对根 (/) 文件系统的镜像**

```
# metastat d0
d0: Mirror
  Submirror 0: d10
    State: Okay
  Submirror 1: d20
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 2100735 blocks (1.0 GB)

d10: Submirror of d0
  State: Okay
  Size: 2100735 blocks (1.0 GB)
  Stripe 0:
    Device      Start Block  Dbase      State Reloc Hot Spare
    c0t3d0s0      0          No         Okay   Yes

d20: Submirror of d0
  State: Okay
  Size: 2100735 blocks (1.0 GB)
  Stripe 0:
    Device      Start Block  Dbase      State Reloc Hot Spare
    c1t3d0s0      0          No         Okay   Yes

# metadetach d0 d20
d0: submirror d20 is detached
# metaroot /dev/dsk/c0t3d0s0
# reboot
...
# metaclear -r d0
d0: Mirror is cleared
d10: Concat/Stripe is cleared
# metaclear d20
d20: Concat/Stripe is cleared
```

在本示例中，根 (/) 文件系统为双向镜像 **d0**。该镜像的子镜像为 **d10** 和 **d20**。子镜像由分片 **/dev/dsk/c0t3d0s0** 和 **/dev/dsk/c1t3d0s0** 组成。**metastat** 命令验证了至少有一个子镜像处于 "Okay"（正常）状态。（必须首先修复其子镜像都不处于 "Okay"（正常）状态的镜像。）子镜像 **d20** 被分离，从而使 **d0** 成为单向镜像。

*rootslice* 为根 (/) 文件系统所在的分片。*metaroot* 命令将使用要从其引导系统的 *rootslice* 来运行。此命令会编辑 */etc/system* 和 */etc/vfstab* 文件。此命令会删除用来指定根 (/) 文件系统的镜像的信息。

重新引导系统之后，通过 *metaclear -r* 命令删除了镜像和另一个子镜像 *d10*。最后一个 *metaclear* 命令清除子镜像 *d20*。

## 示例 11-22 取消对 swap 文件系统的镜像

```
# metastat d1
d1: Mirror
    Submirror 0: d11
        State: Okay
    Submirror 1: d21
        State: Okay
...
# metadetach d1 d21
d1: submirror d21 is detached
    (Edit the /etc/vfstab file to change the entry for swap from metadvice to slice name)
# reboot
...
# metaclear -r d1
d1: Mirror is cleared
d11: Concat/Stripe is cleared
# metaclear d21
d21: Concat/stripes is cleared
```

在本示例中，swap 文件系统由双向镜像 *d1* 组成。该镜像的子镜像为 *d11* 和 *d21*。子镜像由分片 */dev/dsk/c0t3d0s1* 和 */dev/dsk/c1t3d0s1* 组成。*metastat* 命令验证了至少有一个子镜像处于 "Okay"（正常）状态。（必须首先修复其子镜像都不处于 "Okay"（正常）状态的镜像。）子镜像 *d21* 被分离，从而使 *d1* 成为单向镜像。接下来，会编辑 */etc/vfstab* 文件，将 swap 条目更改为引用子镜像 *d21* 中的分片。

在本示例中，*/etc/vfstab* file 包含对应于 swap 文件系统的以下条目：

```
/dev/md/dsk/d4  /dev/md/rdisk/d4  /opt ufs  2  yes  -
/dev/md/dsk/d1  - - swap  - no  -
```

将此项更改为如下内容：

```
/dev/dsk/c0t3d0s1 - - swap  - no  -
```

重新引导系统之后，通过 *metaclear -r* 命令删除了镜像和另一个子镜像 *d11*。最后一个 *metaclear* 命令清除子镜像 *d21*。

## 备份 RAID-1 卷上的数据

Solaris Volume Manager 并不打算成为“备份产品”。但 Solaris Volume Manager 确实提供了一种备份镜像数据且不会导致以下任何一种情况的方法：

- 卸载镜像
- 使整个镜像脱机
- 停止系统
- 拒绝用户访问数据

Solaris Volume Manager 通过首先使其中一个子镜像脱机来备份镜像数据。在备份过程中，镜像将暂时不可用。一旦备份完成，子镜像将立即恢复联机并重新同步。

---

注 - UFS 快照功能提供了一种不使文件系统脱机即可备份系统的备选方法。您可以在不分离子镜像的情况下执行备份，这样也避免了以后由于重新同步镜像而导致的性能损失。在使用 UFS 快照功能执行备份之前，请确保 UFS 文件系统中有足够的可用空间。有关更多信息，请参见《[System Administration Guide: Devices and File Systems](#)》中的第 20 章“[Using UFS Snapshots \(Tasks\)](#)”。

---

### ▼ 如何对 RAID-1 卷执行联机备份

您可以针对除根 (/) 文件系统之外的任何文件系统使用此过程。请注意，这种类型的备份会创建活动文件系统的“快照”。根据文件系统处于写入锁定状态时的使用方式，所备份的某些文件可能与磁盘上的实际文件不对应。

对于此过程，请注意以下限制：

- 如果针对双向镜像使用此过程，请注意，在使一个子镜像脱机以进行备份时，会失去数据冗余功能。多向镜像不会出现此问题。
- 备份完成之后，对重新附加的子镜像进行重新同步时，会产生一些系统开销。

本过程中的概括性步骤如下所示：

- 对文件系统进行写入锁定（仅适用于 UFS）。请勿锁定根 (/) 文件系统。
- 将高速缓存中的所有数据刷新到磁盘。
- 使用 `metadetach` 命令从镜像中分离某个子镜像
- 解除对文件系统的锁定
- 使用 `fsck` 命令检查已分离的子镜像中的文件系统
- 备份分离的子镜像中的数据
- 使用 `metattach` 命令将分离的子镜像重新放回镜像中

---

注 - 如果要定期使用上述过程，请将其置入脚本中以方便使用。

---

---

**提示** – 对于此过程，比较安全的做法是向镜像中附加第三个或第四个子镜像，允许该子镜像重新同步，然后使用该子镜像进行备份。此方法可确保一直保持数据冗余功能。

---

**1 验证镜像是否处于 "Okay" ( 正常 ) 状态。**

应首先恢复处于 "Maintenance" ( 维护 ) 状态的镜像。

```
# metastat mirror
```

**2 将高速缓存中的数据 and UFS 日志记录数据刷新到磁盘上，并对文件系统进行写入锁定。**

```
# /usr/sbin/lockfs -w mount-point
```

只有 UFS 卷需要进行写入锁定。如果将该卷设置为数据库管理软件或其他某个应用程序的原始设备，则不必运行 `lockfs` 命令。但是，您可能希望运行由供应商提供的适当实用程序来刷新所有缓冲区并锁定访问。




---

**注意** – 请不要对根 (/) 文件系统进行写入锁定。对根 (/) 文件系统进行写入锁定会导致系统挂起。如果要备份根 (/) 文件系统，请跳过此步骤。

---

**3 从镜像分离一个子镜像。**

```
# metadetach mirror submirror
```

*mirror*            是镜像的卷名。

*submirror*        是要分离的子镜像 ( 卷 ) 的卷名。

系统将继续从另一个子镜像中进行读取。第一次写入之后，镜像将不再同步。当在 [步骤 7](#) 中重新附加已分离的子镜像后，这种不一致性问题将得到更正。

**4 解除对文件系统的锁定，从而允许继续写入。**

```
# /usr/sbin/lockfs -u mount-point
```

根据 [步骤 2](#) 中所使用的与供应商相关的实用程序，您可能需要执行必要的解除锁定过程。

**5 使用 `fsck` 命令检查已分离的子镜像中的文件系统此步骤可确保执行“干净”的备份。**

```
# fsck /dev/md/rdisk/name
```

**6 对脱机的子镜像执行备份。**

使用 `ufsdump` 命令或常用的备份实用程序。有关使用 `ufsdump` 命令执行备份的信息，请参见第 275 页中的“使用 `ufsdump` 命令对已挂载的文件系统执行备份”。

---

注 – 要确保正确的备份，请使用**原始**卷名（如 `/dev/md/rdisk/d4`）。使用原始卷名可访问大于 2 GB 的存储。

---

## 7 附加子镜像。

```
# metattach mirror submirror
```

Solaris Volume Manager 会自动开始将子镜像与镜像重新进行同步。

### 示例 11-23 对 RAID-1 卷执行联机备份

本示例使用镜像 `d1`。该镜像由子镜像 `d2`、`d3` 和 `d4` 构成。子镜像 `d3` 将在子镜像 `d2` 和 `d4` 保持联机时进行分离和备份。镜像中的文件系统为 `/home1`。

```
# metastat d1
d1: Mirror
   Submirror 0: d2
       State: Okay
   Submirror 1: d3
       State: Okay
   Submirror 1: d4
       State: Okay
...

# /usr/sbin/lockfs -w /home1
# metadetach d1 d3
# /usr/sbin/lockfs -u /home1
# /usr/sbin/fsck /dev/md/rdisk/d3
(Perform backup using /dev/md/rdisk/d3)
# metattach d1 d3
```



## 软分区（概述）

---

本章提供了有关 Solaris Volume Manager 软分区的信息。有关相关任务的信息，请参见第 13 章，软分区（任务）。

本章包含以下信息：

- 第 135 页中的“软分区概述”
- 第 136 页中的“软分区的配置原则”

### 软分区概述

随着磁盘存储容量增加，磁盘阵列需向 Solaris 系统提供更大的逻辑设备。要创建更便于管理的文件系统或分区大小，用户可能需要将磁盘或逻辑卷细分为八个以上的分区。Solaris Volume Manager 的软分区功能可以满足此需求。

Solaris Volume Manager 对于每个磁盘集最多支持 8192 个逻辑卷。此值包括本地磁盘集（或未指定的磁盘集）。Solaris Volume Manager 会在需要时动态配置卷。

您可以通过软分区将磁盘分片或逻辑卷根据需要划分为多个分区。您必须为所划分的每个分区（或称**软分区**）提供一个名称，就像为其他存储卷（如条带或镜像）提供名称一样。对软分区进行命名后，只要该软分区不包括在其他卷中，应用程序（包括文件系统）就可以访问它。一旦将该软分区包括在某个卷中，就不应再直接访问该软分区。

可以将软分区直接放置在磁盘分片之上，也可以放置在镜像、条带或 RAID-5 卷之上。软分区不能同时位于其他卷之上和之下。例如，如果在某个条带上构建了一个软分区，则在该软分区上不允许再构建镜像。

对于文件系统和其他应用程序来说，软分区显示为一个连续的逻辑卷。但实际上，软分区是由一系列**盘区**组成的，这些盘区可能位于底层介质上的任意位置。除了软分区之外，磁盘上的盘区标头（也称为**系统恢复数据区域**）也会记录有关软分区的信息，以便于在发生灾难性系统故障进行恢复。

## 软分区的配置原则

- 用于软分区的分片不能用于其他用途。
- 如果对某一磁盘进行分区并在所生成的分片上构建文件系统，则日后要扩展分片，必须修改或销毁该磁盘格式。而软分区的行为则与此不同。您可以在不移动或销毁其他软分区上的数据的情况下将软分区最多扩展至基础设备上的空间量。
- 虽然从技术上来说可以手动将软分区的盘区放置在磁盘上的任意位置，但应允许系统自动放置它们。有关手动放置的盘区的示例，请参见第 198 页中的“[查看 Solaris Volume Manager 配置](#)”中的 `metastat` 命令输出。
- 您可以在任意分片上构建软分区。但是，在磁盘级别使用软分区的最高效方式是，先创建一个占用整个磁盘空间的分片，然后在该分片上创建软分区。
- 软分区的最大大小不能超过构建它的分片或逻辑卷的大小。鉴于这一限制，您应首先在磁盘分片上构建一个卷，然后再在该卷上构建软分区。通过这一策略，您可以在日后向卷中添加组件，然后根据需要扩展软分区。
- 为了最大限度地提高灵活性和高可用性，请先在磁盘分片上构建 RAID-1（镜像）或 RAID-5 卷，然后再在该镜像或 RAID-5 卷上创建软分区。

## 方案—软分区

软分区可提供一些工具，可将较大的存储空间细分为多个更便于管理的空间。例如，在其他方案（如第 93 页中的“[方案—RAID-1 卷（镜像）](#)”或第 149 页中的“[方案—RAID-5 卷](#)”）中，大型存储聚合会产生数万兆字节的冗余存储。但是，许多可能的方案最初并不需要这么多的空间。通过软分区，您可以将这一存储空间细分为多个更便于管理的分区。其中每个分区都可以包含一个完整的文件系统。例如，您可以在 RAID-1 或 RAID-5 卷上创建 1000 个软分区，以使每个用户都可以在一个单独的文件系统上有一个起始目录。如果用户需要更多空间，只需扩展软分区即可。

## 软分区（任务）

---

本章提供有关执行与 Solaris Volume Manager 软分区关联的任务的信息。有关软分区的概念性信息，请参见第 12 章，软分区（概述）。

### 软分区（任务列表）

以下任务列表列出了管理 Solaris Volume Manager 软分区所需的过程。

任务	说明	参考
创建软分区	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建软分区。	<a href="#">第 137 页中的“如何创建软分区”</a>
检查软分区的状态	使用 Solaris Volume Manager GUI 或 <code>metastat</code> 命令检查软分区的状态。	<a href="#">第 139 页中的“如何检查软分区的状态”</a>
扩展软分区	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令扩展软分区。	<a href="#">第 140 页中的“如何扩展软分区”</a>
删除软分区	使用 Solaris Volume Manager GUI 或 <code>metaclear</code> 命令删除软分区。	<a href="#">第 140 页中的“如何删除软分区”</a>

### 创建软分区

#### ▼ 如何创建软分区

开始之前 请查看第 136 页中的“软分区的配置原则”。

- 使用以下方法之一创建软分区：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择 "Action"（操作）⇒ "Create Volume"（创建卷）。然后，按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要创建软分区，请使用以下格式的 `metainit` 命令：

```
# metainit [-s diskset] soft-partition -p [-e] component size
```

<code>-sdiskset</code>	指定要使用的磁盘集。如果未指定 <code>-s</code> ，将使用本地（缺省）磁盘集。
<code>-p</code>	指定要配置软分区。
<code>-e</code>	指定应重新格式化整个磁盘。格式化磁盘后会提供占用大部分磁盘空间的分片 0。格式化磁盘后还会提供一个最小为 4 MB 的分片 7。分片 7 包含一个状态数据库副本。
<code>soft-partition</code>	指定软分区的名称。该名称采用 <code>dnnn</code> 的形式，其中， <code>nnn</code> 是一个介于 0 到 8192 之间的数字。
<code>component</code>	指定要在其中创建软分区的磁盘、分片或逻辑卷。由于软分区标头写在该组件的开头，因此组件上的所有现有数据都将被销毁。
<code>size</code>	指定软分区的大小。该大小应指定为一个后面跟有以下各项之一的数值： <ul style="list-style-type: none"> <li>■ M 或 m，表示兆字节</li> <li>■ G 或 g，表示千兆字节</li> <li>■ T 或 t，表示兆兆字节</li> <li>■ B 或 b，表示块（扇区）</li> </ul>

有关更多信息，请参见以下示例和 [metainit\(1M\)](#) 手册页。

### 示例 13-1 创建软分区

在以下示例中，将在 `c1t3d0s2` 上创建一个名为 `d20` 的 4 GB 软分区。

```
# metainit d20 -p c1t3d0s2 4g
```

### 示例 13-2 将整个磁盘用作软分区

以下示例将创建一个软分区并格式化磁盘 `c1t2d0`。此操作会销毁该磁盘上的所有数据，并在分片 0 上创建新的软分区。

```
# metainit d7 -p -e c1t2d0 1G
```

# 维护软分区

维护软分区与维护其他的逻辑卷没有任何不同。

## ▼ 如何检查软分区的状态

开始之前 请阅读第 136 页中的“软分区的配置原则”。

- 使用以下方法之一检查软分区的状态：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择要监视的软分区。然后，选择 "Action"（操作）⇒ "Properties"（属性）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 要查看现有配置，请使用以下格式的 `metastat` 命令：

```
# metastat soft-partition
soft-partition 指定要检查的分区名称。
```

### 示例 13-3 检查软分区的状态

在以下示例中，将检查软分区 `d1` 的状态。该软分区包含两个盘区且构建于 RAID-1 卷 `d100` 上。

```
# metastat d1
d1: soft partition
  component: d100
  state: OKAY
  size: 42674285 blocks
      Extent      Start Block      Block Count
      0              10234          40674285
      1             89377263         2000000
d100: Mirror
  Submirror 0: d10
  State: OKAY
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 426742857 blocks
d10: Submirror of d100
  State: OKAY
  Hot spare pool: hsp002
  Size: 426742857 blocks
  Stripe 0: (interlace: 32 blocks)
      Device      Start Block  Dbase State      Hot Spare
      c3t3d0s0      0           No      Okay
```

## ▼ 如何扩展软分区

如果某个软分区上未构建其他逻辑卷，则可以向该软分区添加空间。查找可用空间并用来扩展该分区。不会移动现有数据。

---

注 - 如果软分区已用于创建其他卷（例如，如果该软分区是某个 RAID-0 卷的组件），则无法扩展该软分区。在大多数情况下，通过将其他卷串联到包含设备上可以实现相同的目标（即，为包含软分区的设备提供更多的空间）。有关更多信息，请参见第 80 页中的“扩展存储容量”。

---

开始之前 请阅读第 136 页中的“软分区的配置原则”。

- 使用以下方法之一扩展软分区：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择要扩展的软分区，然后选择 "Action"（操作）⇒ "Properties"（属性）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 要向软分区添加空间，请使用以下格式的 `metattach` 命令：

```
# metattach [-s diskset] soft-partition size
diskset      指定软分区所在的磁盘集的名称。
soft-partition 指定现有软分区的名称。
size         指定要添加的存储空间量。
```

### 示例 13-4 扩展软分区

以下示例说明如何将空间附加到软分区。然后可在该软分区联机并已挂载时使用 `growfs` 命令扩展文件系统。

```
# mount /dev/md/dsk/d20 /home2
# metattach d20 10g
# growfs -M /home2 /dev/md/rdsk/d20
```

有关 `growfs` 命令的更多信息，请参见第 206 页中的“使用 `growfs` 命令扩展文件系统”。

## ▼ 如何删除软分区

开始之前 请阅读第 136 页中的“软分区的配置原则”。

- 使用以下方法之一删除软分区：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择要删除的软分区。然后选择 "Action"（操作）⇒ "Properties"（属性）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 要删除软分区，请使用以下任意一种格式的 `metaclear` 命令：

```
# metaclear [-s diskset] component
# metaclear [-s diskset] -r soft-partition
# metaclear [-s diskset] -p component
```

*diskset*            指定软分区所在的磁盘集。

*soft-partition*    指定要删除的软分区。

-r                    指定按递归方式删除逻辑卷，但不删除其他卷所依赖的卷。

-p                    指定清除指定组件上的所有软分区，但打开的软分区除外。

*component*        指定要清除其中所有软分区的组件。

### 示例 13-5    删除软分区

本示例说明如何删除 `c1t4d2s0` 上的所有软分区。

```
# metaclear -p c1t4d2s0
```



## RAID-5 卷（概述）

---

本章提供了有关 Solaris Volume Manager 的 RAID-5 卷的概念性信息。有关执行相关任务的信息，请参见第 15 章，[RAID-5 卷（任务）](#)。

本章包含以下信息：

- 第 143 页中的“RAID-5 卷概述”
- 第 146 页中的“创建 RAID-5 卷的背景信息”
- 第 147 页中的“检查 RAID-5 卷状态概述”
- 第 149 页中的“在 RAID-5 卷中替换和启用分片概述”
- 第 149 页中的“方案—RAID-5 卷”

### RAID-5 卷概述

RAID 级别 5 类似于条带化，但其奇偶校验数据分布在所有组件（磁盘或逻辑卷）中。如果某个组件出现故障，则可以根据其他组件上分布的数据和奇偶校验信息来重新生成该组件上的数据。在 Solaris Volume Manager 中，**RAID-5 卷**是支持 RAID 级别 5 的卷。

RAID-5 卷使用相当于卷中的一个组件的存储容量来存储冗余信息（奇偶校验）。该奇偶校验信息包含有关在 RAID-5 卷的其余组件上存储的用户数据的信息。也就是说，如果您有三个组件，则会使用与一个组件同等的容量来存储奇偶校验信息。如果您有五个组件，则会使用与一个组件同等的容量来存储奇偶校验信息。奇偶校验信息分布在卷的所有组件中。与镜像类似，RAID-5 卷也可以提高数据可用性，但其所需的硬件成本最低，并且对写操作的影响适中。但是，您不能对根 (/)、/usr 和 swap 文件系统或其他现有文件系统使用 RAID-5 卷。

在替换现有组件时，Solaris Volume Manager 将自动重新同步 RAID-5 卷。如果系统发生故障或出现紧急情况，则 Solaris Volume Manager 也会在重新引导期间重新同步 RAID-5 卷。

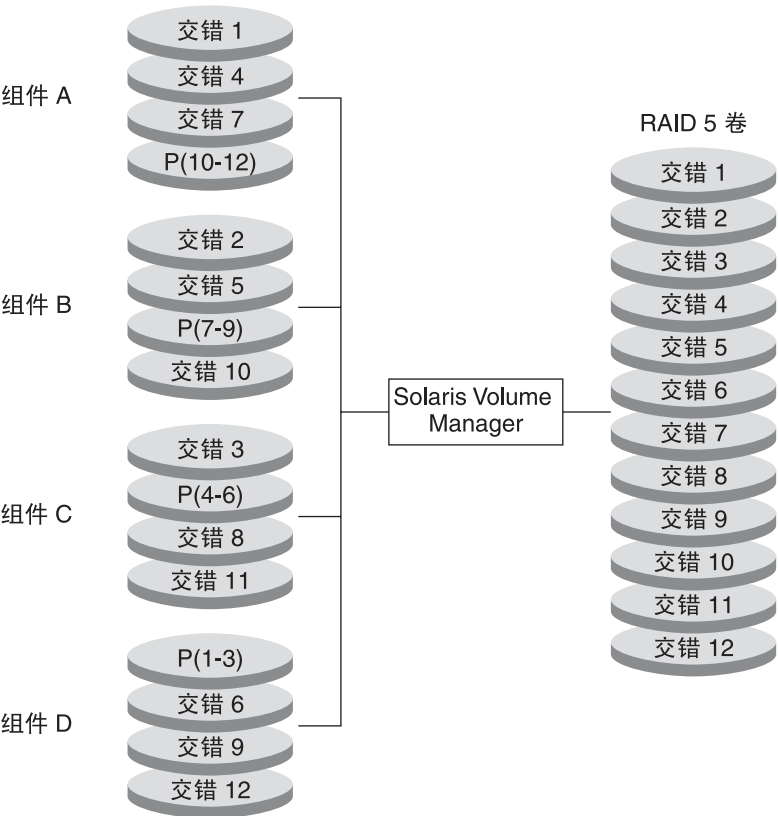
# 示例—RAID-5 卷

图 14-1 显示了一个由四个磁盘（组件）组成的 RAID-5 卷。

前三个数据段将写入到组件 A（交错 1）、组件 B（交错 2）和组件 C（交错 3）中。而下一个要写入的数据段则为奇偶校验段。此奇偶校验段将写入组件 D(P 1-3) 中。此段通过对前三个数据段进行异或运算而构成。下面三个数据段将写入到组件 A（交错 4）、组件 B（交错 5）和组件 D（交错 6）中。然后，另一个奇偶校验段将写入组件 C(P 4-6) 中。

通过这种写入数据段和奇偶校验段的模式，数据和奇偶校验将分布在 RAID-5 卷的所有磁盘中。每个驱动器可以单独进行读取。奇偶校验可防止发生单磁盘故障。如果本示例中的每个磁盘都为 2 GB，则 RAID-5 卷的总容量将为 6 GB。相当于向奇偶校验分配一个驱动器的空间。

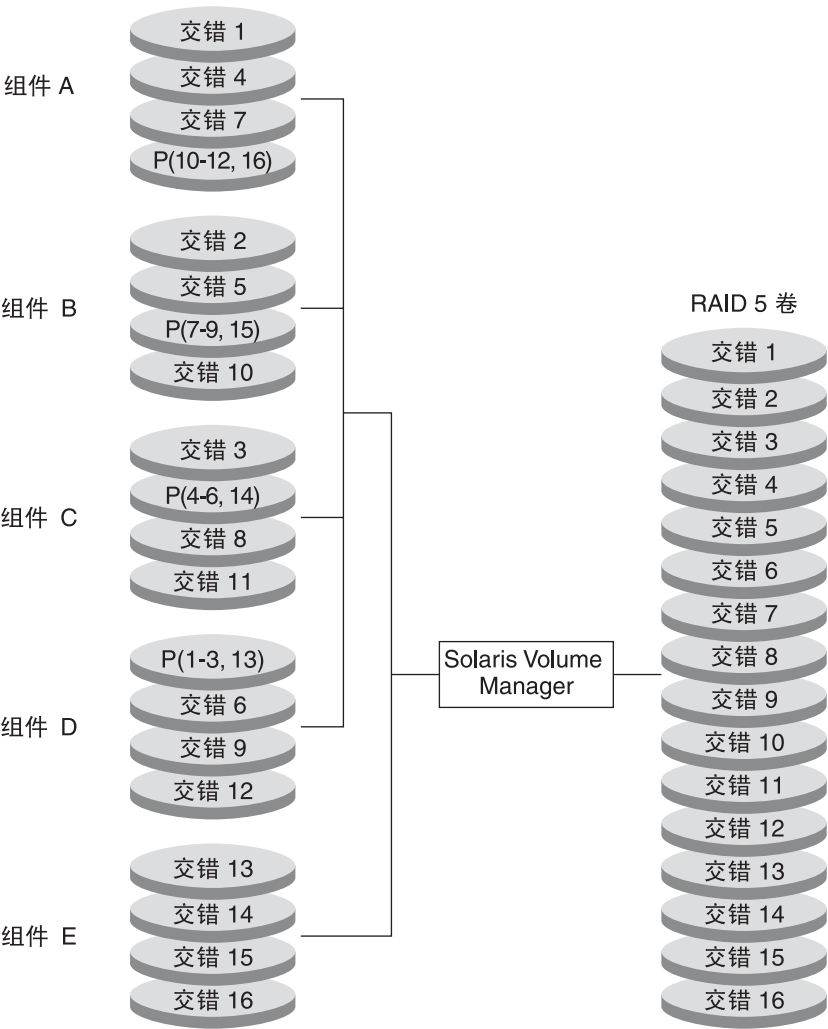
图 14-1 RAID-5 卷示例



# 示例—串联（扩展）RAID-5 卷

下图显示了一个 RAID-5 卷的示例，该卷最初包含四个磁盘（组件）。该图已将第五个磁盘动态串联到该卷，以扩展 RAID-5 卷。

图 14-2 扩展的 RAID-5 卷示例



在最初创建 RAID-5 卷时会分配奇偶校验区域。虽然实际的奇偶校验块分布在所有原始组件中以分散 I/O，但系统仅会为奇偶校验分配相当于一个组件的空间。在向 RAID-5

卷串联其他组件后，附加的空间将全部用于存储数据。不会分配新的奇偶校验块。但是，所串联组件上的数据会包括在奇偶校验计算中，因此该数据可以受到保护，以防止发生单设备故障。

串联的 RAID-5 卷不适合长期使用。可以先使用串联的 RAID-5 卷，直到可以重新配置更大的 RAID-5 卷为止。之后，可以将数据复制到更大的卷中。

---

注 - 向 RAID-5 卷添加新组件时，Solaris Volume Manager 会将该组件中的所有块清零。此过程可确保奇偶校验可以保护新数据。在向附加的空间写入数据时，Solaris Volume Manager 会将该数据包括在奇偶校验计算中。

---

## 创建 RAID-5 卷的背景信息

使用 RAID-5 卷时，请考虑第 146 页中的“RAID-5 卷的要求”和第 146 页中的“RAID-5 卷的原则”。许多条带化原则也适用于 RAID-5 卷配置。请参见第 75 页中的“RAID-0 卷要求”。

### RAID-5 卷的要求

- RAID-5 卷必须至少包含三个组件。但是，RAID-5 卷包含的组件越多，在某个组件发生故障时读写操作所花费的时间就越长。
- 不能对 RAID-5 卷进行条带化、串联或镜像。
- 不要使用包含现有文件系统的组件创建 RAID-5 卷。否则会在 RAID-5 初始化过程中删除数据。
- 创建 RAID-5 卷时，可以定义交错值。如果未指定交错值，则该值将缺省为 16 KB。该值对于大多数应用程序来说是合理的。
- RAID-5 卷（无热备件）只能处理单组件故障。
- 创建 RAID-5 卷时，请使用不同控制器中的组件。控制器以及相关电缆比磁盘更容易发生故障。
- 使用大小相同的组件。使用大小不同的组件创建 RAID-5 卷会出现未使用的磁盘空间。

### RAID-5 卷的原则

- 鉴于奇偶校验计算的复杂性，写入量超过大约 20% 的卷或许不应为 RAID-5 卷。如果需要对频繁写入的卷上实现数据冗余，请考虑使用镜像。
- 如果 RAID-5 卷中的不同组件驻留在不同的控制器上，并且主要对该卷进行大型顺序访问，则将交错值设置为 32 KB 可能会提高性能。

- 可以通过向卷中串联更多组件来扩展 RAID-5 卷。向现有的 RAID-5 卷中串联新组件会降低卷的总体性能，因为串联的数据是顺序的。数据不会在所有组件之间进行条带化。该卷的原始组件会在所有组件之间对数据和奇偶校验进行条带化。而串联的组件则缺少这种条带化。但是，在出现错误后仍可恢复数据，因为在组件 I/O 过程中会使用奇偶校验。生成的 RAID-5 卷将继续处理单组件故障。  
此外，串联的组件还具有一个不同之处，即，它们不会在任何区域中对奇偶校验进行条带化。因此，该组件的所有内容都可用来存储数据。  
串联组件后，对于大型或顺序写入来说，任何性能增强功能都将丢失。
- 您无需在创建 RAID-5 卷时“清空”数据块。为此，请执行以下操作之一：
  - 使用带有 -k 选项的 `metainit` 命令。-k 选项会重新创建 RAID-5 卷，而不对其进行初始化，然后将磁盘块设置为 "Okay"（正常）状态。该选项存在潜在危险，因为卷中磁盘块上如果存在任何错误，将导致 Solaris Volume Manager 出现不可预测的行为，包括可能会出现虚构数据。
  - 初始化该设备并从磁带恢复数据。有关更多信息，请参见 `metainit(1M)` 手册页。

## 检查 RAID-5 卷状态概述

您可以通过查看 RAID-5 卷的卷状态和分片状态来检查该卷的状态。在对 RAID-5 卷错误进行故障排除时，分片状态可提供最具体的信息。而 RAID-5 卷状态则仅提供一般状态信息，例如，"Okay"（正常）或 "Maintenance"（维护）。

如果 RAID-5 卷状态报告 "Maintenance"（维护）状态，请参阅分片状态。分片状态会具体报告分片是处于 "Maintenance"（维护）状态，还是处于 "Last Erred"（最近出错）状态。根据分片是处于 "Maintenance"（维护）状态还是处于 "Last Erred"（最近出错）状态，您可以执行不同的恢复操作。如果分片只处于 "Maintenance"（维护）状态，则可以对分片进行修复，而不会丢失数据。如果有的分片处于 "Maintenance"（维护）状态，而有的分片处于 "Last Erred"（最近出错）状态，则数据可能已损坏。必须首先修复处于 "Maintenance"（维护）状态的分片，然后再修复处于 "Last Erred"（最近出错）状态的分片。

下表说明了 RAID-5 卷状态。

表 14-1 RAID-5 卷状态

状态	含义
Initializing	分片正在将所有磁盘块清零。鉴于 RAID-5 卷在数据和奇偶校验交错条带化方面的特性，此过程是必需的。  一旦状态变为 "Okay"（正常），初始化过程即完成，此时您可以打开设备。在此之前，应用程序会收到错误消息。
Okay	此设备可供使用，并且当前没有错误。

表 14-1 RAID-5 卷状态 (续)

状态	含义
Maintenance	由于 I/O 或打开错误，已将某个分片标记为出现故障。这些错误是在读写操作中出现的。

下表说明了 RAID-5 卷的分片状态以及可能采取的操作。

表 14-2 RAID-5 分片状态

状态	含义	操作
Initializing	分片正在将所有磁盘块清零。鉴于 RAID-5 卷在数据和奇偶校验交错条带化方面的特性，此过程是必需的。	通常无需执行任何操作。如果在此过程中发生 I/O 错误，则此设备将进入 "Maintenance"（维护）状态。如果初始化失败，则卷将处于 "Initialization Failed"（初始化失败）状态，而分片将处于 "Maintenance"（维护）状态。此时，请清除该卷并重新创建它。
Okay	此设备可供使用，并且当前没有错误。	无。可以添加或替换分片（如果需要）。
Resyncing	表明分片正在进行重新同步。发生错误但已更正、已启用分片或已添加分片。	如果需要，请监视 RAID-5 卷的状态，直到完成重新同步为止。
Maintenance	由于 I/O 或打开错误，已将单个分片标记为出现故障。这些错误是在读写操作中出现的。	启用或替换出现故障的分片。请参见第 155 页中的“如何在 RAID-5 卷中启用组件”，或第 156 页中的“如何在 RAID-5 卷中替换组件”。 <code>metastat</code> 命令将显示一条 <code>invoke</code> 恢复消息，其中包含需要使用 <code>metareplace</code> 命令执行的相应操作。
Maintenance/Last Erred	多个分片遇到错误。发生故障的分片的状态为 "Maintenance"（维护）或 "Last Erred"。在这种情况下，不会对处于 "Maintenance"（维护）状态的分片尝试任何 I/O 操作。但是，会对标记为 "Last Erred"（最近出错）的分片尝试 I/O 操作，而操作结果将是此 I/O 请求的总体状态。	启用或替换出现故障的分片。请参见第 155 页中的“如何在 RAID-5 卷中启用组件”，或第 156 页中的“如何在 RAID-5 卷中替换组件”。 <code>metastat</code> 命令将显示一条 <code>invoke</code> 恢复消息，其中包含需要使用 <code>metareplace</code> 命令执行的相应操作。此命令必须使用 <code>-f</code> 标志来运行。此状态表明，由于多个分片发生故障，因此数据可能是虚构的。

## 在 RAID-5 卷中替换和启用分片概述

Solaris Volume Manager 可以替换和启用镜像与 RAID-5 卷中的组件。执行此操作所涉及的问题和要求对于镜像和 RAID-5 卷来说是相同的。有关更多信息，请参见第 208 页中的“替换和启用 RAID-1 和 RAID-5 卷中的组件概述”。

## 方案—RAID-5 卷

通过 RAID-5 卷，您可以使用冗余存储，而不会产生 RAID-1 卷的开销，而要提供数据冗余，则需要两倍于总存储空间的能力。通过设置 RAID-5 卷，您可以在同一组磁盘组件上提供比使用 RAID-1 卷获得的容量更大的冗余存储。此外，借助于热备件（请参见第 16 章，热备用池（概述），尤其是第 160 页中的“热备件的工作原理”），您还可以获得几乎相同的安全级别。其缺点是会增加写入时间，并且在组件出现故障时会显著降低性能。但是，对于大多数情形来说，上述优缺点可能都无关紧要。以下示例（即，在第 5 章，配置和使用 Solaris Volume Manager（方案）中所述的样例方案）说明了 RAID-5 卷是如何提供额外存储容量的。

RAID-0 和 RAID-1 卷的其他方案使用了 6 个分片 (c1t1d0, c1t2d0, c1t3d0, c2t1d0, c2t2d0, c2t3d0)，这些分片位于 6 个磁盘上并分布在 2 个控制器上，从而提供 27 GB 的冗余存储。通过在 RAID-5 配置中使用相同的分片，可以提供 45 GB 的存储。此外，该配置还可以承受单组件故障，而不会丢失数据或中断访问。通过在该配置中添加热备件，RAID-5 卷还可以承受更多组件故障。此方案最明显的缺点是，单控制器故障会导致该 RAID-5 卷丢失数据，而对于在第 93 页中的“方案—RAID-1 卷（镜像）”中所述的 RAID-1 卷，则不会出现这种情况。



## RAID-5 卷（任务）

---

本章提供有关执行与 RAID-5 卷关联的 Solaris Volume Manager 任务的信息。有关这些任务中所涉及的概念的信息，请参见第 14 章，[RAID-5 卷（概述）](#)。

### RAID-5 卷（任务列表）

以下任务列表列出了管理 Solaris Volume Manager RAID-5 卷所需的过程。

任务	说明	参考
创建 RAID-5 卷	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建 RAID-5 卷。	<a href="#">第 152 页中的“如何创建 RAID-5 卷”</a>
检查 RAID-5 卷的状态	使用 Solaris Volume Manager GUI 或 <code>metastat</code> 命令检查 RAID-5 卷的状态。	<a href="#">第 153 页中的“如何检查 RAID-5 卷的状态”</a>
扩展 RAID-5 卷	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令扩展 RAID-5 卷。	<a href="#">第 154 页中的“如何扩展 RAID-5 卷”</a>
启用 RAID-5 卷中的分片	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令启用 RAID-5 卷中的分片。	<a href="#">第 155 页中的“如何在 RAID-5 卷中启用组件”</a>
替换 RAID-5 卷中的分片	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令替换 RAID-5 卷中的分片。	<a href="#">第 156 页中的“如何在 RAID-5 卷中替换组件”</a>

# 创建 RAID-5 卷



注意 - 如果希望运行具有 32 位内核的 Solaris 软件，或希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，请勿创建大于 1 TB 的卷。有关 Solaris Volume Manager 中的大型卷支持的更多信息，请参见第 41 页中的“Solaris Volume Manager 中的多 TB 支持概述”。

## ▼ 如何创建 RAID-5 卷

**开始之前** 请查看第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 146 页中的“创建 RAID-5 卷的背景信息”。

- **要创建 RAID-5 卷，请使用以下方法之一：**

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。选择 "Action"（操作）⇒ "Create Volume"（创建卷）。然后，按照向导中的步骤操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -r component component component -i interlace-value
volume-name    指定要创建的卷的名称。
-r             指定创建 RAID-5 卷。
component      指定要包含在 RAID-5 卷中的分片或软分区。至少需要 3 个组件。
-i            指定交错值。
```

有关更多信息，请参见 `metainit(1M)` 手册页。

### 示例 15-1 创建一个包含三个分片的 RAID-5 卷

在本示例中，RAID-5 卷 d45 是使用 `-r` 选项通过 3 个分片创建的。由于未指定交错值，d45 将使用缺省的 16 KB。系统可验证 RAID-5 卷是否已设置并开始初始化该卷。

您必须先等待初始化完成，然后才能使用 RAID-5 卷。

```
# metainit d45 -r c2t3d0s2 c3t0d0s2 c4t0d0s2
d45: RAID is setup
```

**另请参见** 要针对某个文件系统准备新创建的 RAID-5 卷，请参见《System Administration Guide: Devices and File Systems》中的第 15 章“Creating and Mounting File Systems (Tasks)”。某些应用程序（例如数据库）不使用文件系统，这些应用程序而是使用原始卷。应用程序必须有自己识别卷的方法。

要将热备用池与 RAID-5 卷相关联，请参见第 166 页中的“如何将热备用池与卷关联”。

## 维护 RAID-5 卷

### ▼ 如何检查 RAID-5 卷的状态

在检查 RAID-5 卷的状态时，需要同时检查 RAID-5 状态和分片状态，才能完全了解该卷的状态以及当卷处于非正常状态时丢失数据的可能性。有关详细信息，请参见第 147 页中的“检查 RAID-5 卷状态概述”。

---

注 – RAID-5 卷初始化或重新同步不能被中断。

---

#### ● 要检查 RAID-5 卷的状态，请使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点并查看卷的状态。选择卷。然后，选择 "Action"（操作）⇒ "Properties"（属性）以查看更详细的信息。有关更多信息，请参见联机帮助。
- 使用 `metastat` 命令显示 RAID-5 卷的状态。

```
# metastat [-s diskset] [volume]
```

`-s diskset`      指定向其执行命令的磁盘集的名称。

`volume`          指定要显示的卷。

对于 RAID-5 卷中的每个分片，`metastat` 命令会显示以下内容：

Device            指定条带中分片的设备名称。

Start Block       指定分片的起始块。

Dbase            指定分片是否包含状态数据库副本。

State            指定分片的状态。

Hot Spare        指定该分片是否正用于热备份某个出现故障的分片

#### 示例 15-2 查看 RAID-5 卷状态

以下示例显示 `metastat` 命令的 RAID-5 卷输出。

```
# metastat d10
d10: RAID
    State: Okay
    Interlace: 32 blocks
```

```
Size: 10080 blocks
Original device:
Size: 10496 blocks
Device          Start Block  Dbase State      Hot Spare
c0t0d0s1        330         No  Okay
c1t2d0s1        330         No  Okay
c2t3d0s1        330         No  Okay
```

此 `metastat` 命令输出说明该卷为 RAID-5 卷。该信息是通过卷名称后的 "RAID" 字样指明的。对于 RAID-5 卷中的每个分片，输出显示了以下内容：

- 条带中分片的名称。
- 分片的起始块。
- 一个指示符，表示这些分片都不包含状态数据库副本。
- 分片的状态。在本示例中，所有的分片都处于 "Okay"（正常）状态。
- 分片是否为出现故障的分片的热备用替换。

## ▼ 如何扩展 RAID-5 卷

一般而言，附加组件是 RAID-5 卷空间不足的短期解决方案。出于性能考虑，最好使用“纯”RAID-5 卷。如果必须扩展现有的 RAID-5 卷以获得额外的存储空间，请使用以下过程。



**注意** - 如果希望运行具有 32 位内核的 Solaris 软件，或希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，请勿创建大于 1 TB 的卷。有关 Solaris Volume Manager 中的多 TB 卷支持的更多信息，请参见第 41 页中的“[Solaris Volume Manager 中的多 TB 支持概述](#)”。

**开始之前** 请阅读第 146 页中的“[创建 RAID-5 卷的背景信息](#)”。

- 1 确保您拥有所有数据的当前备份并且具有超级用户访问权限。
- 2 要附加其他组件到 RAID-5 卷，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后打开该 RAID-5 卷。选择 "Components"（组件）窗格。然后，选择 "Attach Component"（附加组件）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metattach` 命令：

```
# metattach volume-name name-of-component-to-add
volume-name          指定要扩展的 RAID-5 卷的名称。
name-of-component-to-add 指定要附加到 RAID-5 卷的组件的名称。
```

有关更多信息，请参见 `metattach(1M)` 手册页。

### 示例 15-3 向 RAID-5 卷添加组件

以下示例说明如何向现有的 RAID-5 卷 d2 添加分片 c2t1d0s2。

```
# metattach d2 c2t1d0s2
d2: column is attached
```

**另请参见** 对于 UFS 文件系统，请针对 RAID-5 卷运行 `growfs` 命令。请参见第 37 页中的“使用 `growfs` 命令扩展卷和磁盘空间”。

某些应用程序（例如数据库）不使用文件系统，这些应用程序而是使用原始卷。在这些情况下，应用程序必须有自己扩展添加的空间的方法。

## ▼ 如何在 RAID-5 卷中启用组件

如果磁盘驱动器有缺陷，则可以使用系统上的其他可用磁盘（及其分片）来替换该磁盘驱动器，如第 156 页中的“如何在 RAID-5 卷中替换组件”中所述。或者，也可以修复该磁盘，为其设置标签，并通过运行带 `-e` 选项的 `metareplace` 命令来重新启用该磁盘。

- 1 确保您拥有所有数据的当前备份并且具有超级用户访问权限。
- 2 要启用 RAID-5 卷中出现故障的组件，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点，然后打开该 RAID-5 卷。选择 "Components"（组件）窗格。然后，选择出现故障的组件。单击 "Enable Component"（启用组件）。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metareplace` 命令：

```
# metareplace -e volume-name component-name
-e                指定将出现故障的组件置于可用状态并对其执行重新同步。
volume-name       指定出现故障的组件所在的卷的名称。
component-name     指定故障组件的名称。
```

`metareplace` 命令会使用 RAID-5 卷的其余部分自动开始重新同步新组件。

### 示例 15-4 在 RAID-5 卷中启用组件

在以下示例中，RAID-5 卷 d20 上的分片 c2t0d0s2 包含软错误。使用带有 `-e` 选项的 `metareplace` 命令可启用分片。

```
# metareplace -e d20 c2t0d0s2
```

## ▼ 如何在 RAID-5 卷中替换组件

该任务将替换仅有一个分片出现故障的 RAID-5 卷上的故障分片。



**注意** - 如果在多个分片包含错误时替换某个出现故障的分片可能会导致虚构数据。在这种情况下，此实例中数据的完整性是可疑的。

您可以对未出现故障的设备使用 `metareplace` 命令以更改磁盘分片或其他组件。此过程对于调优 RAID-5 卷的性能非常有用。

### 1 确保您拥有所有数据的当前备份并且具有超级用户访问权限。

### 2 使用以下方法之一确定需要替换 RAID-5 卷的哪一个分片：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。然后打开 RAID-5 卷。选择 "Components"（组件）窗格。查看各个组件的状态。有关更多信息，请参见联机帮助。
- 使用 `metastat` 命令。

```
# metastat volume
```

`volume` 指定 RAID-5 卷的名称。

查找关键字短语 "Needs Maintenance" 以确定 RAID-5 卷的状态。查找关键字 "Maintenance"（维护）以确定出现故障的分片。

### 3 通过以下方法之一，用另一分片替换出现故障的分片：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。然后打开 RAID-5 卷。选择 "Components"（组件）窗格。选择出现故障的组件。单击 "Replace Component"（取代组件）并遵循屏幕说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metareplace` 命令：

```
# metareplace volume-name failed-component new-component
```

- `volume-name` 是出现故障的组件所在的 RAID-5 卷的名称。
- `failed-component` 用于指定要替换的组件的名称。
- `new-component` 指定要添加到该卷（以替代出现故障的组件）的组件的名称。

有关更多信息，请参见 [metareplace\(1M\)](#) 手册页。

### 4 要验证替换分片的状态，请使用步骤 2 中所述的方法之一。

替换分片的状态应为 "Resyncing" 或 "Okay"。

示例 15-5 替换 RAID-5 组件

```
# metastat d1
d1: RAID
State: Needs Maintenance
  Invoke: metareplace d1 c0t14d0s6 <new device>
  Interlace: 32 blocks
  Size: 8087040 blocks
Original device:
  Size: 8087520 blocks
  Device          Start Block  Dbase State      Hot Spare
  c0t9d0s6         330        No   Okay
  c0t13d0s6        330        No   Okay
  c0t10d0s6        330        No   Okay
  c0t11d0s6        330        No   Okay
  c0t12d0s6        330        No   Okay
  c0t14d0s6        330        No   Maintenance

# metareplace d1 c0t14d0s6 c0t4d0s6
d1: device c0t14d0s6 is replaced with c0t4d0s6
# metastat d1
d1: RAID
State: Resyncing
Resync in progress: 98% done
Interlace: 32 blocks
Size: 8087040 blocks
Original device:
  Size: 8087520 blocks
  Device          Start Block  Dbase State      Hot Spare
  c0t9d0s6         330        No   Okay
  c0t13d0s6        330        No   Okay
  c0t10d0s6        330        No   Okay
  c0t11d0s6        330        No   Okay
  c0t12d0s6        330        No   Okay
  c0t4d0s6         330        No   Resyncing
```

在本示例中，metastat 命令显示了 RAID-5 卷 d1 中出现故障的分片。找到可用分片后，将运行 metareplace 命令，先指定出现故障的分片，然后指定替换分片。

如果没有其他可用分片，则运行带 -e 选项的 metareplace 命令，以通过重新同步出现故障的设备来尝试从可能的软错误中恢复。有关此过程的更多信息，请参见第 155 页中的“如何在 RAID-5 卷中启用组件”。如果存在多个错误，则必须先替换或启用状态为 "Maintenance"（维护）的分片，然后才可以修复状态为 "Last Erred"（最近出错）的分片。运行 metareplace 命令后，可以使用 metastat 命令监视重新同步的进度。替换过程中，卷和新分片的状态为 "Resyncing"。您可以在卷处于以上状态时继续使用该卷。



## 热备用池（概述）

---

本章说明 Solaris Volume Manager 如何使用热备用池。有关执行相关任务的信息，请参见第 17 章，热备用池（任务）。

本章包含以下信息：

- 第 159 页中的“热备件和热备用池概述”
- 第 162 页中的“方案—热备件”

### 热备件和热备用池概述

热备用池是分片（热备件）的集合，Solaris Volume Manager 可使用热备用池提高 RAID-1（镜像）和 RAID-5 卷的数据可用性。当子镜像或 RAID-5 卷发生分片故障时，Solaris Volume Manager 会自动使用热备件替换发生故障的分片。

---

注 - 热备件不适用于 RAID-0 卷或单向镜像。要使用自动替换功能，必须具有冗余数据。

---

不能使用处于空闲状态的热备件来存放数据或状态数据库副本。热备件必须保持就绪状态，以便在与其关联的卷中出现分片故障时，能够立即投入使用。要使用热备件，除了系统实际运行所需要的磁盘之外，您还必须另外购买附加磁盘。

通过 Solaris Volume Manager，可以动态添加、删除、替换和启用在热备用池中的热备件。可以使用 Solaris Management Console 或命令行实用程序来管理热备件和热备用池。有关这些任务的详细信息，请参见第 17 章，热备用池（任务）。

## 热备件

热备件是指可正常运行且可随时使用但目前未使用的分片（而不是卷）。热备件是保留资源，也就是说，它始终处于就绪状态，可随时替换子镜像或 RAID-5 卷中发生故障的分片。

热备件可在出现硬件故障时提供保护。RAID-1 和 RAID-5 卷中的分片可在发生故障时自动由热备件替换。热备件已进行重新同步，可在卷中使用。在发生故障的子镜像或 RAID-5 卷分片进行修复或替换之前，可以临时使用热备件。

可以在热备用池中创建热备件。各个热备件均可包括在一个或多个热备用池中。例如，您可能具有两个子镜像和两个热备件。这些热备件可以排列为两个热备用池，其中，每个热备用池具有两个按不同优先顺序排列的热备件。通过此策略，您可以指定首先使用哪个热备件。同时，此策略还通过提供多个热备件而提高了可用性。

子镜像或 RAID-5 卷只能使用等于或大于子镜像或 RAID-5 卷中发生故障的分片的热备件。例如，如果子镜像包含 1 GB 的驱动器，则该子镜像的热备件必须等于或大于 1 GB。

## 热备用池

热备用池是指经过排序的热备件列表（集合）。

可以将热备件放置在一个或多个热备用池中，以便可以使用最少的分片来获得最大的灵活性和保护。可以将指定为热备件的一个分片放置在多个热备用池中，使每个热备用池具有不同的分片和特征。然后，可以将热备用池分配给任意数量的子镜像卷或 RAID-5 卷。

---

注- 可以将一个热备用池分配给多个子镜像或 RAID-5 卷。但是，一个子镜像或 RAID-5 卷只能与一个热备用池相关联。

---

## 热备件的工作原理

发生 I/O 错误时，Solaris Volume Manager 会根据向热备用池添加热备件的顺序搜索热备用池中的热备件。Solaris Volume Manager 会检查热备用池中第一个等于或大于要替换的分片的可用热备件。一旦 Solaris Volume Manager 发现第一个足够大的热备件，该热备件将用来替换。Solaris Volume Manager 会将此热备件的状态更改为 "In-Use"（使用中），并在需要时自动重新同步数据。进行替换时，不会更改热备用池中的热备件的顺序。

对于镜像，热备件会与可正常运行的子镜像中的数据进行重新同步。对于 RAID-5 卷，热备件会与卷中的其他分片进行重新同步。如果在热备件列表中未找到具有足够

大小的分片，则发生故障的子镜像或 RAID-5 卷就会进入故障状态，而热备件将保持未使用状态。对于子镜像，该子镜像将无法再完整地复制数据。对于 RAID-5 卷，数据冗余将不再可用。

**提示** – 在向热备用池中添加热备件时，请按从小到大的顺序进行添加。此策略可避免使用较大的热备件来替换较小的分片而造成浪费。

如果分片遇到 I/O 错误，则发生故障的分片将处于 "Broken"（中断）状态。要解决这个问题，请首先修复或替换发生故障的分片。然后，使用 Solaris Management Console 中增强的存储工具将该分片恢复为 "Available"（可用）状态。或者，也可以使用 `metahs -e` 命令。

子镜像或 RAID-5 卷将使用热备件来代替发生故障的分片，直到启用或替换发生故障的分片为止。之后，该热备件将在热备用池中标记为 "Available"（可用）。此时，该热备件再次处于就绪状态，从而可供使用。

## 热备用池状态

下表说明了热备用池状态以及可能要执行的操作。

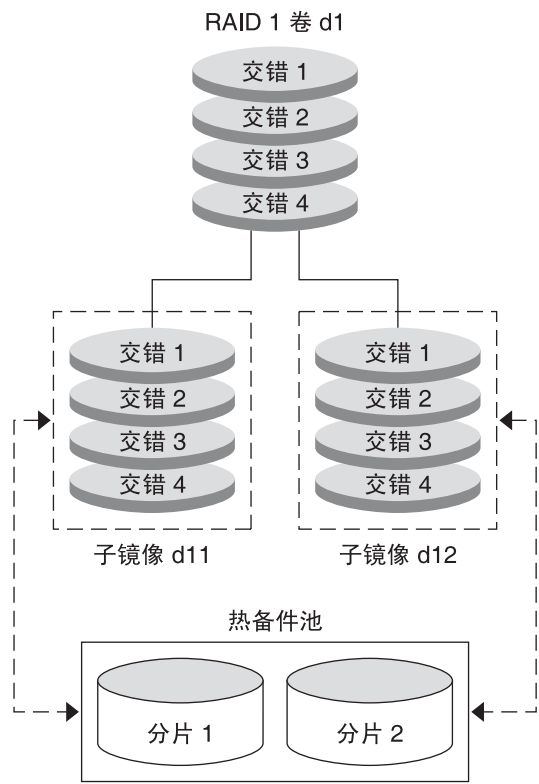
表 16-1 热备用池状态（命令行）

状态	含义	操作
Available	热备用池中的热备件正在运行，并可随时接受数据。当前未对热备件进行写入或读取。	无。
In-Use	此热备用池中的热备件正用于替换冗余卷中发生故障的分片。	诊断热备件的使用方式。然后，修复卷中正在使用使用热备件的分片。
Broken	热备件或热备用池存在问题。但是，不存在立即丢失数据的危险。如果所有热备件都在使用中，则也会显示此状态。	诊断热备件的使用方式或它们发生中断的原因。如果需要，可以向热备用池中添加更多热备件。

## 示例—热备用池

图 16-1 显示了与镜像 d1 中的子镜像 d11 和 d12 相关联的热备用池。如果其中任何一个子镜像中的分片发生故障，则会使用热备件自动替换发生故障的分片。热备用池本身与每个子镜像卷（而不是镜像）相关联。如果需要，热备用池还可以与其他子镜像或 RAID-5 卷相关联。

图 16-1 热备用池示例



# 方案－热备件

热备件可以为冗余卷（RAID-1 和 RAID-5）提供额外保护，从而帮助防止数据丢失。通过将热备件与构成 RAID-0 子镜像或 RAID-5 配置的底层分片相关联，可以使系统自动使用热备用池中的可用分片来替换发生故障的分片。这些已交换使用的分片将使用它们所需要的信息进行更新。然后，这些分片就可以像原始分片一样正常运行了。您可以在方便时替换发生故障的分片。

## 热备用池（任务）

---

本章介绍如何使用 Solaris Volume Manager 的热备件和热备用池。有关相关概念的信息，请参见第 16 章，[热备用池（概述）](#)。

### 热备用池（任务列表）

以下任务列表列出了管理 Solaris Volume Manager 热备用池所需的过程。

任务	说明	参考
创建热备用池	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建热备用池。	<a href="#">第 164 页中的“如何创建热备用池”</a>
向热备用池添加分片	使用 Solaris Volume Manager GUI 或 <code>metahs</code> 命令向热备用池添加分片。	<a href="#">第 165 页中的“如何向热备用池中添加更多分片”</a>
将热备用池与卷关联	使用 Solaris Volume Manager GUI 或 <code>metaparam</code> 命令将热备用池与卷关联。	<a href="#">第 166 页中的“如何将热备用池与卷关联”</a>
更改与卷关联的热备用池	使用 Solaris Volume Manager GUI 或 <code>metaparam</code> 命令更改与卷关联的热备用池。	<a href="#">第 167 页中的“如何更改关联的热备用池”</a>
检查热备件和热备用池的状态	使用 Solaris Volume Manager GUI、 <code>metastat</code> 命令或 <code>metahs -i</code> 命令检查热备件或热备用池的状态。	<a href="#">第 168 页中的“如何检查热备件和热备用池的状态”</a>
替换热备用池中的热备件	使用 Solaris Volume Manager GUI 或 <code>metahs</code> 命令替换热备用池中的热备件。	<a href="#">第 169 页中的“如何替换热备用池中的热备件”</a>
从热备用池中删除热备件	使用 Solaris Volume Manager GUI 或 <code>metahs</code> 命令从热备用池中删除热备件。	<a href="#">第 170 页中的“如何从热备用池中删除热备件”</a>
启用热备件	使用 Solaris Volume Manager GUI 或 <code>metahs</code> 命令启用热备用池中的热备件。	<a href="#">第 171 页中的“如何启用热备件”</a>

# 创建热备用池

## ▼ 如何创建热备用池



注意 – 如果希望运行具有 32 位内核的 Solaris 软件，或希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，请勿创建大于 1 TB 的卷或热备件。有关 Solaris Volume Manager 中的多 TB 卷支持的更多信息，请参见第 41 页中的“[Solaris Volume Manager 中的多 TB 支持概述](#)”。



注意 – 即使创建的热备件不够大，Solaris Volume Manager 也不会发出警告。如果热备件小于其所附加到的卷，热备件将无法正常运行。

开始之前 检查第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要创建热备用池，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Hot Spare Pools"（联机备用磁盘群组）节点。然后选择 "Action"（操作）⇒ "Create Hot Spare Pool"（创建联机备用磁盘群组）。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metainit` 命令：

```
# metainit hot-spare-pool-name ctds-for-slice
```

`hot-spare-pool-name` 指定热备用池的名称。

`ctds-for-slice` 指定要添加到热备用池中的分片。对要添加到热备用池中的每个分片，重复执行该选项。

有关更多信息，请参见 [metainit\(1M\)](#) 手册页。

注 – 还可以使用 `metahs` 命令创建热备用池。

### 示例 17-1 创建热备用池

```
# metainit hsp001 c2t2d0s2 c3t2d0s2
hsp001: Hotspare pool is setup
```

在本示例中，热备用池 `hsp001` 包含两个作为热备件的磁盘。系统确认了热备用池已设置。

另请参见 要将更多热备件添加到热备用池中，请参见第 165 页中的“如何向热备用池中添加更多分片”。创建热备用池后，需要将其与子镜像或 RAID-5 卷关联。请参见第 166 页中的“如何将热备用池与卷关联”。

## ▼ 如何向热备用池中添加更多分片

开始之前 检查第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要向现有的热备用池中添向现有的热备用池中添加分片，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Hot Spare Pools"（联机备用磁盘群组）节点。选择要更改的热备用池。选择 "Action"（操作）⇒ "Properties"（属性）。然后选择 "Hot Spares"（联机备用磁盘）面板。有关更多信息，请参见联机帮助。
  - 使用以下形式之一的 `metahs` 命令：

```
# metahs -a hot-spare-pool-name slice-to-add
# metahs -a -all hot-spare-pool-name slice-to-add
-a hot-spare-pool-name    指定向指定的热备用池中添加分片。
-a all                    指定向所有热备用池中添加分片。
slice-to-add              指定要添加到热备用池中的分片。
```

有关更多信息，请参见 `metahs(1M)` 手册页。

---

注 - 可以向一个或多个热备用池中添加热备件。向热备用池添加热备件时，该热备件将添加到热备用池中分片列表的末尾。

---

### 示例 17-2 向某个热备用池中添加热备用分片

在本示例中，`-a` 选项用于将分片 `/dev/dsk/c3t0d0s2` 添加到热备用池 `hsp001` 中。系统将验证该分片是否已添加到热备用池中。

```
# metahs -a hsp001 /dev/dsk/c3t0d0s2
hsp001: Hotspare is added
```

### 示例 17-3 向所有热备用池中添加热备用分片

在本示例中，与 `all` 一起使用的 `-a` 选项用于将分片 `/dev/dsk/c3t0d0s2` 添加到系统上已配置的所有热备用池中。系统将验证该分片是否已添加到所有热备用池中。

```
# metahs -a -all /dev/dsk/c3t0d0s2
hsp001: Hotspare is added
hsp002: Hotspare is added
hsp003: Hotspare is added
```

## 将热备用池与卷关联

### ▼ 如何将热备用池与卷关联

开始之前 检查第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要将热备用池与 RAID-5 卷或子镜像关联，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）并选择一个卷。选择 "Action"（操作）⇒ "Properties"（属性）。然后选择 "Hot Spare Pool"（联机备用磁盘群组）面板。最后，选择 "Attach HSP"（附加 HSP）。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metaparam` 命令：

```
# metaparam -h hot-spare-pool component
-h                指定修改指定的热备用池。
hot-spare-pool    指定热备用池的名称。
component         指定要与热备用池关联的子镜像或 RAID-5 卷的名称。
```

有关更多信息，请参见 [metaparam\(1M\)](#) 手册页。

#### 示例 17-4 将热备用池与子镜像关联

在以下示例中，通过 `-h` 选项将热备用池 `hsp100` 与镜像 `d0` 的两个子镜像 `d10` 和 `d11` 关联。`metastat` 命令显示该热备用池已与两个子镜像关联。

```
# metaparam -h hsp100 d10
# metaparam -h hsp100 d11
# metastat d0
d0: Mirror
   Submirror 0: d10
       State: Okay
   Submirror 1: d11
       State: Okay
...
d10: Submirror of d0
```

```

        State: Okay
        Hot spare pool: hsp100
    ...

d11: Submirror of d0
    State: Okay
    Hot spare pool: hsp100
    ...

```

### 示例 17-5 将热备用池与 RAID-5 卷关联

在以下示例中，通过 `-h` 选项将热备件 `hsp001` 与 RAID-5 卷 `d10` 关联。`metastat` 命令显示该热备用池已与 RAID-5 卷关联。

```

# metaparam -h hsp001 d10
# metastat d10
d10: RAID
    State: Okay
    Hot spare pool: hsp001
    ...

```

## ▼ 如何更改关联的热备用池

开始之前 检查第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要更改与热备用池关联的卷，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点并选择卷。选择 "Action"（操作）⇒ "Properties"（属性）。然后选择 "Hot Spare Pool"（联机备用磁盘群组）面板。分离不需要的热备用池，并按照屏幕上的说明附加新的热备用池。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metaparam` 命令：

```

# metaparam -h hot-spare-pool-name RAID5-volume-or-submirror-name
-h                指定修改命名的热备用池。

hot-spare-pool    指定新的热备用池的名称，或者使用特殊关键字 none 删除热备用池关联。

component        指定要与热备用池连接的子镜像或 RAID-5 卷的名称。

```

有关更多信息，请参见 [metaparam\(1M\)](#) 手册页。

### 示例 17-6 更改热备用池关联

在以下示例中，最初是热备用池 hsp001 与 RAID-5 卷 d4 关联。然后卷的热备用池关联更改到 hsp002。metastat 命令显示了此更改前后的热备用池关联。

```
# metastat d4
d4: RAID
    State: Okay
    Hot spare pool: hsp001
...
# metaparam -h hsp002 d4
# metastat d4
d4: RAID
    State: Okay
    Hot spare pool: hsp002
...
```

### 示例 17-7 删除卷与热备用池的关联

在以下示例中，最初是热备用池 hsp001 与 RAID-5 卷 d4 关联。然后热备用池关联更改到 none，表示任何热备用池都不应与此卷关联。metastat 命令显示了此更改前后的热备用池关联。

```
# metastat d4
d4: RAID
    State: Okay
    Hot spare pool: hsp001
...
# metaparam -h none d4
# metastat d4
d4: RAID
    State: Okay
    Hot spare pool:
...
```

## 维护热备用池

以下各节说明如何针对热备用池执行维护任务。

### ▼ 如何检查热备件和热备用池的状态

- 要查看热备用池及其热备件的状态，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Hot Spare Pools"（联机备用磁盘群组）节点并选择一个热备用池。选择 "Action"（操作）⇒ "Properties"（属性）以查看详细的状态信息。有关更多信息，请参见联机帮助。

- 运行以下格式的 `metastat` 命令：

```
# metastat hot-spare-pool-name
```

### 示例 17-8 查看热备用池的状态

以下示例显示了针对热备用池运行 `metastat` 命令时的样例输出。

```
# metastat hsp001
hsp001: 1 hot spare
        c1t3d0s2          Available      16800 blocks
```

还可以使用 `metahs` 命令检查热备用池的状态。

有关热备用池状态和可执行的操作的信息，请参见第 161 页中的“热备用池状态”。

## ▼ 如何替换热备用池中的热备件

### 1 成为超级用户。

### 2 通过使用以下方法之一，验证当前是否正在使用热备件：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Hot Spare Pools"（联机备用磁盘群组）节点并选择一个热备用池。选择 "Action"（操作）⇒ "Properties"（属性）。然后选择 "Hot Spares"（联机备用磁盘）面板。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metastat` 命令查看热备用池的状态：

```
# metastat hot-spare-pool-name
```

有关更多信息，请参见 [metastat\(1M\)](#) 手册页。

### 3 要替换热备件，请使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Hot Spare Pools"（联机备用磁盘群组）节点并选择一个热备用池。选择 "Action"（操作）⇒ "Properties"（属性）。然后选择 "Hot Spares"（联机备用磁盘）面板。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metahs` 命令：

```
# metahs -r hot-spare-pool-name current-hot-spare replacement-hot-spare
```

`-r` 指定替换指定的热备用池中的磁盘。

`hot-spare-pool-name` 指定热备用池的名称。还可以使用特殊关键字 `all` 更改所有热备用池关联。

*current-hot-spare*      指定要替换的当前热备件的名称。

*replacement-hot-spare*      指定将替换指定热备用池中当前热备件的分片的名称。

有关更多信息，请参见 [metahs\(1M\)](#) 手册页。

### 示例 17-9 替换某个热备用池中的热备件

在以下示例中，`metastat` 命令显示该热备件未被使用。`metahs -r` 命令将热备用池 `hsp003` 中的热备件 `/dev/dsk/c0t2d0s2` 替换为热备件 `/dev/dsk/c3t1d0s2`。

```
# metastat hsp003
hsp003: 1 hot spare
        c0t2d0s2          Broken          5600 blocks
# metahs -r hsp003 c0t2d0s2 c3t1d0s2
hsp003: Hotspare c0t2d0s2 is replaced with c3t1d0s2
```

### 示例 17-10 在所有关联的热备用池中替换某个热备件

在以下示例中，关键字 `all` 用于在所有关联的热备用池中将热备件 `/dev/dsk/c1t0d0s2` 替换为热备件 `/dev/dsk/c3t1d0s2`。

```
# metahs -r all c1t0d0s2 c3t1d0s2
hsp001: Hotspare c1t0d0s2 is replaced with c3t1d0s2
hsp002: Hotspare c1t0d0s2 is replaced with c3t1d0s2
hsp003: Hotspare c1t0d0s2 is replaced with c3t1d0s2
```

## ▼ 如何从热备用池中删除热备件

- 1 成为超级用户。
- 2 通过使用以下方法之一，验证当前是否正在使用热备件：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Hot Spare Pools"（联机备用磁盘群组）节点并选择一个热备用池。选择 "Action"（操作）⇒ "Properties"（属性）。然后选择 "Hot Spares"（联机备用磁盘）面板。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metastat` 命令查看热备用池的状态：
 

```
# metastat hot-spare-pool-name
```

 请参见 [metastat\(1M\)](#) 手册页。
- 3 要删除热备件，请使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Hot Spare Pools"（联机备用磁盘群组）节点并选择一个热备用池。选择 "Action"（操作）⇒ "Properties"（属性）。然后选择 "Hot Spares"（联机备用磁盘）面板。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metahs` 命令：

```
# metahs -d hot-spare-pool-name current-hot-spare
-d                指定从指定的热备用池中删除热备件。
hot-spare-pool    指定热备用池的名称。还可以使用特殊关键字 all 从所有热备用池中删除热备件。
current-hot-spare 指定将删除的当前热备件的名称。
```

有关更多信息，请参见 [metahs\(1M\)](#) 手册页。

### 示例 17-11 从某个热备用池中删除热备件

在以下示例中，`metastat` 命令显示该热备件未被使用。`metahs -d` 命令将在热备用池 `hsp003` 中删除热备件 `/dev/dsk/c0t2d0s2`。

```
# metastat hsp003
hsp003: 1 hot spare
          c0t2d0s2                Broken          5600 blocks
# metahs -d hsp003 c0t2d0s2
```

## ▼ 如何启用热备件

- 1 成为超级用户。
- 2 要使热备件回到 "Available"（可用）状态，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Hot Spare Pools"（联机备用磁盘群组）节点并选择一个热备用池。选择 "Action"（操作）⇒ "Properties"（属性）。然后选择 "Hot Spares"（联机备用磁盘）面板。请按照屏幕上的说明进行操作。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metahs` 命令：

```
# metahs -e hot-spare-slice
-e                指定启用热备件。
hot-spare-slice   指定要启用的分片的名称。
```

有关更多信息，请参见 [metahs\(1M\)](#) 手册页。

### 示例 17-12 启用热备件

在以下示例中，`metahs` 命令在修复热备件 `/dev/dsk/c0t0d0s2` 后会将其置于 "Available"（可用）状态。不必指定热备用池。

```
# metahs -e c0t0d0s2
```

## 磁盘集（概述）

---

本章提供了有关磁盘集的概念性信息。有关执行相关任务的信息，请参见第 19 章，[磁盘集（任务）](#)。

本章包括以下信息：

- 第 173 页中的“磁盘集方面的新增功能”
- 第 173 页中的“磁盘集介绍”
- 第 176 页中的“Solaris Volume Manager 磁盘集管理”
- 第 181 页中的“磁盘集使用原则”
- 第 182 页中的“磁盘集中的异步共享存储”
- 第 182 页中的“方案—磁盘集”

## 磁盘集方面的新增功能

本节介绍本 Solaris 发行版中新增的磁盘集功能。

有关 Solaris 新增功能的完整列表和 Solaris 发行版的说明，请参见《[Oracle Solaris 10 1/13 新增功能](#)》。

## 磁盘集介绍

磁盘集是指一组包含逻辑卷和热备件的物理存储卷。必须从该磁盘集内部的驱动器上构建卷和热备用池。一旦在该磁盘集内部创建了卷，您就可以像使用物理分片一样来使用该卷了。可以使用该卷创建和挂载文件系统并存储数据。

---

注 – 基于 SPARC 和 x86 的平台均支持磁盘集。

---

# 磁盘集类型

本节介绍在 Solaris Volume Manager 中提供的不同类型的磁盘集。

## 本地磁盘集

每台主机都有一个本地磁盘集。本地磁盘集由主机上不属于已命名磁盘集的所有磁盘组成。本地磁盘集专属于特定主机。本地磁盘集包含用于该特定主机配置的状态数据库。本地磁盘集中的卷和热备用池仅包含该本地磁盘集内部的驱动器。

## 已命名磁盘集

除了本地磁盘集之外，主机还可以加入已命名磁盘集。已命名磁盘集是指不在本地磁盘集中的任何磁盘集。根据您的系统配置，您可以通过下列类型的已命名磁盘集来管理卷。

## 共享磁盘集

**共享磁盘集**可由多台主机共享。虽然共享磁盘集对于所有加入该磁盘集的主机来说都是可见的，但只有该磁盘集的属主才能访问它。每台主机都可以控制共享磁盘集，但一次只能有一台主机进行控制。此外，共享磁盘集还会提供一个可用于管理卷的不同名称空间。

共享磁盘集支持数据冗余和数据可用性。如果一台主机出现故障，则另一台主机可以接管出现故障的主机的磁盘集（此类配置称为**故障转移配置**）。

---

注 – 共享磁盘集在某种程度上可供 Sun Cluster、Solstice HA（High Availability，高可用性）或其他受支持的第三方 HA 框架使用。Solaris Volume Manager 本身并不提供实现故障转移配置所必需的所有功能。

---

尽管每台主机都可以控制磁盘集，但一次只能有一台主机进行控制。

## 自动获取 (autotake) 磁盘集

在 Solaris 9 4/04 发行版提供自动获取功能之前，Solaris Volume Manager 不支持通过 `/etc/vfstab` 文件在磁盘集上自动挂载文件系统。Solaris Volume Manager 要求系统管理员使用 `metaset -s setname -t` 命令手动发出磁盘集获取命令，然后才能访问该磁盘集上的文件系统。

通过自动获取功能，您可以使用 `metaset -s setname -Aenable` 命令设置磁盘集，以便在引导时自动访问它。您可以利用此功能在引导时在 `/etc/vfstab` 文件中为文件系统定义挂载选项。您还可以使用此功能在 `/etc/vfstab` 文件中为已启用的磁盘集中的卷上的文件系统定义挂载选项。

只有单主机磁盘集才支持自动获取功能。自动获取功能要求该磁盘集不能与任何其他系统共享。不能将共享的磁盘集设置为使用自动获取功能，并且 `metaset -A` 命令将失败。但是，从该磁盘集中删除其他主机之后，可以将其设置为使用自动获取功能。同样，不能在自动获取磁盘集中添加其他主机。如果已禁用自动获取功能，则可以向该磁盘集中添加其他主机。

---

注 – 在 Sun Cluster 环境中，自动获取功能处于禁用状态。Sun Cluster 可处理磁盘集的获取和释放。

---

有关自动获取功能的更多信息，请参见 `metaset(1M)` 命令的 `-A` 选项。

## 多属主磁盘集

在 Sun Cluster 环境中创建的已命名磁盘集称为多属主磁盘集。多属主磁盘集可以使多个节点共享磁盘集的所有权，并能够同时访问共享磁盘。群集中的所有节点都可以直接访问多属主磁盘集中的所有磁盘和卷。每个多属主磁盘集都包含一系列已添加到该磁盘集中的主机。因此，一个群集配置中的每个多属主磁盘集都可以具有一组不同的（有时会重叠的）主机。

每个多属主磁盘集都有一个主节点。主节点的功能是管理和更新状态数据库副本更改。由于每个磁盘集都有一个主节点，因此可以同时具有多个主节点。可以通过两种方式选择主节点。第一种方式是，第一个向磁盘集中添加磁盘的节点将成为主节点。第二种方式则是，在主节点出现紧急情况和失败时，节点 ID 最低的节点将成为主节点。

多属主磁盘集功能只能在 Sun Cluster 环境中启用以管理多属主磁盘集存储。自 Sun Cluster 10/04 软件集合起，Sun Cluster 发行版可使用 Solaris Volume Manager for Sun Cluster 功能，同时，诸如 Oracle Real Applications Clusters 等应用程序也可以使用该功能。有关 Solaris Volume Manager for Sun Cluster 的更多信息，请参见第 4 章“Solaris Volume Manager for Sun Cluster（概述）”。

在配置多属主磁盘集之前，除了安装 Solaris OS 之外，还必须安装以下软件：

- Sun Cluster 初始群集框架
- Sun Cluster Support for Oracle Real Application Clusters 软件
- Oracle Real Application Clusters 软件

---

注 – 有关设置 Sun Cluster 和 Oracle Real Application Clusters 软件的信息，请参见《[Oracle Solaris Cluster 软件安装指南](#)》和《[适用于 Oracle Real Application Clusters 的 Oracle Solaris Cluster 数据服务指南](#)》。

---

## Solaris Volume Manager 磁盘集管理

与本地磁盘集管理不同，您无需手动创建或删除磁盘集状态数据库。Solaris Volume Manager 会在磁盘集的每个磁盘上放置一个状态数据库副本（位于分片 7），磁盘集的所有磁盘上最多总共放置 50 个副本。

在向磁盘集添加磁盘时，Solaris Volume Manager 会在磁盘集上自动创建状态数据库副本。在磁盘集接受所添加的磁盘后，Solaris Volume Manager 可能会重新对该磁盘进行分区，以便可以将该磁盘集的状态数据库副本放置在该磁盘上（请参见第 178 页中的“自动磁盘分区”）。

驻留在磁盘集中某个卷上的文件系统通常不会在引导时使用 `/etc/vfstab` 文件自动挂载。所需的 Solaris Volume Manager RPC 守护进程（`rpc.metad` 和 `rpc.metamhd`）不会在引导过程中过早启动，因此不允许执行该操作。此外，在重新引导期间，磁盘集的所有权会丢失。请不要在 `/etc/inetd.conf` 文件中禁用 Solaris Volume Manager RPC 守护进程。已将它们配置为在缺省情况下启动。这些守护进程必须始终保持启用状态，Solaris Volume Manager 才能使用其完整功能。

如果使用 `metaset` 命令的 `-A` 选项启用了自动获取功能，则在引导时会自动获取磁盘集。在这种情况下，驻留在磁盘集中某个卷上的文件系统可以使用 `/etc/vfstab` 文件自动挂载。要在引导过程中启用自动获取功能，磁盘集必须只与一台主机建立关联，并且已启用自动获取功能。可以在创建磁盘集期间或之后启用该磁盘集。有关自动获取功能的更多信息，请参见第 174 页中的“自动获取 (autotake) 磁盘集”。

---

**注**— 尽管在单主机配置中支持磁盘集，但这些磁盘集通常不适合在“本地”（并非双连接）使用。两种常见的例外情况是：使用磁盘集为逻辑卷提供一个更便于管理的名称空间，以及使用磁盘集更方便地管理存储区域网络 (Storage Area Network, SAN) 光纤网络上的存储（请参见第 182 页中的“方案一—磁盘集”）。

---

可以使用 Solaris Volume Manager 命令行界面（`metaset` 命令）或 Solaris Management Console 中增强的存储工具创建和配置磁盘集。

在向磁盘集添加磁盘之后，该磁盘集中的主机可以**保留**（或**获取**）和**释放**该磁盘集。如果磁盘集被某一主机保留，则该磁盘集中的其他主机将无法访问该磁盘集中的磁盘上的数据。要对磁盘集执行维护操作，必须有一台主机是该磁盘集的属主或保留该磁盘集。主机可以通过向磁盘集中放置第一个磁盘来获取该磁盘集的隐式所有权。

可以使用 `metaimport` 命令将磁盘集（包括在其他系统上创建的磁盘集）导入到现有的 Solaris Volume Manager 配置中。

## 保留磁盘集

主机必须先保留磁盘集，才能使用该磁盘集中的磁盘。保留磁盘集的方式有两种：

- **安全**—如果以安全方式保留磁盘集，则 Solaris Volume Manager 会尝试获取该磁盘集，而其他主机则尝试释放该磁盘集。释放操作可能会失败，因此保留操作也可能失败。
- **强制**—如果以强制方式保留磁盘集，则无论其他主机当前是否保留该磁盘集，Solaris Volume Manager 将会保留该磁盘集。通常，当磁盘集中的某一主机已关闭或未进行通信时会使用此方法。磁盘集中的所有磁盘都将被接管。在执行保留操作的主机上将读入状态数据库，并且在该磁盘集中配置的共享卷将可访问。此时，如果另一台主机曾经保留该磁盘集，则该主机可能会由于丢失保留而出现紧急情况。

通常，一个磁盘集中的两台主机会彼此配合以确保磁盘集中的磁盘一次只由一台主机保留。正常情况下，这两台主机都会处于启动状态，并且彼此之间可以进行通信。

---

注—如果无意中确定某一磁盘未被保留（可能的原因是，使用该磁盘集的另一台主机以强制方式接管了该磁盘），则该主机将出现紧急情况。此行为有助于最大限度地减少在两台主机同时访问同一磁盘时所造成的数据丢失。

---

有关获取或保留磁盘集的更多信息，请参见第 190 页中的“如何获取磁盘集”。

## 释放磁盘集

在对磁盘集中的物理磁盘执行维护操作时，释放磁盘集可能会很有用。释放某一磁盘集后，主机便无法访问该磁盘集。如果在某一磁盘集中有两台主机同时释放了该磁盘集，则该磁盘集中的这两台主机均无法访问该磁盘集中的磁盘。

有关释放磁盘集的更多信息，请参见第 192 页中的“如何释放磁盘集”。

## 导入磁盘集

从 Solaris 9 9/04 发行版开始，可以使用 `metaimport` 命令将磁盘集（包括复制的磁盘集）导入到支持在磁盘集中使用设备 ID 的现有 Solaris Volume Manager 配置中。您还可以使用 `metaimport` 命令报告可供导入的磁盘集。

复制的磁盘集是通过远程复制软件创建的。要使用 `metaimport` 命令导入复制的磁盘集，包含该磁盘集中每个磁盘的状态数据库副本的分片也必须复制到复制的磁盘集的相同分片中。对于非 EFI 磁盘，该分片为分片 7，而对于 EFI 磁盘，该分片为分片 6。在复制磁盘集之前，请确保要复制的数据的磁盘配置与远程站点的磁盘配置相匹配。此步骤可以确保状态数据库副本和数据均可准确地进行复制。

如果磁盘集中的某个磁盘不包含卷或状态数据库副本，则 `metaimport` 命令也不会导入该磁盘。如果卷或状态数据库副本尚未添加到磁盘中或已从磁盘删除，则适用于这种情况。在这种情况下，如果将该磁盘集导入到另一系统中，则会发现该磁盘集中缺

少该磁盘。例如，每个 Solaris Volume Manager 磁盘集所允许的最多状态数据库副本数为 50。如果某一磁盘集有 60 个磁盘，则会有 10 个磁盘不包含状态数据库副本，这些磁盘必须包含卷，才能随该磁盘集一起导入。

有关与导入磁盘集相关的任务，请参见第 195 页中的“导入磁盘集”。

## 自动磁盘分区

在向磁盘集添加新磁盘时，Solaris Volume Manager 将检查磁盘格式。如果需要，Solaris Volume Manager 会重新对磁盘分区，以确保该磁盘的分片 7 配置正确，并且具有足够的空间来容纳状态数据库副本。分片 7 的精确大小取决于磁盘几何参数。但是，此大小不会小于 4 MB，并且可以接近于 6 MB（具体取决于柱面边界所在位置）。

缺省情况下，Solaris Volume Manager 会将一个状态数据库副本放置在分片 7 上。您可以增加分片 7 的缺省大小或减小状态数据库副本的大小，以便在该分片中容纳多个状态数据库副本。

---

**注** – 分片 7 的最小大小可能会在日后因各种因素（包括状态数据库副本大小以及要存储在状态数据库副本中的信息）而发生变化。多属主磁盘集中的状态数据库副本的缺省大小为 16 MB。

---

为了能够在磁盘集中使用，磁盘的分片 7 必须满足以下条件：

- 从扇区 0 开始
- 具有足够的空间来容纳磁盘标签和状态数据库副本
- 不能进行挂载
- 不与其他任何分片（包括分片 2）重叠

如果现有分区表不满足以上条件，则 Solaris Volume Manager 将重新对磁盘分区。每个驱动器会在分片 7 中保留一小部分供 Solaris Volume Manager 使用，而每个驱动器上的其余空间则放置在分片 0 中。重新分区时，磁盘上的任何现有数据都会丢失。

---

**提示** – 在向磁盘集添加驱动器后，您可以根据需要重新对该驱动器分区，除非分片 7 未以任何方式发生更改。

---

prtvtoc 命令的以下输出显示的是添加到磁盘集之前的磁盘。

```
[root@lexicon:apps]$ prtvtoc /dev/rdisk/clt6d0s0
* /dev/rdisk/clt6d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   133 sectors/track
```

```
*      27 tracks/cylinder
*      3591 sectors/cylinder
*      4926 cylinders
*      4924 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*
* Partition Tag  Flags      First      Sector      Last
* Partition Tag  Flags      Sector      Count      Sector  Mount Directory
*   0         2    00          0        4111695    4111694
*   1         3    01      4111695    1235304    5346998
*   2         5    01          0    17682084    17682083
*   3         0    00      5346999    4197879    9544877
*   4         0    00      9544878    4197879    13742756
*   5         0    00    13742757    3939327    17682083
```

以上输出显示该磁盘不包含分片 7。因此，在向磁盘集中添加该磁盘时，Solaris Volume Manager 会重新对该磁盘分区。prtvtoc 命令的以下输出显示的是添加到磁盘集之后的磁盘。

```
[root@lexicon:apps]$ prtvtoc /dev/rdisk/clt6d0s0
* /dev/rdisk/clt6d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   133 sectors/track
*   27 tracks/cylinder
*   3591 sectors/cylinder
*   4926 cylinders
*   4924 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*
* Partition Tag  Flags      First      Sector      Last
* Partition Tag  Flags      Sector      Count      Sector  Mount Directory
*   0         0    00      10773    17671311    17682083
*   7         0    01          0      10773      10772
```

以上输出显示该磁盘已进行重新分区，其中包含分片 7，该分片从柱面 0 开始，并且具有足够的空间来容纳状态数据库副本。如果添加到磁盘集的每个磁盘都有一个可接受的分片 7，则这些磁盘就不会进行重新格式化。

---

注 – 如果磁盘集是从 Solstice DiskSuite 软件升级的，则这些磁盘集上的缺省状态数据库副本大小为 1034 个块，而不是 Solaris Volume Manager 中的 8192 个块。此外，通过 Solstice DiskSuite 软件添加的磁盘上的分片 7 会相应地小于通过 Solaris Volume Manager 添加的磁盘上的分片 7。

---

## 磁盘集名称要求

磁盘集卷名与其他 Solaris Volume Manager 组件名称类似。但是，在磁盘集卷名中会包括磁盘集名称。例如，在卷路径名中的 `/dev/md/` 之后以及实际卷名之前会包括磁盘集名称。

下表显示了一些磁盘集卷名示例。

表 18-1 磁盘集卷名示例

<code>/dev/md/blue/dsk/d0</code>	磁盘集 <code>blue</code> 中的块卷 <code>d0</code>
<code>/dev/md/blue/dsk/d1</code>	磁盘集 <code>blue</code> 中的块卷 <code>d1</code>
<code>/dev/md/blue/rdsk/d126</code>	磁盘集 <code>blue</code> 中的原始卷 <code>d126</code>
<code>/dev/md/blue/rdsk/d127</code>	磁盘集 <code>blue</code> 中的原始卷 <code>d127</code>

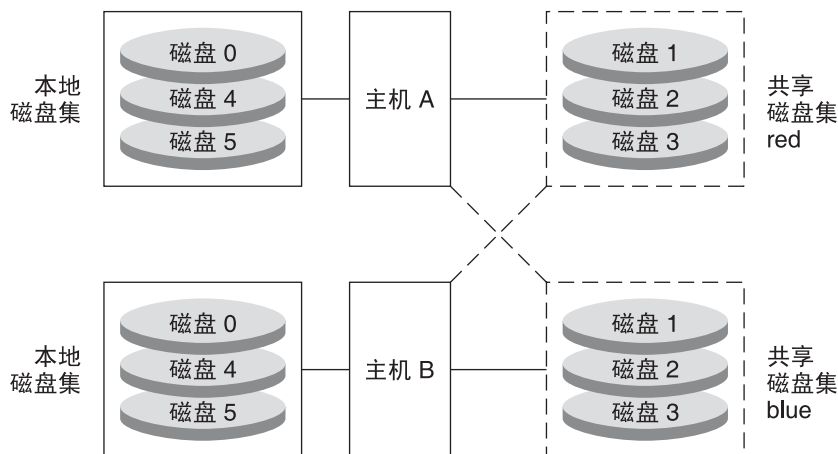
同样，热备用池会在热备件名称中包括磁盘集名称。

## 示例一 两个共享磁盘集

图 18-1 显示了一个使用两个磁盘集的配置示例。

在此配置中，主机 A 和主机 B 共享磁盘集 `red` 和 `blue`。这两个主机都有各自的本地磁盘集，并且这些本地磁盘集未共享。如果主机 A 出现故障，主机 B 可以接管主机 A 的共享磁盘集（即磁盘集 `red`）的控制权。同样，如果主机 B 出现故障，主机 A 可以接管主机 B 的共享磁盘集（即磁盘集 `blue`）的控制权。

图 18-1 磁盘集示例



## 磁盘集使用原则

使用磁盘集时，请考虑以下原则：

- 必须在要连接到该磁盘集的每台主机上配置 Solaris Volume Manager。
- 在创建磁盘集之前，每台主机都必须安装有自己的本地状态数据库。
- 创建磁盘集并为该磁盘集创建卷的步骤顺序为：首先创建磁盘集，然后向磁盘集中添加磁盘，最后在该磁盘集中创建卷。
- 要在群集环境中创建并使用磁盘集，root 必须是组 14 的成员。或者，每台主机上的 `/etc/rhosts` 文件都必须包含相应的条目来说明与该磁盘集关联的其他主机名。

---

注 - 在 SunCluster 3.x 环境中，不需要此步骤。

---

- 要对磁盘集执行维护操作，必须有一台主机是该磁盘集的属主或保留该磁盘集。主机可以通过向磁盘集中放置第一个驱动器来获取该磁盘集的隐式所有权。
- 不能向文件系统、数据库或任何其他应用程序正在使用的磁盘集中添加驱动器。在添加驱动器之前，请确保此驱动器当前未被使用。
- 不要向磁盘集中添加包含要保留的现有数据的驱动器。向磁盘集添加磁盘这一过程会重新对磁盘分区并销毁现有数据。
- 与本地卷管理不同，您无需手动创建或删除磁盘集中的状态数据库副本。Solaris Volume Manager 会尝试在磁盘集中的所有驱动器之间平均分布合理数量的状态数据库副本。
- 向磁盘集添加驱动器之后，Solaris Volume Manager 会在其余驱动器之间重新平均分布状态数据库副本。如果需要，您可以稍后使用 `metadb` 命令更改副本布局。

## 磁盘集中的异步共享存储

在先前版本的 Solaris Volume Manager 中，计划在磁盘集中的主机之间共享的所有磁盘必须连接到每台主机。并且这些磁盘还必须在每台主机上具有完全相同的路径、驱动程序和名称。具体来说，两台主机必须能够在同一位置 (`/dev/rdisk/c#t#d#`) 识别共享磁盘驱动器。此外，共享磁盘还必须使用相同的驱动程序名称 (`ssd`)。

在当前 Solaris OS 发行版中，对于以不同方式访问通用存储的各个系统，可通过非并行方式共享访问磁盘集。为磁盘集引入设备 ID 支持后，Solaris Volume Manager 可自动跟踪磁盘在已命名磁盘集中的移动情况。

升级到最新的 Solaris OS 后，要启用磁盘跟踪功能，需要执行一次磁盘集获取。有关获取磁盘集的更多信息，请参见第 190 页中的“[如何获取磁盘集](#)”。

如果未启用自动获取功能，则必须手动获取每个磁盘集。如果已启用此功能，则此步骤将在系统重新引导时自动执行。有关自动获取功能的更多信息，请参见第 174 页中的“[自动获取 \(autotake\) 磁盘集](#)”。

通过这一扩展的设备 ID 支持，您还可以导入磁盘集，即使这些磁盘集是在不同系统上创建的。有关导入磁盘集的更多信息，请参见第 177 页中的“[导入磁盘集](#)”。

## 方案一 磁盘集

以下示例（即，在第 5 章，[配置和使用 Solaris Volume Manager（方案）](#)中显示的样例系统图）说明了应如何使用磁盘集来管理驻留在 SAN（Storage Area Network，存储区域网络）光纤网络上的存储。

假设该样例系统具有一个附加控制器，用于连接到光纤交换机和 SAN 存储。与其他设备（如 SCSI 和 IDE 磁盘）一样，SAN 光纤网络上的存储在引导过程的初期便不能用于系统。此外，Solaris Volume Manager 还会将此光纤网络上的逻辑卷报告为引导时不可用。但是，通过将该存储添加到磁盘集中，然后使用磁盘集工具管理该存储，即可避免这一引导时可用性问题。另外，还可以在本地存储中的单独的、由磁盘集控制的名称空间中方便地管理与光纤网络连接的存储。

## 磁盘集（任务）

---

本章提供有关执行与磁盘集关联的任务的信息。有关这些任务中所涉及概念的信息，请参见第 18 章，[磁盘集（概述）](#)。

### 磁盘集（任务列表）

以下任务列表列出了管理 Solaris Volume Manager 磁盘集和 Solaris Volume Manager for Sun Cluster 多属主磁盘集所需的过程。除非另有说明，否则所有命令都适用于这两种类型的磁盘集。Solaris Volume Manager GUI 不适用于与多属主磁盘集关联的任务。

任务	说明	参考
创建磁盘集	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令创建磁盘集。  使用 <code>metaset -M</code> 命令创建 多属主磁盘集。	<a href="#">第 184 页中的“如何创建磁盘集”</a>
向磁盘集添加磁盘	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令向磁盘集添加磁盘。	<a href="#">第 185 页中的“如何向磁盘集添加磁盘”</a>
向磁盘集添加主机	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令向磁盘集添加主机。	<a href="#">第 187 页中的“如何向磁盘集添加其他主机”</a>
在磁盘集中创建 Solaris Volume Manager 卷	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令在磁盘集中创建卷。	<a href="#">第 188 页中的“如何在磁盘集中创建 Solaris Volume Manager 组件”</a>
检查磁盘集的状态	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令检查磁盘集的状态。	<a href="#">第 189 页中的“如何检查磁盘集的状态”</a>
从磁盘集中删除磁盘	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令从磁盘集中删除磁盘。	<a href="#">第 190 页中的“如何从磁盘集中删除磁盘”</a>

任务	说明	参考
获取磁盘集	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令获取磁盘集。	<a href="#">第 190 页中的“如何获取磁盘集”</a>
释放磁盘集	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令释放磁盘集。	<a href="#">第 192 页中的“如何释放磁盘集”</a>
从磁盘集中删除主机	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令从磁盘集中删除主机。	<a href="#">第 193 页中的“如何删除主机或磁盘集”</a>
删除磁盘集	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令从磁盘集中删除最后一台主机，从而删除磁盘集。	<a href="#">第 193 页中的“如何删除主机或磁盘集”</a>
导入磁盘集	使用 <code>metainport</code> 命令针对磁盘集运行报告以确定可导入的磁盘集，并将磁盘集从一个系统导入另一个系统中。	<a href="#">第 195 页中的“导入磁盘集”</a>

# 创建磁盘集

## ▼ 如何创建磁盘集

开始之前 查看 [第 181 页中的“磁盘集使用原则”](#)。

1 要创建磁盘集，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中打开 "Disk Sets"（磁盘组）节点。选择 "Action"（操作）⇒ "Create Disk Set"（创建磁盘组）。然后，按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要使用命令行从头创建磁盘集，请使用以下格式的 `metaset` 命令：  

```
# metaset -s diskset-name -a -h -M hostname
```

`-s diskset-name` 指定将针对其运行 `metaset` 命令的磁盘集的名称。

`-a` 将主机添加到指定的磁盘集。Solaris Volume Manager 支持每个磁盘集最多接受四台主机。

`-M` 指定要创建的磁盘集是多属主磁盘集。

`-h hostname` 指定要添加到磁盘集中的一台或多台主机。添加第一台主机会创建磁盘集。随后可以添加第二台主机。但是，如果在指定的 `hostname` 中无法找到磁盘集中的所有磁盘，则不接受第二台主机。`hostname` 与在 `/etc/nodename` 文件中找到的名称相同。

有关更多信息，请参见 [metaset\(1M\)](#) 手册页。

## 2 检查新的磁盘集的状态。

```
# metaset
```

### 示例 19-1 创建磁盘集

在以下示例中，将从主机 `host1` 创建名为 `blue` 的共享磁盘集。`metaset` 命令将显示状态信息。此时，磁盘集没有属主。缺省情况下，向磁盘集添加磁盘的主机将变成属主。

```
# metaset -s blue -a -h host1
# metaset
Set name = blue, Set number = 1

Host                Owner
host1
```

### 示例 19-2 创建多属主磁盘集

在以下示例中，将创建名为 `red` 的多属主磁盘集。`metaset` 命令输出的第一行显示了 "Multi-owner"，表示该磁盘集是多属主磁盘集。

```
# metaset -s red -a -M -h nodeone
# metaset -s red
Multi-owner Set name = red, Set number = 1, Master =

Host                Owner                Member
nodeone
```

## 扩展磁盘集

### ▼ 如何向磁盘集添加磁盘



**注意** – 如果希望运行具有 32 位内核的 Solaris 软件，或者使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，请不要向磁盘集添加大于 1 TB 的磁盘。有关 Solaris Volume Manager 中的多 TB 卷支持的更多信息，请参见第 41 页中的“[Solaris Volume Manager 中的多 TB 支持概述](#)”。

只有满足以下条件的磁盘才能添加到磁盘集中：

- 磁盘不得在卷或热备用池中使用。
- 磁盘不得包含状态数据库副本。
- 磁盘当前不能已挂载或者处于交换或打开状态以供某应用程序使用。

开始之前 查看第 181 页中的“磁盘集使用原则”。

1 要向磁盘集添加磁盘，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中打开 "Disk Sets"（磁盘组）节点。选择要修改的磁盘集。然后单击鼠标右键并选择 "Properties"（属性）。选择 "Disks"（磁盘）选项卡。单击 "Add Disk"（增加磁盘）。然后，按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要从命令行向磁盘集添加磁盘，请使用以下格式的 `metaset` 命令：

```
# metaset -s diskset-name -a disk-name
```

`-s diskset-name` 指定将针对其运行 `metaset` 命令的磁盘集的名称。

`-a` 向已命名磁盘集添加磁盘。

`disk-name` 指定要向磁盘集添加的磁盘。磁盘名称的形式为 `cxtxdx`。向磁盘集添加磁盘时，不包含 "sx" 分片标识符。

有关更多信息，请参见 [metaset\(1M\)](#) 手册页。

向磁盘集添加磁盘的第一台主机将成为该磁盘集的属主。



**注意** – 请勿将带有数据的磁盘添加到磁盘集。将带有数据的磁盘添加到磁盘集的过程可能会导致重新为磁盘设置分区，从而销毁数据。

2 验证磁盘集和磁盘的状态。

```
# metaset
```

示例 19-3 向磁盘集添加磁盘

```
# metaset -s blue -a clt6d0
# metaset
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes
Drive	Dbase
clt6d0	Yes

在此示例中，主机名为 `host1`。共享磁盘集为 `blue`。只有磁盘 `clt6d0` 被添加到磁盘集 `blue` 上。

或者，也可以通过在命令行中列出各个磁盘来同时添加多个磁盘。例如，您可以使用以下命令同时向磁盘集添加两个磁盘：

```
# metaset -s blue -a clt6d0 c2t6d0
```

## ▼ 如何向磁盘集添加其他主机

此过程说明如何向只有一台主机的现有磁盘集添加另一台主机。Solaris Volume Manager 支持每个磁盘集最多接受四台主机。

开始之前 查看第 181 页中的“磁盘集使用原则”。

### 1 要向磁盘集添加主机，请使用以下方法之一：

- 在 Solaris Management Console 中增强的存储工具中，打开 "Disk Sets"（磁盘组）节点并选择要修改的磁盘集。选择要修改的磁盘集。然后单击鼠标右键并选择 "Properties"（属性）。选择 "Hosts"（主机）选项卡。单击 "Add Host"（添加主机）。然后，按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要从命令行向磁盘集添加主机，请使用以下格式的 `metaset` 命令：

```
# metaset -s diskset-name -a -h hostname
-s diskset-name    指定向其添加主机的磁盘集的名称。
-a                向已命名磁盘集添加主机。
-h hostname        指定要添加到磁盘集中的一台或多台主机的名称。添加第一台主
                  机会创建磁盘集。该主机名与 /etc/nodename 文件中的名称相
                  同。
```

有关更多信息，请参见 `metaset(1M)` 手册页。

### 2 验证主机是否已添加到磁盘集中。

```
# metaset
```

#### 示例 19-4 向磁盘集添加其他主机

```
# metaset -s blue -a -h host2
# metaset
Set name = blue, Set number = 1

Host                Owner
host1               Yes
host2

Drive               Dbase
c1t6d0              Yes
c2t6d0              Yes
```

本示例说明如何将主机 `host2` 添加到磁盘集 `blue`。

## ▼ 如何在磁盘集中创建 Solaris Volume Manager 组件

创建磁盘集后，可以使用添加到磁盘集中的磁盘创建卷和热备用池。您可以使用 Solaris Management Console 中增强的存储工具或命令行实用程序。

- 要在磁盘集中创建卷或其他 Solaris Volume Manager 组件，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）、"State Database Replicas"（状态数据库复制）或 "Hot Spare Pools"（联机备用磁盘群组）节点。选择 "Action"（操作）⇒ "Create"（创建）。然后，按照向导中的说明操作。有关更多信息，请参见联机帮助。
  - 使用相同的命令行实用程序和相同的基本语法来创建卷、状态数据库副本或热备用池。但是，在每个命令后面都还要紧跟 `-s disk-set`。

`# command -s disk-set`

### 示例 19-5 在磁盘集中创建 Solaris Volume Manager 卷

以下示例说明如何在磁盘集 `blue` 中创建镜像 `d10`。该镜像由子镜像（RAID-0 卷）`d11` 和 `d12` 组成。

```
# metainit -s blue d11 1 1 c1t6d0s0
blue/d11: Concat/Stripe is setup
# metainit -s blue d12 1 1 c2t6d0s0
blue/d12: Concat/Stripe is setup
# metainit -s blue d10 -m d11
blue/d10: Mirror is setup
# metattach -s blue d10 d12
blue/d10: submirror blue/d12 is attached

# metastat -s blue
blue/d10: Mirror
  Submirror 0: blue/d11
    State: Okay
  Submirror 1: blue/d12
    State: Resyncing
  Resync in progress: 0 % done
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 17674902 blocks

blue/d11: Submirror of blue/d10
  State: Okay
  Size: 17674902 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Reloc  Hot Spare
    c1t6d0s0          0          No    Okay

blue/d12: Submirror of blue/d10
  State: Resyncing
```

```
Size: 17674902 blocks
Stripe 0:
  Device          Start Block  Dbase State      Reloc Hot Spare
  c2t6d0s0        0           No    Okay
```

# 维护磁盘集

## ▼ 如何检查磁盘集的状态

- 使用以下方法之一检查磁盘集的状态。
  - 从 Solaris Management Console 中增强的存储工具中打开 "Disk Sets"（磁盘组）节点。在要监视的磁盘集上单击鼠标右键。然后从菜单中选择 "Properties"（属性）。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metaset` 命令查看磁盘集状态。

```
# metaset -s diskset-name
```

有关更多信息，请参见 [metaset\(1M\)](#) 手册页。

---

注 - 仅在所属主机上显示磁盘集所有权。

---

### 示例 19-6 检查指定磁盘集的状态

以下示例显示了 `-s` 选项后跟磁盘集名称 `blue` 的 `metaset` 命令。该命令的输出显示了该磁盘集的状态信息。输出表示 `host1` 是该磁盘集的属主。`metaset` 命令还显示磁盘集中的磁盘。

```
red# metaset -s blue

Set name = blue, Set number = 1

Host          Owner
host1         Yes

Drive         Dbase
c1t6d0        Yes
c2t6d0        Yes
```

如果只使用 `metaset` 命令，将显示所有磁盘集的状态。

## ▼ 如何从磁盘集中删除磁盘

1 要从磁盘集中删除磁盘，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中打开 "Disk Sets"（磁盘组）节点。在要释放的磁盘集上单击鼠标右键。然后从菜单中选择 "Properties"（属性）。单击 "Disks"（磁盘）选项卡。按照联机帮助中的说明操作。
- 使用以下格式的 `metaset` 命令从磁盘集中删除磁盘。

```
# metaset -s diskset-name -d disk-name
```

`-s diskset-name`      指定要从中删除磁盘的磁盘集名称。

`-d disk-name`          指定要从磁盘集中删除的磁盘。磁盘名称的形式为 `cxtxdx`。从磁盘集删除磁盘时，不包含 "sx" 分片标识符。

有关更多信息，请参见 [metaset\(1M\)](#) 手册页。

2 验证是否已从磁盘集中删除了磁盘。

```
# metaset -s diskset-name
```

---

注 - 要删除磁盘集，首先必须从磁盘集中删除所有磁盘。

---

### 示例 19-7 从磁盘集中删除磁盘

以下示例说明如何从磁盘集 `blue` 中删除磁盘 `c1t6d0`。

```
host1# metaset -s blue -d c1t6d0
host1# metaset -s blue

Set name = blue, Set number = 1

Host                               Owner
  host1
  host2

Drive                               Dbase
  c2t6d0                             Yes
```

## ▼ 如何获取磁盘集

---

注 - 此选项不可用于多属主磁盘集。

---

- 通过以下方法之一获取磁盘集：

- 从 Solaris Management Console 中增强的存储工具中打开 "Disk Sets"（磁盘组）节点。在要获取的磁盘集上单击鼠标右键。然后从菜单中选择 "Take Ownership"（取得所有权）。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metaset` 命令：

```
# metaset -s diskset-name -t -f
-s diskset-name    指定要获取的磁盘集的名称。
-t                指定获取磁盘集。
-f                指定强制获取磁盘集。
```

有关更多信息，请参见 [metaset\(1M\)](#) 手册页。

一个磁盘集在一段时间内只能由一台主机所有。当某个磁盘集中的一台主机获取该磁盘集后，该磁盘集中的其他主机将无法访问该磁盘集中各磁盘上的数据。

`metaset` 命令的缺省行为是：只有拥有磁盘集的主机释放该磁盘集时，才允许其他主机获取该磁盘集。使用 `-f` 选项可强制获取磁盘集。无论该磁盘集当前是否为另一台主机所有，此选项都会获取该磁盘集。当磁盘集中的一台主机已关闭或者无法与之通信时，请使用此方法。如果此时另一台主机已经获取了该磁盘集，那么，当该主机尝试针对磁盘集执行 I/O 操作时，将会发生紧急情况。

---

注- 仅在所属主机上显示磁盘集所有权。

---

### 示例 19-8 获取磁盘集

在以下示例中，主机 `host1` 与主机 `host2` 进行通信。此通信可以确保主机 `host2` 在主机 `host1` 尝试获取磁盘集前释放该磁盘集。

```
host1# metaset
...
Set name = blue, Set number = 1

Host                Owner
  host1
  host2
...
host1# metaset -s blue -t
host1# metaset
...
Set name = blue, Set number = 1

Host                Owner
  host1                Yes
  host2
...

```

如果 host2 拥有磁盘集 blue，则上面输出中的 "Owner"（属主）列将仍然为空。metaset 命令仅显示发出命令的主机是否拥有磁盘集。

### 示例 19-9 强制获取磁盘集

在以下示例中，正在获取该磁盘集的主机不会与另一台主机通信，而是通过 -f 选项在不发出警告的情况下强制获取该磁盘集中的磁盘。如果另一台主机拥有该磁盘集，则当该主机尝试对该磁盘集执行 I/O 操作时，该主机将会发生紧急情况。

```
# metaset -s blue -t -f
```

## ▼ 如何释放磁盘集

在对磁盘集中的物理磁盘执行维护操作时，释放磁盘集会很有用。释放某一磁盘集后，主机便无法访问该磁盘集。如果磁盘集中的两台主机都释放了该磁盘集，则这两台主机均不能直接访问该磁盘集中所定义的卷或热备用池。但是，如果两台主机都释放了磁盘集，则这两台主机可以通过磁盘的 c\*t\*d\* 名称直接访问磁盘。

---

注-此选项不可用于多属主磁盘集。

---

开始之前 查看第 181 页中的“磁盘集使用原则”。

#### 1 通过以下方法之一释放磁盘集：

- 从 Solaris Management Console 中增强的存储工具中打开 "Disk Sets"（磁盘组）节点。在要释放的磁盘集上单击鼠标右键。然后从菜单中选择 "Release Ownership"（释放所有权）。有关更多信息，请参见联机帮助。
- 要释放对磁盘集的所有权，请使用以下格式的 metaset 命令：

```
# metaset -s diskset-name -r
```

-s diskset-name 指定将针对其运行 metaset 命令的磁盘集的名称。

-r 释放对磁盘集的所有权。将取消对磁盘集中所有磁盘的保留。磁盘集中的卷将不再可访问。

有关更多信息，请参见 [metaset\(1M\)](#) 手册页。

---

注-仅在所属主机上显示磁盘集所有权。

---

#### 2 验证是否已在该主机上释放磁盘集。

```
# metaset
```

### 示例 19-10 释放磁盘集

以下示例说明如何释放磁盘集 `blue`。请注意，该磁盘集没有属主。从主机 `host1` 查看状态可能具有误导性。主机只能确定它自己是否拥有磁盘集。例如，如果主机 `host2` 将拥有磁盘集的所有权，则不会对主机 `host1` 显示所有权。只有主机 `host2` 会显示 `host2` 拥有该磁盘集的所有权。

```
host1# metaset -s blue -r
host1# metaset -s blue

Set name = blue, Set number = 1

Host                Owner
  host1
  host2

Drive              Dbase
  c1t6d0            Yes
  c2t6d0            Yes
```

## ▼ 如何删除主机或磁盘集

如果要删除某个磁盘集，要求该磁盘集中不包含磁盘，并且没有其他主机连接到该磁盘集。删除最后一台主机将销毁该磁盘集。

### 1 使用以下方法之一从磁盘集中删除主机，或者删除磁盘集：

- 从 Solaris Management Console 中增强的存储工具中打开 "Disk Sets"（磁盘组）节点。在要释放的磁盘集上单击鼠标右键，然后从菜单中选择 "Delete"（删除）。按照联机帮助中的说明操作。
- 要删除主机，请使用以下格式的 `metaset` 命令。

```
metaset -s diskset-name -d -h hostname
-s diskset-name    指定将针对其运行 metaset 命令的磁盘集的名称。
-d                从磁盘集中删除主机。
-h hostname        指定要删除的主机的名称。
```

要删除磁盘集，请使用上述 `metaset` 命令形式。如果要删除某个磁盘集，要求该磁盘集中不包含磁盘，并且没有其他主机拥有该磁盘集。删除最后一台主机将销毁该磁盘集。

有关更多信息，请参见 [metaset\(1M\)](#) 手册页。

- 2 使用 **metaset** 命令来验证该主机是否已从磁盘集中删除。请注意，仅显示当前（所属）主机。其他主机已被删除。

```
# metaset -s disk-set
```

### 示例 19-11 从磁盘集中删除主机

以下示例说明如何从磁盘集 **blue** 中删除主机 **host2**。

```
# metaset -s blue
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes
..host2	

Drive	Dbase
clt2d0	Yes
clt3d0	Yes
clt4d0	Yes
clt5d0	Yes
clt6d0	Yes
c2t1d0	Yes

```
# metaset -s blue -d -h host2
```

```
# metaset -s blue
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes

Drive	Dbase
clt2d0	Yes
clt3d0	Yes
clt4d0	Yes
clt5d0	Yes
clt6d0	Yes
c2t1d0	Yes

### 示例 19-12 从磁盘集中删除最后一台主机

以下示例说明如何从磁盘集 **blue** 中删除最后一台主机。

```
host1# metaset -s blue -d -h host1
host1# metaset -s blue

metaset: host: setname "blue": no such set
```

# 导入磁盘集

## 导入磁盘集

通过 `metaimport` 命令，可以将磁盘集从一个系统导入另一个系统。

### ▼ 如何打印有关可导入的磁盘集的报告

- 1 成为超级用户。
- 2 获取有关可导入的磁盘集的报告。

```
# metaimport -r -v
```

-r 提供可导入到系统上的未配置磁盘集的报告。

-v 提供有关以下内容的详细信息：状态数据库 (metadb) 副本的位置；可导入到系统上的未配置磁盘集中各磁盘的状态。

#### 示例 19-13 报告可导入的磁盘集

以下示例说明如何打印有关可导入磁盘集的报告。`metaimport` 命令的输出对常规磁盘集和复制的磁盘集进行了区分。

```
# metaimport -r
# metaimport -r
Drives in regular diskset including disk c1t2d0:
  c1t2d0
  c1t3d0
More info:
  metaimport -r -v c1t2d0
Import: metaimport -s <newsetname> c1t2d0
Drives in replicated diskset including disk c1t4d0:
  c1t4d0
  c1t5d0
More info:
  metaimport -r -v c1t4d0
Import: metaimport -s <newsetname> c1t4d0
# metaimport -r -v c1t2d0
Import: metaimport -s <newsetname> c1t2d0
Last update: Mon Dec 29 14:13:35 2003
Device      offset      length replica flags
c1t2d0       16         8192      a      u
c1t3d0       16         8192      a      u
```

## ▼ 如何将磁盘集从一个系统导入到另一个系统

1 成为超级用户。

2 验证磁盘集是否可导入。

```
# metainport -r -v
```

3 导入可用磁盘集。

```
# metainport -s diskset-name disk-name
```

-s *diskset-name*      指定要创建的磁盘集的名称。

*disk-name*              标识要导入的磁盘集中包含状态数据库副本的磁盘 (c#t#d#)。

4 验证该磁盘集是否已导入。

```
# metaset -s diskset-name
```

### 示例 19-14 导入磁盘集

以下示例说明如何导入磁盘集。

```
# metainport -s red c1t2d0
Drives in diskset including disk c1t2d0:
  c1t2d0
  c1t3d0
  c1t8d0
More info:
  metainport -r -v c1t2d0
# metaset -s red
```

```
Set name = red, Set number = 1
```

Host	Owner
host1	Yes

Drive	Dbase
c1t2d0	Yes
c1t3d0	Yes
c1t8d0	Yes

## 维护 Solaris Volume Manager（任务）

---

本章提供有关使用 Solaris Volume Manager 执行一般存储管理维护任务的信息。

以下是本章中信息的列表：

- 第 197 页中的“Solaris Volume Manager 维护（任务列表）”
- 第 198 页中的“查看 Solaris Volume Manager 配置”
- 第 201 页中的“重命名卷”
- 第 204 页中的“使用配置文件”
- 第 206 页中的“更改 Solaris Volume Manager 缺省值”
- 第 206 页中的“使用 `growfs` 命令扩展文件系统”
- 第 208 页中的“替换和启用 RAID-1 和 RAID-5 卷中的组件概述”

### Solaris Volume Manager 维护（任务列表）

以下任务列表列出了维护 Solaris Volume Manager 所需的过程。

任务	说明	参考
查看 Solaris Volume Manager 配置	使用 Solaris Volume Manager GUI 或 <code>metastat</code> 命令查看系统配置。	第 198 页中的“如何查看 Solaris Volume Manager 卷配置”
重命名卷	使用 Solaris Volume Manager GUI 或 <code>metarename</code> 命令重命名卷。	第 203 页中的“如何重命名卷”
创建配置文件	使用 <code>metastat -p</code> 命令和 <code>metadb</code> 命令创建配置文件。	第 204 页中的“如何创建配置文件”
根据配置文件初始化 Solaris Volume Manager	使用 <code>metainit</code> 命令根据配置文件初始化 Solaris Volume Manager。	第 204 页中的“如何根据配置文件初始化 Solaris Volume Manager”

任务	说明	参考
扩展文件系统	可以使用 <code>growfs</code> 命令扩展文件系统。	<a href="#">第 207 页中的“如何扩展文件系统”</a>
启用组件	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令启用组件。	<a href="#">第 208 页中的“启用组件”</a>
替换组件	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令替换组件。	<a href="#">第 209 页中的“将一个组件替换为另一个可用组件”</a>

## 查看 Solaris Volume Manager 配置

提示 `-metastat` 命令不会对输出进行排序。将 `metastat -p` 命令的输出传输到 `sort` 或 `grep` 命令以获得更易管理的配置列表。

### ▼ 如何查看 Solaris Volume Manager 卷配置

- 要查看卷配置，请使用以下方法之一：
  - 在 Solaris Management Console 中增强的存储工具中，打开 "Volumes"（卷）节点。有关更多信息，请参见联机帮助。
  - 使用以下格式的 `metastat` 命令：

```
# metastat -p -i component-name
```

`-p`                    指定以在概括性摘要中显示输出。此输出适用于创建 `md.tab` 文件。

`-i`                    指定以验证可以访问 RAID-1（镜像）卷、RAID-5 卷和热备件。

`component-name`    指定要查看的卷的名称。如果未指定卷名称，将显示组件的完整列表。

#### 示例 20-1 查看 Solaris Volume Manager 卷配置

以下示例显示 `metastat` 命令的输出。

```
# metastat
d50: RAID
    State: Okay
    Interlace: 32 blocks
    Size: 20985804 blocks
Original device:
```

```
Size: 20987680 blocks
Device      Start Block  Dbase State      Reloc Hot Spare
c1t4d0s5    330         No  Okay      Yes
c1t5d0s5    330         No  Okay      Yes
c2t4d0s5    330         No  Okay      Yes
c2t5d0s5    330         No  Okay      Yes
c1t1d0s5    330         No  Okay      Yes
c2t1d0s5    330         No  Okay      Yes

d1: Concat/Stripe
Size: 4197879 blocks
Stripe 0:
Device      Start Block  Dbase Reloc
c1t2d0s3    0           No   Yes

d2: Concat/Stripe
Size: 4197879 blocks
Stripe 0:
Device      Start Block  Dbase Reloc
c2t2d0s3    0           No   Yes

d80: Soft Partition
Device: d70
State: Okay
Size: 2097152 blocks
Extent      Start Block      Block count
0           1                2097152

d81: Soft Partition
Device: d70
State: Okay
Size: 2097152 blocks
Extent      Start Block      Block count
0           2097154          2097152

d70: Mirror
Submirror 0: d71
State: Okay
Submirror 1: d72
State: Okay
Pass: 1
Read option: roundrobin (default)
Write option: parallel (default)
Size: 12593637 blocks

d71: Submirror of d70
State: Okay
Size: 12593637 blocks
Stripe 0:
Device      Start Block  Dbase State      Reloc Hot Spare
c1t3d0s3    0           No  Okay      Yes
Stripe 1:
Device      Start Block  Dbase State      Reloc Hot Spare
c1t3d0s4    0           No  Okay      Yes
Stripe 2:
Device      Start Block  Dbase State      Reloc Hot Spare
c1t3d0s5    0           No  Okay      Yes
```

```
d72: Submirror of d70
State: Okay
Size: 12593637 blocks
Stripe 0:
Device          Start Block  Dbase State      Reloc Hot Spare
c2t3d0s3        0           No  Okay       Yes
Stripe 1:
Device          Start Block  Dbase State      Reloc Hot Spare
c2t3d0s4        0           No  Okay       Yes
Stripe 2:
Device          Start Block  Dbase State      Reloc Hot Spare
c2t3d0s5        0           No  Okay       Yes

hsp010: is empty

hsp014: 2 hot spares
Device          Status      Length      Reloc
c1t2d0s1        Available  617652 blocks Yes
c2t2d0s1        Available  617652 blocks Yes

hsp050: 2 hot spares
Device          Status      Length      Reloc
c1t2d0s5        Available  4197879 blocks Yes
c2t2d0s5        Available  4197879 blocks Yes

hsp070: 2 hot spares
Device          Status      Length      Reloc
c1t2d0s4        Available  4197879 blocks Yes
c2t2d0s4        Available  4197879 blocks Yes

Device Relocation Information:
Device          Reloc      Device ID
c1t2d0          Yes       id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N1S200002103AF29
c2t2d0          Yes       id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0P64Z00002105Q6J7
c1t1d0          Yes       id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N1EM00002104NP2J
c2t1d0          Yes       id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N93J000071040L3S
c0t0d0          Yes       id1,dad@s53554e575f4154415f5f53543339313430412525415933
```

示例 20-2 查看多 TB Solaris Volume Manager 卷

以下示例说明了针对多 TB 存储卷（11 TB）的 `metastat` 命令的输出。

```
# metastat d0
d0: Concat/Stripe
Size: 25074708480 blocks (11 TB)
Stripe 0: (interlace: 32 blocks)
Device          Start Block  Dbase Reloc
c27t8d3s0        0           No  Yes
c4t7d0s0        12288       No  Yes
Stripe 1: (interlace: 32 blocks)
Device          Start Block  Dbase Reloc
c13t2d1s0        16384       No  Yes
c13t4d1s0        16384       No  Yes
```

c13t6d1s0	16384	No	Yes
c13t8d1s0	16384	No	Yes
c16t3d0s0	16384	No	Yes
c16t5d0s0	16384	No	Yes
c16t7d0s0	16384	No	Yes
c20t4d1s0	16384	No	Yes
c20t6d1s0	16384	No	Yes
c20t8d1s0	16384	No	Yes
c9t1d0s0	16384	No	Yes
c9t3d0s0	16384	No	Yes
c9t5d0s0	16384	No	Yes
c9t7d0s0	16384	No	Yes
Stripe 2: (interlace: 32 blocks)			
Device	Start Block	Dbase	Reloc
c27t8d2s0	16384	No	Yes
c4t7d1s0	16384	No	Yes
Stripe 3: (interlace: 32 blocks)			
Device	Start Block	Dbase	Reloc
c10t7d0s0	32768	No	Yes
c11t5d0s0	32768	No	Yes
c12t2d1s0	32768	No	Yes
c14t1d0s0	32768	No	Yes
c15t8d1s0	32768	No	Yes
c17t3d0s0	32768	No	Yes
c18t6d1s0	32768	No	Yes
c19t4d1s0	32768	No	Yes
c1t5d0s0	32768	No	Yes
c2t6d1s0	32768	No	Yes
c3t4d1s0	32768	No	Yes
c5t2d1s0	32768	No	Yes
c6t1d0s0	32768	No	Yes
c8t3d0s0	32768	No	Yes

## 下一步执行的操作

有关更多信息，请参见 [metastat\(1M\)](#) 手册页。

# 重命名卷

## 重命名卷的背景信息

Solaris Volume Manager 允许您在任何时候重命名大多数卷类型（遵循某些限制）。可以使用 Solaris Management Console 中增强的存储工具或命令行（[metarename\(1M\)](#) 命令）重命名卷。

重命名卷或切换卷名可对卷名管理提供方便。例如，可以在需要的数字范围内排列所有文件系统挂载点。您可能会重命名卷以维护逻辑卷的命名模式，或允许事务卷使用相同名称作为底层卷的名称。

---

注 – 事务卷在 Solaris Volume Manager 中不再有效。可以重命名事务卷以替换它们。

---

重命名卷前，请确保其当前未在使用中。对于文件系统，请确保其未挂载或作为 `swap` 使用。其他使用原始设备的应用程序，如数据库，应当具备它们自己的方法来防止访问数据。

有关重命名卷的具体注意事项包括以下内容：

- 可以重命名任何卷，除以下卷之外：
  - 软分区
  - 直接在其上生成软分区的卷
  - 用作日志设备的卷
  - 热备用池
- 可以重命名磁盘集中的卷。但是，不可以重命名卷，以将它们从一个磁盘集移动到另一个磁盘集。

## 交换卷名称

使用带有 `-x` 选项的 `metarename` 命令可交换具有父子关系的卷的名称。有关更多信息，请参见第 203 页中的“如何重命名卷”和 `metarename(1M)` 手册页。现有卷的名称将与其某个子组件的名称进行交换。例如，这种类型的交换可以出现在镜像与其某个子镜像之间。使用 `metarename -x` 命令可以更加方便地镜像或取消镜像现有卷。

---

注 – 必须使用命令行交换卷名称。此功能当前在 Solaris Volume Manager GUI 中不可用。但是，可以使用命令行或 GUI 重命名卷。

---

要重命名卷时，请考虑以下原则：

- 您不能重命名当前正在使用的卷。此限制包括用作挂载的文件系统、`swap` 或用作应用程序或数据库的活动存储区的卷。因此，在使用 `metarename` 命令之前，停止对要重命名的卷的所有访问。例如，取消挂载已挂载的文件系统。
- 不能交换处于故障状态的卷。
- 不能交换使用热备件替换的卷。
- 交换只能发生在具有直接父子关系的卷之间。
- 不能交换（或重命名）日志设备。解决方法是分离日志设备，并附加另一个要求的名称的日志设备。
- 只能交换卷。不能交换分片或热备件。

## ▼ 如何重命名卷

开始之前 检查卷名称要求（第 38 页中的“卷名称”），和第 201 页中的“重命名卷的背景信息”。

### 1 取消挂载使用此卷的文件系统。

```
# umount /filesystem
```

### 2 要重命名卷，请使用下列方法之一：

- 从 Solaris Management Console 中增强的存储工具打开卷。选择要重命名的卷。在图标上单击鼠标右键。选择 "Properties"（属性）选项。然后，按照屏幕说明进行操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metarename` 命令：

```
# metarename old-volume-name new-volume-name
```

`old-volume-name` 指定现有卷的名称。

`new-volume-name` 指定现有卷的新名称。

有关更多信息，请参见 [metarename\(1M\)](#) 手册页。

### 3 如果需要，编辑 `/etc/vfstab` 文件以指向新的卷名称。

### 4 重新挂载文件系统。

```
# mount /filesystem
```

## 示例 20-3 重命名用于文件系统的卷

在以下示例中，卷 `d10` 已重命名为 `d100`。

```
# umount /home
# metarename d10 d100
d10: has been renamed to d100
      (Edit the /etc/vfstab file so that the file system references the new volume)
# mount /home
```

因为 `d10` 包含一个挂载的文件系统，该文件系统必须取消挂载才可以重命名卷。如果卷用于在 `/etc/vfstab` 文件中具有条目的文件系统，该条目必须更改为引用新的卷名称。

例如，如果 `/etc/vfstab` file 包含以下用于文件系统的项：

```
/dev/md/dsk/d10 /dev/md/rdisk/d10 /docs home 2 yes -
```

将此项更改为如下内容：

```
/dev/md/dsk/d100 /dev/md/rdisk/d100 /docs home 2 yes -
```

然后，重新挂载文件系统。

如果您有现有的镜像或事务卷，则可以使用 `metarename -x` 命令删除镜像或事务卷，并将数据保留在底层卷中。对于事务卷，只要主设备是卷（RAID-0、RAID-1 或 RAID-5 卷），就可以将数据保留在该卷上。

## 使用配置文件

Solaris Volume Manager 配置文件包含基本的 Solaris Volume Manager 信息以及重新构建配置所需的大部分数据。以下过程介绍如何使用这些文件。

### ▼ 如何创建配置文件

- 一旦定义了用于 Solaris Volume Manager 环境的所有合适的参数，请使用 `metastat -p` 命令创建 `/etc/lvm/md.tab` 文件。

```
# metastat -p > /etc/lvm/md.tab
```

该文件包含 `metainit` 命令和 `metabs` 命令使用的所有参数。如果需要设置多个类似的环境或需要在系统故障后重新创建配置，请使用该文件。

有关 `md.tab` 文件的更多信息，请参见第 280 页中的“[md.tab 文件概述](#)”和 `md.tab(4)` 手册页。

### ▼ 如何根据配置文件初始化 Solaris Volume Manager



注意 – 在以下情况中使用此过程：

- 如果您已经完全丢失了 Solaris Volume Manager 配置
- 如果尚未建立配置，且希望从保存的配置文件创建配置

有时，您的系统会丢失在状态数据库中维护的信息。例如，如果在删除所有状态数据库副本后重新引导系统，则会发生这样的丢失。只要在丢失状态数据库后未创建任何卷，就可以使用 `md.cf` 或 `md.tab` 文件恢复 Solaris Volume Manager 配置。

注 – `md.cf` 文件不会维护活动的热备件上的信息。因此，如果丢失 Solaris Volume Manager 配置时热备件正在使用中，那些正在使用活动的热备件的卷可能会损坏。

有关这些文件的更多信息，请参见 [md.cf\(4\)](#) 和 [md.tab\(4\)](#) 手册页。

## 1 创建状态数据库副本。

有关更多信息，请参见第 64 页中的“创建状态数据库副本”。

## 2 创建或更新 `/etc/lvm/md.tab` 文件。

- 如果尝试恢复上一个已知的 Solaris Volume Manager 配置，请将 `md.cf` 文件复制到 `/etc/lvm/md.tab` 文件。
- 如果基于保留的 `md.tab` 文件的副本创建新的 Solaris Volume Manager 配置，请将保留的文件复制到 `/etc/lvm/md.tab` 文件。

## 3 编辑“新的”`/etc/lvm/md.tab` 文件并执行以下操作：

- 如果在崩溃后创建新的配置或恢复配置，请将镜像配置为单向镜像。例如：

```
d80 -m d81 1
d81 1 1 clt6d0s3
```

如果镜像的子镜像大小不相同，请确保将最小的子镜像用于此单向镜像。否则，数据可能会丢失。

- 如果正在恢复现有的配置并且已正常停止了 Solaris Volume Manager，将镜像配置为多向镜像。例如：

```
d70 -m d71 d72 1
d71 1 1 clt6d0s2
d72 1 1 clt5d0s0
```

- 使用 `-k` 选项指定 RAID-5 以防止设备重新初始化。例如：

```
d45 -r clt3d0s5 clt3d0s3 clt3d0s4 -k -i 32b
```

有关更多信息，请参见 [metainit\(1M\)](#) 手册页。

## 4 使用 `metainit` 命令的以下格式之一，在不提交更改的情况下检查 `/etc/lvm/md.tab` 文件条目的语法。

```
# metainit -n md.tab-entry
```

```
# metainit -n -a
```

`metainit` 命令不会维护使用 `-n` 选项运行时创建的设备的假定状态，因此在使用 `-n` 选项创建卷时，如果这些卷依赖于其他不存在的卷，将会导致错误，但在不使用 `-n` 选项的情况下，命令有可能成功。

`-n`                      指定不实际创建设备。使用该选项验证结果是否和预期一样。

`md.tab-entry`          指定要初始化的组件的名称。

`-a`                      指定以检查所有组件。

- 5 如果前面的步骤中没有明显的问题，请从 `md.tab` 文件重新创建卷和热备用池：

```
# metainit -a  
-a 指定以激活 /etc/lvm/md.tab file 中的条目。
```

- 6 如果需要，使用 `metattach` 命令将单向镜像变为多向镜像。

```
# mettach mirror submirror
```

- 7 验证卷上的数据，确认已正确地重新构建了配置。

```
# metastat
```

## 更改 Solaris Volume Manager 缺省值

在 Solaris 10 发行版中，Solaris Volume Manager 已经增强，可以动态配置卷。您无需编辑 `/kernel/drv/md.conf` 文件中的 `nmd` 和 `md_nsets` 参数。新卷会根据需要动态创建。

最大 Solaris Volume Manager 配置值保持不变：

- 支持的最大卷数为 8192。
- 支持的最大磁盘集数为 32。

## 使用 growfs 命令扩展文件系统

扩展包含 UFS 文件系统的卷（意味着添加了更多的空间）后，还需要扩展文件系统以便识别添加的空间。必须使用 `growfs` 命令手动扩展文件系统。`growfs` 命令可以扩展文件系统，即使文件系统已挂载。但是，`growfs` 命令运行时，无法对文件系统进行写入访问。

使用原始设备的应用程序（如数据库）必须具有自己的方法来纳入添加的空间。Solaris Volume Manager 不提供该功能。

`growfs` 命令在扩展文件系统时会对挂载的文件系统进行“写入锁定”。可以通过分阶段扩展文件系统来缩短文件系统的写入锁定时间。例如，要将 1 GB 的文件系统扩展为 2 GB，可以使用 `-s` 选项以 16 MB 大小的阶段来扩展该文件系统。此选项指定新文件系统在每个阶段的总大小。

扩展期间，由于使用了写入锁定功能，文件系统无法接受写入访问。写入访问透明地暂停，并在 `growfs` 命令解锁文件系统时重新启动。读取访问不受影响。但是，在锁定生效期间不会保留访问时间。

## 有关扩展分片和卷的背景信息

注 – Solaris Volume Manager 卷可以扩展。但是，卷的大小不能缩小。

- 无论用于文件系统、应用程序或数据库，卷均可扩展。可以扩展 RAID-0（条带和串联）卷、RAID-1（镜像）卷、RAID-5 卷和软分区。
- 可以串联一个包含正在使用的现有文件系统的卷。只要文件系统是 UFS 文件系统，就可以扩展该文件系统（使用 `growfs` 命令）以占据更大的空间。可以在不中断数据读取访问的情况下扩展该文件系统。
- 由于 UFS 文件系统的限制，一旦扩展了一个文件系统，其大小就不能缩小。
- 使用原始设备的应用程序和数据库必须自身具备扩展增加空间的方法，以便它们可识别这些空间。Solaris Volume Manager 不提供此功能。
- 组件添加到 RAID-5 卷时，即成为该卷的串联。新的组件不包含奇偶校验信息。但是，新组件上的数据受到对卷执行的整体奇偶校验计算的保护。
- 可以通过添加其他组件扩展日志设备。无需运行 `growfs` 命令，因为 Solaris Volume Manager 会在重新引导时自动识别增加的空间。
- 可以通过从底层卷或分片添加空间来扩展软分区。可以通过添加分片来扩展所有其他卷。

## ▼ 如何扩展文件系统

开始之前 检查第 41 页中的“创建 Solaris Volume Manager 组件的先决条件”。

### 1 查看与文件系统相关的磁盘空间。

```
# df -hk
```

有关更多信息，请参见 [df\(1M\)](#) 手册页。

### 2 在逻辑卷上扩展 UFS 文件系统。

```
# growfs -M /mount-point /dev/md/rdisk/volume-name
```

`-M/mount-point` 为要扩展的文件系统指定挂载点。

`/dev/md/rdisk/volume-name` 指定要扩展到其上的卷的名称。

有关更多信息，请参见 [growfs\(1M\)](#) 手册页。

## 示例 20-4 扩展文件系统

在以下示例中，新的分片已添加到卷 `d10`，该卷包含挂载的文件系统 `/home2`。growfs 命令使用 `-M` 选项指定挂载点为 `/home2`，该文件系统已扩展到原始卷

/dev/md/rdsk/d10。growfs 命令完成时，文件系统将跨整个卷。在扩展文件系统之前和之后，可以使用 df -hk 命令验证总的磁盘容量。

```
# df -hk
Filesystem            kbytes    used    avail capacity  Mounted on
...
/dev/md/dsk/d10       69047    65426         0   100%    /home2
...
# growfs -M /home2 /dev/md/rdsk/d10
/dev/md/rdsk/d10:      295200 sectors in 240 cylinders of 15 tracks, 82 sectors
      144.1MB in 15 cyl groups (16 c/g, 9.61MB/g, 4608 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
   32, 19808, 39584, 59360, 79136, 98912, 118688, 138464, 158240, 178016, 197792,
  217568, 237344, 257120, 276896,
# df -hk
Filesystem            kbytes    used    avail capacity  Mounted on
...
/dev/md/dsk/d10       138703    65426    59407     53%    /home2
...
```

对于镜像卷，始终在顶层卷上运行 growfs 命令。不要在子镜像或主设备上运行命令，即使空间添加到子镜像或主设备也是如此。

## 替换和启用 RAID-1 和 RAID-5 卷中的组件概述

Solaris Volume Manager 可以替换并启用 RAID-1（镜像）和 RAID-5 卷中的组件。

在 Solaris Volume Manager 术语中，替换组件是一种使用系统上可用的组件来替换子镜像或 RAID-5 卷中的选定组件的方法。可以将此过程认为是逻辑替换，而不是物理替换组件。有关更多信息，请参见第 209 页中的“将一个组件替换为另一个可用组件”。

启用组件是指“激活”组件，或组件以自身进行替换（即组件名称不变）。有关更多信息，请参见第 208 页中的“启用组件”。

注 – 从磁盘错误中恢复时，扫描 /var/adm/messages 可查看出现了何种错误。如果错误是暂时的，且磁盘本身没有错误，尝试启用故障组件。也可以使用 format 命令测试磁盘。

## 启用组件

存在以下任何情况时即可启用组件：

- Solaris Volume Manager 无法访问物理驱动器。出现此问题的原因可能是断电或驱动器电缆松动。在这种情况下，Solaris Volume Manager 将组件置于 "Maintenance"（维护）状态。您需要确保驱动器可以访问（恢复供电、重新连接电缆等），然后启用卷中的组件。
- 您怀疑物理驱动器出现了和磁盘不相关的暂时性问题。您可以只通过启用组件来修复处于 "Maintenance"（维护）状态的组件。如果启用组件没有修复该问题，则需要执行以下操作之一：
  - 物理替换该磁盘驱动器并启用组件
  - 使用系统上其他可用的组件替换该组件

物理替换磁盘时，务必要与被替换的磁盘同样地对该磁盘进行分区，确保在每一个使用的组件上均有足够的空间。

---

注 – 始终检查被替换的磁盘上的状态数据库副本和热备件。替换磁盘前，应当删除处于出错状态的所有状态数据库副本。然后，在启用组件后，使用相同的大小重新创建状态数据库副本。您应当以同样方式来处理热备件。

---

## 将一个组件替换为另一个可用组件

要将现有的组件替换或切换为系统上可用但尚未使用的不同组件，可以使用 `metareplace` 命令。

存在以下任何情况时即可使用该命令：

- 磁盘驱动器出现问题，并且没有替换驱动器。但是，在系统中其他位置确实有可用的组件。  
当替换绝对必要，但又不想关闭系统时，可能需要使用该策略。
- 在物理磁盘上发现软错误。  
物理磁盘可能会报告软错误，即使 Solaris Volume Manager 显示镜像/子镜像或 RAID-5 卷处于 "Okay"（正常）状态。通过将存在问题的组件替换为另一个可用的组件，您可以执行预防维护，还可能防止产生不可更正的错误。
- 您想进行性能调优。

有一种评估组件的方法是使用 Solaris Management Console 中增强的存储工具提供的性能监视功能。例如，您可以看到 RAID-5 卷中某一特定组件的平均负载很高，即使它处于 "Okay"（正常）状态。要平衡卷上的负载，您可以将该组件替换为较少使用的磁盘中的组件。您可以在不中断对卷提供服务的情况下联机执行此类替换。

## "Maintenance"（维护）和 "Last Erred"（最近出错）状态

RAID-1 或 RAID-5 卷中的组件出现错误时，Solaris Volume Manager 会将组件置于 "Maintenance"（维护）状态。对于处于 "Maintenance"（维护）状态的组件，不执行进一步的读取或写入。

有时组件会进入 "Last Erred"（最近出错）状态。对于 RAID-1 卷，这种情况通常伴随单向镜像出现。卷出现错误。但是，没有冗余组件可进行读取。对于 RAID-5 卷，这种情况在一个组件进入 "Maintenance"（维护）状态，而另一个组件又发生故障后出现。发生故障的第二个组件将进入 "Last Erred"（最近出错）状态。

当 RAID-1 卷或 RAID-5 卷有某一组件处于 "Last Erred"（最近出错）状态时，仍会尝试对标记为 "Last Erred"（最近出错）的组件执行 I/O 操作。进行这种 I/O 尝试是因为 "Last Erred"（最近出错）状态的组件包含最近一次的正确数据副本（从 Solaris Volume Manager 的角度来看）。对于处于 "Last Erred"（最近出错）状态的组件，卷的行为类似于正常的设备（磁盘），会向应用程序返回 I/O 错误。通常，此时已丢失一些数据。

相同卷中其他组件上的后续错误会以不同方式进行处理，具体取决于卷的类型。

### RAID-1 Volume（RAID-1 卷）

RAID-1 卷可能容许多个组件处于 "Maintenance"（维护）状态，且仍然可以进行读取和写入。如果组件处于 "Maintenance"（维护）状态，则未丢失任何数据。您可以按任何顺序安全地替换或启用这些组件。如果组件处于 "Last Erred"（最近出错）状态，则无法替换它，除非您先替换那些处于 "Maintenance"（维护）状态的组件。替换或启用处于 "Last Erred"（最近出错）状态的组件通常意味着某些数据已丢失。请确保在修复后验证镜像上的数据。

### RAID-5 Volume（RAID-5 卷）

RAID-5 卷可以容许单个组件处于 "Maintenance"（维护）状态。您可以在不丢失数据的情况下安全地替换处于 "Maintenance"（维护）状态的单个组件。如果在其他组件上出现错误，该组件将置于 "Last Erred"（最近出错）状态。此时，RAID-5 卷为只读设备。您需要执行某些类型的错误恢复，以便稳定 RAID-5 卷的状态和减少数据丢失的可能性。如果一个 RAID-5 卷到达 "Last Erred"（最近出错）状态，则它很可能已丢失数据。请确保在修复后验证 RAID-5 卷上的数据。

始终先替换处于 "Maintenance"（维护）状态的组件，然后再替换处于 "Last Erred"（最近出错）状态的组件。替换和重新同步组件后，使用 `metastat` 命令验证其状态。然后，验证数据。

## 替换和启用 RAID-1 和 RAID-5 卷中组件的背景信息

替换 RAID-1 卷或 RAID-5 卷中的组件时，请遵循以下这些原则：

- 始终先替换处于 "Maintenance"（维护）状态的组件，然后再替换处于 "Last Erred"（最近出错）状态的组件。
- 替换和重新同步组件后，使用 `metastat` 命令验证该卷的状态。然后，验证数据。替换或启用处于 "Last Erred"（最近出错）状态的组件通常意味着某些数据已丢失。请确保在修复后验证卷上的数据。对于 UFS，运行 `fsck` 命令验证“元数据”（文件系统的结构）。然后，检查实际的用户数据。（实际上，用户必须检查他们的文件。）数据库或其他应用程序必须有自己的方法来验证其内部数据结构。
- 替换组件时始终检查状态数据库副本和热备件。替换物理磁盘前，应当删除处于出错状态的所有状态数据库副本。启用组件前应当添加回状态数据库副本。相同过程也适用于热备件。
- 在替换 RAID-5 卷组件的期间，可使用两种方法中的任一种恢复数据。或者从当前正在使用的热备件恢复数据，或者使用 RAID-5 奇偶校验恢复数据（没有热备件处于使用中时）。
- 替换 RAID-1 卷的组件时，Solaris Volume Manager 会自动启动新组件与其他卷的重新同步。重新同步完成后，替换的组件即可读取且可写入。如果使用来自热备件的数据替换出现故障的组件，该热备件将置于 "Available"（可用）状态，并可用于其他热备件替换。
- 新的组件必须足够大，才能替换旧的组件。
- 作为预防措施，在替换 "Last Erred"（最近出错）设备之前，请备份所有数据。



## Solaris Volume Manager 的最佳做法

---

本章通过使用 Solaris Volume Manager 的实际存储方案提供一般的最佳做法信息。在本章中，将给出典型配置，然后进行分析，并提供满足相同需求的建议（“最佳做法”）配置。

本章包括以下信息：

- 第 213 页中的“部署小型服务器”
- 第 215 页中的“将 Solaris Volume Manager 与网络存储设备结合使用”

### 部署小型服务器

分布式计算环境通常需要在多个位置部署类似或相同的服务器。这些环境包括 ISP、地理位置分散的销售机构和电信服务提供商。分布式计算环境中的服务器可能提供以下某些服务：

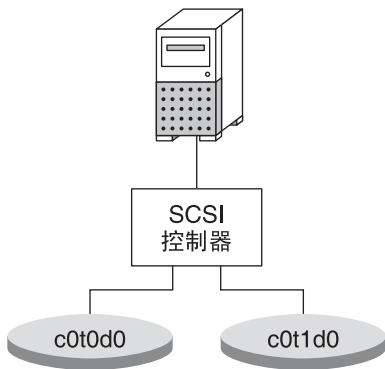
- 路由器或防火墙服务
- 电子邮件服务
- DNS 高速缓存
- Usenet（网络新闻）服务器
- DHCP 服务
- 在不同位置以最佳方式提供的其他服务

这些小型服务器具有几个共同特征：

- 高可靠性要求
- 高可用性要求
- 常规硬件和性能要求

在开始时，请考虑具有单个 SCSI 总线和两个内部磁盘的 Netra 服务器。此现成的配置是分布式服务器的良好起点。使用 Solaris Volume Manager 可以轻松地镜像某些或所有分片，从而提供冗余存储，帮助防止磁盘故障。有关此小型系统配置的示例，请参见下图。

图 21-1 小型系统配置



此配置可能包括根 (/)、/usr、swap、/var 和 /export 文件系统以及状态数据库副本（每个磁盘有一个副本）的镜像。因此，任何镜像的任一端出现故障都不一定会导致系统故障。而且，最多可允许出现五个独立故障。但是，这不能充分保护系统免受磁盘或分片故障的影响。各种潜在故障均可能导致整个系统出现故障，因而需要操作员进行干预。

尽管此配置的确提供了防止灾难性磁盘故障的某些保护机制，但是仍有可能出现重要的单点故障：

- 单个 SCSI 控制器成为一个潜在的故障点。如果控制器出现故障，整个系统会停止运行，必须更换部件。
- 两个磁盘不能提供状态数据库副本的足够分布水平。多数一致算法要求有一半数量的状态数据库副本可用时，系统才能继续运行。此算法还要求有一半数量再加一个的状态数据库副本可用时，才能重新引导。因此，如果每个磁盘上有一个状态数据库副本，且一个磁盘或包含该副本的分片出现故障，则系统无法重新引导。因此，镜像的根 (/) 文件系统将处于无效状态。如果每个磁盘上有两个或多个状态数据库副本，则单个分片出现故障可能就不会成为问题。但是，磁盘故障仍可能会妨碍重新引导。如果每个磁盘上的副本数不同，一个磁盘可能具有一半以上的副本，另一个磁盘可能具有一半以下的副本。如果具有较少副本的磁盘出现故障，则系统可以重新引导并继续运行。但是，如果具有较多副本的磁盘出现故障，则系统可能会立即出现紧急情况。

“最佳做法”是再添加一个控制器和一个硬盘驱动器来修改配置。最终的配置将更具弹性。

## 将 Solaris Volume Manager 与网络存储设备结合使用

Solaris Volume Manager 可以与网络存储设备良好配合，尤其是那些提供可配置的 RAID 级别和灵活选项的设备。通常，将 Solaris Volume Manager 与此类设备结合使用所获得的性能和灵活性优于任一单个产品。

一般情况下，请不要在提供冗余（例如 RAID-1 和 RAID-5 卷）的任何硬件存储设备上创建 Solaris Volume Manager 的 RAID-5 卷。除非有非常特殊的情况，否则性能将受到影响。而且，在冗余性或更高可用性方面受益很小。

另一方面，对底层硬件存储设备配置 RAID-5 卷非常有效。这样可以为 Solaris Volume Manager 卷提供良好的基础。硬件 RAID-5 可为 Solaris Volume Manager 的 RAID-1 卷、软分区或其他卷提供额外的冗余性。

---

注 - 请勿配置同类的软件和硬件设备。例如，请不要在硬件 RAID-1 设备之上构建软件 RAID-1 卷。在硬件和软件上配置同类的设备会降低性能，但不影响可靠性。

---

Solaris Volume Manager 中在底层硬件存储设备之上构建的 RAID-1 卷不是 RAID-1+0。因为 Solaris Volume Manager 不能完全识别底层存储设备，从而不能提供 RAID-1+0 功能。

在硬件 RAID-5 设备上构建 Solaris Volume Manager RAID-1 卷，然后在 RAID-1 卷上再配置软分区是一种非常灵活且很有弹性的配置。



## 自上而下创建卷（概述）

---

本章提供了有关在 Solaris Volume Manager 中自上而下创建卷的概念性信息。

本章包含以下信息：

- 第 217 页中的“自上而下创建卷概述”
- 第 218 页中的“使用磁盘集实现自上而下创建卷”
- 第 218 页中的“自上而下的卷创建过程”
- 第 220 页中的“确定可用于自上而下创建卷的磁盘”

有关执行相关任务的信息，请参见第 23 章，自上而下创建卷（任务）。

### 自上而下创建卷概述

通过自上而下创建卷，您可以使用 `metassist` 命令自动创建 Solaris Volume Manager 卷配置。您不再需要手动完成对磁盘分区、创建 RAID-0 卷（作为子镜像）、创建热备用池和热备件以及最终创建镜像的全过程，而是可以执行 `metassist` 命令来创建卷。Solaris Volume Manager 会为您执行其余操作。

通过 `metassist` 命令，您只需使用一个命令即可创建 Solaris Volume Manager 卷配置。您可以在**服务质量**方面指定卷的特征。指定服务质量特征意味着，您无需指定要在卷中使用的硬件组件，而是可以利用 `metassist` 命令的输入来提供以下信息：

- 卷大小
- 冗余级别（即，数据副本数）
- 卷的数据路径数
- 故障恢复（用于指示该卷是否与热备用池相关联）

您可以通过命令行选项或在命令行上指定的输入文件来指定卷的服务质量。

在某些情况下，需要更具体地定义卷的特征或在创建卷时应满足的约束条件。此时，您还可以指定以下特征：

- 卷类型，例如，RAID-0（串联）卷或 RAID-0（条带）卷。

- 要在特定卷中使用的组件。
- 可用或不可用的组件。
- 要使用的组件的数量。
- 要创建的卷类型特有的详细信息。这些详细信息包括条带、镜像的读取策略以及相似特征。

如果您希望更详细地指定卷的名称、大小和组件，请使用输入文件。输入文件包括卷请求文件和卷规范文件。有关如何使用输入文件的更多信息，请参见第 218 页中的“自上而下的卷创建过程”。

最后，可以对 `metassist` 命令加以限制，以便使用（或不使用）特定磁盘或路径。

## 使用磁盘集实现自上而下创建卷

`metassist` 命令使用 Solaris Volume Manager 磁盘集来管理可用于自上而下创建卷的卷和可用磁盘。对于任何给定的自上而下创建卷过程，用作生成块的所有磁盘都必须位于磁盘集中，或者可添加到磁盘集中。您可以通过自上而下创建过程在不同磁盘集中创建卷。但是，可用的磁盘和组件会受到磁盘集功能的约束。

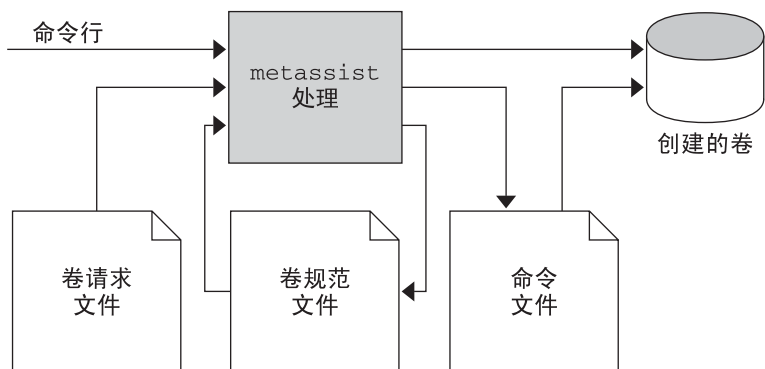
## 自上而下的卷创建过程

自上而下创建过程通过以下过程提高了灵活性：

- 您可以通过一个完全自动的端对端过程来指定所需的约束条件，并在命令完成时创建所需的卷。
- 您可以通过一个更精确的过程利用断点写入基于 XML 的文件。

下图显示了 `metassist` 命令如何基于命令行输入和输入文件来支持端对端处理。图中还显示了 `metassist` 命令如何支持部分处理，通过此处理方式可提供基于文件的数据或检查卷的特征。

图 22-1 自上而下创建卷的处理选项



要在无人工参与的情况下自动创建卷，请使用命令行指定所需的服务质量特征。`metassist` 命令会自动为您创建所请求的卷。例如：

```
# metassist create -s storagepool -S 10Gb
```

此命令可在 `storagepool` 磁盘集中创建一个大小为 10 GB 的条带卷。此命令会使用 `storagepool` 磁盘集中的可用存储。

或者，您也可以使用 **卷请求文件** 定义卷的特征。然后，可以使用 `metassist -F request-file` 命令创建具有这些特征的卷。

您可以使用 `metassist -d` 命令生成卷规范文件。可以使用此文件评估预期实现情况，并在需要时编辑此文件。然后，可以将此卷规范文件用作 `metassist` 命令的输入来创建卷。

最后，可以使用 `metassist -c` 命令创建命令文件。**命令文件**是指一个 shell 脚本，该脚本用于实现由 `metassist` 命令指定的 Solaris Volume Manager 设备配置。您可以使用此文件来重复创建卷，并根据需要对此文件进行编辑。

在使用 `metassist` 命令创建这些文件时，您可以了解到 `metassist` 命令会执行什么功能以及如何决策。在对以下问题进行故障排除时，该信息可能会很有用：

- 为什么卷是以某种特定方式创建的
- 为什么没有创建卷
- `metassist` 命令会创建什么卷，而实际上却未创建

## 确定可用于自上而下创建卷的磁盘

`metassist` 命令会检查磁盘以确定哪些磁盘未被使用。此命令会尝试保守地确定哪些磁盘可以使用。所有正在使用的磁盘或分片都不能由 `metassist` 命令使用。`metassist` 命令将检查以下内容：

- 在其他磁盘集中使用的磁盘
- 已挂载的分片
- 包含文件系统超级块的分片，用于指示可挂载的文件系统
- 在其他 Solaris Volume Manager 卷中使用的分片

满足以上条件之一的任何分片都不能用于自上而下创建卷。

# 自上而下创建卷（任务）

本章提供与使用 `metassist` 命令自上而下创建 Solaris Volume Manager 卷相关联的任务。

以下是本章中信息的列表：

- 第 221 页中的“自上而下创建卷（任务列表）”
- 第 222 页中的“自上而下创建卷的先决条件”
- 第 222 页中的“自动创建卷”
- 第 226 页中的“使用 `metassist` 命令处理基于文件的数据”
- 第 232 页中的“更改 `metassist` 命令的缺省行为”

有关自上而下创建卷的概念性信息，请参见第 22 章，自上而下创建卷（概述）。

## 自上而下创建卷（任务列表）

以下任务列表列出了使用 `metassist` 命令创建 Solaris Volume Manager 卷所需的过程。使用该命令可以基于服务质量特征指定卷以及通过单一命令创建多组分层卷。

任务	说明	参考
自动创建卷	允许您使用 <code>metassist</code> 命令创建一个或多个 Solaris Volume Manager 卷。  此外，还允许您控制 <code>metassist</code> 命令为故障排除和问题诊断而提供的关于卷创建过程的信息的数量。	第 222 页中的“自动创建卷”  第 223 页中的“通过指定输出的详细级别来分析卷的创建过程”
创建命令文件	帮助您使用 <code>metassist</code> 命令创建 <code>shell</code> 脚本以生成该命令所指定的卷。	第 230 页中的“使用 <code>metassist</code> 命令创建卷配置文件”

任务	说明	参考
使用 shell 脚本创建卷	说明如何通过 <code>metassist</code> 命令以前生成的 shell 脚本创建该命令所指定的 Solaris Volume Manager 卷。	第 229 页中的“使用保存的 shell 脚本（由 <code>metassist</code> 命令创建）创建卷”
创建卷配置文件	帮助您创建卷配置文件，描述要创建的卷的特征。	第 230 页中的“使用 <code>metassist</code> 命令创建卷配置文件”
更改卷缺省文件	允许您设置缺省卷特征以定制 <code>metassist</code> 命令的行为。	第 232 页中的“更改卷缺省文件”

## 自上而下创建卷的先决条件

如果要使用 `metassist` 命令自动创建卷和卷配置，则要求 Solaris Volume Manager 配置能够正常运行。在开始之前，应具备以下各项：

- 超级用户访问权限，或能够承担等效的基于角色的访问控制 (Role-Based Access Control, RBAC) 角色。有关更多信息，请参见《Oracle Solaris 管理：基本管理》中的“成为超级用户 (root) 或承担角色”。
- 状态数据库副本，相应地分布于您的系统中。有关状态数据库副本的更多信息，请参见第 57 页中的“关于 Solaris Volume Manager 状态数据库和副本”。
- 用于创建卷的可用磁盘。`metassist` 命令使用磁盘集来帮助管理存储。必须具有完全未使用的磁盘（或现有的磁盘集），才能使用 `metassist` 命令创建新卷。有关磁盘可用性的更多信息，请参见第 220 页中的“确定可用于自上而下创建卷的磁盘”。

除了这些最低要求以外，还不能在 `/etc/inetd.conf` 文件中禁用 Solaris Volume Manager RPC 守护进程（`rpc.metad`、`rpc.metamhd` 和 `rpc.metamedd`）。已将这些守护进程配置为在缺省情况下启动。它们必须保持启用状态，以便允许 Solaris Volume Manager 使用共享磁盘集。

## 自动创建卷

使用 `metassist` 命令可以基于服务质量条件创建 Solaris Volume Manager 卷以及多组卷。仅使用 `metassist` 命令就可以创建卷，而不像以前，Solaris Volume Manager 需要使用一系列命令来创建卷。

可以使用 `metassist` 命令直接创建 RAID-1（镜像）卷。因此，不必先创建用作 RAID-1（镜像）卷的组件的子镜像（串联或条带）。

## 通过指定输出的详细级别来分析卷的创建过程

在运行 `metassist` 命令时，可以指定输出的详细级别。更详细的输出有助于诊断问题，比如确定磁盘为什么被（未被）选中用于卷中，或确定在某次尝试执行命令时为什么会失败。不太详细的输出可减少必须查看的无关信息的量。

当指定输出详细级别时，可以了解 `metassist` 命令执行哪些操作以及该命令如何做出决定。在对以下问题进行故障排除时，该信息会很有用：

- 为什么卷是以某种特定方式创建的
- 为什么没有创建卷
- `metassist` 命令会创建什么卷，而实际上却未创建

## ▼ 如何使用 `metassist` 命令创建 RAID-1（镜像）卷

开始之前 查看第 222 页中的“自上而下创建卷的先决条件”。

### 1 确定要在其上创建卷的可用存储。

如果未显式指定存储，则 Solaris Volume Manager 会识别系统上未使用的存储并根据需要使用存储。如果选择指定存储，无论是广泛地指定（例如，控制器 1 上的所有存储）还是明确地指定（例如，使用 `c1t4d2`，但不使用 `c1t4d1`），Solaris Volume Manager 都会使用指定的存储。

### 2 使用 `metassist` 命令和适用于您的任务的相应选项。

- 要从命令行创建卷，请使用以下格式的 `metassist` 命令。

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... -S size -v verbosity
```

<code>create</code>	是用于创建卷的子命令。
<code>-s diskset-name</code>	指定要用于卷的磁盘集的名称。
<code>-f</code>	指定该卷与热备件相关联。
<code>-r redundancy</code>	指定要创建的冗余级别（数据副本的数量）。
<code>-a device1, device2...</code>	指定可用于创建卷的设备。
<code>-S size</code>	以 KB（千字节）、MB（兆字节）、GB（千兆字节）或 TB（兆兆字节）为单位，指定要创建的卷的大小。
<code>-v verbosity</code>	指定输出的详细级别。允许的值介于 0（几乎无提示的输出）和 2（大量输出）之间。缺省级别为 1（中等输出）。

- 要使用输入文件创建卷以指定卷特征，请使用以下任意一种格式的 `metassist` 命令。

```
# metassist create [-v n] [-c] -F config_file
# metassist create [-v n] [-c | -d] -F request_file
```

- c 指定输出将实现指定的或所生成的卷配置的命令脚本。该命令脚本将不运行，而且处理将停止在该阶段。
- d 指定输出满足指定的或所生成的卷请求的卷配置。不会生成或执行任何命令脚本，而且处理将停止在该阶段。
- F *config\_file* | *request\_file* 指定要处理的卷请求文件或卷配置文件。如果将 *config\_file* 或 *request\_file* 指定为短划线 (-)，则会从标准输入中读取该文件。输入文件为卷配置文件时无法指定 -d 选项。  
  
卷配置文件介绍了要创建的卷的详细配置，而卷请求文件则提供了要生成的卷的特征。有关更多信息，请参见 [volume-config\(4\)](#)、[volume-config\(4\)](#) 和 [volume-request\(4\)](#) 手册页。
- v *verbosity* 指定输出的详细级别。允许的值介于 0（几乎无提示的输出）和 2（大量输出）之间。缺省级别为 1（中等输出）。

有关更多信息，请参见以下示例和 [metassist\(1M\)](#) 手册页。

### 3 创建卷后，可查看新卷。

```
# metastat -s diskset-name
```

#### 示例 23-1 使用 metassist 命令创建双向镜像

以下示例说明如何创建 10 MB 大小的双向镜像。metassist 命令可识别未使用的磁盘，并使用这些磁盘创建尽可能最佳的镜像。-s myset 参数指定在 myset 磁盘集中创建卷。如有必要，将创建该磁盘集。

```
# metassist create -s myset -r 2 -S 10mb
```

#### 示例 23-2 使用 metassist 命令创建双向镜像和热备件

以下示例说明如何使用 metassist 命令来创建大小为 10 MB 且具有可提供额外容错功能的热备件的双向镜像。-f 选项指定容错功能。

```
# metassist create -s myset -f -r 2 -S 10mb
```

#### 示例 23-3 使用 metassist 命令通过特定的控制器创建条带

以下示例说明如何使用 metassist 命令通过控制器 1 上的可用磁盘来创建条带。-a 选项指定可用的控制器。

```
# metassist create -s myset -a c1 -S 10mb
```

### 示例 23-4 指定 metassist 命令的输出详细级别

以下示例说明如何使用 `metassist` 命令来创建大小为 10 MB 且具有可提供额外容错功能的热备件的双向镜像。-f 选项指定容错功能。最后一个参数 (-v 2) 指定详细级别为 2，这是最大级别，将尽可能多地提供关于 `metassist` 命令如何运行的信息。

```
# metassist create -s myset -f -r 2 -S 10mb -v 2
Scanning system physical device configuration...

These HBA/Controllers are known:..
c0                                /pci@1f,0/pci@1,1/ide@3
c1                                /pci@1f,0/pci@1/pci@2/SUNW,isptwo@4

These disks are known:

c0t0d0                            id1,dad@AST34342A=_____VGD97101
c1t1d0                            id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L88P000021097XNL
c1t2d0                            id1,sd@SSEAGATE_ST39102LCSUN9.0GLJW22867000019171JDF
c1t3d0                            id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L7RV00007108TG0H
c1t4d0                            id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0LDFR000021087R1T
c1t5d0                            id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L0M200002109812L
c1t6d0                            id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L8K8000021087R0Z
.
.
.
(output truncated)
```

以下示例说明如何使用 `metassist` 命令来创建大小为 10 MB 且具有可提供额外容错功能的热备件的双向镜像。-f 选项指定容错功能。最后一个参数 (-v 0) 指定详细级别为 0，这是最小级别，在命令运行时将提供几乎无提示的输出。

```
# metassist create -s myset -f -r 2 -S 10mb -v 0
myset/hsp000: Hotspare pool is setup
myset/hsp000: Hotspare is added
myset/d2: Concat/Stripe is setup
myset/d1: Concat/Stripe is setup
myset/d0: Mirror is setup
myset/d0: submirror myset/d1 is attached
```

### 示例 23-5 使用输入文件创建卷

以下示例说明如何通过 `metassist` 命令使用输入文件来创建卷。

```
# metassist create -F request.xml
```

有关通过 `metassist` 命令使用输入文件的更多信息，请参见第 226 页中的“使用 `metassist` 命令处理基于文件的数据”。

## 使用 metassist 命令处理基于文件的数据

使用 metassist 命令可以创建用于评估卷特征或实际创建卷的文件。

### 使用 metassist 命令创建命令文件（shell 脚本）

通过运行带 `-c` 参数的 metassist 命令，可以生成一个 Bourne shell 脚本，其中包含将来创建卷配置的命令。通过该技术可以在实际创建卷之前检查命令，甚至可以根据具体需要稍微修改脚本。

### ▼ 如何使用 metassist 命令创建命令文件（shell 脚本）

开始之前 查看第 222 页中的“自上而下创建卷的先决条件”。

#### 1 确定要在其上创建卷的可用存储。

如果未显式指定存储，则 Solaris Volume Manager 会识别系统上未使用的存储并根据需要使用存储。如果选择指定存储，无论是广泛地指定（例如，控制器 1 上的所有存储）还是明确地指定（例如，使用 `c1t4d2`，但不使用 `c1t4d1`），Solaris Volume Manager 都会使用指定的存储。

#### 2 使用 metassist 命令和适用于您的任务的相应选项。

使用 `-c` 选项指定不应实际创建卷。

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... \
  -S size -v verbosity [-c]
```

<code>create</code>	是用于创建卷的子命令。
<code>-s diskset-name</code>	指定要用于卷的磁盘集的名称。
<code>-f</code>	指定该卷与热备件相关联。
<code>-r redundancy</code>	指定要创建的冗余级别（数据副本的数量）。
<code>-a device1, device2...</code>	指定可用于创建卷的设备。
<code>-S size</code>	以 KB（千字节）、MB（兆字节）、GB（千兆字节）或 TB（兆兆字节）为单位，指定要创建的卷的大小。
<code>-v verbosity</code>	指定输出的详细级别。允许的值介于 0（几乎无提示的输出）和 2（大量输出）之间。缺省级别为 1（中等输出）。
<code>-c</code>	指定不应实际创建卷，而应将可用来创建指定配置的 shell 脚本发送到标准输出中。

---

注 -c 参数所需的 shell 脚本将发送到标准输出中，而 metassist 命令的其余输出将转至标准错误中。您可以在进行选择时重定向输出流。

---

有关更多信息，请参见以下示例和 [metassist\(1M\)](#) 手册页。

### 示例 23-6 使用 metassist 命令创建命令文件（shell 脚本）

以下示例说明如何使用 metassist 命令来创建大小为 10 MB 且具有可提供额外容错功能的热备件的双向镜像。-f 选项指定容错功能。最后一个参数 (-c) 指定不应实际创建卷，而应将可用来创建指定配置的 shell 脚本发送到标准输出中。

```
# metassist create -s myset -f -r 2 -S 10mb -c
(output truncated)
.
.
.
Volume request completed successfully.
#!/bin/sh

#
# Environment
#

# Amend PATH
PATH="/usr/sbin:/usr/bin:$PATH"
export PATH

# Disk set name
diskset='myset'

#
# Functions
#

# Echo (verbose) and exec given command, exit on error
execho () {
    test -n "$verbose" && echo "$@"
    "$@" || exit
}

# Get full /dev/rdisk path of given slice
fullpath () {
    case "$1" in
        /dev/dsk/*|/dev/did/dsk/*) echo "$1" | sed 's/dsk/rdsk/' ;;
        /*) echo "$1" ;;
        *) echo /dev/rdsk/"$1" ;;
    esac
}

# Run fmfthard, ignore partboot error, error if output
```

```

fmthard_special () {
    ignore='Error writing partboot'
    out='fmthard "$@" 2>&1'
    result=$?
    echo "$out" |
    case "$out" in
        *"$ignore"*) grep -v "$ignore"; return 0 ;;
        '') return "$result" ;;
        *) cat; return 1 ;;
    esac >&2
}

#
# Main
#

# Verify root
if [ "$(id | sed 's/^[^]*(\[^\]*\).*\/1/'" != root ]
then
    echo "This script must be run as root." >&2
    exit 1;
fi

# Check for verbose option
case "$1" in
    -v) verbose=1 ;;
    *) verbose= ;;
esac

# Does the disk set exist?
if metaset -s "$diskset" >/dev/null 2>&1
then
    # Take control of disk set
    execho metaset -s "$diskset" -t
else
    # Create the disk set
    autotakeargs=
    /usr/sbin/clinfo || autotakeargs='-A enable'
    execho metaset -s "$diskset" $autotakeargs -a -h 'uname -n | cut -f1 -d.'
fi

# Format slices
execho fmthard_special -d 7:0:0:0:0 'fullpath clt3d0s7'
execho fmthard_special -d 7:0:0:0:0 'fullpath clt6d0s7'
execho fmthard_special -d 7:0:0:0:0 'fullpath clt4d0s7'

# Add disks to set
execho metaset -s "$diskset" -a clt3d0
execho metaset -s "$diskset" -a clt6d0
execho metaset -s "$diskset" -a clt4d0

# Format slices
execho fmthard_special -d 0:4:0:10773:17649765 'fullpath clt3d0s0'
execho fmthard_special -d 0:4:0:10773:17649765 'fullpath clt6d0s0'
execho fmthard_special -d 0:4:0:10773:17649765 'fullpath clt4d0s0'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath clt3d0s1'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath clt4d0s1'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath clt6d0s1'

```

```
# Does hsp000 exist?
metahs -s "$diskset" -i hsp000 >/dev/null 2>&1 || {
    # Create hsp hsp000
    execho metainit -s "$diskset" hsp000
}

# Add slices to hsp000
execho metahs -s "$diskset" -a hsp000 clt3d0s1

# Create concat d2
execho metainit -s "$diskset" d2 1 1 clt4d0s1

# Associate concat d2 with hot spare pool hsp000
execho metaparam -s "$diskset" -h hsp000 d2

# Create concat d1
execho metainit -s "$diskset" d1 1 1 clt6d0s1

# Associate concat d1 with hot spare pool hsp000
execho metaparam -s "$diskset" -h hsp000 d1

# Create mirror d0
execho metainit -s "$diskset" d0 -m d2 1
execho metattach -s "$diskset" d0 d1
#
```

### 示例 23-7 使用 metassist 命令保存命令文件（shell 脚本）

以下示例说明如何使用 metassist 命令来创建大小为 10 MB 且具有可提供额外容错功能的热备件的双向镜像。-f 选项指定容错功能。最后一个参数 (-c) 指定不应实际创建卷，而应将可用来创建指定配置的 shell 脚本发送到标准输出中。该命令的结尾将重定向到标准输出，以便创建 /tmp/metassist-shell-script.sh shell 脚本，该脚本稍后可用于创建指定的卷。

```
# metassist create -s myset -f -r 2 -S 10mb -c > \
/tmp/metassist-shell-script.sh
```

## 使用保存的 shell 脚本（由 metassist 命令创建）创建卷

使用 metassist 命令创建 shell 脚本后，可使用该脚本来创建在创建 shell 脚本时所指定的卷。



**注意** – 在创建脚本时，metassist 命令所创建的命令脚本在很大程度上依赖于在其上创建该脚本的系统的特定系统配置。在不同的系统上使用该脚本或者在更改系统配置之后使用该脚本可能会导致数据损坏或丢失。

## ▼ 如何执行保存的 metassist 命令 shell 脚本

开始之前 查看第 222 页中的“自上而下创建卷的先决条件”。

- 1 确保自创建 shell 脚本以来系统配置未进行过更改，而且是在创建脚本的同一系统上执行该脚本。

- 2 执行保存的 shell 脚本。

```
# sh ./metassist-shell-script-name
```

- 3 查看新卷。

```
# metastat -s diskset-name
```

### 示例 23-8 执行保存的 metassist 命令 shell 脚本

以下示例说明如何通过 metassist 命令使用 shell 脚本来创建卷。

```
# sh ./tmp/metassist-shell-script.sh
myset/hsp000: Hotspare pool is setup
myset/hsp000: Hotspare is added
myset/d2: Concat/Stripe is setup
myset/d1: Concat/Stripe is setup
myset/d0: Mirror is setup
myset/d0: submirror myset/d1 is attached
```

## 使用 metassist 命令创建卷配置文件

通过运行带 -d 参数的 metassist 命令，可以生成基于 XML 的卷配置文件，该文件详细指定了卷及其组件，包括所有的选项和关于卷的相关信息。查看该文件有助于理解 metassist 命令建议使用的配置。如果对该文件进行适当的更改，还可以优化配置，然后可以使用该卷配置文件作为 metassist 命令的输入以实际创建卷。

## ▼ 如何使用 metassist 命令创建卷配置文件

开始之前 查看第 222 页中的“自上而下创建卷的先决条件”。

- 1 确定要在其上创建卷的可用存储。

如果未显式指定存储，则 Solaris Volume Manager 会识别系统上未使用的存储并根据需要使用存储。如果选择指定存储，无论是广泛地指定（例如，控制器 1 上的所有存储）还是明确地指定（例如，使用 c1t4d2，但不使用 c1t4d1），Solaris Volume Manager 都会使用指定的存储。

## 2 使用 metassist 命令和适用于您的任务的相应选项。

使用 `-d` 选项指定不应实际创建卷。而应将基于 XML 的卷配置文件发送到标准输出中：

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... \
  -S size -v verbosity [-d]
```

<code>create</code>	是用于创建卷的子命令。
<code>-s diskset-name</code>	指定要用于卷的磁盘集的名称。
<code>-f</code>	指定该卷与热备件相关联。
<code>-r redundancy</code>	指定要创建的冗余级别（数据副本的数量）。
<code>-a device1, device2...</code>	指定可用于创建卷的设备。
<code>-S size</code>	以 KB（千字节）、MB（兆字节）、GB（千兆字节）或 TB（兆兆字节）为单位，指定要创建的卷的大小。
<code>-v verbosity</code>	指定输出的详细级别。允许的值介于 0（几乎无提示的输出）和 2（大量输出）之间。缺省级别为 1（中等输出）。
<code>-d</code>	指定不应实际创建卷，

---

注 – `-d` 参数所需的基于 XML 的卷配置文件将发送到标准输出中。但是，`metassist` 命令的其余输出将转至标准错误中。您可以在进行选择时重定向输出流。

---

有关更多信息，请参见以下示例和 [metassist\(1M\)](#) 手册页。

### 示例 23-9 使用 metassist 命令创建卷配置文件

以下示例说明如何使用 `metassist` 命令来创建大小为 10 MB 且具有可提供额外容错功能的热备件的双向镜像。`-f` 选项指定容错功能。最后一个参数 (`-d`) 指定不应实际创建卷，而应将最终可用来创建指定配置的卷配置文件发送到标准输出中。

```
# metassist create -s myset -f -r 2 -S 10mb -d

.(output truncated)
.
.
Volume request completed successfully.
<?xml version="1.0"?>
<!DOCTYPE volume-config SYSTEM "/usr/share/lib/xml/dtd/volume-config.dtd">
  <volume-config>
    <diskset name="myset">
      <disk name="clt3d0"/>
      <disk name="clt6d0"/>
      <disk name="clt4d0"/>
    <slice name="clt3d0s7" sizeinblocks="0"/>
  </volume-config>
</volume-config>
```

```

<slice name="clt3d0s0" sizeinblocks="17649765" startsector="10773"/>
<slice name="clt6d0s7" sizeinblocks="0"/>
<slice name="clt6d0s0" sizeinblocks="17649765" startsector="10773"/>
<slice name="clt4d0s7" sizeinblocks="0"/>
<slice name="clt4d0s0" sizeinblocks="17649765" startsector="10773"/>
<hsp name="hsp000">
<slice name="clt3d0s1" sizeinblocks="21546" startsector="17660538"/>
</hsp>
<mirror name="d0" read="ROUNDROBIN" write="PARALLEL" passnum="1">
<concat name="d2">
<slice name="clt4d0s1" sizeinblocks="21546" startsector="17660538"/>
<hsp name="hsp000"/>
</concat>
<concat name="d1">
<slice name="clt6d0s1" sizeinblocks="21546" startsector="17660538"/>
<hsp name="hsp000"/>
</concat>
</mirror>
</volume-config>
#

```

### 示例 23-10 使用 metassist 命令保存卷配置文件

以下示例说明如何使用 metassist 命令来创建大小为 10 MB 且具有可提供额外容错功能的热备件的双向镜像。-f 选项指定容错功能。最后一个参数 (-d) 指定不应实际创建卷，而应将最终可用来创建指定配置的卷配置文件发送到标准输出中。命令的结尾将重定向标准输出，以便创建 /tmp/metassist-volume-config.xml 卷配置文件，该配置文件稍后可用于创建指定的卷。

```
# metassist create -s myset -f -r 2 -S 10mb -d > \
/tmp/metassist-volume-config.xml
```

## 更改 metassist 命令的缺省行为

可以使用卷缺省文件 (/etc/defaults/metassist.xml) 更改 metassist 命令的缺省行为。通过更改缺省文件，可以显式地将特定的磁盘或控制器排除或包括在考虑范围之内。还可以指定 metassist 命令所使用的大多数卷设置的要求。

/etc/defaults/metassist.xml 的格式由 /usr/share/lib/xml/dtd/volume-defaults.dtd 文档类型定义 (Document Type Definition, DTD) 指定。此格式在 [volume-defaults\(4\)](#) 手册页中进行了介绍。

## 更改卷缺省文件

编辑卷缺省文件 (/etc/defaults/metassist.xml) 以指定 metassist 命令的行为方式。

---

注 – 在编辑该文件时，必须确保该文件仍然符合 /usr/share/lib/xml/dtd/volume-defaults.dtd 文档类型定义 (Document Type Definition, DTD)。如果该 XML 文件不符合 DTD，则 metassist 命令将失败，并显示一条错误消息。

---

示例 23-11 通过 metassist 命令使用更改后的缺省配置创建卷

在创建卷之前，请编辑 /etc/default/metassist.xml 文件以指定缺省设置，这些设置将应用于使用 metassist 命令所创建的所有卷。在本示例中，metassist 命令仅在控制器 c1 上创建卷，且在创建条带时仅创建正好包含四个组件且交错值为 512KB 的条带。这些约束适用于 metassist 命令的所有应用场合，直至 /etc/default/metassist.xml 文件再次发生更改。

```
# cat /etc/default/metassist.xml
<!DOCTYPE volume-defaults SYSTEM \
"/usr/share/lib/xml/dtd/volume-defaults.dtd">

<volume-defaults>
<available name="c1" />
<stripe mincomp="4" maxcomp="4" interlace="512KB" ></stripe>
</volume-defaults>

# metassist create -s myset -S 10Gb
```

metassist 命令会创建一个 10 GB 的条带，如 /etc/default/metassist.xml 文件中所指定，该条带将恰好使用四个分片且交错值为 512 KB。



## 监视和错误报告（任务）

有时 Solaris Volume Manager 会遇到问题，比如由于分片级的物理错误而无法写入卷。出现问题时，Solaris Volume Manager 会更改卷的状态，从而通知系统管理员。但是，除非通过 Solaris Management Console 在 Solaris Volume Manager GUI 中或者通过运行 `metastat` 命令来定期检查状态，否则可能不会立即看到这些状态更改。

本章提供了有关 Solaris Volume Manager 可用的各种监视工具的信息。其中一种工具是 Solaris Volume Manager SNMP 代理，该工具是 Solstice Enterprise Agents 监视软件的子代理。除了配置该工具以报告 SNMP 陷阱以外，还可以创建 shell 脚本来主动监视多个 Solaris Volume Manager 功能。该 shell 脚本可以作为 cron 作业运行，而且可用来确定潜在的问题。

以下是本章中信息的列表：

- 第 235 页中的“Solaris Volume Manager 监视与报告（任务列表）”
- 第 236 页中的“配置 `mdmonitord` 命令以定期检查错误”
- 第 237 页中的“Solaris Volume Manager SNMP 代理概述”
- 第 237 页中的“配置 Solaris Volume Manager SNMP 代理”
- 第 239 页中的“Solaris Volume Manager SNMP 代理的限制”
- 第 240 页中的“使用 cron 作业监视 Solaris Volume Manager”

## Solaris Volume Manager 监视与报告（任务列表）

以下任务列表列出了管理 Solaris Volume Manager 错误报告所需的过程。

任务	说明	参考
配置 <code>mdmonitord</code> 守护进程以定期检查错误	通过编辑 <code>/lib/svc/method/svc-mdmonitor</code> 脚本来配置 <code>mdmonitord</code> 守护进程所用的错误检查时间间隔。	第 236 页中的“配置 <code>mdmonitord</code> 命令以定期检查错误”

任务	说明	参考
配置 Solaris Volume Manager SNMP 代理	编辑 /etc/snmp/conf 目录中的配置文件，使 Solaris Volume Manager 相应地抛出陷阱到正确的系统。	<a href="#">第 237 页中的“配置 Solaris Volume Manager SNMP 代理”</a>
通过使用 cron 命令运行脚本来监视 Solaris Volume Manager	创建或改写脚本以检查错误，然后通过 cron 命令运行该脚本。	<a href="#">第 240 页中的“使用 cron 作业监视 Solaris Volume Manager”</a>

## 配置 mdmonitord 命令以定期检查错误

Solaris Volume Manager 包含 /usr/sbin/mdmonitord 守护进程。当磁盘出现故障时，Solaris Volume Manager 会检测到该故障并生成错误。该错误事件会触发 mdmonitord 守护进程对 RAID-1（镜像）卷、RAID-5 卷和热备件执行检查。不过，也可以将该程序配置为按指定的时间间隔主动检查错误。

### ▼ 如何配置 mdmonitord 命令以定期检查错误

编辑 /lib/svc/method/svc-mdmonitor 脚本以添加定期检查的时间间隔。

- 1 成为超级用户。
- 2 在所选的编辑器中打开 /lib/svc/method/svc-mdmonitor 脚本。在脚本中查找以下部分：

```
$MDMONITORD
error=$?
case $error in
0)      exit 0
        ;;

*)      echo "Could not start $MDMONITORD. Error $error."
        exit 0
```

- 3 更改 mdmonitord 命令的起始行，添加 -t 标志和两次检查之间的秒数。

```
$MDMONITORD -t 3600
error=$?
case $error in
0)      exit 0
        ;;

*)      echo "Could not start $MDMONITORD. Error $error."
        exit 0
        ;;
esac
```

#### 4 重新启动 `mdmonitord` 命令以激活更改。

```
# svcadm restart system/mdmonitor
```

有关更多信息，请参见 [mdmonitord\(1M\)](#) 手册页。

## Solaris Volume Manager SNMP 代理概述

Solaris Volume Manager SNMP 陷阱代理需要核心软件包 `SUNWlvmr` 以及 `SUNWlvma` 和 Solstice Enterprise Agents 的软件包。这些核心软件包包含以下内容：

- `SUNWmibii`
- `SUNWsacom`
- `SUNWsadmi`
- `SUNWsasnm`

这些软件包是 Solaris 操作系统的一部分。缺省情况下通常会安装这些软件包，除非在安装时修改了软件包选择或安装了最少的软件包集合。要确认这些软件包是否可用，请使用 `pkginfo pkgname` 命令（如 `pkginfo SUNWsasnm`）。确认所有五个软件包均可用后，需要按下节所述配置 Solaris Volume Manager SNMP 代理。

## 配置 Solaris Volume Manager SNMP 代理

缺省情况下，不启用 Solaris Volume Manager SNMP 代理。请使用以下过程启用 SNMP 陷阱。

每次升级 Solaris 操作系统后，都可能需要编辑 `/etc/snmp/conf/enterprises.oid` 文件并再次附加 [步骤 6](#) 中的行，然后重新启动 Solaris Enterprise Agents 服务器。

完成该过程后，系统会向指定的一个或多个主机发出 SNMP 陷阱。您需要使用相应的 SNMP 监视程序（例如 Solstice Enterprise Agents 软件）来查看所发出的陷阱。

设置 `mdmonitord` 命令以定期探测系统可有助于确保在出现问题时能够接收陷阱。请参见第 236 页中的“配置 `mdmonitord` 命令以定期检查错误”。有关其他错误检查选项，另请参阅第 240 页中的“使用 `cron` 作业监视 Solaris Volume Manager”。

### ▼ 如何配置 Solaris Volume Manager SNMP 代理

- 1 成为超级用户。
- 2 将 `/etc/snmp/conf/mdlogd.rsrc` 配置文件移到 `/etc/snmp/conf/mdlogd.rsrc` 中。

```
# mv /etc/snmp/conf/mdlogd.rsrc- /etc/snmp/conf/mdlogd.rsrc
```

- 3 编辑 `/etc/snmp/conf/mdlogd.acl` 文件以指定哪些主机应接收 SNMP 陷阱。在该文件中查找以下内容：

```
trap = {
{
    trap-community = SNMP-trap
    hosts = corsair
    {
        enterprise = "Solaris Volume Manager"
        trap-num = 1, 2, 3
    }
}
```

更改包含 `hosts = corsair` 的行以指定要接收 Solaris Volume Manager SNMP 陷阱的主机名。例如，要将 SNMP 陷阱发送到 `lexicon`，需要将该行更改为 `hosts = lexicon`。如果要包含多个主机，请提供以逗号分隔的主机名列表（如 `hosts = lexicon, idiom`）。

- 4 还需编辑 `/etc/snmp/conf/snmpdx.acl` 文件以指定哪些主机应接收 SNMP 陷阱。

找到以 `trap =` 开头的块，然后添加上一步添加的同一主机列表。这部分可能会使用 `#` 注释掉。在这种情况下，必须删除本部分中所必需的各行开头的 `#`。陷阱部分中的附加行也会注释掉。但是，可以保留这些行，或者为清楚起见将其删除。取消注释所必需的行并更新主机行后，本部分可能类似于以下内容：

```
#####
# trap parameters #
#####

trap = {
{
    trap-community = SNMP-trap
    hosts =lexicon
    {
        enterprise = "sun"
        trap-num = 0, 1, 2-5, 6-16
    }
}
# {
#     enterprise = "3Com"
#     trap-num = 4
# }
# {
#     enterprise = "snmp"
#     trap-num = 0, 2, 5
# }
# }
# {
#     trap-community = jerry-trap
#     hosts = jerry, nanak, hubble
#     {
#         enterprise = "sun"
#         trap-num = 1, 3
#     }
#     {
#         enterprise = "snmp"
#         trap-num = 1-3
#     }
# }
}
```

---

注 – 请确保 `/etc/snmp/conf/snmpdx.ac1` 文件中左括号和右括号的数量相同。

---

- 5 在 `/etc/snmp/conf/snmpdx.ac1` 文件内上一步取消注释的部分中添加新的 Solaris Volume Manager 部分。

```
trap-community = SNMP-trap
hosts = lexicon
{
    enterprise = "sun"
    trap-num = 0, 1, 2-5, 6-16
}
{
    enterprise = "Solaris Volume Manager"
    trap-num = 1, 2, 3
}
```

请注意，添加的四行紧接着 `enterprise = "sun"` 块放在后面。

- 6 在 `/etc/snmp/conf/enterprises.oid` 文件中追加以下行：

```
"Solaris Volume Manager" "1.3.6.1.4.1.42.104"
```

- 7 停止并重新启动 Solstice Enterprise Agents 服务器。

```
# /etc/init.d/init.snmpdx stop
# /etc/init.d/init.snmpdx start
```

## Solaris Volume Manager SNMP 代理的限制

Solaris Volume Manager SNMP 代理不会针对系统管理员需要知道的所有 Solaris Volume Manager 问题发出陷阱。具体来说，该代理仅在以下情况下发出陷阱：

- 某个 RAID-1 或 RAID-5 子组件进入了 "Needs Maintenance"（需要维护）状态
- 某个热备件因交换而开始服务
- 某个热备件开始重新同步
- 某个热备件完成了重新同步
- 某个镜像脱机
- 某一磁盘集被另一台主机所占用，且当前主机发生紧急情况

许多问题（例如包含 RAID-0 卷的磁盘或该磁盘上的软分区不可用）都不会导致产生 SNMP 陷阱，甚至在尝试读取和写入设备时也是如此。在这些情况下，通常会报告 SCSI 或 IDE 错误。但是，对于要报告给监视控制台的那些错误，其他 SNMP 代理必须发出陷阱。

# 使用 cron 作业监视 Solaris Volume Manager

## ▼ 如何自动检查卷中的错误

- 要自动检查 Solaris Volume Manager 配置中的错误，请创建一个 cron 实用程序可定期运行的脚本。

以下示例介绍了一个脚本，您可以根据需要对其进行改写和修改。

---

注 - 自动对 Solaris Volume Manager 进行错误检查的第一步是运行此脚本。您可能需要根据自己的配置修改此脚本。

---

```
#
#!/bin/ksh
#ident "@(#)metacheck.sh 1.3 96/06/21 SMI"
# ident='%Z%M% %I% %E% SMI'
#
# Copyright (c) 1999 by Sun Microsystems, Inc.
#
# metacheck
#
# Check on the status of the metadevice configuration. If there is a problem
# return a non zero exit code. Depending on options, send email notification.
#
# -h
# help
# -s setname
# Specify the set to check. By default, the 'local' set will be checked.
# -m recipient [recipient...]
# Send email notification to the specified recipients. This
# must be the last argument. The notification shows up as a short
# email message with a subject of
# "Solaris Volume Manager Problem: metacheck.who.nodename.setname"
# which summarizes the problem(s) and tells how to obtain detailed
# information. The "setname" is from the -s option, "who" is from
# the -w option, and "nodename" is reported by uname(1).
# Email notification is further affected by the following options:
# -f to suppress additional messages after a problem
# has been found.
# -d to control the suppression.
# -w to identify who generated the email.
# -t to force email even when there is no problem.
# -w who
# indicate who is running the command. By default, this is the
# user-name as reported by id(1M). This is used when sending
# email notification (-m).
# -f
# Enable filtering. Filtering applies to email notification (-m).
# Filtering requires root permission. When sending email notification
# the file /etc/lvm/metacheck.setname.pending is used to
# controll the filter. The following matrix specifies the behavior
```

```

#   of the filter:
#
#   problem_found   file_exists
#   yes             no       Create file, send notification
#   yes             yes      Resend notification if the current date
#                           (as specified by -d datefmt) is
#                           different than the file date.
#   no               yes      Delete file, send notification
#                           that the problem is resolved.
#   no               no       Send notification if -t specified.
#
# -d datefmt
#   Specify the format of the date for filtering (-f). This option
#   controls the how often re-notification via email occurs. If the
#   current date according to the specified format (strftime(3C)) is
#   identical to the date contained in the
#   /etc/lvm/metacheck.setname.pending file then the message is
#   suppressed. The default date format is "%D", which will send one
#   re-notification per day.
# -t
#   Test mode. Enable email generation even when there is no problem.
#   Used for end-to-end verification of the mechanism and email addresses.
#
# These options are designed to allow integration of metacheck
# into crontab. For example, a root crontab entry of:
#
# 0,15,30,45 * * * * /usr/sbin/metacheck -f -w SVMcron \
# -d '\%D \%h' -m notice@example.com 2148357243.8333033@pager.example.com
#
# would check for problems every 15 minutes, and generate an email to
# notice@example.com (and send to an email pager service) every hour when
# there is a problem. Note the \ prior to the '%' characters for a
# crontab entry. Bounced email would come back to root@nodename.
# The subject line for email generated by the above line would be
# Solaris Volume Manager Problem: metacheck.SVMcron.nodename.local
#

# display a debug line to controlling terminal (works in pipes)
decho()
{
    if [ "$debug" = "yes" ] ; then
        echo "DEBUG: $" < /dev/null > /dev/tty 2>&1
    fi
}

# if string $1 is in $2-* then return $1, else return ""
strstr()
{
    typeset    look="$1"
    typeset    ret=""

    shift

    # decho "strstr LOOK .$look. FIRST .$1."
    while [ $# -ne 0 ] ; do
        if [ "$look" = "$1" ] ; then
            ret="$look"
        fi
        shift
    done
}

```

```

done
echo "$ret"
}

# if string $1 is in $2-* then delete it. return result
strdstr()
{
    typeset    look="$1"
    typeset    ret=""

    shift
#   decho "strdstr LOOK .$look. FIRST .$1."
    while [ $# -ne 0 ] ; do
        if [ "$look" != "$1" ] ; then
            ret="$ret $1"
        fi
    shift
done
echo "$ret"
}

merge_continued_lines()
{
    awk -e '\
BEGIN { line = ""; } \
$NF == "\\\" { \
    $NF = ""; \
    line = line $0; \
    next; \
} \
$NF != "\\\" { \
    if ( line != "" ) { \
        print line $0; \
        line = ""; \
    } else { \
        print $0; \
    } \
}'
}

# trim out stuff not associated with metadevices
find_meta_devices()
{
    typeset    devices=""

#   decho "find_meta_devices .$*."
    while [ $# -ne 0 ] ; do
        case $1 in
            d+([0-9]) )    # metadevice name
                devices="$devices $1"
                ;;
        esac
    shift
done
echo "$devices"
}

# return the list of top level metadevices
toplevel()

```

```

{
    typeset      comp_meta_devices=""
    typeset      top_meta_devices=""
    typeset      devices=""
    typeset      device=""
    typeset      comp=""

    metastat$setarg -p | merge_continued_lines | while read line ; do
    echo "$line"
    devices='find_meta_devices $line'
    set -- $devices
    if [ $# -ne 0 ] ; then
        device=$1
        shift
        # check to see if device already referred to as component
        comp='strstr $device $comp_meta_devices'
        if [ -z $comp ] ; then
            top_meta_devices="$top_meta_devices $device"
        fi
        # add components to component list, remove from top list
        while [ $# -ne 0 ] ; do
            comp=$1
            comp_meta_devices="$comp_meta_devices $comp"
            top_meta_devices='strdstr $comp $top_meta_devices'
            shift
        done
    fi
    done > /dev/null 2>&1
    echo $top_meta_devices
}

#
# - MAIN
#
METAPATH=/usr/sbin
PATH=/usr/bin:$METAPATH
USAGE="usage: metacheck [-s setname] [-h] [[-t] [-f [-d datefmt]] \
    [-w who] -m recipient [recipient...]"

datefmt="%D"
debug="no"
filter="no"
mflag="no"
set="local"
setarg=""
testarg="no"
who='id | sed -e 's/^uid=[0-9][0-9]*(/' -e 's/).*/''

while getopts d:Dfms:tw: flag
do
    case $flag in
    d)    datefmt=$OPTARG;
        ;;
    D)    debug="yes"
        ;;
    f)    filter="yes"
        ;;
    m)    mflag="yes"
        ;;
    )
    ;;

```

```

s)      set=$OPTARG;
if [ "$set" != "local" ] ; then
    setarg="-s $set";
fi
;;
t)      testarg="yes";
;;
w)      who=$OPTARG;
;;
\?)     echo $USAGE
exit 1
;;
esac
done

# if mflag specified then everything else part of recipient
shift 'expr $OPTIND - 1'
if [ $mflag = "no" ] ; then
    if [ $# -ne 0 ] ; then
        echo $USAGE
        exit 1
    fi
else
    if [ $# -eq 0 ] ; then
        echo $USAGE
        exit 1
    fi
fi
recipients="$*"

curdate_filter='date +%datefmt'
curdate='date'
node='uname -n'

# establish files
msg_f=/tmp/metacheck.msg.$$
msgs_f=/tmp/metacheck.msgs.$$
metastat_f=/tmp/metacheck.metastat.$$
metadb_f=/tmp/metacheck.metadb.$$
metahs_f=/tmp/metacheck.metahs.$$
pending_f=/etc/lvm/metacheck.$set.pending
files="$metastat_f $metadb_f $metahs_f $msg_f $msgs_f"

rm -f $files > /dev/null 2>&1
trap "rm -f $files > /dev/null 2>&1; exit 1" 1 2 3 15

# Check to see if metadb is capable of running
have_metadb="yes"
metadb$setarg > $metadb_f 2>&1
if [ $? -ne 0 ] ; then
    have_metadb="no"
fi
grep "there are no existing databases" < $metadb_f > /dev/null 2>&1
if [ $? -eq 0 ] ; then
    have_metadb="no"
fi
grep "/dev/md/admin" < $metadb_f > /dev/null 2>&1
if [ $? -eq 0 ] ; then
    have_metadb="no"

```

```

fi

# check for problems accessing metadbs
retval=0
if [ "$have_metadb" = "no" ] ; then
    retval=1
    echo "metacheck: metadb problem, can't run '$METAPATH/metadb$setarg'\" \
        >> $msgs_f
else
    # snapshot the state
    metadb$setarg 2>&1 | sed -e '1d' | merge_continued_lines > $metadb_f
    metastat$setarg 2>&1 | merge_continued_lines > $metastat_f
    metahs$setarg -i 2>&1 | merge_continued_lines > $metahs_f

    #
    # Check replicas for problems, capital letters in the flags
    # indicate an error, fields are seperated by tabs.
    #
    problem='awk < $metadb_f -F\t '{if ($1 ~ /[A-Z]/) print $1;}''
    if [ -n "$problem" ] ; then
        retval='expr $retval + 64'
        echo "\
metacheck: metadb problem, for more detail run:\n\t$METAPATH/metadb$setarg -i\" \
            >> $msgs_f
    fi

    #
    # Check the metadvice state
    #
    problem='awk < $metastat_f -e \
        '/State:/ {if ($2 != "Okay" && $2 != "Resyncing") print $0;}''
    if [ -n "$problem" ] ; then
        retval='expr $retval + 128'
        echo "\
metacheck: metadvice problem, for more detail run:\" \
            >> $msgs_f

    # refine the message to toplevel metadvicees that have a problem
    top='toplevel'
    set -- $top
    while [ $# -ne 0 ] ; do
        device=$1
        problem='metastat $device | awk -e \
            '/State:/ {if ($2 != "Okay" && $2 != "Resyncing") print $0;}''
        if [ -n "$problem" ] ; then
            echo "\t$METAPATH/metastat$setarg $device" >> $msgs_f
            # find out what is mounted on the device
            mp='mount|awk -e '/\/dev\/md\/dsk\/'$device'[ \t]/{print $1;}''
            if [ -n "$mp" ] ; then
                echo "\t\t$mp mounted on $device" >> $msgs_f
            fi
        fi
        shift
    done
fi

#
# Check the hotspares to see if any have been used.
#

```

```

        problem=""
        grep "no hotspare pools found" < $metahs_f > /dev/null 2>&1
        if [ $? -ne 0 ] ; then
            problem='awk < $metahs_f -e \
                '/blocks/ { if ( $2 != "Available" ) print $0;}'
        fi
        if [ -n "$problem" ] ; then
            retval='expr $retval + 256'
            echo "\
metacheck: hot spare in use, for more detail run:\n\t$METAPATH/metahs$setarg -i" \
                >> $msgs_f
        fi
    fi

# If any errors occurred, then mail the report
if [ $retval -ne 0 ] ; then
    if [ -n "$recipients" ] ; then
        re=""
        if [ -f $pending_f ] && [ "$filter" = "yes" ] ; then
            re="Re: "
            # we have a pending notification, check date to see if we resend
            penddate_filter='cat $pending_f | head -1'
            if [ "$curdate_filter" != "$penddate_filter" ] ; then
                rm -f $pending_f > /dev/null 2>&1
            else
                if [ "$debug" = "yes" ] ; then
                    echo "metacheck: email problem notification still pending"
                    cat $pending_f
                fi
            fi
        fi
        if [ ! -f $pending_f ] ; then
            if [ "$filter" = "yes" ] ; then
                echo "$curdate_filter\n\tDate:$curdate\n\tTo:$recipients" \
                    > $pending_f
            fi
            echo "\
Solaris Volume Manager: $node: metacheck$setarg: Report: $curdate" >> $msg_f
            echo "\
-----" >> $msg_f
            cat $msg_f $msgs_f | mailx -s \
                "${re}Solaris Volume Manager Problem: metacheck.$who.$set.$node" $recipients
        fi
    else
        cat $msgs_f
    fi
else
    # no problems detected,
    if [ -n "$recipients" ] ; then
        # default is to not send any mail, or print anything.
        echo "\
Solaris Volume Manager: $node: metacheck$setarg: Report: $curdate" >> $msg_f
        echo "\
-----" >> $msg_f
        if [ -f $pending_f ] && [ "$filter" = "yes" ] ; then
            # pending filter exists, remove it and send OK
            rm -f $pending_f > /dev/null 2>&1
            echo "Problem resolved" >> $msg_f
            cat $msg_f | mailx -s \

```

```
    "Re: Solaris Volume Manager Problem: metacheck.$who.$node.$set" $recipients
elif [ "$testarg" = "yes" ] ; then
    # for testing, send mail every time even though there is no problem
    echo "Messaging test, no problems detected"      >> $msg_f
    cat $msg_f | mailx -s \
        "Solaris Volume Manager Problem: metacheck.$who.$node.$set" $recipients
fi
else
    echo "metacheck: Okay"
fi
fi

rm -f $files                                > /dev/null 2>&1
exit $retval
```

有关使用 cron 实用程序来调用脚本的信息，请参见 [cron\(1M\)](#) 手册页。



## Solaris Volume Manager 故障排除（任务）

---

本章介绍如何对 Solaris Volume Manager 相关问题进行故障排除。本章同时提供了一般的故障排除原则以及解决某些已知问题的特定过程。

本章包括以下信息：

- 第 249 页中的“Solaris Volume Manager 故障排除（任务列表）”
- 第 250 页中的“系统故障排除概览”
- 第 251 页中的“替换磁盘”
- 第 253 页中的“从磁盘移动问题中恢复”
- 第 255 页中的“升级到 Solaris 10 发行版之后设备 ID 的改变”
- 第 256 页中的“从引导问题中恢复”
- 第 263 页中的“从状态数据库副本故障中恢复”
- 第 265 页中的“从软分区问题中恢复”
- 第 267 页中的“从另一个系统恢复存储”
- 第 273 页中的“从磁盘集问题中恢复”
- 第 275 页中的“使用 `ufsdump` 命令对已挂载的文件系统执行备份”
- 第 276 页中的“执行系统恢复”

本章介绍了一些 Solaris Volume Manager 问题及相应的解决方案。本章无意涵盖所有的故障情形，仅提供一些常见的场景和恢复过程。

## Solaris Volume Manager 故障排除（任务列表）

以下任务列表列出了对 Solaris Volume Manager 进行故障排除所需的一些过程。

任务	说明	参考
替换出现故障的磁盘	替换磁盘，然后在新磁盘上更新状态数据库副本和逻辑卷。	第 252 页中的“如何替换故障磁盘”

任务	说明	参考
从磁盘移动问题中恢复	将磁盘恢复到其原始位置或联系产品支持人员。	<a href="#">第 253 页中的“从磁盘移动问题中恢复”</a>
从错误的 <code>/etc/vfstab</code> 条目中恢复	针对镜像使用 <code>fsck</code> 命令，然后编辑 <code>/etc/vfstab</code> 文件，以便系统可以正确引导。	<a href="#">第 257 页中的“如何从不正确的 <code>/etc/vfstab</code> 项恢复”</a>
从引导设备故障中恢复	从另一个子镜像进行引导。	<a href="#">第 259 页中的“如何从引导设备故障中恢复”</a>
从不充分的状态数据库副本中恢复	使用 <code>metadb</code> 命令删除不可用的副本。	<a href="#">第 263 页中的“如何从不足的状态数据库副本中恢复”</a>
恢复已丢失软分区的配置数据	使用 <code>metarecover</code> 命令恢复软分区的配置数据。	<a href="#">第 265 页中的“如何恢复软分区的配置数据”</a>
从挽救的磁盘中恢复 Solaris Volume Manager 配置	将磁盘连接到新系统，并让 Solaris Volume Manager 从现有的状态数据库副本中重新生成配置。	<a href="#">第 267 页中的“如何从本地磁盘集恢复存储”</a>
从另一个系统恢复存储	将已知磁盘集中的存储导入到不同的系统中。	<a href="#">第 267 页中的“从另一个系统恢复存储”</a>
清除无法访问的磁盘集。	使用 <code>metaset</code> 命令清除无法获取或使用的磁盘集的内容。	<a href="#">第 273 页中的“从磁盘集问题中恢复”</a>
恢复存储在 Solaris Volume Manager 卷上的系统配置。	使用 Solaris OS 安装介质恢复存储在 Solaris Volume Manager 卷上的系统配置。	<a href="#">第 276 页中的“执行系统恢复”</a>

# 系统故障排除概览

## 系统故障排除先决条件

要对与 Solaris Volume Manager 相关的存储管理问题进行故障排除，需要执行以下操作：

- 具有 root 用户权限
- 具有所有数据的最新备份

## Solaris Volume Manager 故障排除的一般原则

对 Solaris Volume Manager 问题进行故障排除时，应该具备以下信息：

- `metadb` 命令的输出
- `metastat` 命令的输出

- `metastat -p` 命令的输出
- `/etc/vfstab` 文件的备份副本
- `/etc/lvm/mddb.cf` 文件的备份副本
- 磁盘分区信息，可通过 `prtvtoc` 命令（SPARC 系统）或 `fdisk` 命令（基于 x86 的系统）获得
- 系统上的 Solaris 版本
- 已安装的 Solaris 修补程序列表
- 已安装的 Solaris Volume Manager 修补程序列表

---

**提示** – 每次更新 Solaris Volume Manager 配置或对系统执行其他与存储或操作系统相关的更改，都会生成此配置信息的全新副本。还可以通过 `cron` 作业自动生成此信息。

---

## 一般故障排除方法

尽管无法通过一个过程来评估所有的 Solaris Volume Manager 问题，但以下过程提供了有助于评估的一般方法。

1. 收集有关当前配置的信息。
2. 查看当前的状态指示器，包括 `metastat` 和 `metadb` 命令的输出。此信息应指示哪个组件存在故障。
3. 检查硬件是否存在明显的故障点：
  - 所有连接是否正常？
  - 最近是否停过电？
  - 是否更改或添加过任何设备？

## 替换磁盘

本节介绍了如何在 Solaris Volume Manager 环境中替换磁盘。



---

**注意** – 如果故障磁盘或者构建在故障磁盘上的卷上有软分区，则必须将新磁盘置于同一物理位置。此外，请使用与要替换的磁盘相同的 `c ntnd n` 编号。

---

## ▼ 如何替换故障磁盘

1 通过检查 `/var/adm/messages` 文件和 `metastat` 命令输出来确定要替换的故障磁盘。

2 查找可能放在故障磁盘上的所有状态数据库副本。

可使用 `metadb` 命令查找这些副本。

对于位于故障磁盘上的状态数据库副本，`metadb` 命令可能会报告错误。在本示例中，出现问题的设备是 `c0t1d0`。

```
# metadb
  flags      first blk      block count
a m      u      16          1034      /dev/dsk/c0t0d0s4
a      u      1050         1034      /dev/dsk/c0t0d0s4
a      u      2084         1034      /dev/dsk/c0t0d0s4
W pc lu0    16          1034      /dev/dsk/c0t1d0s4
W pc lu0    1050         1034      /dev/dsk/c0t1d0s4
W pc lu0    2084         1034      /dev/dsk/c0t1d0s4
```

输出显示，在每个本地磁盘（`c0t0d0` 和 `c0t1d0`）的分片 4 上都有三个状态数据库副本。`c0t1d0s4` 分片的标志字段中的 `W` 表示设备出现写入错误。`c0t0d0s4` 分片上的三个副本仍处于良好状态。

3 记录状态数据库副本所在的分片名称和状态数据库副本的个数。然后，删除状态数据库副本。

状态数据库副本的个数可通过计算 `metadb` 命令输出中某个分片的出现次数而得出。在本示例中，将删除 `c0t1d0s4` 上的三个状态数据库副本。

```
# metadb -d c0t1d0s4
```



注意 - 删除有问题的状态数据库副本后，如果只剩下三个或更少的副本，请先[添加更多的状态数据库副本](#)，然后再继续。执行此操作有助于确保配置信息保持完整。

4 查找并删除故障磁盘上的所有热备件。

可使用 `metastat` 命令查找热备件。在本示例中，热备用池 `hsp000` 包含 `c0t1d0s6`（随后将从池中删除）。

```
# metahs -d hsp000 c0t1d0s6
hsp000: Hotspare is deleted
```

5 替换出现故障的磁盘。

此步骤可能需要使用 `cfgadm` 命令、`luxadm` 命令或其他适用于您的硬件和环境的命令。执行此步骤时，请确保按照硬件文档中的过程正确地处理该磁盘的 Solaris 状态。

6 对新磁盘重新分区。

通过 `format` 命令或 `fmthard` 命令使用与故障磁盘相同的分片信息对该磁盘进行分区。如果有故障磁盘的 `prtvtoc` 输出，则可以使用 `fmthard -s /tmp/failed-disk-prtvtoc-output` 命令格式化替换磁盘。

- 7 如果您删除了一些状态数据库副本，请向相应的分片重新添加相同数量的副本。  
在此示例中，使用了 `/dev/dsk/c0t1d0s4`。  

```
# metadb -a -c 3 c0t1d0s4
```
- 8 如果磁盘上的某些分片是 RAID-5 卷或 RAID-0 卷的组件，而这些卷又是 RAID-1 卷的子镜像，请针对每个分片运行 `metareplace -e` 命令。  
在此示例中，使用了 `/dev/dsk/c0t1d0s4` 和镜像 `d10`。  

```
# metareplace -e d10 c0t1d0s4
```
- 9 如果在替换磁盘的分片上直接构建了软分区，请针对软分区所在的每个分片运行 `metarecover -m -p` 命令。此命令将在磁盘上重新生成盘区标头。  
在本示例中，`/dev/dsk/c0t1d0s4` 需要重新生成磁盘上的软分区标记。系统会扫描该分片并基于状态数据库副本中的信息重新应用这些标记。  

```
# metarecover c0t1d0s4 -m -p
```
- 10 如果磁盘上的某些软分区是 RAID-5 卷或 RAID-0 卷的组件，而这些卷又是 RAID-1 卷的子镜像，请针对每个分片运行 `metareplace -e` 命令。  
在此示例中，使用了 `/dev/dsk/c0t1d0s4` 和镜像 `d10`。  

```
# metareplace -e d10 c0t1d0s4
```
- 11 如果在 RAID-0 卷上构建了软分区，请针对每个 RAID-0 卷运行 `metarecover` 命令。  
在本示例中，RAID-0 卷 `d17` 上构建了软分区。  

```
# metarecover d17 -m -p
```
- 12 替换已删除的热备件，然后将其添加到相应的热备用池或其他池中。  
在本示例中，热备用池 `hsp000` 包含 `c0t1d0s6`。此分片将添加至热备用池。  

```
# metabs -a hsp000 c0t1d0s6
```

`hsp000: Hotspare is added`
- 13 如果软分区或非冗余卷受到故障影响，请通过备份恢复数据。如果仅冗余卷受到影响，请对数据进行验证。  
检查所有卷上的用户和应用程序数据。可能必须运行应用程序级别的一致性检验器，或者使用其他方法检查数据。

## 从磁盘移动问题中恢复

本节介绍了在 Solaris Volume Manager 环境中移动磁盘之后如何从意外问题中恢复。

## 磁盘移动和设备 ID 概述

Solaris Volume Manager 使用与特定磁盘相关联的设备 ID 跟踪 Solaris Volume Manager 配置中使用的所有磁盘。当磁盘移动到不同的控制器或 SCSI 目标编号发生更改时，Solaris Volume Manager 通常可以正确地识别该移动并相应地更新所有相关的 Solaris Volume Manager 记录。这不需要系统管理员进行干预。在个别情况下，Solaris Volume Manager 无法全面更新记录，会在系统引导时报告错误。

## 解决未命名设备错误消息

如果添加了新的硬件或移动了硬件（例如，将一组磁盘从一个控制器移动到另一个控制器），则 Solaris Volume Manager 将检查与移动的磁盘相关联的设备 ID，然后相应地更新内部 Solaris Volume Manager 记录中的 *cnt ndn* 名称。如果无法更新这些记录，则 `svc:/system/mdmonitor` 服务所产生的引导过程会在引导时向控制台报告错误：

```
Unable to resolve unnamed devices for volume management.  
Please refer to the Solaris Volume Manager documentation,  
Troubleshooting section, at http://docs.sun.com or from  
your local copy.
```

该问题既不会导致数据丢失，也不会直接产生任何负面影响。此错误消息表明仅更新了部分 Solaris Volume Manager 名称记录。`metastat` 命令的输出中将显示一些之前使用过的 *cn tndn* 名称，还将显示一些可反映移动后状态的 *cn tndn* 名称。

在这种情况下，如果需要更新 Solaris Volume Manager 配置，则在发出任何 `meta*` 命令时，必须使用由 `metastat` 命令报告的 *cntnd n* 名称。

如果出现这种错误状态，可以执行以下操作之一来解决此问题：

- 将所有磁盘恢复回其原始位置。然后，执行重新配置重新引导，或者运行以下命令（作为单个命令）：

```
/usr/sbin/devfsadm && /usr/sbin/metadevadm -r
```

这些命令完成后，该错误状态将得以解决。

- 请与您的技术支持代表联系以寻求帮助。

---

注—此错误状态极少出现。如果确实出现了，则很有可能会影响与光纤通道连接的存储。

---

## 升级到 Solaris 10 发行版之后设备 ID 的改变

从 Solaris 10 发行版开始，设备 ID 输出将以新的格式显示。Solaris Volume Manager 可能以新格式或旧格式显示设备 ID 输出，具体取决于将设备 ID 信息添加到状态数据库副本中的时间。

以前，设备 ID 以十六进制值显示。新格式的设备 ID 显示为 ASCII 字符串。在很多情况下，这种变化极其微小，如以下示例所示：

旧格式：`id1,ssd@w600c0ff00000000007ecd255a9336d00`

新格式：`id1,ssd@n600c0ff00000000007ecd255a9336d00`

在另一些情况下，这种变化较为显著，如以下示例所示：

旧格式：`id1,sd@w4849544143484920444b3332454a2d33364e4320202020203433334239383939`

新格式：`id1,ssd@n600c0ff00000000007ecd255a9336d00`

升级至 Solaris 10 发行版后，对于与以前的 Solaris 发行版中创建的现有磁盘集相关联的设备 ID，不会在 Solaris Volume Manager 配置中更新其格式。如果需要恢复到以前的 Solaris 发行版，那么在升级之后对磁盘集进行的配置更改在以前的发行版中可能会不可用。这些配置更改包括：

- 向升级前已存在的磁盘集中添加新磁盘
- 创建新磁盘集
- 创建状态数据库副本

这些配置更改会影响能够在 Solaris Volume Manager 中创建的所有磁盘集，包括本地磁盘集。例如，如果对 Solaris 10 发行版中创建的磁盘集实施了上述任何更改，则无法将该磁盘集导入到之前的 Solaris 发行版中。另一个示例是，您可能会将镜像根的一面升级到 Solaris 10 发行版，然后对本地集进行配置更改。如果随后又将该子镜像重新合并到以前的 Solaris 发行版中，系统将无法识别这些更改。

Solaris 10 OS 配置始终以新格式显示设备 ID，即便是在升级期间也是如此。可以使用 `prtconf -v` 命令显示此信息。而 Solaris Volume Manager 会以旧格式或新格式显示。Solaris Volume Manager 中具体显示哪种格式取决于开始使用该磁盘时所运行的 Solaris OS 版本。要确定 Solaris Volume Manager 所显示的设备 ID 是否与 Solaris OS 配置中的设备 ID 格式不同但等效，请比较 `metastat` 命令的输出与 `prtconf -v` 命令的输出。

在以下示例中，对于同一个磁盘 `c1t6d0`，`metastat` 命令输出中显示的设备 ID 与 `prtconf -v` 命令输出中的设备 ID 格式不同但等效。

```
# metastat
d127: Concat/Stripe
    Size: 17629184 blocks (8.4 GB)
    Stripe 0:
        Device      Start Block  Dbase   Reloc
```

```

c1t6d0s2      32768      Yes      Yes

Device Relocation Information:
Device Reloc Device ID c1t6d0      Yes      id1,sd@w4849544143484920444b3332454a2d3336
4e4320202020203433334239383939

# prtconf -v
.(output truncated)

.
.
sd, instance #6
  System properties:
    name='lun' type=int items=1
    value=00000000
    name='target' type=int items=1
    value=00000006
    name='class' type=string items=1
    value='scsi'
  Driver properties:
    name='pm-components' type=string items=3 dev=none
    value='NAME=spindle-motor' + '0=off' + '1=on'
    name='pm-hardware-state' type=string items=1 dev=none
    value='needs-suspend-resume'
    name='ddi-failfast-supported' type=boolean dev=none
    name='ddi-kernel-ioctl' type=boolean dev=none
  Hardware properties:
    name='devid' type=string items=1
    value='id1,@THITACHI_DK32EJ-36NC_____433B9899'

.
.
.
.(output truncated)
```

prtconf -v 命令输出中包含 "instance #6" 的行与 metastat 命令输出中的磁盘 c1t6d0 相对应。prtconf -v 命令输出中的设备 ID id1,@THITACHI\_DK32EJ-36NC\_\_\_\_\_433B9899 与 metastat 命令输出中的设备 ID id1,sd@w4849544143484920444b3332454a2d33364e43202020203433334239383939 相对应。这两个命令输出之间的区别表明，Solaris Volume Manager 在 metastat 命令的输出中以十六进制格式显示设备 ID，而 Solaris 10 OS 配置在 prtconf 命令的输出中以 ASCII 字符串的形式显示设备 ID。

# 从引导问题中恢复

由于 Solaris Volume Manager 允许您镜像根 (/)、swap 和 /usr 目录，因此在引导系统时可能会出现特殊问题。这些问题可能由硬件故障引起，也可能由操作错误引起。本节中的过程针对此类潜在问题提供了解决方案。

下表列出了这些问题，并提供了指向相应解决方案的链接。

表 25-1 Solaris Volume Manager 的常见引导问题

出现引导问题的原因	参考
/etc/vfstab 文件包含错误的信息。	第 257 页中的“如何从不正确的 /etc/vfstab 项恢复”
未定义足够的状态数据库副本。	第 263 页中的“如何从不足的状态数据库副本中恢复”
引导设备（磁盘）出现故障。	第 259 页中的“如何从引导设备故障中恢复”

## 引导问题的背景信息

- 如果 Solaris Volume Manager 由于错误而使卷脱机，请卸载发生故障的磁盘上的所有文件系统。  
由于每个磁盘分片都是独立的，因此可以在单个磁盘上挂载多个文件系统。如果软件遇到了故障，则同一磁盘上的其他分片有可能很快也会遇到故障。直接挂载在磁盘分片上的文件系统不具备 Solaris Volume Manager 错误处理的保护功能。如果使这样的文件系统保持挂载，则很可能会出现系统崩溃和数据丢失的情况。
- 请尽可能缩短使用已禁用或脱机的子镜像运行的时间。在重新同步和联机备份间隔期间，镜像将失去所有的保护。

## 如何从不正确的 /etc/vfstab 项恢复

如果在 /etc/vfstab 文件中创建了错误的条目（例如，在镜像根 (/) 文件系统时），则系统最初看起来引导正确，但随后会出现故障。要解决此问题，需要在单用户模式下编辑 /etc/vfstab 文件。

从错误的 /etc/vfstab 文件条目中恢复的概括性步骤如下所示：

1. 将系统引导到单用户模式
2. 针对镜像卷运行 fsck 命令
3. 重新挂载文件系统并启用读写选项
4. 可选：针对根 (/) 镜像运行 metaroot 命令
5. 验证 /etc/vfstab 文件是否正确引用了与该文件系统条目对应的卷
6. 重新引导系统

### ▼ 恢复根 (/) RAID-1（镜像）卷

在以下示例中，根 (/) 文件系统使用双向镜像 d0 进行镜像。/etc/vfstab 文件中的根 (/) 条目由于某种原因又恢复为文件系统的原始分片。但是，/etc/system 文件中的信息仍显示将从镜像 d0 引导。最可能的原因就是，没有使用 metaroot 命令来维护 /etc/system 和 /etc/vfstab 文件。另外一个原因可能是，/etc/vfstab 文件的旧副本被复制回当前的 /etc/vfstab 文件中。

不正确的 `/etc/vfstab` 文件与以下内容类似：

```
#device      device      mount      FS      fsck  mount  mount
#to mount    to fsck     point      type    pass  at boot options
#
/dev/dsk/c0t3d0s0 /dev/rdsk/c0t3d0s0 /      ufs      1      no      -
/dev/dsk/c0t3d0s1 -           -         swap     -       no      -
/dev/dsk/c0t3d0s6 /dev/rdsk/c0t3d0s6 /usr    ufs      2      no      -
#
/proc        -           /proc     proc     -       no      -
swap         -           /tmp      tmpfs    -       yes     -
```

由于这些错误，在系统引导时，将自动进入单用户模式：

```
ok boot
...
configuring network interfaces: hme0.
Hostname: host1
mount: /dev/dsk/c0t3d0s0 is not this fstype.
setmnt: Cannot open /etc/mnttab for writing

INIT: Cannot create /var/adm/utmp or /var/adm/utmpx

INIT: failed write of utmpx entry:" "

INIT: failed write of utmpx entry:" "

INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): <root-password>
```

此时，根 (/) 和 /usr 文件系统都以只读模式挂载。请执行以下步骤：

1 在根 (/) 镜像上运行 `fsck` 命令。

---

注 – 请小心谨慎，务必对根 (/) 镜像使用正确的卷。

---

```
# fsck /dev/md/rdsk/d0
** /dev/md/rdsk/d0
** Currently Mounted on /
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2274 files, 11815 used, 10302 free (158 frags, 1268 blocks,
0.7% fragmentation)
```

2 重新以读/写方式挂载根 (/) 文件系统，以便于编辑 `/etc/vfstab` 文件。

```
# mount -o rw,remount /dev/md/dsk/d0 /
mount: warning: cannot lock temp file </etc/.mnt.lock>
```

3 运行metaroot命令。

```
# metaroot d0
```

此命令将编辑 /etc/system 和 /etc/vfstab 文件，以指定根 (/) 文件系统当前位于卷 d0 上。

4 验证 /etc/vfstab 文件是否包含正确的卷条目。

/etc/vfstab 文件中的根 (/) 条目应显示为以下内容，文件系统条目才能正确引用 RAID-1 卷：

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
/dev/md/dsk/d0	/dev/md/rdisk/d0	/	ufs	1	no	-
/dev/dsk/c0t3d0s1	-	-	swap	-	no	-
/dev/dsk/c0t3d0s6	/dev/rdisk/c0t3d0s6	/usr	ufs	2	no	-
#						
/proc	-	/proc	proc	-	no	-
swap	-	/tmp	tmpfs	-	yes	-

5 重新引导系统。

系统将恢复到正常的运行状态。

▼ 如何从引导设备故障中恢复

如果您有一个根 (/) 镜像，那么，当引导设备出现故障时，您需要设置备用引导设备。

本任务中的概括性步骤如下所示：

- 从备用根 (/) 子镜像引导
- 确定出错的状态数据库副本和卷
- 修复出现故障的磁盘
- 将状态数据库副本和卷恢复到其原始状态

最初，当引导设备出现故障时，您将看到类似于以下内容的消息。在不同的体系结构中，此消息可能会有所不同。

```
Rebooting with command:
Boot device: /iommu/sbus/dma@f,81000/esp@f,80000/sd@3,0
The selected SCSI device is not responding
Can't open boot device
...
```

看到此消息时，请记住该设备。然后，请执行以下步骤：

1 从其他根 (/) 子镜像引导。

由于本示例中的六个状态数据库副本中只有两个出错，因此仍可以引导系统。如果不是这种情况，则需要在单用户模式下删除不可访问的状态数据库副本。此过程在[第 263 页](#)中的“如何从不足的状态数据库副本中恢复”中介绍。

在为根 (/) 文件系统创建镜像时，应在执行过程中记录备用引导设备。在本示例中，disk2 是备用引导设备。

```
ok boot disk2
SunOS Release 5.9 Version s81_51 64-bit
Copyright 1983-2001 Sun Microsystems, Inc. All rights reserved.
Hostname: demo
...
demo console login: root
Password: <root-password>
Dec 16 12:22:09 host1 login: ROOT LOGIN /dev/console
Last login: Wed Dec 12 10:55:16 on console
Sun Microsystems Inc. SunOS 5.9 s81_51 May 2002
...
```

2 使用 metadb 命令确定已出现故障的状态数据库副本的个数。

```
# metadb
      flags          first blk    block count
M      p          unknown      unknown      /dev/dsk/c0t3d0s3
M      p          unknown      unknown      /dev/dsk/c0t3d0s3
a m    p    luo     16          1034      /dev/dsk/c0t2d0s3
a      p    luo    1050         1034      /dev/dsk/c0t2d0s3
a      p    luo     16          1034      /dev/dsk/c0t1d0s3
a      p    luo    1050         1034      /dev/dsk/c0t1d0s3
```

在本示例中，系统无法再检测到故障磁盘上的分片 /dev/dsk/c0t3d0s3 上的状态数据库副本。

3 使用 metastat 命令确定一半的根 (/)、swap 和 /usr 镜像已出现故障。

```
# metastat
d0: Mirror
  Submirror 0: d10
    State: Needs maintenance
  Submirror 1: d20
    State: Okay
...

d10: Submirror of d0
  State: Needs maintenance
  Invoke: "metareplace d0 /dev/dsk/c0t3d0s0 <new device>"
  Size: 47628 blocks
  Stripe 0:
    Device          Start Block  Dbase State          Hot Spare
    /dev/dsk/c0t3d0s0      0      No    Maintenance

d20: Submirror of d0
  State: Okay
  Size: 47628 blocks
  Stripe 0:
    Device          Start Block  Dbase State          Hot Spare
    /dev/dsk/c0t2d0s0      0      No    Okay

d1: Mirror
  Submirror 0: d11
    State: Needs maintenance
  Submirror 1: d21
```

```
State: Okay
...

d11: Submirror of d1
State: Needs maintenance
Invoke: "metareplace d1 /dev/dsk/c0t3d0s1 <new device>"
Size: 69660 blocks
Stripe 0:
Device          Start Block Dbase State      Hot Spare
/dev/dsk/c0t3d0s1      0      No   Maintenance

d21: Submirror of d1
State: Okay
Size: 69660 blocks
Stripe 0:
Device          Start Block Dbase State      Hot Spare
/dev/dsk/c0t2d0s1      0      No   Okay

d2: Mirror
Submirror 0: d12
State: Needs maintenance
Submirror 1: d22
State: Okay
...

d12: Submirror of d2
State: Needs maintenance
Invoke: "metareplace d2 /dev/dsk/c0t3d0s6 <new device>"
Size: 286740 blocks
Stripe 0:
Device          Start Block Dbase State      Hot Spare
/dev/dsk/c0t3d0s6      0      No   Maintenance

d22: Submirror of d2
State: Okay
Size: 286740 blocks
Stripe 0:
Device          Start Block Dbase State      Hot Spare
/dev/dsk/c0t2d0s6      0      No   Okay
```

在本示例中，metastat 命令显示以下子镜像需要维护：

- 子镜像 d10、设备 c0t3d0s0
- 子镜像 d11、设备 c0t3d0s1
- 子镜像 d12、设备 c0t3d0s6

4 停止系统，然后替换磁盘。使用 format 命令或 fmthard 命令按照磁盘出现故障之前的状态对磁盘进行分区。

提示 - 如果新磁盘与现有磁盘（在本示例中为镜像完整的一面）相同，请快速格式化新磁盘。为此，可使用 prtvtoc /dev/rdisk/c0t2d0s2 | fmthard -s - /dev/rdisk/c0t3d0s2 命令（在本示例中为 c0t3d0）。

```
# halt
...
Halted
...
ok boot
...
# format /dev/rdisk/c0t3d0s0
```

5 重新引导系统。

请注意，必须从另一半根 (/) 镜像重新引导。在创建镜像时，您应该已经记录了备用引导设备。

```
# halt
...
ok boot disk2
```

6 要删除出现故障的状态数据库副本重新添加它们，请使用 `metadb` 命令。

```
# metadb
      flags      first blk  block count
M      p      unknown    unknown    /dev/dsk/c0t3d0s3
M      p      unknown    unknown    /dev/dsk/c0t3d0s3
a m  p  luo     16        1034    /dev/dsk/c0t2d0s3
a      p  luo    1050     1034    /dev/dsk/c0t2d0s3
a      p  luo     16        1034    /dev/dsk/c0t1d0s3
a      p  luo    1050     1034    /dev/dsk/c0t1d0s3
# metadb -d c0t3d0s3
# metadb -c 2 -a c0t3d0s3
# metadb
      flags      first blk  block count
a m  p  luo     16        1034    /dev/dsk/c0t2d0s3
a      p  luo    1050     1034    /dev/dsk/c0t2d0s3
a      p  luo     16        1034    /dev/dsk/c0t1d0s3
a      p  luo    1050     1034    /dev/dsk/c0t1d0s3
a          u     16        1034    /dev/dsk/c0t3d0s3
a          u    1050     1034    /dev/dsk/c0t3d0s3
```

7 使用 `metareplace` 命令重新启用子镜像。

```
# metareplace -e d0 c0t3d0s0
Device /dev/dsk/c0t3d0s0 is enabled

# metareplace -e d1 c0t3d0s1
Device /dev/dsk/c0t3d0s1 is enabled

# metareplace -e d2 c0t3d0s6
Device /dev/dsk/c0t3d0s6 is enabled
```

一段时间之后，重新同步将完成。现在您可以恢复为从原始设备引导。

# 从状态数据库副本故障中恢复

如果无法满足状态数据库副本定额（例如，由于驱动器故障），则系统无法重新引导至多用户模式。当 Solaris Volume Manager 发现可用的状态数据库副本不足一半时，可能会引发系统紧急情况。如果使用恰好一半或者更少的正常运行的状态数据库副本重新引导系统，也可能出现这种情况。按 Solaris Volume Manager 术语来说，状态数据库已“过时”。以下过程说明如何从此问题中恢复。

## ▼ 如何从不足的状态数据库副本中恢复

- 1 引导系统。

- 2 确定哪些状态数据库副本不可用。

```
# metadb -i
```

- 3 如果已知有一个或多个磁盘不可用，请删除这些磁盘上的状态数据库副本。否则，请删除足够数量的出错的状态数据库副本（通过 `metadb` 所报告的 W、M、D、F 或 R 状态标志表明），以确保大多数的现有状态数据库副本都没有错误。

```
# metadb -d disk-slice
```

---

提示 – 带有大写状态标志的状态数据库副本是出错的副本。带有小写状态标志的状态数据库副本是正常运行的副本。

---

- 4 验证是否已删除副本。

```
# metadb
```

- 5 重新引导系统。

```
# reboot
```

- 6 如有必要，请替换磁盘，正确格式化该磁盘，然后添加该磁盘所需的任何状态数据库副本。

请按照第 64 页中的“创建状态数据库副本”中的说明操作。

在拥有替换磁盘后，停止系统，替换故障磁盘，然后再次重新引导系统。使用 `format` 命令或 `fmthard` 命令按照磁盘出现故障之前的配置对磁盘进行分区。

### 示例 25-1 从过时的状态数据库副本中恢复

在以下示例中，包含七个副本的磁盘已出现错误。因此，系统只有三个良好的副本。系统将发生紧急情况，然后无法重新引导至多用户模式。

```
panic[cpu0]/thread=70a41e00: md: state database problem

403238a8 md:mddb_commitrec_wrapper+6c (2, 1, 70a66ca0, 40323964, 70a66ca0, 3c)
%l0-7: 0000000a 00000000 00000001 70bbcce0 70bbcd04 70995400 00000002 00000000
40323908 md:alloc_entry+c4 (70b00844, 1, 9, 0, 403239e4, ff00)
%l0-7: 70b796a4 00000001 00000000 705064cc 70a66ca0 00000002 00000024 00000000
40323968 md:md_setdevname+2d4 (7003b988, 6, 0, 63, 70a71618, 10)
%l0-7: 70a71620 00000000 705064cc 70b00844 00000010 00000000 00000000 00000000
403239f8 md:setnm_ioctl+134 (7003b968, 100003, 64, 0, 0, ffbffc00)
%l0-7: 7003b988 00000000 70a71618 00000000 00000000 000225f0 00000000 00000000
40323a58 md:md_base_ioctl+9b4 (157ffff, 5605, ffbffa3c, 100003, 40323ba8, ffb5470)
%l0-7: ff3f2208 ff3f2138 ff3f26a0 00000000 00000000 00000064 ff1396e9 00000000
40323ad0 md:md_admin_ioctl+24 (157ffff, 5605, ffbffa3c, 100003, 40323ba8, 0)
%l0-7: 00005605 ffbffa3c 00100003 0157ffff 0aa64245 00000000 7efeff 81010100
40323b48 md:mdioctl+e4 (157ffff, 5605, ffbffa3c, 100003, 7016db60, 40323c7c)
%l0-7: 0157ffff 00005605 ffbffa3c 00100003 0003ffff 70995598 70995570 0147c800
40323bb0 genunix:ioctl+1dc (3, 5605, ffbffa3c, ffffffff8, fffffffe0, ffbffa65)
%l0-7: 0114c57c 70937428 ff3f26a0 00000000 00000001 ff3b10d4 0aa64245 00000000

panic:
stopped at      edd000d8:      ta      %icc,%0 + 125
Type 'go' to resume

ok boot -s
Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), No Keyboard
OpenBoot 3.11, 128 MB memory installed, Serial #9841776.
Ethernet address 8:0:20:96:2c:70, Host ID: 80962c70.


Rebooting with command: boot -s
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a File and args: -s
SunOS Release 5.9 Version s81_39 64-bit


Copyright 1983-2001 Sun Microsystems, Inc. All rights reserved.
configuring IPv4 interfaces: hme0.
Hostname: dodo

metainit: dodo: stale databases

Insufficient metadvice database replicas located.

Use metadb to delete databases which are broken.
Ignore any "Read-only file system" error messages.
Reboot the system when finished to reload the metadvice database.
After reboot, repair any broken database replicas which were deleted.

Type control-d to proceed with normal startup,
(or give root password for system maintenance): root-password
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode

Jun  7 08:57:25 su: 'su root' succeeded for root on /dev/console
Sun Microsystems Inc. SunOS 5.9      s81_39 May 2002
# metadb -i
      flags      first blk      block count
a m p lu      16      8192      /dev/dsk/c0t0d0s7
```

```

      a      p      l      8208      8192      /dev/dsk/c0t0d0s7
      a      p      l      16400     8192      /dev/dsk/c0t0d0s7
      M      p      16      unknown    /dev/dsk/c1t1d0s0
      M      p      8208     unknown    /dev/dsk/c1t1d0s0
      M      p      16400    unknown    /dev/dsk/c1t1d0s0
      M      p      24592    unknown    /dev/dsk/c1t1d0s0
      M      p      32784    unknown    /dev/dsk/c1t1d0s0
      M      p      40976    unknown    /dev/dsk/c1t1d0s0
      M      p      49168    unknown    /dev/dsk/c1t1d0s0
# metadb -d c1t1d0s0
# metadb
      flags      first blk      block count
      a m p l u      16      8192      /dev/dsk/c0t0d0s7
      a      p      l      8208     8192      /dev/dsk/c0t0d0s7
      a      p      l      16400    8192      /dev/dsk/c0t0d0s7
#
```

系统发生紧急情况，原因是它无法再检测到分片 `/dev/dsk/c1t1d0s0` 上的状态数据库副本。此分片位于故障磁盘上或者与故障控制器连接。第一个 `metadb -i` 命令将此分片上的副本标识为主块出现问题。

删除过时的状态数据库副本后，根 (`/`) 文件系统将成为只读系统。您可以忽略所显示的 `mddb.cf` 错误消息。

此时，尽管系统拥有的状态数据库副本可能不足，但仍然会重新正常运行。任何使用故障存储中某个部分的卷也将出现故障、出错或成为热备用卷。这些问题应该尽快解决。

# 从软分区问题中恢复

本节说明如何恢复软分区的配置信息。仅当所有的状态数据库副本都已丢失，而且没有下列任一信息时，才应该使用以下过程：

- `metastat -p` 输出的当前或正确副本
- `md.cf` 文件的当前或正确副本
- 最新的 `md.tab` 文件

## ▼ 如何恢复软分区的配置数据

在每个软分区盘区的开头，有一个用于标记软分区盘区开始的扇区。这些隐藏的扇区称为**盘区标头**。这些标头不对软分区的用户显示。如果所有的 Solaris Volume Manager 配置数据都已丢失，则可以通过扫描磁盘来尝试生成配置数据。

不到万不得已，请不要使用此过程来恢复所丢失的软分区配置信息。仅当 `metadb` 和 `md.cf` 文件均已丢失，并且 `md.tab` 文件也已丢失或过期时，才应使用 `metarecover` 命令。

注- 此过程仅用于恢复软分区信息，无法从丢失其他配置中进行恢复或者恢复其他 Solaris Volume Manager 卷的配置信息。

注- 如果您的配置中包括在软分区之上构建的其他 Solaris Volume Manager 卷，则在尝试恢复其他卷之前应该先恢复软分区。

有关软分区的配置信息存储在设备上和状态数据库中。由于这两个源都可能会损坏，因此必须向 `metarecover` 命令指示哪个源可靠。

首先，使用 `metarecover` 命令来确定这两个源是否一致。如果它们一致，就不能使用 `metarecover` 命令执行任何更改。但是，如果 `metarecover` 命令报告存在不一致，则必须仔细检查其输出，以确定磁盘或状态数据库是否已损坏。然后，应使用 `metarecover` 命令基于适当的源重新生成配置。

- 1 请阅读第 136 页中的“软分区的配置原则”。
- 2 使用 `metarecover` 命令查看软分区恢复信息。

```
# metarecover component -p -d
component      指定原始组件的 cnt ndnsn 名称
-p             指定重新生成软分区
-d             指定扫描物理分片以查找软分区的盘区标头
```

示例 25-2 从磁盘上的盘区标头中恢复软分区

```
# metarecover clt1d0s1 -p -d
The following soft partitions were found and will be added to
your metadevice configuration.
  Name      Size      No. of Extents
    d10      10240          1
    d11      10240          1
    d12      10240          1
# metarecover clt1d0s1 -p -d
The following soft partitions were found and will be added to
your metadevice configuration.
  Name      Size      No. of Extents
    d10      10240          1
    d11      10240          1
    d12      10240          1
WARNING: You are about to add one or more soft partition
metadevices to your metadevice configuration.  If there
appears to be an error in the soft partition(s) displayed
above, do NOT proceed with this recovery operation.
Are you sure you want to do this (yes/no)?yes
clt1d0s1: Soft Partitions recovered from device.
```

```
bash-2.05# metastat
d10: Soft Partition
  Device: cltld0s1
  State: Okay
  Size: 10240 blocks
    Device      Start Block  Dbase Reloc
    cltld0s1      0         No    Yes

    Extent      Start Block      Block count
    0           1           10240

d11: Soft Partition
  Device: cltld0s1
  State: Okay
  Size: 10240 blocks
    Device      Start Block  Dbase Reloc
    cltld0s1      0         No    Yes

    Extent      Start Block      Block count
    0          10242      10240

d12: Soft Partition
  Device: cltld0s1
  State: Okay
  Size: 10240 blocks
    Device      Start Block  Dbase Reloc
    cltld0s1      0         No    Yes

    Extent      Start Block      Block count
    0          20483      10240
```

在本示例中，状态数据库副本被意外删除后，从磁盘恢复了三个软分区。

# 从另一个系统恢复存储

即使在不同于原始系统的系统上也可以恢复 Solaris Volume Manager 配置。

## ▼ 如何从本地磁盘集恢复存储

如果遇到系统故障，则可以将存储连接到另一个系统，然后从本地磁盘集恢复完整的配置。例如，假设您的系统有一个由六个磁盘组成的外部磁盘组，某些磁盘上具有 Solaris Volume Manager 配置（至少包括一个状态数据库副本）。如果系统出现故障，则可以将这个磁盘组以物理方式移到新系统中，并使新系统能够识别该配置。此过程说明如何将磁盘移到另一个系统以及如何从本地磁盘集恢复配置。

---

注 – 此恢复过程仅适用于 Solaris 9 及更高版本的 Solaris Volume Manager 卷。

---

- 1 将包含 Solaris Volume Manager 配置的一个或多个磁盘连接到原本没有 Solaris Volume Manager 配置的系统。

- 2 执行重新配置重新引导，以确保系统能够识别新添加的磁盘。

```
# reboot -- -r
```

- 3 确定新添加的磁盘上状态数据库副本所在的分片的主/从设备号。

运行 `ls -ll`，并记下组名和日期之间的两个数字。这两个数字就是此分片的主/从设备号。

```
# ls -ll /dev/dsk/clt9d0s7
brw-r----- 1 root sys 32, 71 Dec 5 10:05 /dev/dsk/clt9d0s7
```

- 4 如有必要，请通过在 `/etc/name_to_major` 中查找主设备号来确定与主设备号相对应的主设备名称。

```
# grep " 32" /etc/name_to_major sd 32
```

- 5 在 `/kernel/drv/md.conf` 文件中更新相应信息，以指示 Solaris Volume Manager 在新磁盘上找到有效的状态数据库副本。

例如，在以 `mddb_bootlist1` 开头的行中，使用在步骤 4 中找到的主设备名称替换 `sd`。使用在步骤 3 中确定的从设备号替换本示例中的 `71`。

```
#pragma ident "@(#)md.conf 2.2 04/04/02 SMI"
#
# Copyright 2004 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# The parameters nmd and md_nsets are obsolete. The values for these
# parameters no longer have any meaning.
name="md" parent="pseudo" nmd=128 md_nsets=4;

# Begin MDD database info (do not edit)
mddb_bootlist1="sd:71:16:id0";
# End MDD database info (do not edit)
```

- 6 重新引导以强制 Solaris Volume Manager 重新装入配置。

您将看到类似于以下内容的消息显示在控制台上。

```
volume management starting.
Dec 5 10:11:53 host1 metadevadm: Disk movement detected
Dec 5 10:11:53 host1 metadevadm: Updating device names in
Solaris Volume Manager
The system is ready.
```

- 7 验证您的配置。使用 `metadb` 命令验证状态数据库副本的状态，并使用 `metastat` 命令查看每个卷的状态。

```
# metadb
      flags          first blk      block count
a m p  lu0          16              8192      /dev/dsk/clt9d0s7
a      lu0          16              8192      /dev/dsk/clt10d0s7
a      lu0          16              8192      /dev/dsk/clt11d0s7
```

```

a      lu0      16      8192      /dev/dsk/clt12d0s7
a      lu0      16      8192      /dev/dsk/clt13d0s7
# metastat
d12: RAID
  State: Okay
  Interlace: 32 blocks
  Size: 125685 blocks
Original device:
  Size: 128576 blocks
    Device      Start Block  Dbase State      Reloc  Hot Spare
    clt11d0s3      330      No  Okay      Yes
    clt12d0s3      330      No  Okay      Yes
    clt13d0s3      330      No  Okay      Yes

d20: Soft Partition
  Device: d10
  State: Okay
  Size: 8192 blocks
    Extent      Start Block      Block count
    0      3592      8192

d21: Soft Partition
  Device: d10
  State: Okay
  Size: 8192 blocks
    Extent      Start Block      Block count
    0      11785      8192

d22: Soft Partition
  Device: d10
  State: Okay
  Size: 8192 blocks
    Extent      Start Block      Block count
    0      19978      8192

d10: Mirror
  Submirror 0: d0
    State: Okay
  Submirror 1: d1
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 82593 blocks

d0: Submirror of d10
  State: Okay
  Size: 118503 blocks
  Stripe 0: (interlace: 32 blocks)
    Device      Start Block  Dbase State      Reloc  Hot Spare
    clt9d0s0      0      No  Okay      Yes
    clt10d0s0      3591      No  Okay      Yes

d1: Submirror of d10
  State: Okay
  Size: 82593 blocks
  Stripe 0: (interlace: 32 blocks)
```

```

Device      Start Block  Dbase State      Reloc  Hot Spare
c1t9d0s1    0          No  Okay      Yes
c1t10d0s1   0          No  Okay      Yes

Device Relocation Information:
Device      Reloc      Device ID
c1t9d0      Yes      id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3487980000U00907AZ
c1t10d0     Yes      id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3397070000W0090A8Q
c1t11d0     Yes      id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3449660000U00904NZ
c1t12d0     Yes      id1,sd@SSEAGATE_ST39103LCSUN9.0GLS32655400007010H04J
c1t13d0     Yes      id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3461190000701001T0
#
# metadb
      flags      first blk      block count
a m p  lu0      16          8192      /dev/dsk/c1t9d0s7
a      lu0      16          8192      /dev/dsk/c1t10d0s7
a      lu0      16          8192      /dev/dsk/c1t11d0s7
a      lu0      16          8192      /dev/dsk/c1t12d0s7
a      lu0      16          8192      /dev/dsk/c1t13d0s7
# metastat
d12: RAID
State: Okay
Interlace: 32 blocks
Size: 125685 blocks
Original device:
Size: 128576 blocks
      Device      Start Block  Dbase State      Reloc  Hot Spare
c1t11d0s3    330          No  Okay      Yes
c1t12d0s3    330          No  Okay      Yes
c1t13d0s3    330          No  Okay      Yes

d20: Soft Partition
Device: d10
State: Okay
Size: 8192 blocks
Extent      Start Block      Block count
0           3592          8192

d21: Soft Partition
Device: d10
State: Okay
Size: 8192 blocks
Extent      Start Block      Block count
0           11785         8192

d22: Soft Partition
Device: d10
State: Okay
Size: 8192 blocks
Extent      Start Block      Block count
0           19978         8192

d10: Mirror
Submirror 0: d0
State: Okay
Submirror 1: d1
State: Okay
```

```
Pass: 1
Read option: roundrobin (default)
Write option: parallel (default)
Size: 82593 blocks

d0: Submirror of d10
State: Okay
Size: 118503 blocks
Stripe 0: (interlace: 32 blocks)
  Device          Start Block  Dbase State      Reloc Hot Spare
  c1t9d0s0         0          No  Okay      Yes
  c1t10d0s0        3591       No  Okay      Yes

d1: Submirror of d10
State: Okay
Size: 82593 blocks
Stripe 0: (interlace: 32 blocks)
  Device          Start Block  Dbase State      Reloc Hot Spare
  c1t9d0s1         0          No  Okay      Yes
  c1t10d0s1         0          No  Okay      Yes

Device Relocation Information:
Device      Reloc   Device ID
c1t9d0      Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3487980000U00907AZ1
c1t10d0     Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3397070000W0090A8Q
c1t11d0     Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3449660000U00904NZ
c1t12d0     Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS32655400007010H04J
c1t13d0     Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3461190000701001T0
# metastat -p
d12 -r c1t11d0s3 c1t12d0s3 c1t13d0s3 -k -i 32b
d20 -p d10 -o 3592 -b 8192
d21 -p d10 -o 11785 -b 8192
d22 -p d10 -o 19978 -b 8192
d10 -m d0 d1 1
d0 1 2 c1t9d0s0 c1t10d0s0 -i 32b
d1 1 2 c1t9d0s1 c1t10d0s1 -i 32b
#
```

# 从已知磁盘集恢复存储

在 Solaris Volume Manager 中引入了对磁盘集设备 ID 的支持，这允许您从已知磁盘集恢复存储并将该磁盘集导入另一个系统。通过 `metainport` 命令，可以将已知磁盘集从一个系统导入另一个系统。这两个系统都必须包含现有的 Solaris Volume Manager 配置（包括设备 ID 支持）。有关设备 ID 支持的详细信息，请参阅第 182 页中的“磁盘集中的异步共享存储”。有关 `metainport` 命令的更多信息，请参见 `metainport(1M)` 手册页。

## ▼ 如何打印有关可导入的磁盘集的报告

- 1 成为超级用户。
- 2 获取有关可导入的磁盘集的报告。

```
# metainport -r -v
```

-r 提供可导入到系统上的未配置磁盘集的报告。

-v 提供有关以下内容的详细信息：状态数据库副本的位置；可导入到系统上的未配置磁盘集中各磁盘的状态。

### 示例 25-3 报告可导入的磁盘集

以下示例说明如何打印有关可导入磁盘集的报告。

```
# metainport -r
Drives in regular diskset including disk clt2d0:
  clt2d0
  clt3d0
More info:
  metainport -r -v clt2d0
Import:  metainport -s <newsetname> clt2d0
Drives in replicated diskset including disk clt4d0:
  clt4d0
  clt5d0
More info:
  metainport -r -v clt4d0
Import:  metainport -s <newsetname> clt4d0
```

```
# metainport -r -v clt2d0
Import: metainport -s <newsetname> clt2d0
Last update: Mon Dec 29 14:13:35 2003
Device      offset      length replica flags
clt2d0       16          8192      a      u
clt3d0       16          8192      a      u
clt8d0       16          8192      a      u
```

## ▼ 如何将磁盘集从一个系统导入到另一个系统

- 1 成为超级用户。
- 2 验证磁盘集是否可导入。

```
# metainport -r -v
```

- 3 导入可用磁盘集。

```
# metainport -s diskset-name drive-name
```

- *s diskset-name*      指定要创建的磁盘集的名称。
- drive-name*            标识要导入的磁盘集中包含状态数据库副本的磁盘 (c#t#d#)。

#### 4 验证该磁盘集是否已导入。

```
# metaset -s diskset-name
```

### 示例 25-4 导入磁盘集

以下示例说明如何导入磁盘集。

```
# metaimport -s red c1t2d0
Drives in diskset including disk c1t2d0:
  c1t2d0
  c1t3d0
  c1t8d0
More info:  metaimport -r -v c1t2d0# metaset -s red
```

```
Set name = red, Set number = 1
```

Host	Owner
host1	Yes

Drive	Dbase
c1t2d0	Yes
c1t3d0	Yes
c1t8d0	Yes

## 从磁盘集问题中恢复

以下各节详细说明了如何从与磁盘集相关的特定问题中恢复。

### 无法获取磁盘集所有权时应执行的操作

如果无法从任何节点获取对磁盘集的所有权（可能由于系统故障、磁盘故障或通信链路故障），从而无法删除磁盘集记录，则可以从当前主机上的 Solaris Volume Manager 状态数据库副本记录中清除磁盘集。

清除磁盘集记录不会影响磁盘集中包含的状态数据库信息，因此稍后可导入磁盘集（使用 `metaimport` 命令，如第 195 页中的“导入磁盘集”中所述）。

如果需要从 Sun Cluster 配置中清除磁盘集，请使用以下过程，但是，在没有 Sun Cluster 配置时，应使用 `-C` 选项代替 `-P` 选项。

## ▼ 如何清除磁盘集

### 1 尝试使用 `metaset` 命令获取磁盘集。

```
# metaset -s setname -t -f
```

此命令将尝试强制 (-f) 获取 (-t) 名为 *setname* 的磁盘集。如果磁盘集可供获取，则此命令将成功。如果此命令运行时该磁盘集属于另一台主机，则该主机将发生紧急情况，以防止数据损坏或丢失。如果此命令成功，则可以彻底删除该磁盘集，而无需清除它。

如果无法获取该磁盘集，则可以清除所有权记录。

### 2 使用带 `-P` 的 `metaset` 命令从当前主机中清除该磁盘集。

```
# metaset -s setname -P
```

此命令将从运行该命令的主机中清除 (-P) 名为 *setname* 的磁盘集。

### 3 使用 `metaset` 命令验证该磁盘集是否已被清除。

```
# metaset
```

## 示例 25-5 清除磁盘集

```
host1# metaset -s red -t -f
metaset: host1: setname "red": no such set
```

```
host2# metaset
```

```
Set name = red, Set number = 1
```

```
Host          Owner
  host2
```

```
Drive  Dbase
```

```
c1t2d0  Yes
```

```
c1t3d0  Yes
```

```
c1t8d0  Yes
```

```
host2# metaset -s red -P
host2# metaset
```

- 另请参见
- [第 18 章，磁盘集（概述）](#)（介绍有关磁盘集的概念信息）。
  - [第 19 章，磁盘集（任务）](#)（介绍有关与磁盘集相关联的任务的信息）。

# 使用 ufsdump 命令对已挂载的文件系统执行备份

以下过程说明在使用 ufsdump 命令来备份 RAID-1 卷上的已挂载文件系统时如何提高该命令的性能。

## ▼ 如何对 RAID-1 卷上的已挂载文件系统执行备份

可以使用 ufsdump 命令来备份 RAID-1 卷上的已挂载文件的文件。当备份实用程序为 ufsdump 时，请将该卷上的读取策略设置为 "first"。这样将提高备份的执行速率。

- 1 成为超级用户。
- 2 运行 **metastat** 命令以确保镜像处于 "Okay" ( 正常 ) 状态。

```
# metastat d40
d40: Mirror
  Submirror 0: d41
    State: Okay
  Submirror 1: d42
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 20484288 blocks (9.8 GB)
```

应首先恢复处于 "Maintenance" ( 维护 ) 状态的镜像。

- 3 将镜像的读取策略设置为 "first"。

```
# metaparam -r first d40
# metastat d40
d40: Mirror
  Submirror 0: d41
    State: Okay
  Submirror 1: d42
    State: Okay
  Pass: 1
  Read option: first
  Write option: parallel (default)
  Size: 20484288 blocks (9.8 GB)
```

- 4 对文件系统执行备份。
- 5 ufsdump 命令执行完之后，将镜像的读取策略设置为 "roundrobin"。

```
# metaparam -r roundrobin d40
# metastat d40
d40: Mirror
  Submirror 0: d41
    State: Okay
  Submirror 1: d42
```

```
State: Okay
Pass: 1
Read option: roundrobin
Write option: parallel (default)
Size: 20484288 blocks (9.8 GB)
```

## 执行系统恢复

有时，从 DVD 或 CD 介质上的 Solaris OS 安装映像进行引导有助于执行系统恢复。例如，对于重置 root 口令，使用安装映像就非常有用。

如果您使用的是 Solaris Volume Manager 配置，则需要挂载 Solaris Volume Manager 卷而不是底层磁盘。如果根 (/) 文件系统已镜像，则此步骤尤其重要。由于 Solaris Volume Manager 是 Solaris OS 的一部分，因此挂载 Solaris Volume Manager 卷将确保所做的更改在镜像的两面都反映出来。

通过以下过程，可以使 Solaris Volume Manager 卷可供 Solaris OS DVD 或 CD-ROM 安装映像访问。

### ▼ 如何使用 Solaris Volume Manager 配置来恢复系统

从 Solaris OS 安装 DVD 或 CD 介质引导系统。从 Solaris miniroot 的 root 提示符处执行此过程。

- 1 以只读方式挂载包含 Solaris Volume Manager 配置的底层磁盘。  

```
# mount -o ro /dev/dsk/c0t0d0s0 /a
```
- 2 将 md.conf 文件复制到 /kernel/drv 目录。  

```
# cp /a/kernel/drv/md.conf /kernel/drv/md.conf
```
- 3 从 miniroot 取消挂载文件系统。  

```
# umount /a
```
- 4 更新 Solaris Volume Manager 驱动程序以装入该配置。忽略 update\_drv 命令所显示的任何警告消息。  

```
# update_drv -f md
```
- 5 配置系统卷。  

```
# metainit -r
```
- 6 如果您的 Solaris Volume Manager 配置中包含 RAID-1 卷，请将其重新同步。  

```
# metasync mirror-name
```

## 7 使用 mount 命令应该可访问 Solaris Volume Manager 卷。

```
# mount /dev/md/dsk/volume-name /a
```

### 示例 25-6 使用 Solaris Volume Manager 配置来恢复系统

```
# mount -o ro /dev/dsk/c0t0d0s0 /a
# cp /a/kernel/drv/md.conf /kernel/drv/md.conf
# umount /a
# update_drv -f md
Cannot unload module: md
Will be unloaded upon reboot.
Forcing update of md.conf.
devfsadm: mkdir failed for /dev 0xled: Read-only file system
devfsadm: inst_sync failed for /etc/path_to_inst.1359: Read-only file system
devfsadm: WARNING: failed to update /etc/path_to_inst
# metainit -r
# metasync d0
# mount /dev/md/dsk/d0 /a
```



## 重要的 Solaris Volume Manager 文件

---

本附录包含有关 Solaris Volume Manager 文件的信息，以供参考。本附录包含以下主题：

- 第 279 页中的“系统文件和启动文件”
- 第 280 页中的“手动配置的文件”

### 系统文件和启动文件

本节介绍了 Solaris Volume Manager 正常运行所必需的文件。除了进行少量专门性的配置更改外，无需访问或修改这些文件。

- `/etc/lvm/mddb.cf`



---

**注意** – 请勿编辑此文件。如果更改此文件，可能会破坏 Solaris Volume Manager 配置。

---

`/etc/lvm/mddb.cf` 文件记录状态数据库副本的位置。如果状态数据库副本位置发生变化，Solaris Volume Manager 就会在 `mddb.cf` 文件中生成一个条目，以记录所有状态数据库的位置。有关更多信息，请参见 [mddb.cf\(4\)](#) 手册页。

- `/etc/lvm/md.cf`

`/etc/lvm/md.cf` 文件包含为缺省（未指定或本地）磁盘集自动生成的配置信息。更改 Solaris Volume Manager 配置时，Solaris Volume Manager 会自动更新 `md.cf` 文件（有关正在使用的热备件的信息除外）。有关更多信息，请参见 [md.cf\(4\)](#) 手册页。



---

**注意** – 请勿编辑此文件。如果更改此文件，可能会破坏或无法恢复 Solaris Volume Manager 配置。

---

如果系统丢失了状态数据库中维护的信息，只要在此期间未更改或创建任何卷，就可以使用 `md.cf` 文件恢复配置。请参见第 204 页中的“[如何根据配置文件初始化 Solaris Volume Manager](#)”。

- `/kernel/drv/md.conf`

Solaris Volume Manager 在启动时会读取 `md.conf` 配置文件。`md.conf` 文件包含状态数据库副本配置信息。自 Solaris 10 开始，不再手动编辑 `nmd` 和 `md_nsets` 参数。Solaris Volume Manager 已增强，可以根据需要动态配置卷。

## 手动配置的文件

### md.tab 文件概述

`/etc/lvm/md.tab` 文件包含可用于重新构建 Solaris Volume Manager 配置的 Solaris Volume Manager 配置信息。Solaris Volume Manager 可以使用此文件作为命令行实用程序 `metainit`、`metadb` 和 `metahs` 的输入来重新构建配置。卷、磁盘集和热备用池可能在此文件中存在相应条目。有关创建此文件（通过使用 `metastat -p > /etc/lvm/md.tab` 命令）的说明，请参见第 204 页中的“[如何创建配置文件](#)”。

---

**注** – `/etc/lvm/md.tab` 文件中的配置信息可能与正在使用中的当前卷、热备件和状态数据库副本不同。系统管理员以手动方式使用此文件来捕获所需的配置。更改 Solaris Volume Manager 配置后，请重新创建此文件并保留一个备份副本。

---

创建并更新此文件后，`metainit`、`metahs` 和 `metadb` 命令就会激活此文件中定义的卷、热备用池和状态数据库副本。

在 `/etc/lvm/md.tab` 文件中，有关单个卷的一个完整配置条目会出现在使用 `metainit`、`metadb` 和 `metahs` 命令语法的每行中。

---

**注** – 如果使用 `metainit -an` 命令模拟初始化 `md.tab` 文件中的所有卷，则可能会显示有关某些卷的错误消息，这些卷依赖于 `md.tab` 中定义的其他卷。出现这些错误消息的原因是，Solaris Volume Manager 不维护在 `metainit -an` 运行期间所创建的卷的状态。每一行将基于现有配置（如果配置存在）进行评估。因此，即使表明 `metainit -an` 命令可能会失败，在不使用 `-n` 选项的情况下该命令也可能会成功。

---

然后，可以运行带有 `-a` 选项的 `metainit` 命令来激活 `/etc/lvm/md.tab` 文件中的所有卷，也可以在该命令中使用与此文件中的特定条目对应的卷名来激活卷。

---

**注** – Solaris Volume Manager 不会在 `/etc/lvm/md.tab` 文件中写入或存储配置信息。必须手动编辑此文件并运行 `metainit`、`metabs` 或 `metadb` 命令才能创建 Solaris Volume Manager 组件。

---

有关更多信息，请参见 [md.tab\(4\)](#) 手册页。



# Solaris Volume Manager 快速参考

本附录提供有关 Solaris Volume Manager 中可用的特征和功能的快速访问信息。

## 命令行参考

此处列出了用于管理 Solaris Volume Manager 的所有命令。有关更多详细信息，请参见手册页。

表 B-1 Solaris Volume Manager 命令

Solaris Volume Manager 命令	说明	手册页
growfs	以非破坏性方式扩展 UFS 文件系统。	<a href="#">growfs(1M)</a>
metaclear	删除活动卷和热备用池。	<a href="#">metaclear(1M)</a>
metadb	创建和删除状态数据库副本。	<a href="#">metadb(1M)</a>
metadetach	将卷与 RAID-0 或 RAID-1（镜像）卷分离，或将日志设备与事务卷分离。 注 - 不再支持事务卷。	<a href="#">metadetach(1M)</a>
metadevadm	请检查设备 ID 配置。	<a href="#">metadevadm(1M)</a>
metahs	管理热备件及热备用池。	<a href="#">metahs(1M)</a>
metaimport	将磁盘集（包括复制的磁盘集）导入在磁盘集中具备设备 ID 支持的现有 Solaris Volume Manager 配置。	<a href="#">metaimport(1M)</a>
metainit	配置卷。	<a href="#">metainit(1M)</a>
metaoffline	使子镜像脱机。	<a href="#">metaoffline(1M)</a>
metaonline	使子镜像联机。	<a href="#">metaonline(1M)</a>
metaparam	修改卷参数。	<a href="#">metaparam(1M)</a>

表 B-1 Solaris Volume Manager 命令 (续)

Solaris Volume Manager 命令	说明	手册页
metarecover	恢复软分区的配置信息。	<a href="#">metarecover(1M)</a>
metarename	重命名和交换卷名称。	<a href="#">metarename(1M)</a>
metareplace	替换子镜像和 RAID-5 卷中的组件。	<a href="#">metareplace(1M)</a>
metaroot	设置系统文件以便对根 (/) 文件系统进行镜像。	<a href="#">metaroot(1M)</a>
metaset	管理磁盘集。	<a href="#">metaset(1M)</a>
metastat	显示卷或热备用池的状态。	<a href="#">metastat(1M)</a>
metasync	重新引导期间重新同步卷。	<a href="#">metasync(1M)</a>
metattach	将组件附加到 RAID-0 或 RAID-1 卷。	<a href="#">metattach(1M)</a>

## Solaris Volume Manager CIM/WBEM API

---

### 管理 Solaris Volume Manager

Solaris Volume Manager CIM/WBEM 应用编程接口 (Application Programming Interface, API) 提供了用于观察和配置 Solaris Volume Manager 的基于标准的公共程序接口。此 API 基于分布式管理任务组 (Distributed Management Task Force, DMTF) 的通用信息模型 (Common Information Model, CIM)。有关 DMTF 的更多信息，请参见 <http://www.dmtf.org>。

CIM 定义了称为“架构”的数据模型，其中描述了以下内容：

- Solaris Volume Manager 设备的属性以及针对这些设备的操作
- 各种 Solaris Volume Manager 设备之间的关系
- Solaris Volume Manager 设备与操作系统的其他方面（如文件系统）之间的关系

此模型通过 Solaris 基于 Web 的企业管理 (Web Based Enterprise Management, WBEM) SDK 提供。WBEM SDK 是一组基于 Java 技术的 API，可用于访问 CIM 表示的系统管理功能。

有关 CIM/WBEM SDK 的更多信息，请参见《[Solaris WBEM Developer's Guide](#)》。



# 索引

---

## C

- 重命名卷, 201
- 重新对磁盘分区, 178–180
- 重新同步
  - 部分, 88
  - 完全, 88
  - 优化, 88
- 传送号
  - 定义的, 91
  - 只读镜像, 88
- cron 命令, 240

## D

DiskSuite 工具, [请参见图形界面](#)

## E

- /etc/lvm/md.cf 文件, 279
- /etc/lvm/mddb.cf 文件, 279
- /etc/vfstab 文件, 128
  - 从错误的条目中恢复, 257

## F

- fmthard 命令, 261, 263
- format 命令, 261, 263

## G

- growfs 命令, 206–208, 283
- growfs 命令, 37, 208
- GUI, 样例, 33

## I

- I/O, 29

## K

- /kernel/drv/md.conf 文件, 280

## L

- lockfs 命令, 132

## M

- md.cf 文件, 280
  - 恢复 Solaris Volume Manager 配置, 204
- md.tab 文件, 205
  - 概述, 280–281
- mdmonitord 命令, 236–237
- metaclear 命令, 283
- metaclear 命令, 84, 126, 127, 128–130
- metadb 命令, 283
- metadb 命令, 66
- metadb 命令, 替换故障磁盘和, 251

metadetach 命令, 283  
metadetach 命令, 117, 126  
metadevadm 命令, 283  
    替换故障磁盘和, 251  
metahs 命令, 283  
    启用, 171–172  
    删除热备件, 170–171  
    替换故障磁盘和, 251  
    替换热备件, 169–170  
    向热备用池添加分片, 165–166  
metaimport 命令, 177–178, 195–196, 271–273, 283  
metainit 命令, 283  
metainit 命令, 205  
metainit 命令, 创建热备用池, 164–165  
metaoffline 命令, 283  
metaoffline 命令, 118  
metaonline 命令, 283  
metaparam 命令, 283  
metaparam 命令, 121  
metaparam 命令  
    更改热备用池关联, 167–168  
    将热备用池与卷关联, 166–168  
metarecover 命令, 284  
    替换故障磁盘和, 251  
metarename 命令, 202, 284  
metarename 命令, 203  
metareplace 命令, 284  
metareplace 命令, 119, 155, 262  
metareplace 命令, 替换故障磁盘和, 251  
metaroot 命令, 284  
metaset 命令, 284  
    创建磁盘集, 184–185  
    从磁盘集中删除磁盘, 190  
    从磁盘集中删除主机, 193–194  
    获取磁盘集, 190–192  
    检查磁盘集的状态, 189  
    删除磁盘集, 193–194  
    释放磁盘集, 192–193  
    向磁盘集添加磁盘, 185  
    向磁盘集添加主机, 187  
metastat 命令, 284  
metastat 命令, 121, 153  
metasync 命令, 284

metattach  
    任务, 100, 104, 109, 113  
metattach 命令, 284  
metattach 命令, 83, 116, 123  
    附加 RAID-5 组件, 154  
    附加子镜像, 206

## O

Oracle Real Application Clusters, 45–46

## P

prtconf 命令, 查看设备 ID, 255–256

## R

RAID, Solaris Volume Manager 中支持的级别, 26  
RAID-0 卷  
    定义, 69  
    使用, 69  
RAID-0 (条带) 卷, 三个分片的示例, 70  
RAID-0+1, 87  
RAID-0 卷  
    类型, 69  
    原则, 75  
RAID-0 (串联) 卷, 创建信息, 75  
RAID-0 (串联条带) 卷  
    交错值, 73  
    三个条带的示例, 73  
RAID-0 (条带) 卷, 70  
    创建信息, 75  
RAID-1+0, 87  
RAID-1 卷, 85–87  
    RAID-0+1, 87  
    RAID-1+0, 87  
    创建, 96–115  
    读写策略, 91  
    镜像, 85–87  
    替换和启用组件, 208–211  
    替换和启用组件的信息, 211  
    维护和最近出错, 210

**RAID-1 卷 (续)**

- 选项, 90-91
- 引导至单用户模式, 93
- 传送号, 91
- 子镜像, 85-87

**RAID-5 卷**

- 创建, 152-153
- 定义, 26, 36
- 分片状态, 147-148
- 概述, 143-146
- 检查状态, 153-154
- 交错, 146
- 扩展, 154-155, 155
- 扩展设备的示例, 145
- 奇偶校验计算, 146
- 奇偶校验信息, 143, 145
- 启用出现故障的分片, 155
- 启用组件, 155
- 任务, 151
- 四个磁盘的示例, 144
- 替换故障分片, 157
- 替换和启用组件, 149, 208-211
- 替换和启用组件的信息, 211
- 替换组件, 156-157
- 维护和最近出错, 210
- 要求, 146-147
- 原则, 146
- 重新同步分片, 143
- 状态, 147-148

**S****SCSI 磁盘**

- 替换, 251, 253

**SMF**

- Solaris Volume Manager 服务, 42-43
- 升级 Solaris Volume Manager, 42-43

**Solaris Volume Manager**

- 请参见 Solaris Volume Manager
- Oracle Real Applications Clusters, 45-46
- Sun Cluster, 45-46
- 恢复配置, 204
- 配置原则, 40
- 网络存储设备和, 215

**Solaris Volume Manager for Sun Cluster**

- 超时, 49
- 基于应用程序的恢复, 50-51
- 配置, 49
- 软件组件, 46
- 数据管理和恢复, 50-51
- 优化的重新同步, 50-51
- 有向镜像读取, 50-51

**Solaris Volume Manager 界面**

- Solaris Management Console, 32
- 命令行, 32
- 样例 GUI, 33

**Solaris Volume Manager 元素, 概述, 34****Sun Cluster, 45-46**

- svc-mdmonitor 脚本, 236-237
- swap, 取消镜像, 130

**V**

- /var/adm/messages 文件, 208
- 替换故障磁盘, 252

**安**

- 安装程序, 系统恢复, 276-277

**保**

- 保留磁盘集, 176-177

**本**

- 本地磁盘集, 173-174

**并**

- 并行访问, 定义, 70
- 并行写入策略, 91

## 查

查看 Solaris Volume Manager 配置, 198–201

## 串

### 串联

创建, 80

定义

**另请参见**RAID-0（串联）卷

扩展, 83

扩展 UFS 文件系统, 72

三个分片的示例, 72

删除, 84

使用, 72

串联的条带, 删除, 84

串联卷, **请参见**RAID-0（串联）卷

串联条带

**请参见**RAID-0（串联条带）卷

定义

**另请参见**RAID-0（串联条带）卷

串行写入策略, 91

## 创

创建磁盘集, 184–185

创建配置文件, 204–206

## 磁

磁盘集, 173–174

保留, 176–177, 192

本地, 173–174

创建, 184–185

创建组件, 188–189

从磁盘集中删除主机, 193–194

导入, 177–178, 195–196

定义, 35, 39

多属主, 173–174

**请参见**多属主磁盘集

方案, 182

共享, 173–174

管理, 176–180

## 磁盘集（续）

获取, 190–192

检查状态, 189

命名约定, 180

删除磁盘, 190

删除磁盘集, 193–194

使用, 173–174

使用两个共享磁盘集的示例, 180

释放, 177, 192–193

添加磁盘, 185

添加主机, 187

已命名, 173–174

异步共享存储, 182

原则, 181

支持的最大数目, 206

自动磁盘分区, 178–180

自动获取, 173–174

## 从

从安装程序恢复系统, 276–277

从磁盘集中删除磁盘, 190

从磁盘集中删除主机, 193–194

## 错

错误, 检查, 使用脚本, 240–247

错误检查, 236–237

## 导

导入磁盘集, 177–178

## 多

多属主磁盘集, 173–174

RAID-1 卷, 50–51

导入, 45–46

定义的, 45–46

任务, 48–49

设备 ID 支持, 45–46

## 多属主磁盘集 (续)

主节点, 47-49  
多数一致算法, 57

## 分

分片, 扩展, 82

## 服

服务管理工具 (Service Management Facility, SMF), 请  
参见SMF

## 副

副本, 38

## 更

更改缺省值, 206

## 共

共享磁盘集, 173-174

## 故

故障排除

metainport 命令, 271-273  
从磁盘移动问题中恢复, 253  
错误的 /etc/vfstab 文件条目, 257  
导入磁盘集, 271-273  
设备 ID 差异, 255-256  
替换故障磁盘, 251  
一般原则, 250-251  
引导问题, 256-262  
故障转移配置, 39

## 管

管理磁盘集, 176-180

## 获

获取磁盘集, 190-192

## 几

几何读取策略, 91

## 基

基于应用程序的恢复, 50-51

## 检

检查磁盘集的状态, 189

## 简

简单卷, 定义, 35

## 交

交错, 指定, 78  
交错值, 70  
交换卷名称, 202

## 界

界面, 请参见Solaris Volume Manager 界面

## 镜

镜像  
请参见RAID-1 卷

## 镜像（续）

另请参见RAID-1 卷

创建, 96-115

定义, 36

读取和写入性能, 28

附加子镜像, 116

根 (/), /usr, 和 swap, 102

根文件系统

SPARC, 103-106

x86, 106-115

使用 GRUB, 106-115

更改选项, 122

更新大小, 122-123

和联机备份, 131

可以取消挂载的文件系统, 101

扩展, 123

两个子镜像的示例, 86

双向镜像, 224, 225, 227-229, 229, 231-232, 232, 233

替换和启用组件, 149

文件系统, 98-102

重新同步, 88

状态输出样例, 120

## 卷

### 卷

定义, 34

概念性概述, 35

交换名称, 202

扩展磁盘空间, 37-38

类型, 35

使用, 36

使用文件系统命令, 36

虚拟磁盘, 32

支持的最大数目, 206

重命名, 204

卷名称切换, 38

## 配置规划

概述, 26

权衡, 28

原则, 26

配置文件, 创建, 204-206

## 启

启用 RAID-5 卷中的分片, 155

启用子镜像中的分片, 119

## 热

### 热备件

定义, 160

概念性概述, 160

启用, 171-172

如何将热备件用于 Solaris Volume Manager, 160

添加到热备用池, 165

在热备用池中替换, 170

### 热备用池, 39

创建, 164-165

定义, 35, 39, 160

概念性概述, 159-161

更改关联, 167-168, 168

关联, 167

基本操作, 39

检查状态, 168-172

镜像示例, 161

删除热备件, 170-171

替换热备件, 169-170

添加分片, 165-166

维护, 168-172

与卷关联, 166-168

状态, 161

## 软

软分区, 135

创建, 137-138

定义, 135

恢复配置, 265

## 配

配置, 查看, 198-201

## 软分区（续）

- 检查状态, 139–141
- 扩展, 140
- 任务, 137
- 删除, 140–141
- 维护, 139–141
- 位置, 135
- 原则, 136

## 删

删除磁盘集, 193–194

## 设

设备 ID, 格式, 255–256

## 释

释放磁盘集, 177, 192–193

## 首

首次读取策略, 91

## 顺

顺序 I/O, 29

## 随

随机 I/O, 29–30

## 添

添加热备件, 165

## 条

### 条带

- 创建, 78
- 定义
  - 另请参见 RAID-0（条带）卷
- 扩展, 83
- 删除, 84

条带化, 定义, 70

条带卷, 请参见 RAID-0（条带）卷

## 图

图形界面, 概述, 32

## 文

### 文件系统

- 扩展, 206–208
- 扩展概述, 37
- 取消镜像, 130
- 通过创建串联扩展, 81–82
- 原则, 40

## 系

系统文件, 279–280

## 向

向磁盘集添加磁盘, 185

向磁盘集添加主机, 187

## 小

小型服务器, 部署的概述, 213–214

## 循

循环读取策略, 91

## 一

一般性能原则, 28

## 已

已命名磁盘集, 173–174

## 引

引导问题, 256–262

引导至单用户模式, 93

## 有

有向镜像读取, 50–51

## 元

元设备, [请参见卷](#)

## 在

在磁盘集中创建组件, 188–189

## 增

增强的存储, [请参见图形界面](#)

## 主

主节点, 47–49

## 状

状态, 189

状态数据库

    定义, 35, 38

## 状态数据库（续）

    概念性概述, 38–39, 58

    损坏, 57

状态数据库副本, 38

    磁盘集条件, 178–180

    错误, 60

    大小, 178–180

    定义, 38

    基本操作, 57

    两个磁盘的配置, 60–61

    使用, 57

    添加更大的副本, 67

    位置, 39, 59–60

    在一个分片上创建多个, 59–60

    最小数量, 59–60

## 子

子镜像, 85–86

[请参见RAID-1 卷](#)

    分离, 86

    附加, 86

    启用出现故障的分片, 119

    替换故障分片, 124

    替换整个, 126

    脱机时操作, 86

## 自

自动磁盘分区, 178–180

自动获取磁盘集, 173–174

自上而下创建卷

    RAID 1 卷, 创建, 223–225

    shell 脚本, 226–232

    卷配置文件, 230

    缺省值, 232–233

## 组

组件, 定义, 69